Title
Architecting Non-volatile Memory for High Bandwidth Systems

Author
Sim, Joonseop

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Architecting Non-volatile Memory for High Bandwidth Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Electrical Engineering (Nanoscale Devices & Systems)

by

Joonseop Sim

Committee in charge:

Professor Tajana Simunic Rosing, Chair
Professor Chung-Kuan Cheng
Professor Bill Lin
Professor Patrick Mercier
Professor Steven Swanson

2019

The Dissertation of Joonseop Sim is approved and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

DEDICATION

All my beloved family
I give thanks, glory and honor to your infinite love, encouragement, prayer and consideration.

EPIGRAPH

In their hearts humans plan their course,
but the Lord establishes their steps.

Proverbs 16:9 (NIV)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I give thanks and glory to God, who has led and guided my life to this day. I confess that there is no part of me without your labor.

I would like to thank my advisor Prof. Tajana Rosing for her guidance, encouragement, admonition, patience and support through my entire doctoral course. Although I came with a different background from my lab, she has guided me through understanding and guidance based on her broad knowledge. Her guidance has helped me to develop my ability to discover and promote research as well as the knowledge of my field. Also, I would like to thank my committee members, Prof. Bill Lin, Prof. Chung-Kuan Cheng, Prof. Patrick Mercier and Prof. Steven Swanson for their valuable discussions, feedback, and contributions to this work.

I would like to thank all my lab mates and friends in SEELab, both current and past, for their comments, feedback, discussions, and lots of good memories. It was a great benefit for me to work with exceptionally smart team with diverse backgrounds. I am grateful to all of them, but I would also like to give special thanks to Yeseong Kim, Mohsen Imani, and Saransh Gupta.

I would like to thank Prof. Shadi Dayeh for helping me during my doctoral course at UCSD and his professional advice in the field of devices. I would also like to thank my collaborators at the IEBL lab, Woojin Choi, Yungoo Ro, Atsunori Tanaka, and Renjie Chen.

I would like to thank SK-Hynix Inc. for supporting my entire PhD program. Also, I would like thank CRISP, one of six centers in JUMP, an SRC program sponsored by DARPA, and also NSF grants 1730158 and 1527034 for their continued support to this research.

I give my thanks and love to all my family. I give thanks and respect to Senior Pastor Kwon and Mrs. Kim, who have been my spiritual parents since I was a child, and now are my parents-in-law. I give thanks and respect to my father who always believed in and cheered me, and my mother who would be happy to see me in heaven. I give love to my sister and her husband, sister-in-law, pastor Kwon, his wife, and newborn prince Min.

Finally, I give my biggest thanks and love to my wife Eunhea for her devotion, love, advice, and prayer. I will devote the rest of my life to the research of her. Also, I give my best

love to my daughter Eunha and my son Eunho who fill all my days with happiness.

Chapter 2 contains material from Joonseop Sim, Mohsen Imani, Yeseong Kim, Tajana Rosing, “Enabling Efficient System Design Using VerticalNanowire Transistor Current Mode Logic”, IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2017 [13]. The dissertation author was the primary investigator and author of this paper.

Chapter 2 contains material from Shadi Dayeh, Renjie Chen, Yungoo Ro, Joonseop Sim, Progress in doping semiconductor nanowires during growth. Materials Science in Semiconductor Processing, 2017 [14]. The dissertation author was the fourth author of this paper.

Chapter 3 contains material from Joonseop Sim, Saransh Gupta, Mohsen Imani, Yeseong Kim, Tajana Rosing, “UPIM : Unipolar Switching Logic for High Density Processing-in-Memory Applications”, ACM Great lakes symposium on VLSI (GLSVLSI), 2019 [15]. The dissertation author was the primary investigator and author of this paper.

Chapter 4 contains material from Joonseop Sim, Mohsen Imani, Woojin Choi, Yeseong Kim, Tajana Rosing, “LUPIS: Latch-Up Based Ultra Efficient Processing-in-Memory System”, IEEE International Symposium on Quality Electronic Design (ISQED), 2018 [10]. The dissertation author was the primary investigator and author of this paper.

Chapter 4 contains material from Joonseop Sim, Mohsen Imani, Woojin Choi, Yeseong Kim, and Tajana Rosing. “Current-Sensing Efficient Adder for Processing in Memory Design.” Non-Volatile Memory Workshop (NVMW), 2019. [16]. The dissertation author was the primary investigator and author of this paper.

Chapter 5 contains material from Joonseop Sim, Minsu Kim, Yeseong Kim, Saransh Gupta, Behnam Khaleghi and Tajana Rosing, “MAPIM: Mat Parallelism for High Performance Processing in Non-volatile Memory Architecture”, IEEE International Symposium on Quality Electronic Design (ISQED), 2019 [17]. The dissertation author was the primary investigator and author of this paper.

Chapter 5 contains material from Joonseop Sim, Minsu Kim, Yeseong Kim, Saransh Gupta, Behnam Khaleghi, and Tajana Rosing. “Multi-bit Parallelized Sensing for Processing

in Non-volatile Memory." Non-Volatile Memory Workshop (NVMW). 2019. The dissertation author was the primary investigator and author of this paper.

Chapter 6 contains material from Joonseop Sim, Minsu Kim, Mohsen Imani, Yeseong Kim, Chris Kim and Tajana Rosing, "MAPLE: Multi-mat Parallelized Execution for Hyperdimensional Computing", which was submitted to ACM Journal of Emerging Technologies in Computing. The dissertation author was the primary investigator and author of this paper.

VITA

2004	B. S. in Material Science and Engineering, Yonsei University, South Korea
2004–2006	M. S. in Material Science and Engineering, Seoul National University, South Korea
2015–2019	Ph. D. in Electrical Engineering (Nano Devices & Systems), University of California, San Diego

PUBLICATIONS

Joonseop Sim, Mohsen Imani, Yeseong Kim, Saransh Gupta, and Tajana Rosing, "Excavating Hidden Parallelism for Processing-in-Memory Architecture" TECHCON SRC Conference, 2019

Joonseop Sim, Minsu Kim, Mohsen Imani, Yeseong Kim, Chris Kim, and Tajana Rosing, "MAPLE: Multi-mat Parallelized Execution for Hyperdimensional Computing ", ACM Journal on Emerging Technologies in Computing (JETC), 2019 (submitted)

Saransh Gupta, Mohsen Imani, Joonseop Sim, Andrew Huang, Fan Wu, and Tajana Rosing, "SCRIMP: A General Stochastic Computing Acceleration Architecture using ReRAM in-Memory Processing", IEEE/ACM International Symposium on Microarchitecture, 2019 (Submitted)

Joonseop Sim, Saransh Gupta, Mohsen Imani, Yeseong Kim, Tajana Rosing, "UPIM: Unipolar Switching Logic for High Density Processing-in-Memory Applications", ACM Great lakes symposium on VLSI (GLSVLSI), 2019.

Joonseop Sim, Minsu Kim, Yeseong Kim, Saransh Gupta, Behnam Khaleghi, and Tajana Rosing. "Multi-bit Parallelized Sensing for Processing in Non-volatile Memory." Non-Volatile Memory Workshop (NVMW). 2019.

Joonseop Sim, Mohsen Imani, Woojin Choi, Yeseong Kim, and Tajana Rosing. "Current-Sensing Efficient Adder for Processing-in-Memory Design." Non-Volatile Memory Workshop (NVMW). 2019.

Joonseop Sim, Minsu Kim, Yeseong Kim, Saransh Gupta, Behnam Khaleghi and Tajana Rosing, "MAPIM: Mat Parallelism for High Performance Processing in Non-volatile Memory Architecture", IEEE International Symposium on Quality Electronic Design (ISQED), 2019.

Alvin Glova, Joonseop Sim, "Electronic device and method for fabricating the same" US Patent App. 10/199,433, 2019.

Joonseop Sim, Mohsen Imani, Woojin Choi, Yeseong Kim, Tajana Rosing, "LUPIS: Latch-Up Based Ultra Efficient Processing-in-Memory System", IEEE International Symposium on Quality

Electronic Design (ISQED), 2018 **(Best Paper Nomination)**

Joonseop Sim, Mohsen Imani, Yeseong Kim, Tajana Rosing, "Enabling Efficient System Design Using VerticalNanowire Transistor Current Mode Logic", IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2017.

Dongjoon Kim, Jaeyun Yi, Joonseop Sim, "Electronic device including switching element and semiconductor memory" US Patent App. 15/333,045, 2017.

Shadi Dayeh, Renjie Chen, Yungoo Ro, Joonseop Sim, "Progress in doping semiconductor nanowires during growth" Materials Science in Semiconductor Processing. 2017.

Joonseop Sim, Seokpyo Song, Jaeyun Yi, "Electronic device and method for fabricating the same" US Patent 9,564,584, 2017.

Joonseop Sim, "Electronic device and method for fabricating the same" US Patent 9,559,145, 2017.

Joonseop Sim, Seokpyo Song, Jaeyun Yi,"Electronic device and method for fabricating the same" US Patent App. 15/075,016, 2016.

Joonseop Sim, "Semiconductor integrated circuit apparatus and method of manufacturing the same" US Patent 9,397,193, 2016.

Joonseop Sim, "Semiconductor integrated circuit apparatus and method of manufacturing the same" US Patent App. 14/946,035, 2016.

YI Jae-Yun, SP Song, JS Sim, "Electronic devices having semiconductor memories" US Patent 9,171,889, 2015.

MS Kim, SG Kim, NK Park, SC Kim, G Sok, JS Sim, HJ Lee, "Semiconductor intergrated circuit device, method of manufacturing the same, and method of driving the same" US Patent 8,917,545, 2014.

JS Sim, JH Son, DW Lee, YH Oh, "Phase-change random access memory device and method of manufacturing the same, and method of driving the same" US Patent App. 13/339,691, 2013.

Joonseop Sim, Jin Shi Zhao, Hyun Ju Lee, Keun Lee and Cheol Seong Hwang, "Characteristics of polycrystalline SrRuO3 thin film electrodes for metal-organic chemical vapor-deposited Pb(Zr0.2Ti0.8)O3 thin films", J. Electrochem. Soc., 153 (11), C777 (2006).

Jin Shi Zhao, Joon Seop Sim, Hyun Ju Lee, Dong-Yeon Park, and , Cheol Seong Hwang, "Investigation of the Deposition Behavior of a Lead Oxide Thin Film on Ir Substrates by Liquid Delivery Metallorganic Chemical Vapor Deposition", Electrochemical and Solid-State Letters, 9

(2) C29-C31 (2006).

Jin Shi Zhao, Joon Seop Sim, Hyun Ju Lee, Dong-Yeon Park, Gyu Weon Hwang, Keun Lee, and Cheol Seong Hwang, "Characterization of PbxPty Alloy Formation on a Pt Substrate during Liquid-Delivery MOCVD of Pb(Zr,Ti)O3 Thin Films", J. Electrochem. Soc., Vol. 153 (5) F81-F86 (2006).

Jin Shi Zhao, Hyun Ju Lee, Keun Lee, Joon Seop Sim, and Cheol Seong Hwang, "Influence of the Pt Top Electrode Annealing Procedure on the Ferroelectric Property of MOCVD Pb(Zr0.2Ti0.8)O3 Thin Films", Electrochemical and Solid-State Letters, 9 (7) F69-F72 (2006).

Jin Shi Zhao, Hyun Ju Lee, Joon Seop Sim, Keun Lee, and Cheol Seong Hwang, "Ferroelelctric and reliability properties of metal-organic chemical vapor deposited Pb(Zr0.15Ti0.85)O3 thin films grown in the self-regulation process window", Appl. Phys. Lett, 88, 172904 (2006).

Zhao, J.S., Joonseop Sim, Lee, H.J., Park, D.Y. and Hwang, C.S., 2005. A study of liquid delivery MOCVD of lead oxide thin films on Pt and Ir substrates. Journal of The Electrochemical Society, 152(5), pp.C277-C282.

ABSTRACT OF THE DISSERTATION

Architecting Non-volatile Memory for High Bandwidth Systems

by

Joonseop Sim

Doctor of Philosophy in Electrical Engineering (Nanoscale Devices & Systems)

University of California San Diego, 2019

Professor Tajana Simunic Rosing, Chair

High-performance processors and emerging deep learning applications require a tremendous amount of data access. Meeting the demands of this bandwidth within the available energy is a big challenge for current memory systems. Processing-in-Memory (PIM) is a promising solution to address this bandwidth bottleneck by performing a portion of computation inside the memory. Existing state-of-the-art PIM technologies implement logic using sensing circuits and cell-based logic-in-memory technology. We present novel designs that improve performance and energy efficiency in each of these two areas.

For the designs using the sensing circuit, we present PIM architecture based on the latch-up effect of thyristors, enabling single-cycle addition, and significantly improving the

performance of multiplication. Our design requires no additional cell array for processing, hence can be an excellent candidate for the storage class memory which has been considered as the main application of memristor-based products. Also, we present the method to carry out multiple bit-line requests under a multiplexer in parallel, dramatically accelerating PIM applications. Our design is 16x and 12.7x faster over state-of-the-art PIM designs [8, 18], respectively.

Considering that cell-based PIM technology is mainly targeted at high-density applications, we present technologies that contribute to energy efficiency and integration. Our UPIM design exploits unipolar switching memristors to offer a sneak current reduction compared to the existing bipolar-based structure and takes advantages of a 3D vertical crossbar array structure to increase memory utilization per unit area for high-density applications. As compared to the state-of-the-art PIM design based on the bipolar switching mode, our design achieves 3.1$\times$ lower energy consumption and 84% area savings. We also extend our cell-based PIM design for very long byte processing by using a multi-block parallelizing method. Proposed design saves 51$\times$ energy consumption and 16% area compared to the state-of-the-art PIM accelerators with a cell-efficiency of 45% that is comparable to commodity DRAM.

Chapter 1

Introduction

Today's tremendous advances in technology have resulted in massive amounts of data being produced. The advent of Internet of Things (IoT) has further increased the amount of data and movement, which has created serious challenges for limited device resources. Today's IoT applications typically process raw data using a machine learning algorithm at the data center. Sending all the data to the cloud is not scalable, does not guarantee real-time response, and is not preferable due to privacy and security concerns. Therefore, a portion of data needs to be processed on the devices at the edge of the internet. However, running data-intensive workloads with large datasets in traditional cores leads to massive data movement between memory and processing units, resulting in high power consumption and slow processing speed. Although new processor technologies have evolved to perform complex tasks using a variety of advanced technologies, they are not sufficient to handle the data throughput required by the IoT system.

A major challenge to large amounts of data processing is memory access. To date, memory technologies have evolved to improve performance by increasing bandwidth, resulting in a bandwidth of hundreds of MB/s in the latest high-bandwidth memory modules [19]. However, future memory products are expected to demand multiple TB/s of memory bandwidth requiring more than the limit of bandwidth that current DRAM devices can provide.

This thesis explores memory technology to overcome the limitations of this extreme data processing requirements. To address the extreme data movement, we first propose novel

processing-in-memory (PIM) technology. Our proposed technologies are based on emerging non-volatile memory (NVM). Although most of our evaluations are based on memristor devices, the proposed techniques can be extended to other resistive memories such as *Spin-Transfer-Torque Random Access Memory* (STT-RAM) and *Phase-Change Random Access Memory* (PCRAM). We thoroughly analyze existing state-of-the-art PIM technologies based on NVMs and identify their limitations. We then propose technologies that can overcome those limitations. We have focused on minimizing the overhead caused by the technologies, along with the effects of the proposed technologies, and demonstrate the novelty of addressing the problems above the architecture with low-level contributions such as circuit and device. We also optimized memory architecture to effectively implement PIM technology as well as PIM logic itself. What distinguishes this study from the previously published research is that it finds hidden parallelism while making full use of resources of commodity memory architectures. The following subsections explain the background needed for understanding the main contributions of this thesis.

1.1 Non-volatile memory (NVM)

Emerging NVM technologies, such as spin-torque-transfer random-access memory (STT-RAM, or MRAM), phase-change random-access memory (PCRAM), and resistive random-access memory (ReRAM) offer fast random access, high storage density, and nonvolatility [1]. They ultimately pursue a universal memory that spans a wide range of highly latency-optimized microprocessor caches to highly density-optimized secondary storage. More detailed description of the characteristics of different NVM technologies follows below.

Spin-torque-transfer random access memory (STT-RAM)

STT-RAM uses a magnetic tunnel junction (MTJ) as memory storage and utilizes changes in magnetic direction to represent information. As shown in Fig. 1.1, MTJ consists of two ferromagnetic layers. One ferromagnetic layer has a fixed magnetization direction and is called a reference layer. Another ferromagnetic layer has a free magnetization direction, called the

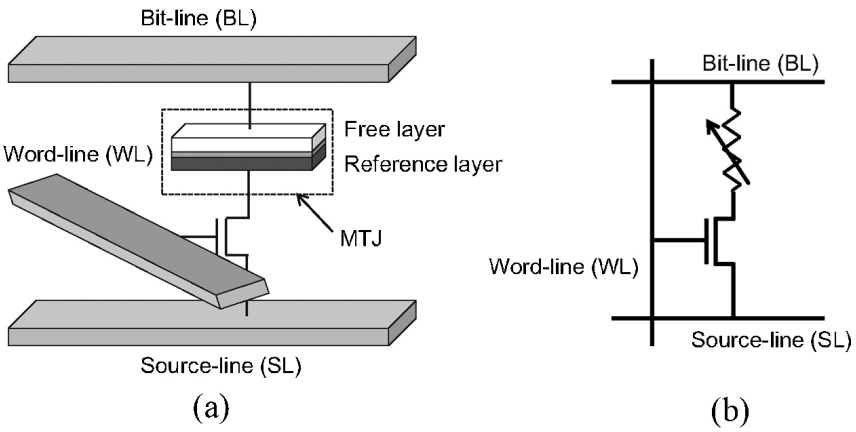


Figure 1.1. Illustration of a STTRAM cell. (a) Structural view. (b) Schematic view (BL = bitline, WL = wordline, SL = sourceline). [1]

free layer, which can change the direction by passing the current. The relative magnetization direction of the two ferromagnetic layers determines the resistance of the MTJ. The relative magnetization direction of the two ferromagnetic layers determines the resistance of the MTJ. If the direction of the ferromagnetic layers is the same, the MTJ has a low resistance and represents a state of '1'. If the directions of the two layers are different, the MTJ has high resistance and expresses a state of '0'. In Fig. 1.1, '0' writing (RESET switching) is performed when a positive bias is applied between SL and BL, and '1' writing (SET switching) is performed in the opposite case. Although the non-volatile nature of STT-RAM has the advantage of lower static power consumption compared to conventional DRAM, its high write power and the large cell size result in poor density [20]. However, compared to PCRAM and ReRAM, STTRAM exhibits much better read / write performance and energy characteristics and also has better write endurance [20]. As a result, STT-RAM is primarily used as SRAM substitute in on-chip cache, not main memory.

Phase change random access memory (PCRAM)

PCRAM uses chalcogenide material (e.g. GST) as memory storage. The chalcogenide material utilizes heat to perform state changes between the crystalline phase (SET state) and the amorphous phase (RESET state). The crystalline phase has low resistance and the amorphous

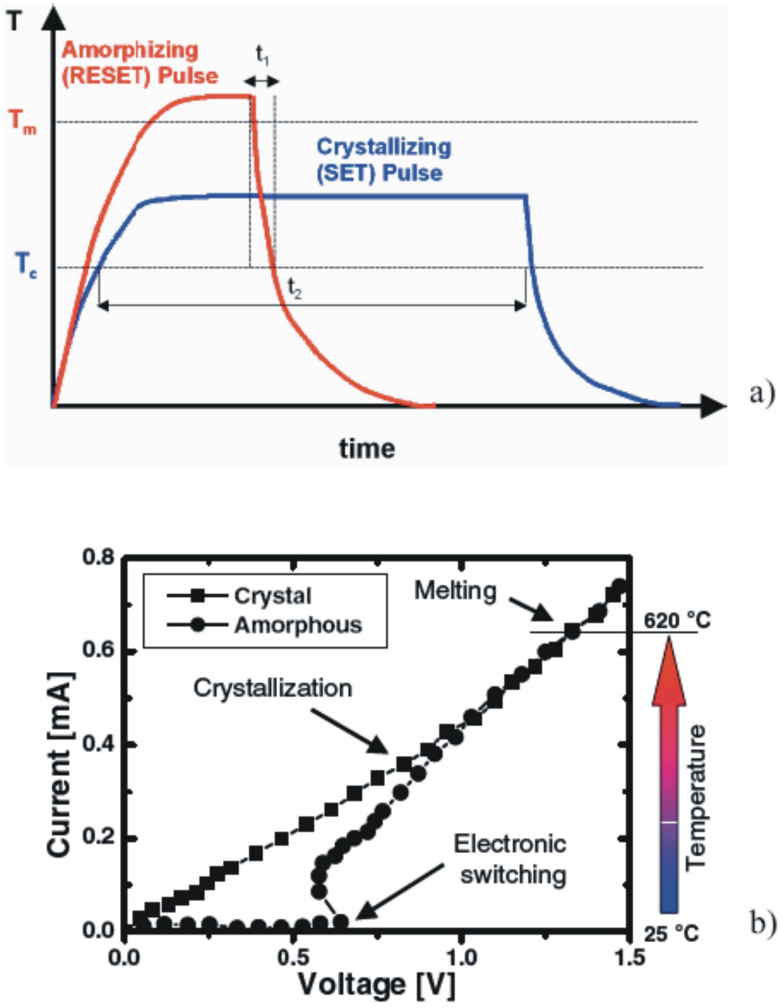


Figure 1.2. (a) Thermal-induced switching of the phase-change material, (b) I-V curves of both the crystalline and amorphous state. [2]

phase has high resistance. As shown in Fig. 1.2(a), SET switching crystallizes GST by applying more heat than crystallization temperature (T_C), and RESET switching converts GST into the amorphous state by melting GST followed by rapid quenching. RESET switching requires a high-power pulse since GST needs to be heated above Tc to melt, and SET switching requires a longer duration pulse with intermediate power between T_m and T_c [2]. Today, PCRAM aims to replace Flash memory. However, its set switching latency, energy consumption, and endurance characteristics are still not enough to replace DRAM in main memory [21].

Resistive random access memory (ReRAM)

Resistive memory (ReRAM) is attracting attention as a basic device for high-performance and high-density applications [22]. ReRAM uses a variety of access devices, but CMOS-based transistors are most widely used to enable fast switching and reduce power consumption [3]. The general structure is the Metal/Oxide/Metal structure as shown in Fig. 1.3. It has metal layers (e.g. Pt) on the top and bottom, and oxide layers based on Tantalum (Ta), Hafnium (Hf), Titanium (Ti) in the middle. The electrical behavior of the metal/oxide interfaces depends on the oxygen vacancy concentration of the oxide layer [1]. In general, the metal/oxide interface exhibits Ohmic behavior in the case of high doping and rectifying behavior in the case of low doping. The oxygen vacancies of the metal oxide layer act as n-type dopants. When the voltage is applied to the memristor cell, the dopant profile changes due to the movement of oxygen vacancies, and cell switching occurs under proper conditions. For example, in case of bipolar memristor, SET switching (OFF $\rightarrow$ ON) occurs when a negative bias is applied to a cell and RESET switching (ON $\rightarrow$ OFF) occurs when a positive bias is applied. Compared to other memories such as STT-RAM and RCRAM, ReRAM is the most promising candidate for the next-generation non-volatile memory due to its simple structure, high switching speed($\sim 1ns$) and high scalability($\sim 10nm$) [23]. Conclusively, PCRAM is the most advanced in the industry such as IBM, Intel, Hitachi and Samsung compared to ReRAM and STTRAM [24].

1.2 NVM-based computation

Most modern computing systems are designed based on the von Neumann architecture, which performs spatial operations on the computation and storage of data. Therefore, data must transfer through a limited bandwidth between processor and memory. This data movement is becoming the biggest bottleneck in modern computing systems in terms of performance and energy. Several approaches have been proposed to address the issue of data movement. Near data computing (NDP) brings the computing unit close to the memory to avoid data transfer

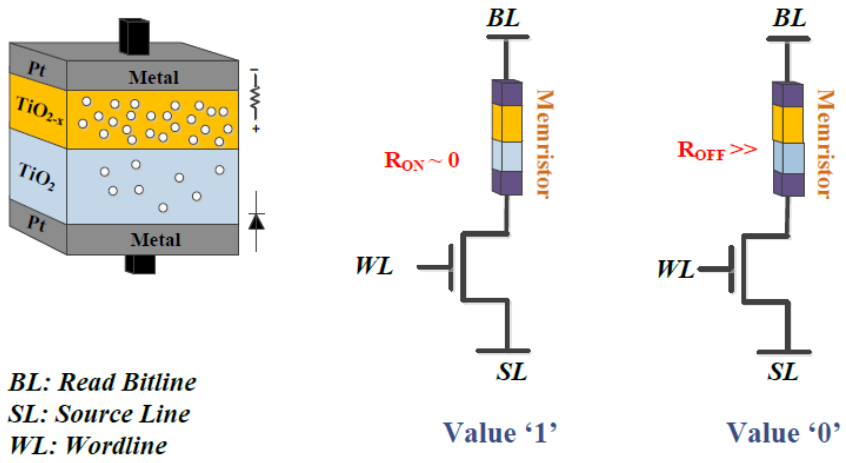


Figure 1.3. The working mechanism of ReRAM structure [3].

across the hierarchy [25, 26]. Such designs need extra processing units near main memory. Some implementations put processing cores in different layers of 3D stacked memories to reduce the data transfer overhead. For example, high bandwidth memory (HBM) uses vertically stacked memory chips interconnected by microscopic wires called "through-silicon vias (TSVs)". However, this increases the energy consumption of the system, while the data still needs to be transferred to the additional processing units.

Processing in memory (PIM) is a promising way to address the data movement issue. Instead of sending all data to the on-chip cache, it offloads a portion of computations near the memory side or directly inside the memory. There have been several proposals to enable computing capabilities inside DRAM. However DRAM is inherently destructive during read operations, that is, the stored bits are invalidated after the read. Thus, the original data should be backed up to another cell before any computation is performed, causing undesired overhead. On the other hand, NVMs are good candidates for PIM due to their high density, scalability, and low power consumption [27, 28]. However, the supported functionality in most of the PIM designs is limited to either bitwise operations or operations derived from basic bitwise operations which require multiple cycles. For example, the work in [29, 9] proposed a sensing circuit to implement the basic bitwise operations such as AND, OR, and INV. However, they

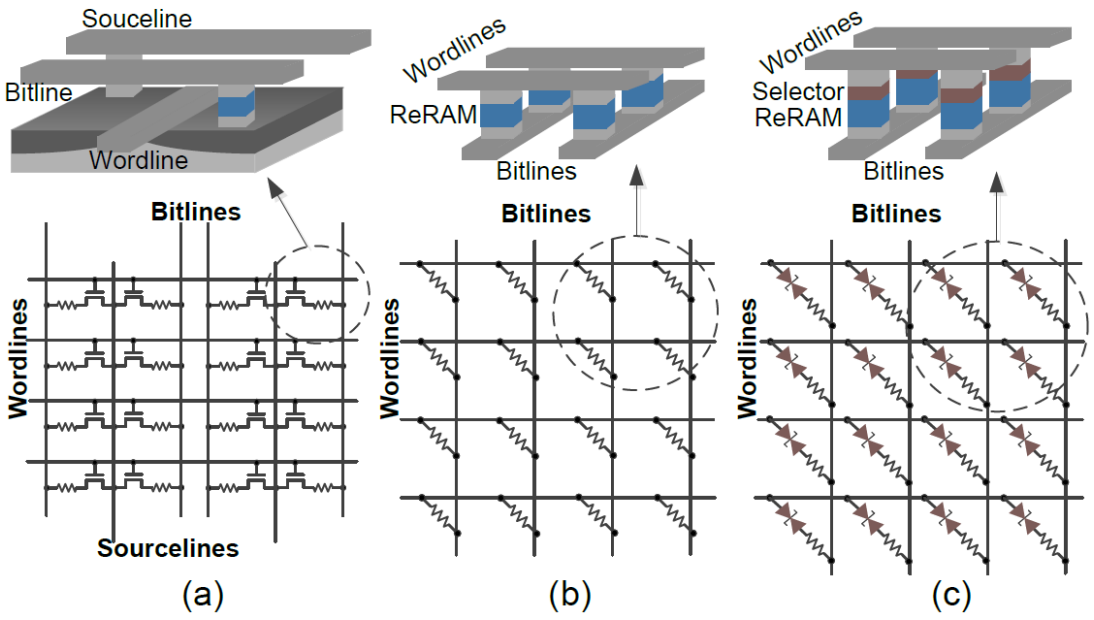


Figure 1.4. Overview of ReRAM array structures: (a) MOSFET accessed structure; (b) access-device-free crossbar structure; (c) diode-accessed crossbar structure [4].

do not support addition and multiplication, which are the key arithmetic functions involved in many applications such as machine learning algorithms and image processing [30]. This thesis identifies problems of existing designs and suggests PIM techniques that improve them. Through Chapter 3 and Chapter 4, we present a new PIM design enabling single-cycle addition, and significantly improve the performance of multiplication. Also, we present a new design offering a sneak current reduction compared to the existing PIM designs and takes advantages of a 3D vertical crossbar array structure for high-density applications.

1.3 State-of-the-art NVM architecture

This thesis also investigates the optimizing method of the architecture environment to achieve maximum performance of PIM function as well as PIM logic. We first examine the current NVM architectures through the following sub-chapters.

1.3.1 Array Architecture

1T1R Structure

ReRAM cells can be organized in 1T1R structure and crossbar structure at the array level. The cell structure is selected considering the energy/delay/cost in the memory system. Fig. 1.4(a) shows the conventional structure in which each cell is connected to one transistor, called '1T1R' structure. This structure is similar to DRAM. When a row is activated, each transistor in the selected row provides exclusive access to the cell without interference from other cells. Due to the complete isolation between the cells by the access transistors, the 1T1R structure is more energy efficient and has shorter operating latency than other structures. However, compared with DRAM, NVM requires high current, hence it requires a large area of the access transistors. This is a critical challenge in the high-density applications NVM is targeting. In addition, since the WL of 1T1R structure is connected to the gate of a transistor that is responsible for turning drivers on and off, PIM operations using analog strategies cannot be performed.

Crossbar Structure

Fig. 1.4(b) and (c) show a crossbar structure in which all cells are connected without access transistors. Due to the absence of access transistors, the crossbar structure has a cell size of 4F2, which is regarded as the smallest area in the two-dimensional structure. In addition, since the ReRAM fabrication is completely different from the silicon-based transistor process, the crossbar structure can maximize the area efficiency by allowing the lower layer of the cell to be used as a peripheral circuit configuration such as decoders or drivers. The crossbar structure is also an optimal structure for PIM design. Its two-terminal node scheme allows floating BLs to be used to propose many PIM designs utilizing the analog characteristics of current passing through the resistors. However, in the case of the crossbar structure, the selected cell is not isolated from the unselected cells, so there maybe various sneak current paths that cause additional power consumption. Also, it is not easy to predict the current path in a large number of cells. To overcome this problem, we implemented PIM logic using unipolar switching mode instead of

bipolar switching, which is presented in Chapter 3.

1.3.2 Bank-level Architecture

Modern processors typically have on-chip memory controllers. Each memory controller is connected to off-chip memory modules via one or two memory channels. According to the JEDEC standard, each channel has a 64-bit data bus, a 17-bit row / column address bus, and an 8-bit command bus [31]. Dual in-line memory modules (DIMMs) can be accessed by a single memory channel and controller. Each DIMM consists of multiple ranks, each rank consisting of multiple chips. We adopt an x8 ReRAM chip configuration with a DDR3-compatible interface and 64-bit internal prefetch width, which is distributed to 32 mats in a subarray of the banks. We used the architecture model of [4] as the baseline bank structure of this thesis. As shown in Fig. 1.5, WL and BL are selected by the WL decoder and BL multiplexer, respectively, and the cell at the intersection of these is selected. Then, the information of the selected cells is moved through the I/O buffer. We used a hierarchical WL structure to minimize delay in cell addressing in high-density cell structure. The hierarchical WL structure is composed of global wordlines (GWLs) using the upper metal layer and lower sub-wordlines (SWLs) using poly silicon. A GWL is selected first by the GWL decoder in Fig. 1.5(a), and a SWL of Fig. 1.5(b) which is directly connected to the cell is selected by the local decoder. In the NVM cell structure, one cell can be selected by applying appropriate potentials to WL and BL connected to the desired cell. Therefore, unlike a DRAM that stores all the cells of one row into a buffer and reads the desired cell, NVM does not have an over-fetch problem. The mat size of our design is 1024 WLs $\times$ 1024 BLs, which is considered the maximum size constrained by parasitic resistance and capacitance.

1.4 Thesis contribution

This thesis explores various ways to maximize performance improvement by addressing the physical and technical limitations of NVM-based PIM designs. We address and improve the problems of existing PIM researches through the range of approaches such as device level

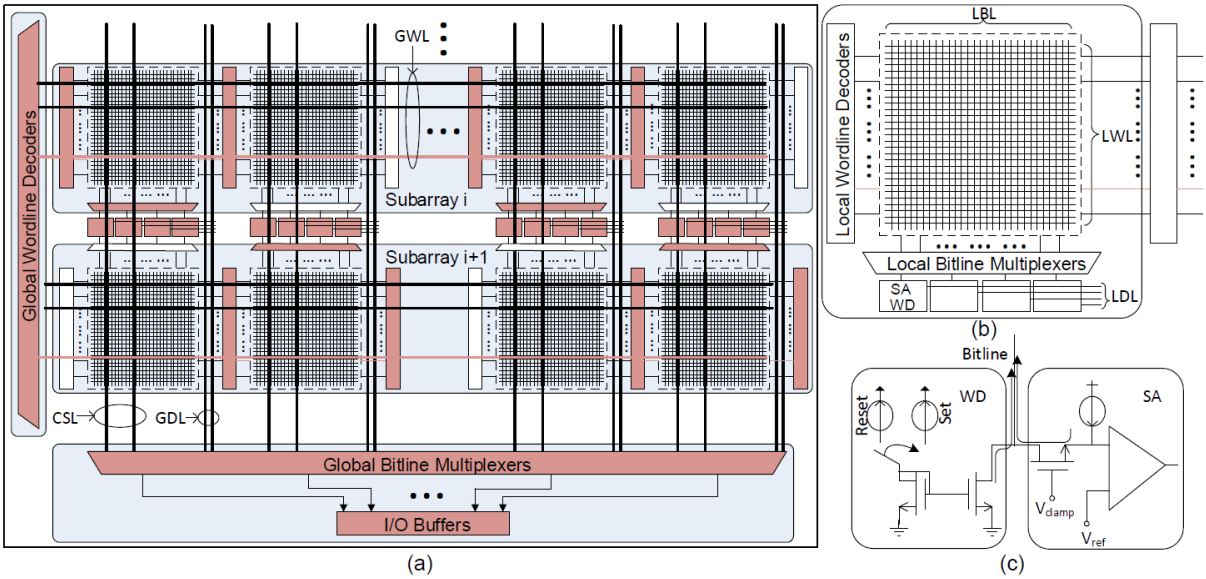


Figure 1.5. (a) Bank structure; (b) Mat organization; (c) Sense amplifier and write driver (CSL: column select line; GWL: global wordline; GDL: global dataline; LBL: local bitline; LWL: local wordline; LDL: local dataline) [4].

as well as an architecture. The following subsections describe our contributions over on the hierarchy of devices, circuits, and architecture.

1.4.1 Device level

Emerging Non-planar Transistor Application

Vertical Nanowire-FET (VNFET) is a promising candidate to succeed in industry mainstream due to its superior suppression of short-channel-effects and area efficiency. However, CMOS is not an appropriate solution for logic gate design due to the process incompatibility with VNFET, which creates a technical challenge for mass production. In this thesis, we present a novel VNFET-based logic design, called *VnanoCML* (Vertical Nanowire Transistor-based Current Mode Logic), which addresses the process issue while significantly improving power and performance of diverse logic designs. Unlike the CMOS-based logic, our design exploits current mode logic to overcome the fabrication issue. Furthermore, we reduce the drain-to-source resistance of *VnanoCML*, which results in higher performance improvement without compromising the subthreshold swing. In order to show the impact of the proposed *VnanoCML*,

we present key logic components including SRAM, full adder and multiplier, and also evaluate the application-level effectiveness of digital designs for image processing and mathematical computation. Our proposed design improves the fundamental circuit characteristics including output swing, delay time, and power consumption compared to conventional planar MOSFET (PFET)-based circuits. Our architecture-level results show that *VnanoCML* can enhance the performance and power by 16.4× and 1.15×, respectively compared to PFET-based designs. Details of this study are described in Chapter 2.

Implementing Unipolar Memristor PIM

Prior techniques [5, 32, 33, 34, 35] that enable the computation in non-volatile memory (NVM) are designed for a bipolar switching mode, which suffers from a high sneak current in a crossbar array (CBA) structure. In this thesis, we present 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. 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 AMD R390 GPU [36]. As compared to the state-of-the-art PIM design based on the bipolar switching mode [5, 6, 7], our design achieves 3.1× lower energy consumption. Details of this study are described in Chapter 3.

1.4.2 Circuit level

Current-based Efficient Adder Design

We also propose *LUPIS* (Latch-Up based Processing In-memory System) for NVM. Unlike existing PIM techniques [9, 29, 32, 35], which mainly focus on bitwise operation based computations and involve considerable latency and area penalty, our design facilitates computations like addition and multiplication with very low latency. This makes the system faster and more efficient as compared to state-of-the-art techniques. We evaluate LUPIS at both

circuit-level and application-level. Our evaluations show that LUPIS can enhance performance and energy efficiency by 62× and 484× respectively as compared to AMD Southern Island GPU, Radeon HD 7970 device [37]. Compared to the state-of-the-art PIM accelerator [8], our design presents 12.7× and 20.9× improvement in latency and energy consumption with an insignificant overhead of 21% for area increase and one cycle for latency delay. Details of this study are described in Chapter 4.

Enabling Multi-bit Parallel Execution in NVM

Many prior studies have enabled various PIM operations on non-volatile memory (NVM) by modifying sense amplifiers (SA) [9, 10, 29]. They exploit a single sense amplifier to handle multiple bitlines with a multiplexer (MUX) since a single SA circuit takes a much larger area than an NVM 1-bit cell. This limits potential parallelism that the PIM techniques can ideally achieve. In this thesis, we present `MAPIM`, mat parallelism for high-performance processing in non-volatile memory architecture. Our design carries out multiple bit-lines (BLs) requests under a MUX in parallel with two novel design components, multi-column/row latch (*MCRL*) and shared SA routing (*SSR*). The *MCRL* allows the address decoder to activate multiple addresses in both column and row directions by buffering the consecutively-requested addresses. The activated bits are simultaneously sensed by the multiple SAs across a MUX based on the *SSR* technique. The experimental results show that `MAPIM` is up to 339× faster and 221× more energy efficient than a modern GPGPU, AMD Radeon R9 390 GPU with 8GB memory [36]. As compared to the state-of-the-art PIM designs [9, 10], our design is 16× faster and 1.8× more energy efficient with an insignificant area overhead. Details of this study are described in Chapter 5.

1.4.3 Architecture level

Excavating Hidden Parallelism using Multi-mat Execution Techniques

Today's emerging computations such as DNN and graph processing require large amounts of data processing with very long words. This inevitably requires an effort to maximize perfor-

mance with limited hardware capacity and energy. We have reduced the cost of data movement through PIM technology and maximized the efficiency of long word operations through the technique of concurrently computing data distributed in several memory mats. In Chapter 6, we present a novel PIM architecture, called Multi-mat Parallelized Execution for Hyper-dimensional Computing (MAPLE).

Hyperdimensional (HD) computing is an energy-efficient brain-inspired alternative to traditional computing. Instead of computing with numbers, HD computing uses hypervectors (HVs) that are high-dimensional (*i.e.* D=10,000). Due to the requirement of massive data access, HD computing needs a memory-centric architecture, which motivates us to utilize processing-in-memory (PIM) architecture. However, the existing memory architecture has physical limitations to fully support the PIM execution for HD computing. To compute over 10,000-bit words corresponding to the dimensions of an HV, tens of memory mats consisting of only 1K bit-rows need to be simultaneously activated. This conflicts with the operation in conventional memory architecture, which executes 8 to 64-bit words, so each mat is designed to be accessed *in sequential order*. Our design works as both memory and PIM within the same cell array. In memory mode, *OpHD* follows the conventional read/write access pattern, with consecutive requests to each mat. In PIM mode, it enables concurrent activation of all mats in a bank using our proposed *global decoder bypass* (GDP), enabling a single cycle operation of 10,000-dimensional HVs. We also propose a *global wordline (WL) search* for the nearest hamming distance, to facilitate a one-shot search for an HV whose bits are located across multiple mats. The experimental results show that *OpHD* is up to 10× faster and 51× more energy efficient than state-of-the-art HD accelerators [11] with an insignificant area overhead. Details of this study are described in Chapter 6.

Chapter 2

High-Performance Design with Vertical MOSFETs

Resistive RAM (ReRAM) has been extensively researched due to its potential for a solution to the scaling issue of charge-trapping based memory [38]. The crossbar structure of ReRAM can realize $4F^2$ which is considered as the minimum cell size in two dimensions. It has also enabled a variety of PIM technologies utilizing the analog property by means of a two terminal access approach [9, 29, 32]. To realize this, the access device needs also be able to implement the $4F^2$ structure. Vertical bipolar junction transistor (BJT) and diode have been attempted as access devices with $4F^2$ density, respectively [39, 40]. However, they exhibit higher leakage currents and are complicated in the process when implemented in CMOS logic. Therefore, the most appropriate way to implement $4F^2$ density is to integrate memory cells into vertically stacked MOSFETs. In this chapter, we design *vertical nanowire field effect transistors* (VNFETs) directly in current mode logic to improve their performance and energy compared to conventional planar-FETs, as well as their area efficiency.

2.1 Introduction

Conventional planar transistors, called PFET, face the challenge in further scaling due to: (i) short channel effects on transistors and (ii) physical limitations of design rules such as complexity of metal routing and shortage of the distance from the metal contact to the gate [41].

To scale down the transistor, the fabrication process should address the inherently shortened distance between a gate and a contact (source/drain). In addition, the degradation of subthreshold swing (SS) should be also addressed due to the increase of the leakage current and latency.

To overcome the scaling issue, three dimensional (3D) gate structures such as FinFET [42, 43] and Nanowire-FET [44] have been proposed. Nanowire FET (NWFET) has many excellent characteristics for scaling such as low power, high density and steeper SS [45]. NWFET can be classified into two groups, called lateral and vertical scheme, according to the physical structure as depicted in Fig. 2.3a. Vertical-NWFET (VNFET) has higher area efficiency and better device performance especially at sub 7nm technology node than the lateral NWFET [46]. Thus, VNFET is considered a promising candidate to succeed in the industry mainstream.

A conventional way to create logic is CMOS-based design. However, since the CMOS logic requires *NMOS* and *PMOS* on a single die, this leads to process incompatibility issue of VNFET, e.g., different dopants of *NMOS* and *PMOS* and gate overlap mismatch [47]. In this chapter, we present a novel VNFET-based logic design, called *VnanoCML*, that mitigates the incompatibility issue by exploiting current mode logic (CML) [48]. Since the CML only uses *NMOS* transistors for logic implementation unlike the conventional CMOS design, the proposed *VnanoCML* scheme highly reduces the fabrication overhead. In addition, utilizing the high density of VNFET, we dramatically decrease the resistance of VNFET so that the proposed design can operate with low supply voltage. The proposed design improves the logic performance and also saves static power consumption which has been considered as an intrinsic limitation of CML. In order to show the effectiveness of our proposed design in diverse architecture layers, we present two key logic designs, SRAM and Arithmetic Logic Units (ALUs), and evaluate additional ASIC designs which exploit the *VnanoCML* logic. In our experiments, we show that the proposed *VnanoCML* improves performance and power consumption significantly for circuit and application levels.

The main contributions of this work are listed as follows:

- To the best of our knowledge, this is the first design which addresses the process compatibility of VNFET devices by employing CML logic which uses only NMOS transistors.

- Our *VnanoCML* design resolves the power penalty of conventional CML circuits by using the multiplication of VNFET which reduces supply voltage for a given operation current and makes *VnanoCML* more efficient.

- Our experimental results show that for the same operating condition, *VnanoCML* can provide 16.4× speedup and 1.15× lower power consumption as compared to PFET-based logic. In addition, we show that the proposed design can be also a good candidate for approximate computing.

2.2 An overview of nanowire

The visionary works of Wagner and Ellis in the 1960s have predicted the evolution of microwire growth and their doping when they stated "Controlled growth can be obtained through appropriate use of impurities in patterns of films on substrate surfaces and on single-crystal seeds of many substrates. P-N junctions and heterojunctions can be made" [49]. In the years that followed, Si[50], Ge[51], and their axial heterostructures[52], as well as compound semiconductors such as GaAs, GaP, InAs [53, 54], and their alloys of InGaAs, GaAsP have already been realized in the period of 1960-1980. The first device on a 1D nanowire (NW) was fabricated in 1991 by Haraguchi et al. who formed p-n junctions from GaAs whiskers grown using selective area metal-organic chemical vapor deposition [55]. A global renaissance in NW growth occurred in 1998 when A.M. Morales and C.M. Lieber utilized VLS to prepare Si and Ge NWs with diameters of 3-20 nm and lengths of 1-30μm [56]. This led to fascinating progress in the formation of axial [57] and radial [58] heterostructure growth with efficient size [59] and alloy [60] bandgap engineering enabling new frontiers in crystal engineering [61], and applications in diverse areas such as electronics, photonics, thermoelectrics, chemical sensing, biosensors,

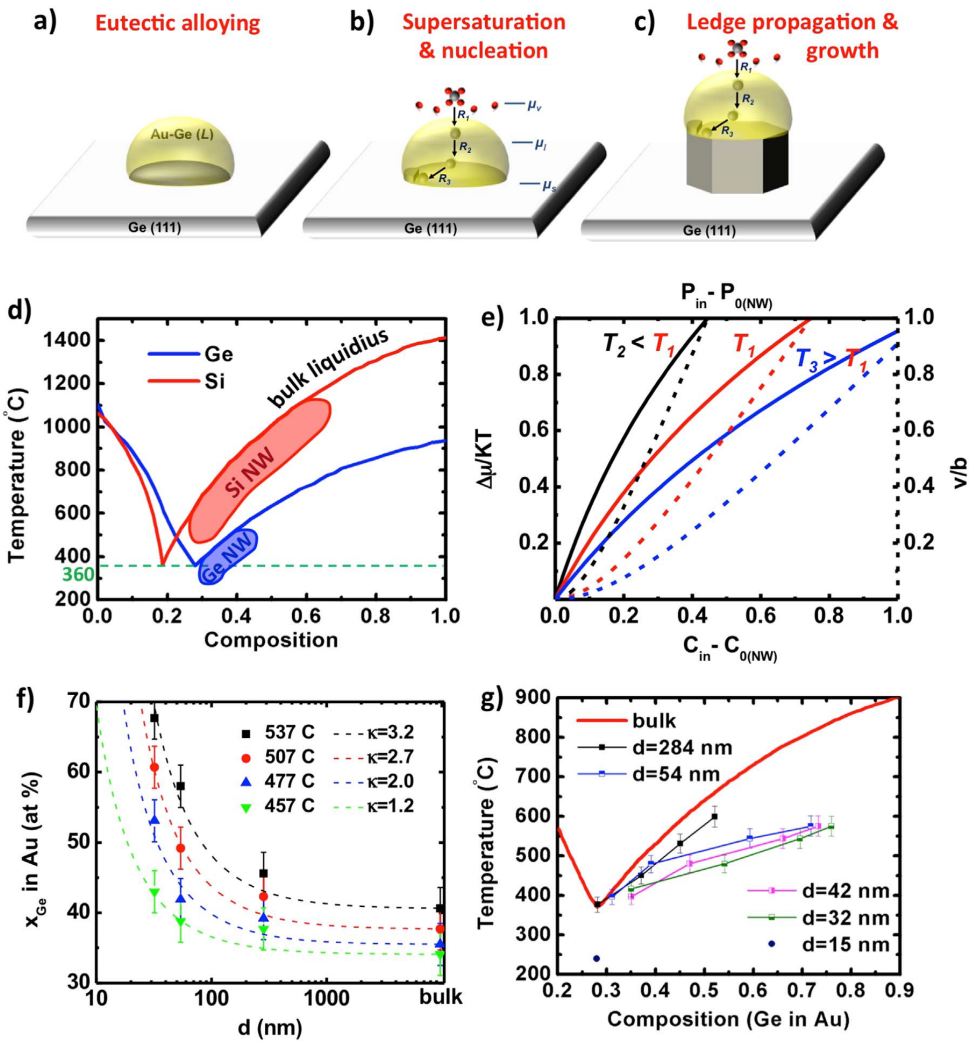


Figure 2.1. (a)-(c) VLS NW growth. (d) Bulk phase diagram (e) NW growth rate (f) Ge composition in Au (g) Suppression of the liquidius line. Copyright American Chemical Society.

and biostimulants to name a few. Semiconductor NWs can be synthesized using a variety of growth techniques including organo-metallic vapor phase epitaxy (OMVPE) [62], selective area OMVPE (SA-OMVPE) [63], molecular and chemical beam epitaxy, wafer annealing, chemical vapor deposition (CVD), laser ablation, and low temperature solution methods. In the presence of foreign catalytic metal particles that seed NW growth, their evolution is interpreted to occur via the Vapor-Liquid-Solid (VLS) growth mechanism. In the absence of foreign metal particles, their growth is interpreted via group-III catalyzed VLS growth, oxide-assisted growth, ligand-aided solution-solid (LSS) growth, reactive Si-assisted growth, and dislocation-driven growth.

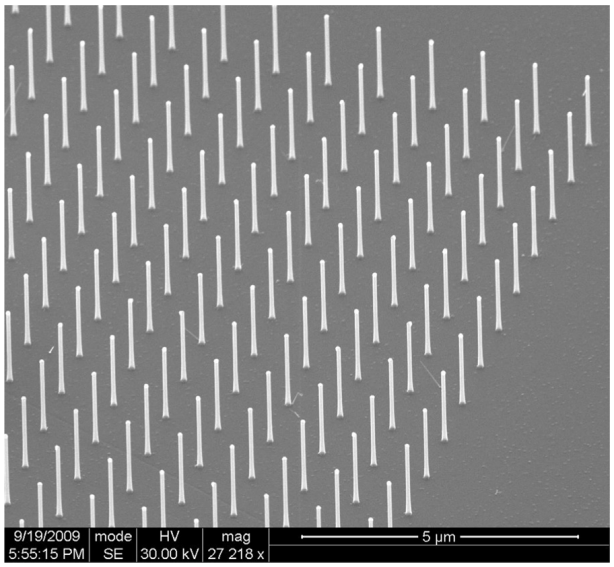


Figure 2.2. SEM image of an array of Ge NWs (d=70nm) grown epitaxially on a Ge(111) substrate from a Au-Ge alloy particle.

The most applicable growth mechanism for semiconductor NWs is the VLS growth mechanism that is briefly overviewed in Fig. 2.1(a)-(c). It is centered around the formation of a supersaturated eutectic droplet that facilitates the NW growth in a layer-by-layer fashion. The prerequisite conditions for epitaxial NW growth include the formation of liquid solution between the metal seed and the material to be grown (Fig. 2.1a) and must usually have a large contact angle (95-120°) with the growth substrate to enable its rise above the surface. The chemical reactions, whether homogeneous or heterogeneous, should be thermodynamically possible but not favored kinetically so that the catalytic and adsorption properties of the liquid growth seed become effective in precursor decomposition and incorporation (Fig. 2.1b) and consequently in the one-dimensional NW growth. In addition, high supersaturations in the growth seed are required especially in the initial stages of growth to enable its rise above the substrate surface, which must be oxide free to enable the vertical epitaxial growth. In steady state, the rate of adatom incorporation at the liquid-solid interface (Fig. 2.1b, R3) is equal to the rate of precursor decomposition and incorporation at the vapor-liquid interface (Fig. 2.1b, R1). The diffusion through the liquid particle (Fig. 2.1b, R2) is not rate limiting. The above requirements determine

the temperature ranges over which NW growth is preferred, which are typically 100-200°C lower than those used in thin film growth. Within the NW growth temperature window, low temperatures may not allow the formation of the liquid growth seed alloy and oxide desorption, whereas high temperatures increase the material solubility in the liquid nanoparticle (NP) and reduce its supersaturation and consequently the NW nucleation and growth rates. Fig. 2.2 shows a scanning electron microscope (SEM) image of an array of Ge NWs with 70 nm diameter grown epitaxially by CVD on a Ge(111) substrate from a 20 nm thick pre-patterned Au disc by electron beam lithography. The experimental growth regimes for Ge and Si NWs are highlighted under the liquidius line of the bulk Au-Si and Au-Ge binary phase diagram in Fig. 2.1(d).

The driving force for growth, supersaturation $\Delta\mu = \mu_v - \mu_s \cong \mu_l - \mu_s$ is the change in the chemical potential during the phase transformation from the liquid droplet to the solid NW. If Φ is the thermodynamic potential, and N is the number of atoms in the crystal during growth, then $\Delta\mu = \delta\varphi/\delta N$. One can compute the changes in $\delta\varphi$ in terms of the variation of the system free energy $-\pi r 2\delta L\Delta\mu_0/\Omega$ where r is the NW radius, δL is the incremental change in the NW length, assuming a cylindrical NW, Ω is the atomic volume of the NW material in the growth seed, and μ_0 is the supersaturation without surface considerations, i.e., in the bulk limit. For every δL in the NW, there is an increase in the surface free energy $\pi r\delta L\alpha_{vs}$ where α_{vs} is the NW surface energy density. Therefore, $\delta\varphi = -\pi r\delta L\Delta\mu/\Omega + 2\pi r\delta L\alpha_{vs}$ and with $\delta N = \pi r 2\delta L/\Omega$, one can therefore write:

$$\Delta\mu = \Delta\mu_0 - \frac{4\Omega\alpha_{vs}}{d} \quad or \quad \frac{\Delta\mu}{kT} = \frac{\Delta\mu_0}{kT} - \frac{4\Omega\alpha_{vs}}{kT}\frac{1}{d} \tag{2.1}$$

which is pressure and temperature dependent ($\Delta\mu_0$) and inversely proportional to diameter d. Alternatively, one can utilize the Gibbs-Thompson effect for the rise of partial pressure at small sizes according to [64]

$$P_0 = P_\infty e^{\frac{4\Omega\alpha_{vl}}{kT\cdot d}} \approx P_\infty e^{\frac{4\Omega\alpha_{vs}}{kT\cdot d}} \tag{2.2}$$

to reach Eq. 2.1 by substituting $\Delta\mu = kTln(P_{in}/P_0)$ where P_0 and P_∞ are the equilibrium partial pressures of the grown material in bulk and NW forms, respectively, and P_{in} is the gas precursor input partial pressure.

If the adatoms have negligible diffusion along the NW sidewalls, the axial growth velocity can be empirically related to supersaturation with $v = b(\Delta\mu/kT)^2$ where b is a temperature independent kinetic coefficient of crystallization [42,43]. Thus, using Eq. 2.1, one can write an equation of the growth velocity as a function of supersaturation as

$$\sqrt{v} = \sqrt{b}\frac{\Delta\mu_0}{kT} - \sqrt{b}\frac{4\Omega\alpha_{vs}}{kT}\frac{1}{d} \tag{2.3}$$

from which $L = t \cdot b \cdot (\Delta\mu_0/kT - 4\Omega\alpha_{vs}/kTd)^2$ can be obtained.

Using these equations, one can summarize the thermodynamic dependencies on the growth in Fig. 2.1(e), which shows enhanced supersaturations and normalized growth rates for higher overdrive pressures and lower temperatures.

One can also deduce from Eq. 2.2 that the phase diagram in Fig. 2.1(d) does not apply to NW growth where for a given temperature, the equilibrium compositions are higher in NWs at smaller d compared to their bulk counterparts. Using in-situ transmission electron microscopy (TEM) heating experiments, one can quantify the equilibrium composition of the growth seed in the absence of any input precursor [65]. Indeed, the experimental measured data follow qualitatively Eq. 2.3 or its composition equivalent by invoking Henry's law to obtain,

$$C_0 = C_\infty exp(k4\Omega\alpha_{vs}/dkT) \tag{2.4}$$

where k is a fit parameter that accounts for the triple-phase curvature changes with temperature. This force-balance at the triple-phase boundary is governed by $\sigma_l cos\beta = \sigma_s cos\alpha - \sigma_{ls} - \tau/r_0$ where β is the contact angle at the solid-liquid interface, α is the angle between the liquid-solid interface and the sidewall of the nanostructure (α=90° for NW), σ_l, σ_s, and σ_{ls} are the

liquid-vapor, solid-vapor, and liquid-solid surface energy densities, τ is the droplet surface line tension, and r_0 is the radius of its contact cross-section or the NW radius. τ/r_0 is negligible compared to the other terms [66]. In the Au-Ge material system, σ_{ls} (460 °C)=0.319 J/m^2 and σ_{ls} (584 °C)=0.211 J/m2 while l (460 °C)=0.798 J/m^2 and σ_l (584 °C)=0.765 J/m^2. This indicates a smaller contact angle with higher temperatures (higher compositions) which in this case changes from 113.56° at 460-106° at 584 °C. This trend is qualitatively in agreement with what is observed experimentally in Ge NWs, where the composition is fit using Eq. 2.4 in Fig. 2.1(f) and the NW phase diagram relative to bulk is constructed in Fig. 2.1(g). It is clear that the liquidius line is suppressed compared to that of bulk [67]. Liquid Au-Ge particles have been observed at temperatures that are 120 °C lower than the bulk eutectic temperature, signifying these effects and the applicability of the VLS mechanism at deep sub-eutectic temperatures.

2.3 Related Work

The innovation of the transistor structure has led to performance improvements. The FinFET structure, which takes up three-dimensional gates, exhibits better static noise margin at lower supply voltage [68, 69], and has been successfully commercialized in industry. As the next generation of technology, NWFET, also known as Gate-All-Around (GAA) FET, is the most promising candidate for sub 10nm scale due to its excellent electrostatic property. Device-level characterization of NWFET has been widely investigated in [70, 71], demonstrating better suppression of the short channel effects than FinFET and PFET. For example, [71] compares NWFET to FinFET, and presents that NWFET shows improved SS and reduced Drain-Induced-Barrier-Lowering. Prior research has studied characteristics of VNFET as a future transistor for sub 7nm, since VNFET has a significantly higher density than lateral NWFET [46].

Despite of the superior characteristics of VNFET, it has not been used for circuit/system design because of not being compatible with mass production. A nanowire fabrication method suitable for the mass production is *top-down etching process* as a counterpart of *bottom-up growth*

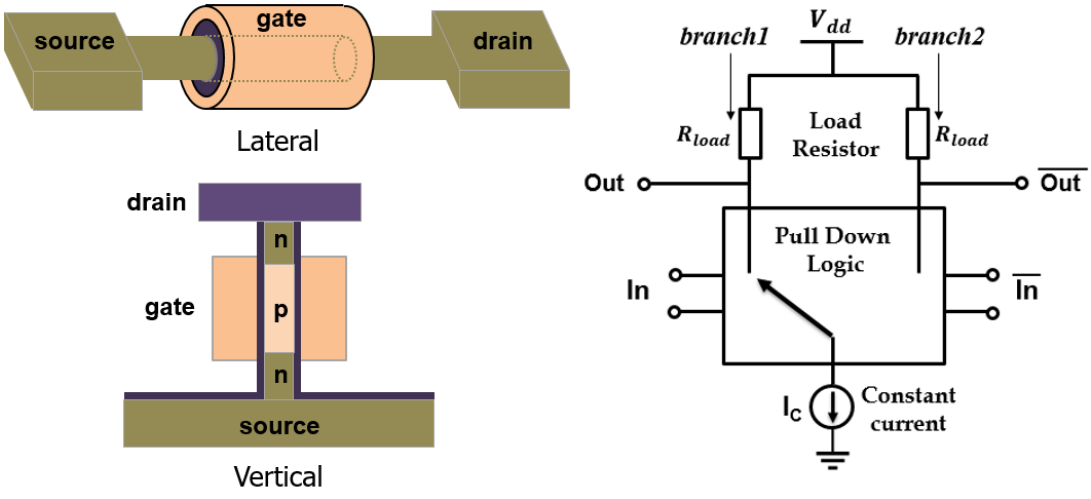
process, since the *top-down method* can yield uniform dimension, better alignment between layers and shorter process time [72]. However, to produce the CMOS logic based on VNFET, the most challenging issue is junction formation due to harsh implantation process and the high variation during subsequent dopant activation [47]. Since the CMOS logic requires both *NMOS* and *PMOS*, each doping step which creates a junction should be processed separately. Inevitably, for processing the gate formation, this incurs gate overlap mismatch to channel regions between N/PMOS transistors. Worse still, the mismatch is more severe since the dopant of each *N* and *P* has different diffusion length variations.

To address this issue, we present a novel design which allows process compatibility of VNFET devices. Instead of using the CMOS-based design, we exploit current mode logic which only requires NMOS-type transistors. Since our design utilizes high density of VNFET, the proposed *VnanoCML* improves the performance of circuits significantly and also mitigates the power issue which has been considered as the main limitation of CML.

2.4 Proposed *VnanoCML*

2.4.1 Current mode logic

In our design of *VnanoCML*, we utilize current mode logic, in short CML, to address the fabrication issue of CMOS-based design. Current mode logic is a technology to construct integrated circuits. Unlike CMOS circuits, CML requires only *NMOS* transistors to build logic. This fact significantly mitigates the process incompatibility which happens in CMOS-based design. As discussed in Section 2.3, using complementary logic creates two major fabrication issues, gate overlap mismatch and dopant variation. However, when producing one type of transistor, i.e., *NMOS*, only a single doping step is required for junction formation, and the subsequent gate process does not incur gate mismatch variations since the same dopant can be exploited for all transistors. The performance of CML is in general better than the complementary logic since the output swing voltage is lower, thus providing faster switching speeds [48].



(a) NWFET Structures (b) Current Mode Logic Schematic

Figure 2.3. Nanowire-FET structures and current mode logic schematic

To better describe these advantages, Fig. 2.3b shows a simple CML schematic. A CML consists of three key parts: (1) load resistance (R_{load}), (2) constant current source (I_c), and (3) pull-down logic designed using *NMOS* transistors which handle the inputs (In and $\overline{In}$). The CML operates based on current differentials of the branch pairs which corresponds to an output voltage, i.e., Out and $\overline{Out}$. For example, when current flows through branch 1, the electric potential of Out is $V_{dd} - \Delta V$ where $\Delta V = I_c \times R_{load}$, while the other branch keeps $\overline{Out}$ by V_{dd}. Since the output voltage swing of a CML, i.e., ΔV, is less than V_{dd}, the dynamic power consumption $P = C\Delta V^2 f$ is lower and the switching speed is consequently faster than a CMOS logic which produces the full swing range of V_{dd}.

One known issue of CML is the relatively high static power consumption, since the current of a constant amount, I_C, keeps flowing through at least one branch during the operation, consuming $P_{static} = V_{dd} \times I_C$. A number of strategies have been proposed to overcome this issue. For example, [73] presented a new circuit design, called near-threshold circuits, which operates in a region of low voltage. However, this approach makes two subsequent problems for PFET-based CML. First, as the supply voltage lowers, the voltage difference between 1 and 0 becomes smaller. Since an output swing range of CML is narrow, the small voltage difference is

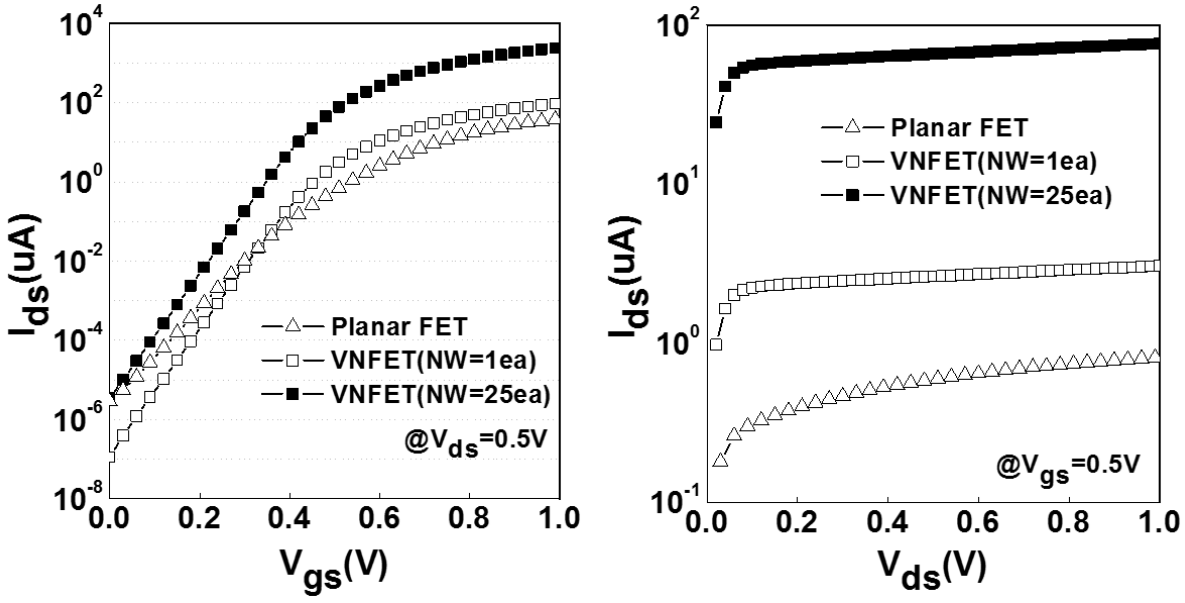
difficult to distinguish in sensing circuits. Furthermore, since NMOS transistors of the pull-down logic are serially connected to each other, the potential at output node, i.e., $V_{dd} - \Delta V$, is split into each transistor. Hence, the applied voltage to each transistor is insufficient to operate them in the device saturation region.

2.4.2 CML integration with VNFET

The two issues discussed above are mitigated in our proposed *VnanoCML* due to VNFET characteristics. Fig. 2.4a illustrates the comparison of SS between PFET and VNFET, while the drain-to-source voltage is at 0.5V. V_{gs} is the applied voltage to the gate of a transistor, and I_{ds} is the current from drain to source. When using one nanowire (NW) for a VNFET transistor, the SS of the VNFET is steeper than PFET. The high on-off current ratio coming from the better SS allows CML to have a higher differential capability. Moreover, VNFET can run at relatively small voltage due to its low resistance. Fig. 2.4b shows $I_{ds} - V_{ds}$ characteristics for different drain-to-source voltages, V_{ds}, when the V_{gs} is 0.5V. Compared to the PFET, we can obtain enough I_{ds} to operate in CML, since the resistance of VNFET is smaller.

In our *VnanoCML* design, we further utilize the characteristics of VNFET so that logic is more compatible with CML. The main advantage of VNFET is its high density. Thus, for the same area of the conventional PFET, we can implement multiple NWs (MNW) to provide smaller resistance, i.e., higher operational current at a given voltage. This provides various advantages, e.g., better output swing, reduced delay, and less power consumption. For example, Fig. 2.5 shows the comparison of PFET and the MNW-based VNFET design. In our experimental setup, that uses *GPDK*45 PEFT model, 25 NWs can be implemented in the area taken by a single PFET. In the integration of CML logic, we create a serial connection in the pull-down logic. Fig. 2.6 illustrates an example structure which integrates two transistors in top and vertical views. A MNW is connected to a silicon (Si) and a metal that form the source and drain respectively[1].

[1]Schottky contact issue can be eliminated by either the silicide process or appropriate metal selection whose work-function is similar to silicon ($\simeq$4.05eV) [74, 75].



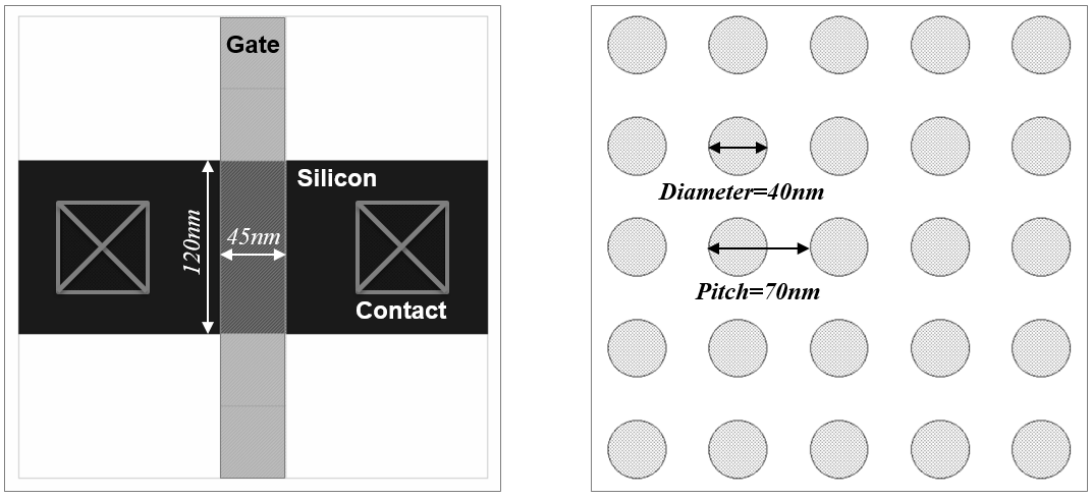
(a) V_g-I_d (b) V_d-I_d

Figure 2.4. Transfer curve analysis of PFET and VNFET

The MNW is surrounded by a gate in the channel region. As shown in Fig. 2.4b, the MNW with 25 nanowires can drop drain-to-source resistance (R_{ds}) by 1/25. In addition, the increase of the number of nanowires does not change the characteristic of SS. The significant increase in the operation current makes *VnanoCML* more efficient due to two main advantages, i) higher output swing in CML, and ii) higher speed by the general characteristic of integrated circuits, $T \propto 1/I_C$ where T is the cycle period.

2.4.3 *VnanoCML* SRAM and ALU

***VnanoCML* SRAM** A single CMOS SRAM consists of two inverters and two *pass* transistors [76]. In our *VnanoCML* SRAM design, we replace all the six transistors of the CMOS SRAM with the *VNFET* transistors. Fig. 2.7a shows the design of a *VnanoCML* inverter. As discussed in Section 2.4.2, since the proposed *VnanoCML* has steep SS and high operation current due to the wire multiplication, the SRAM which uses the *VnanoCML* inverters has high differential ability, i.e., high static noise margin. In Section 2.5.2, we evaluate the static noise



(a) PFET (b) VNFET

Figure 2.5. Layout comparison between PFET and VNFET

margin of the proposed SRAM design in detail.

***VnanoCML* ALU** Fig. 2.7b and c show the design of a *VnanoCML* one-bit full adder (FA). The one-bit FA has two components, sum and carry-out. Fig. 2.7 shows the design of each part based on *VnanoCML*. In the pull-down logic, three stacked transistors and current source I_c are serially connected. For both sides of sum and carry-out, 300KΩ resistors are loaded into the pull-up network (denoted R) to balance the resistance of the pull-down logic. Based on the one-bit FA, we designed an n-bit FA and an n-bit multiplier. For the n-bit FA, we serially put the one-bit FAs as shown in Fig. 2.7d. Fig. 2.7e illustrates the n-bit multiplier design which uses a shift logic gate beside an n-bit FA. Given two n-bit operands, the shift gate takes each bit of the first operand and produces n bits of either shifted n bits of the second operand or all 0s. Then, the n-bit FA accumulates the output of the shift logic to compute the final output. We verify the detailed operation of the designed *VnanoCML* ALUs with a comparison with PFET-based CML in Section 2.5.2.

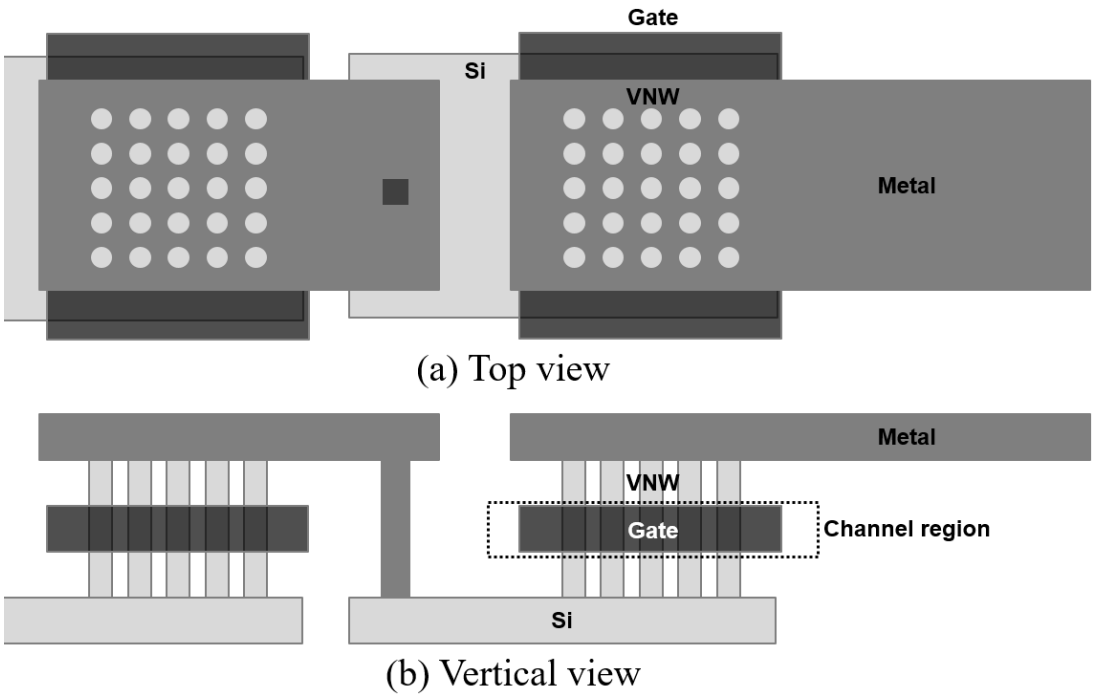


Figure 2.6. VNFET structure used in *VnanoCML*

2.4.4 Application design using *VnanoCML*

Since the *VnanoCML* circuits are compatible with CMOS-based logic in terms of the logic functionalities, *VnanoCML* can be easily integrated with various system components such as memory and processors. In addition, as discussed in Section 2.4.1, the use of CML mitigates the fabrication issues, and thus the *VnanoCML* can be a practical and viable solution for general architecture designs. To investigate how the *VnanoCML* circuits perform in the architecture level, we design *VnanoCML*-based ASICs. For many ASIC designs, which mostly compute and assimilate a stream of data adder, the adders and multipliers are the main building blocks. In addition, more complex arithmetic computation, e.g. square root, also can be approximated using the blocks. Thus, we replaced the logic gates of the ASICs using *VnanoCML* ALUs.

An important aspect that we have also considered for designing circuits is process variation which most of today's technology suffers. In small feature size, process variation degrades the stability of the design and increases the failure rates. To avoid the impact of process

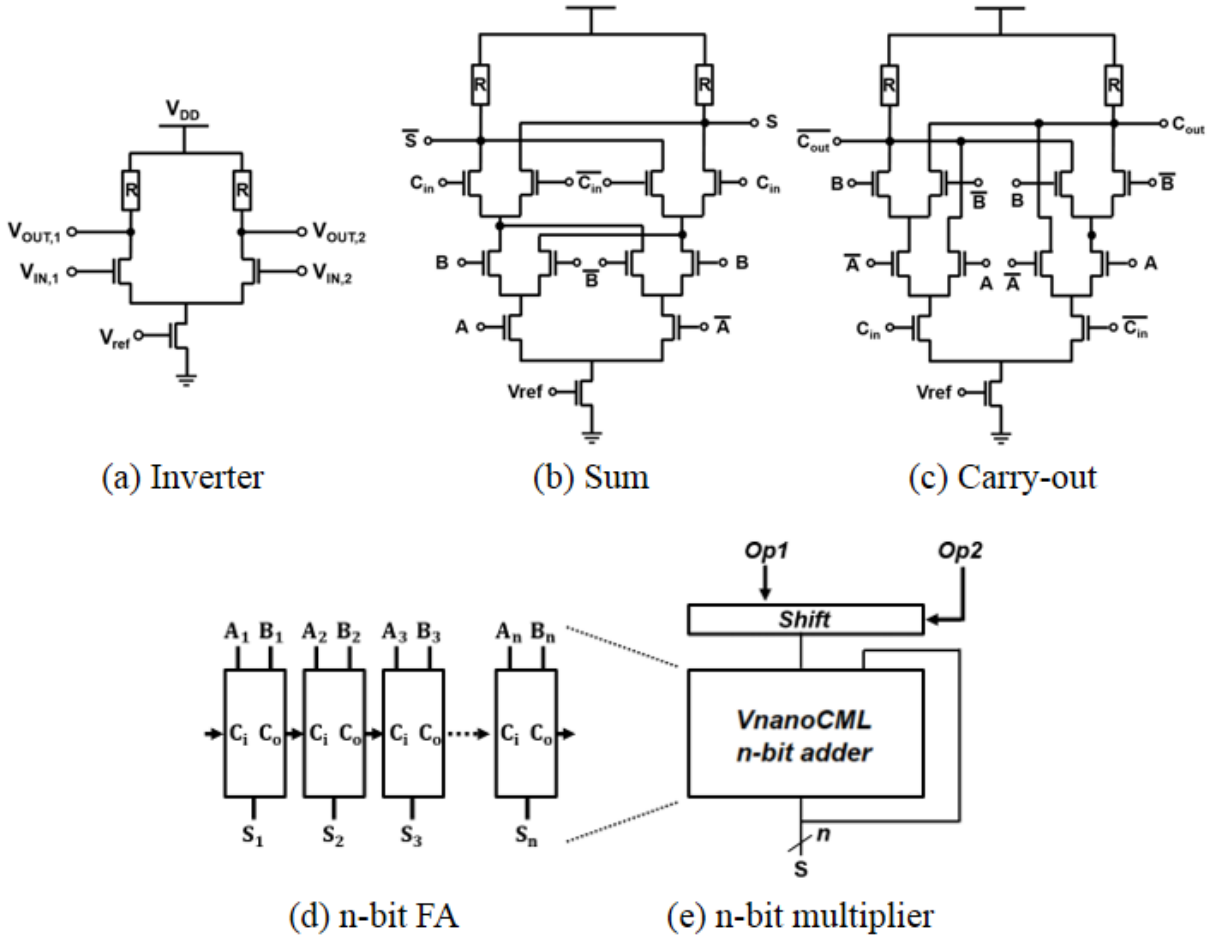


Figure 2.7. VNANOCML ALU design

variation on computation accuracy, designers consider the variation bounds to guarantee the correct functionality even in the worst-case scenarios. To take explicitly into account the process variation, we use Monte Carlo simulation with 10% Gaussian distribution ($3\sigma = 10\%$) on the transistor gate length and diameter. In circuit and logic structures, the output signals are sampled in a specific moment defined by the clock frequency. Thus, given the process variation, we set the clock frequency, f_{ref}, so that it guarantees the correct functionality even for the circuit which has the longest output delay.

Note that, increasing the clock frequency above f_{ref} may result in incorrect signal sampling for a part of circuits, thus degrading the accuracy of application outputs. However, for error-tolerant applications, such accuracy degradation would be acceptable and compensated

by the speedup. For example, in multimedia and vision applications, the accuracy is limited by the human ability to perceive and respond. In addition, there are some applications which are stochastic in nature, e.g., machine learning algorithms [77, 11]. To explore this feasibility of application approximation, in Section 2.5.3, we also evaluate the ASIC designs by relaxing the design constraint.

2.5 Experimental Results

2.5.1 Experimental Setup

We used *BSIMCMG* model [78] for VNFET and *GPDK*45 model [79] for the conventional PFET to estimate power and performance. We compute performance and energy consumption of the proposed design from circuit-level simulations with Cadence Virtuoso and Spectre simulators. For both technologies, we use gate length of 45nm. For a fair comparison, the gate width and diameter of VNFET is set to 120nm and 40nm respectively. In this configuration, a *VnanoCML* transistor can have 25 NWs at most as discussed in Section 2.4.2. One minor issue of *VnanoCML* fabrication would be the formation of multiple NWs since the edge line of an NW array may not be well-formed due to dry etch damage caused by the reactive ion etching process. However, several solutions such as inserting the sacrificial wire and optical proximity correction method [80] can highly minimize the damage. Moreover, even considering a severe case that the last rows of NW array are damaged, only around 15% of the resistance degradation occurs, still providing sufficiently lower resistance compared to PFET-based design.

We evaluate the efficiency and functionality of the proposed *VnanoCML* circuits, compared to the PFET-based CML logic. In order to better show the practical value of the *VnanoCML* design, we also experiment with four ASIC designs running different applications: *Sobel, Robert, Blackscholes* and *FFT*. For image processing applications (*Sobel* and *Robert*), the input data have been randomly chosen from Caltech 101 Library [81]. For the other applications (*Blackscholes* and *FFT*), the input data are given by streaming randomly generated data. Each ASIC design

has been implemented using the *VnanoCML* circuits in Verilog RTL. We extract performance, switching activity and accuracy of each application during post-synthesis simulations with *ModelSim*.

2.5.2 Circuit-level efficiency of *VnanoCML*

As discussed in Section 2.4.1, one technical challenge in using CML is that it has lower output swing than CMOS logic, making voltage sensing difficult. To understand how the proposed *VnanoCML* exhibits differential ability for the output voltage swing at the circuit level, we first evaluate the static noise margin (SNM) of *VnanoCML* SRAM described in 2.4.3. Fig. 2.8a shows that *VnanoCML* SRAM has higher SNM than the circuits using PFET for all the evaluated range of supply voltage. To explain the SNM difference in detail, Fig. 2.8b and 2.8c demonstrate the transfer curves of PFET-based CML and the *VnanoCML* SRAM, respectively. The results show that the PFET-based SRAM circuit has no SNM with V_{dd}=0.5V. In contrast, a *VnanoCML* SRAM shows sufficient SNM for the same V_{dd}, and the SNM is still acceptable even at V_{dd}=0.4V.

We also verify the differential ability of the *VnanoCML* full adder. Fig. 2.9 presents the transmission waveforms of the carry-out circuit. For all simulations we use the input waveforms illustrated in Fig. 2.9a. In order to quantify the sensing ability of different design approaches, we define a distance between the lowest upper and highest lower signal as *Output Swing* (*W* denoted in Fig. 2.9b). The larger the output swing, the better their differential ability. As shown in Fig. 2.9b, PFET-based circuit shows the output swing of 4.63mV at V_{dd}=0.4V. In full adder design which uses one NW, it shows a slightly better output swing of 16.25mV for the same V_{dd}. In our *VnanoCML* design which uses 25 NWs, the output swing is 344mV for the same V_{dd}, i.e., guaranteeing sufficient sensing resolution.

Fig. 2.10 shows the comparison of PFET-based CML and *VnanoCML*. Fig. 2.10a summarizes the results of the output swing. The results show that to achieve the same output swing of 100mV, the available supply voltage V_{dd} are 0.58V and 0.47V, and 0.36V for the PFET-based,

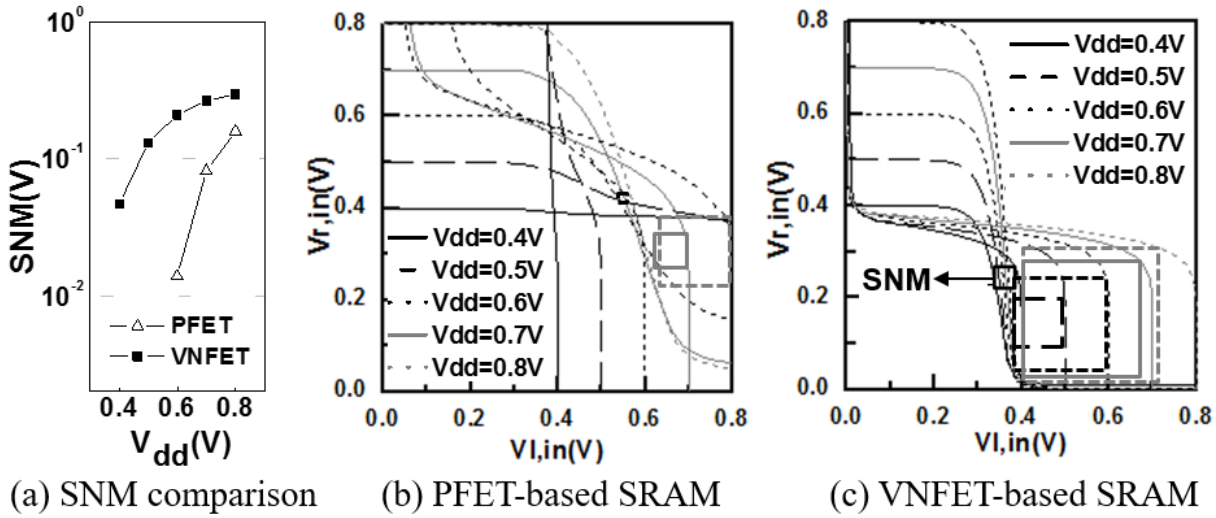


Figure 2.8. SNM comparison of CML-based SRAM

one NW-based, and *VMTCML* circuits, respectively. In Fig. 2.10b, we compare the constant current, I_c, for PFET-based CML and *VMTCML*. The proposed adder has higher I_c value than PFET-based adder due to its lower drain-to-source resistance. Note that, in CML circuits, the performance increases as the current grows. Thus, this result implies that the *VMTCML* design exhibits better performance than PFET-based CML. This fact is observed in Fig. 2.10c which presents the delay from an input signal to an output signal of the carry-out circuit, which is the critical path delay of the full adder. The results show that *VnanoCML* achieves lower delay than PFET-based circuit due to the lower device resistance which is a key factor in signal delay. The *VMTCML* adder achieves 45.6× lower delay time on average for the tested V_{dd} range. The performance improvement enables better power efficiency. Fig. 2.10d illustrates this observation. For example, to drive the same current level, I_c=4uA, i.e., same performance, *VnanoCML* presents 1.16× better power efficiency than the PFET case. Table 2.1 summarizes the performance comparison of *VnanoCML* with different technologies including CMOS and FinFET. Our result shows that *VnanoCML* outperforms all other technologies in terms of performance.

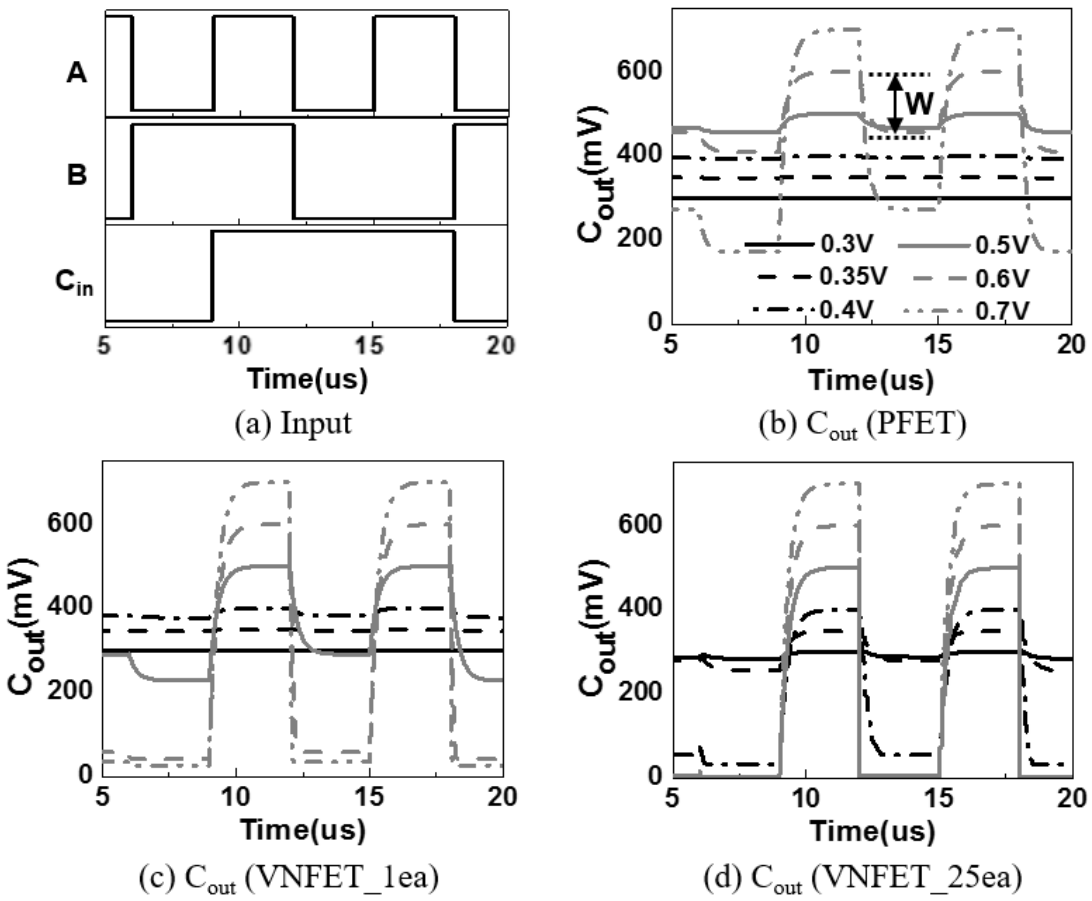


Figure 2.9. Waveform of transmission in CML Full Adder

2.5.3 VNFET Efficiency in Application

In this section, we evaluate the impact of *VnanoCML* on the architecture level using four ASIC designs. We compare to PFET-based design for two scenarios, i) when both technology consume the same power ii) provide the same performance. Fig. 2.11a first shows the comparison for the same power consumption. We adjust V_{dd} to produce the same power consumption. The results show that *VnanoCML*-based ASICs achieves significantly higher performance than the designs which use PFET-based CML. This advantage is due to lower resistance of VNFET which provides higher I_C current at lower supply voltage. For example, the proposed *VnanoCML* ASICs can achieve on average 16.4× performance speedup compared to the PFET-based design at the same power consumption. This significant performance improvement stems from the low

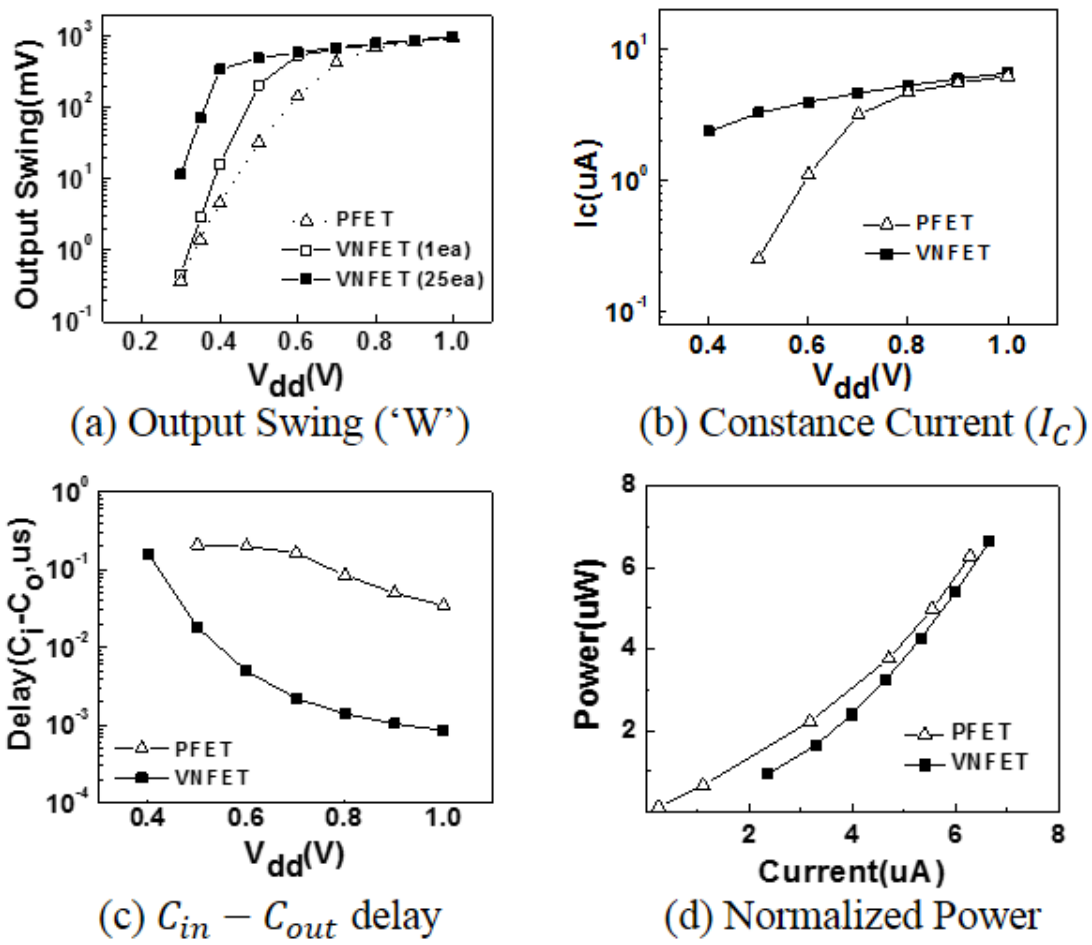
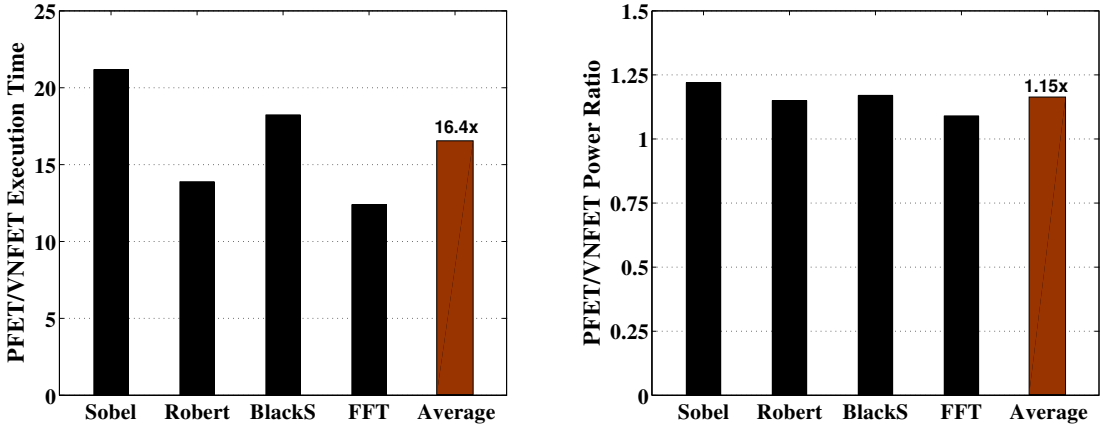


Figure 2.10. Comparison of *VnanoCML* to PFET-based CML for full adder design

resistance characteristic of our design, which creates high operation current and consequent high speed. Fig. 2.11b compares the power efficiency of the two designs when the performance is controlled to the same level by adjusting V_{dd}. The result shows that *VnanoCML* can also achieve higher power efficiency than the PFET-based design. For example, our design can provide 1.15× improvement in terms of average power consumption for the four applications. Since the proposed design provides better SS and lower resistance than PFET, the device can work on lower supply voltage while providing the same current.

As discussed in Section 2.4.4, although ASICs can be performed precisely under f_{ref} which considers the worst-case circuit delay, we may further improve the design efficiency for error-tolerant applications by relaxing the clock frequency constraint. Fig. 2.12 compares



(a) Same power

(b) Same performance

Figure 2.11. Normalized energy consumption and execution time of PFET-based design and *VnanoCML* at the same performance and power consumption.

the energy-delay product (EDP) of the four ASICs on the supply voltage of 0.6V for different error rates adjusted by increasing the clock frequency above f_{ref}. Since the process variation simulation creates a distribution function of the circuit delay, we define the error rate $\alpha\%$ by upper $(100-\alpha)$ percentile of the distribution function. The clock frequency is chosen by the delay at the percentile.

Our evaluation shows that, using the *precise ASICs* which set the clock frequency by f_{ref}, the EDP of *VnanoCML*-based design dramatically outperforms the PFET-based design, by $38.5\times$ on average for the four ASICs. In addition, if we allow the application approximation at $\alpha = 10\%$, the EDP improvement is $48.3\times$.

Table 2.1. Performance Comparison of *VnanoCML* with different device technologies

Vdd (V)	Delay (ns)			
	CMOS_PFET	CML_PFET	CML_FinFET	VnanoCML
0.5	494.4	207.0	209.3	**18.0**
0.6	125.9	198.1	149.3	**4.9**
0.7	44.8	162.6	60.7	**2.2**
0.8	22.9	83.7	35.4	**1.4**
0.9	15.2	49.9	25.0	**1.0**
1	11.5	34.7	19.8	**0.9**

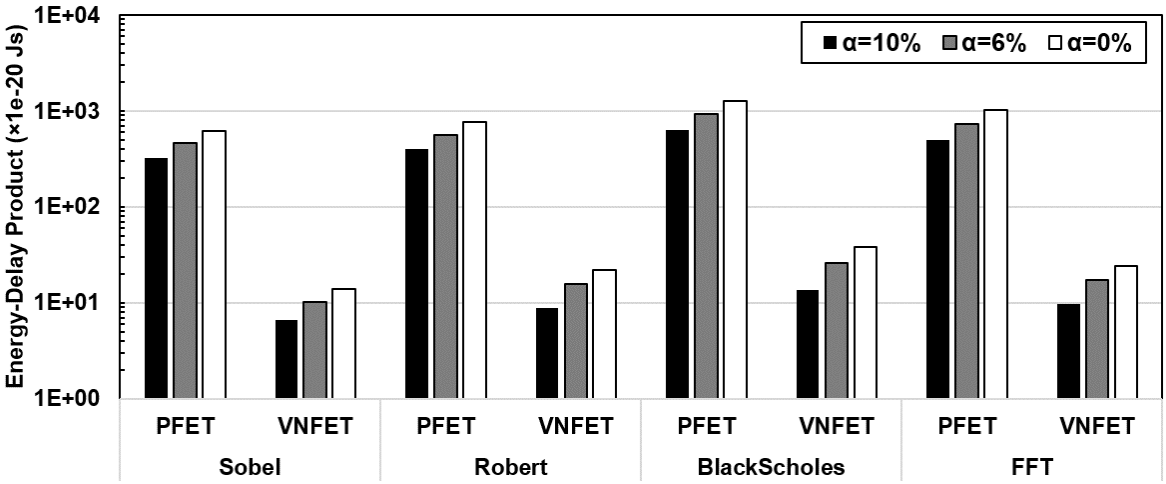


Figure 2.12. Impact of error rates on EDP improvement

In fact, our experiment shows that the approximated results would be still acceptable. Table 2.2 summarizes the quality loss of each application for different approximation levels. For image processing applications, the quality loss is defined by the Peak Signal-to-Noise Ratio (PSNR), and for the other applications, Average Relative Error (ARE) is used as a quality loss metric. It is known that 30dB of PSNR and 10% of ARE are acceptable quality loss [30]. The results show, for all the four ASIC designs, provide acceptable accuracy when $\alpha = 6\%$. For this acceptable accuracy, our design achieves 1.6× EDP improvement compared to the *precise ASICs* of *VnanoCML*.

Table 2.2. Quality loss of applications for different levels of approximation.

Error Rate (α)	**10%**	**8%**	**6%**	**4%**	**2%**	**1%**
Sobel	17*dB*	23*dB*	32*dB*	40*dB*	54*dB*	59*dB*
Robert	25*dB*	34*dB*	38*dB*	44*dB*	51*dB*	62*dB*
BlackScholes	13.1%	10.6%	8.3%	5.2%	2.1%	0.9%
FFT	10.5%	7.7%	5.3%	3.3%	2.6%	1.0%

2.6 Conclusion

We have presented a novel design which allows process compatibility of VNFET devices by utilizing the current mode logic. The proposed design also addresses the existing power issues of CML circuits using the high density of VNFET. The experimental results show that, as compared to conventional design, the proposed logic can achieve 16.4× speedup and 38.5× EDP improvement for the four ASICs. Furthermore, our design shows an advantage to approximate computation of error-tolerant applications. Compared to PFET-based approximation, our design achieves 48.3× EDP improvement while guaranteeing acceptable quality loss. In the following chapter, we will show how to implement PIM logic with low power consumption using the $4F^2$ access device presented in this chapter.

This chapter contains material from Joonseop Sim, Mohsen Imani, Yeseong Kim, Tajana Rosing, "Enabling Efficient System Design Using VerticalNanowire Transistor Current Mode Logic", IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2017. The dissertation author was the primary investigator and author of this paper.

This chapter contains material from Shadi Dayeh, Renjie Chen, Yungoo Ro, Joonseop Sim, "Progress in doping semiconductor nanowires during growth" Materials Science in Semiconductor Processing, 2017. The dissertation author was the fourth author of this paper.

Chapter 3

Unipolar Switching Logic for High Density PIM Applications

In the previous chapter, we presented a CMOS-type access device used in a crossbar array structure. In this chapter, we present how to implement PIM logic in crossbar stucture with low power consumption using this $4F^2$ access device. Prior studies show that non-volatile memories (NVMs) have great potential to enable the PIM functionality by exploiting analog characteristics of the memory devices [5, 6, 8, 9, 82, 83]. The various PIM technologies can be divided into two types. The first approach is a cell-based computation in which the result of the operation of multiple input cells is stored in the output cell as a resistance value. Another method is to implement the logic function by modifying the current sensing circuit of the NVM in addition to the function of determining the 0 and 1 of the data. The cell-based computation does not require peripheral circuits in the computation process, but massive column parallel computation is possible. However, the crossbar structure, in which cell-based computation mainly operates, consumes more power than the 1T1R structure and becomes more vulnerable as density increases. In this chapter, we first analyze power consumption factors in existing cell-based computation and then present a new PIM technology that can overcome it through novel device technology.

3.1 Introduction

There are several approaches to address the issue of data movements between the processor and memory. One notable way, often referred to near-data computing (NDC), is to integrate extra computing units close to the memory in order to locally process the data [84, 85]. However, this approach demands considerable costs to implement the additional logic stack into the memory stack, including e.g. Through Silicon Vias(TSVs), micro-bumps, etc. As an alternative solution of the NDC, several works proposed PIM architectures which migrate computation in the memory without using additional CMOS processing cores. There have been several proposals to enable computing capabilities inside DRAM [86, 87, 26, 88, 89]. However DRAM is inherently destructive in read operations, that is, the stored bits are invalidated after the read operations. Thus, the original data should be backed up to another cell before any computations, causing undesired overhead in PIM operations [26].

NVMs are good candidates for PIM due to their high density, scalability, and low power consumption [28, 27]. 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 [33]. 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 [5, 34]. This hinders area efficiency for the high-density applications.

In this chapter, we present *UPIM*, a novel processing-in-memory architecture, which enables PIM functions using unipolar-switching mode with 3D-layered structure. Unipolar-switching behavior of *UPIM* allows each memristor to be stacked with a diode, commonly known as 1-diode-1-resistor (1D1R) structure. It significantly reduces the sneak current by rectifying

reverse-direction current. Our UPIM design implements fundamental Boolean operations, including NOR, NAND, and NOT, taking the advantages of lower sneak current in the 1D1R structures. We also show how our PIM logic can be integrated with the 3D CBA structure. Our design under/over-laps the layer of the computation cells to the memory cells in a vertical direction to increase the cell density and gain the area efficiency.

The main contributions of this chapter 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.
- We evaluate the proposed UPIM design for a wide range of applications. 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.

3.2 Related Work

The work in [8, 9] modified sensing circuits to implement fundamental bitwise operations, such as AND and OR. However, they do not support arithmetic operations, e.g. addition and multiplication, which are the key functions involved in many applications such as deep learning algorithms and image processing. To support arithmetic operations, several designs have been also presented in [5, 8, 18]. Their arithmetic PIM operations are computed using bitwise operations that run on a bipolar switching mode. However, the bipolar memristors

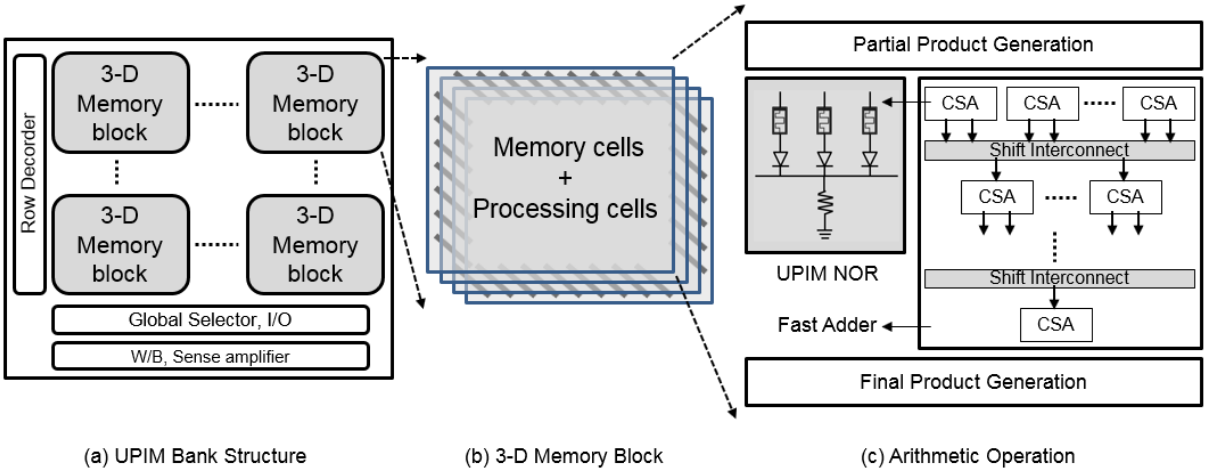


Figure 3.1. The overall structure of UPIM design

incur significant static power consumption due to sneak current dissipation. The work in [82] introduces PIM techniques for unipolar memristors. However, it requires multiple backup stages to preserve the inputs.

Our proposed UPIM is different from previous work since we enable non-destructive in-memory processing in unipolar 1D1R crossbar architecture. This significantly reduces the static energy in PIM operations by eliminating sneak current in the memory. In addition, unlike the previous work, we integrate our design in 3D CBA to increase the computing density per area.

3.3 UPIM Design

Fig. 3.1 shows the overview of UPIM. A bank structure in UPIM has multiple memory blocks, while each memory block consists of memory cells and processing cells. The cells are integrated with a 3D-layered structure to increase the memory density and minimize area overhead in supporting PIM functions. Memory and processing layers are connected with the configurable interconnects which implement shift operations. Each memory cell in UPIM works with a unipolar-switching mode that exploits a diode to rectify the current direction and reduce the sneak current. UPIM cells support fundamental Boolean logic, e.g., NOR, NOT and

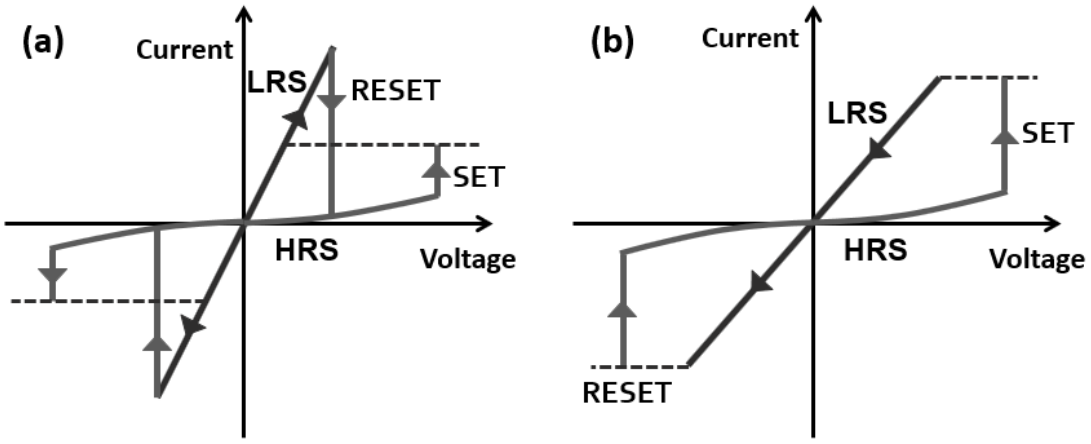


Figure 3.2. (a) Unipolar and (b) Bipolar switching mechanisms of memristor

NAND. The arithmetic operations are implemented based on the Boolean logic.[1] We implement arithmetic operations with three key components: partial product generation, fast adder, and final product generation. The fast adder utilizes shift interconnect and carry save addition (CSA) to optimize the latency of the addition. We describe the UPIM logic in Section 3.3.2, and show the 3D integration of UPIM for the high-density applications in Section 3.3.3.

3.3.1 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 as shown in Fig. 3.2(a), and the other is 'bipolar', where the reset switching (LRS $\rightarrow$ HRS) and set switching (HRS $\rightarrow$ LRS) take place with the opposite of the bias polarity as shown in Fig. 3.2(b) [90]. Unipolar switching has 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 [82]. Fig. 3.3 is the schematic of a memory array in the read operation. Consider an M$\times$N array, there are $(N-1)(M-1)$ sneak current paths (one of them is shown with a blue line at Fig. 3.3) when a single cell on the black line is intended to

[1] In this work, we focus on NOR-based PIM arithmetic operations. Note that the NAND-based operations can be also implemented with minimal modification of the PIM logic.

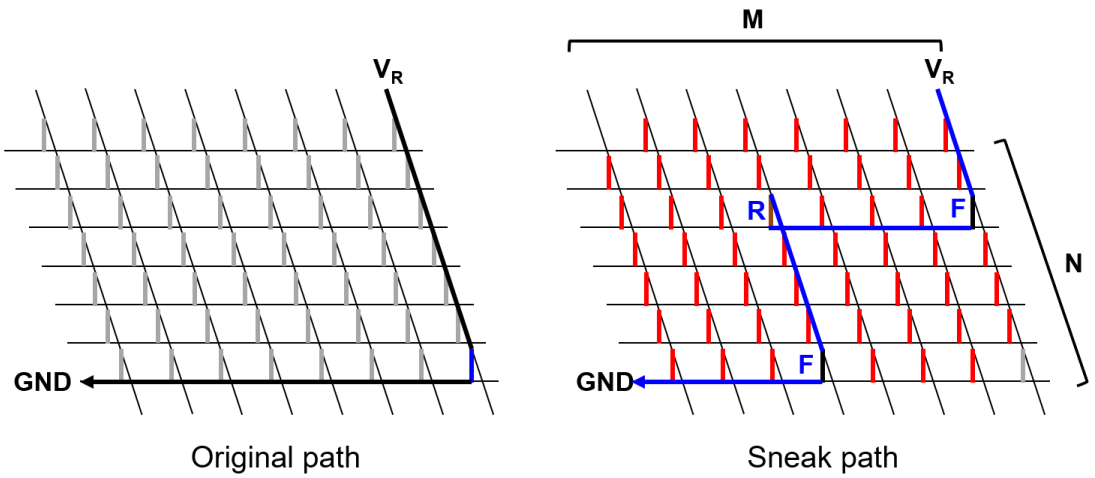


Figure 3.3. Sneak current influence on the total current in a memory array

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. (3.1) [33].

$$I_{SNEAK} = V_R \times \left(\frac{R_F}{N-1} + \frac{R_R}{(N-1)(M-1)} + \frac{R_F}{M-1} \right)^{-1} \tag{3.1}$$

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. (3.1) has a significant implication that since majority 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 a bipolar device [6, 8, 5], In this chapter, we present 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.

3.3.2 Unipolar-based logic within NVM

NOR operation design

Fig. 3.4(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 logic 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 model shown in [91], 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 3.4.3. All four resistors are connected to the BL. Fig. 3.4(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. 3.4(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 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} keeps the high resistance state representing the logical 0. Fig. 3.4(d) shows the resistance behavior of the UPIM NOR gate. R_{OUT} and $R_{OUT\prime}$ 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_G, all the other cases keep the R_G as low resistance state, presenting NOR operation.

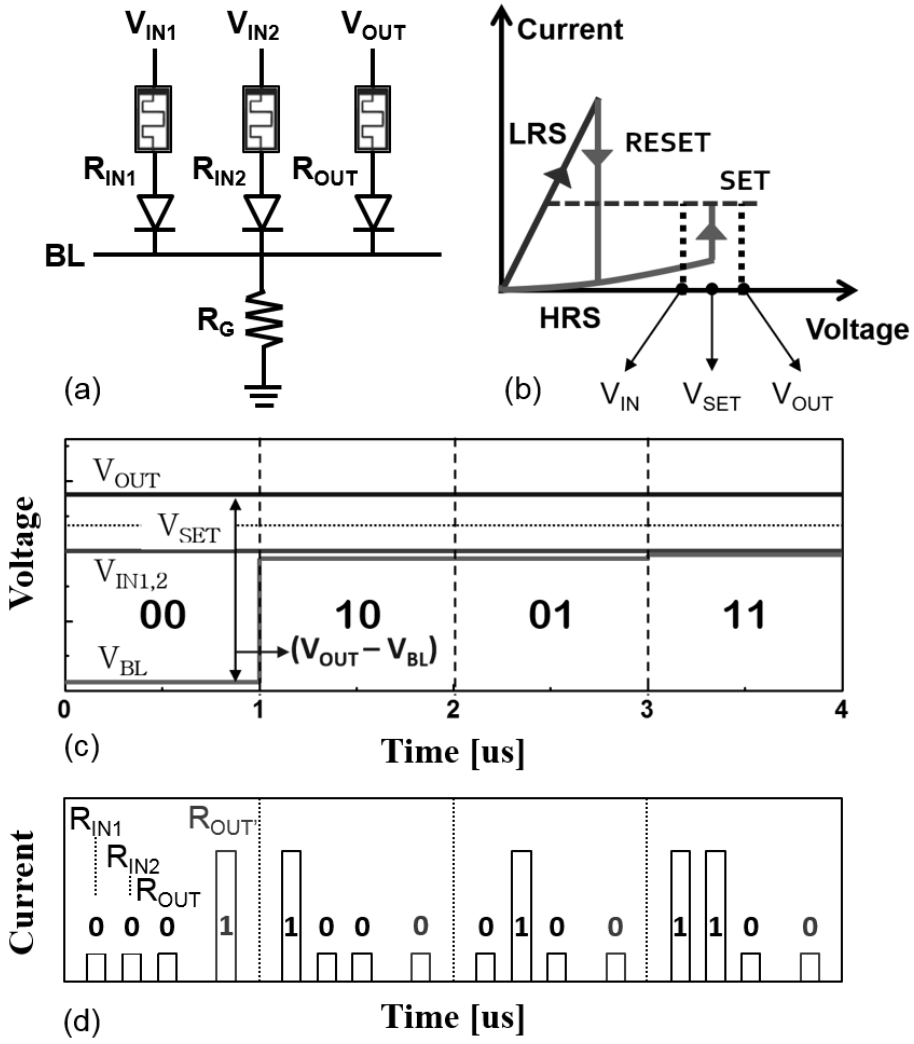
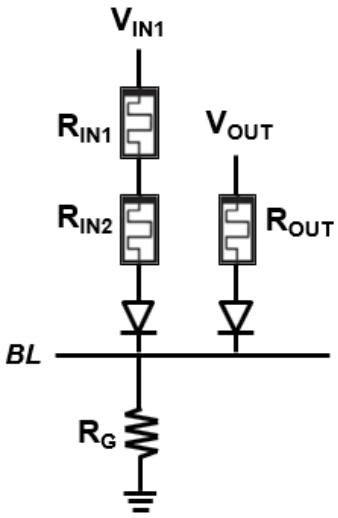


Figure 3.4. Proposed unipolar–based NOR logic: (a) Schematic of the NOR gate (b) Voltage conditions (c) NOR gate simulation result (d) Resistance behaviors depending on input states

NOT and NAND operation design

The unipolar–based design can support other Boolean logic. For example, a NOT operation can be performed using one input cell in the same way to the NOR design. Fig. 3.5 shows how we can support the NAND operation. Since input resistors, R_{IN1} and R_{IN2}, are connected in series, if at least one resistor has R_{HRS}, V_{BL} is pulled down to ground, and thus R_{OUT} is switched to the low resistance state. In the other case when both resistors are R_{LRS}, R_{OUT} maintains the initial high resistance state.



NAND Truth Table

R_{IN1}	R_{IN2}	R_{OUT}
0	0	1
1	0	1
0	1	1
1	1	0

Figure 3.5. unipolar–based NAND logic

3.3.3 Integration to 3D CBA structure

The proposed design executes arithmetic functions using NOR operations. As discussed in Section 3.1, 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. 3.6(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. 3.6(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. 3.7 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 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

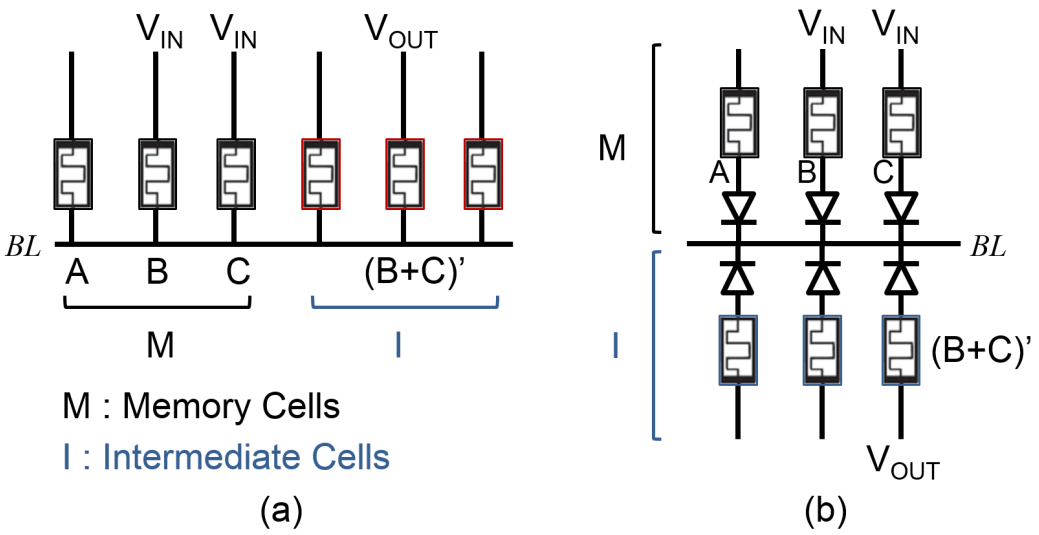


Figure 3.6. Schematic of (a) Prior 2D and (b) Proposed 3D logic in memory

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. 3.8 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 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 case of Fig. 3.8, the UPIM NOR operations of A and B, D and E can be executed in parallel with a single PIM operation.

Table. 3.1 summarizes the comparison of the proposed UPIM to existing technologies. Since UPIM performs logic operations in 1D1R cell structure, we achieve higher power efficiency than other PIM technologies based on the bipolar switching mode. Moreover, when implementing UPIM into the 3D CBA structure, it further overcomes the issue of the area overhead existing in

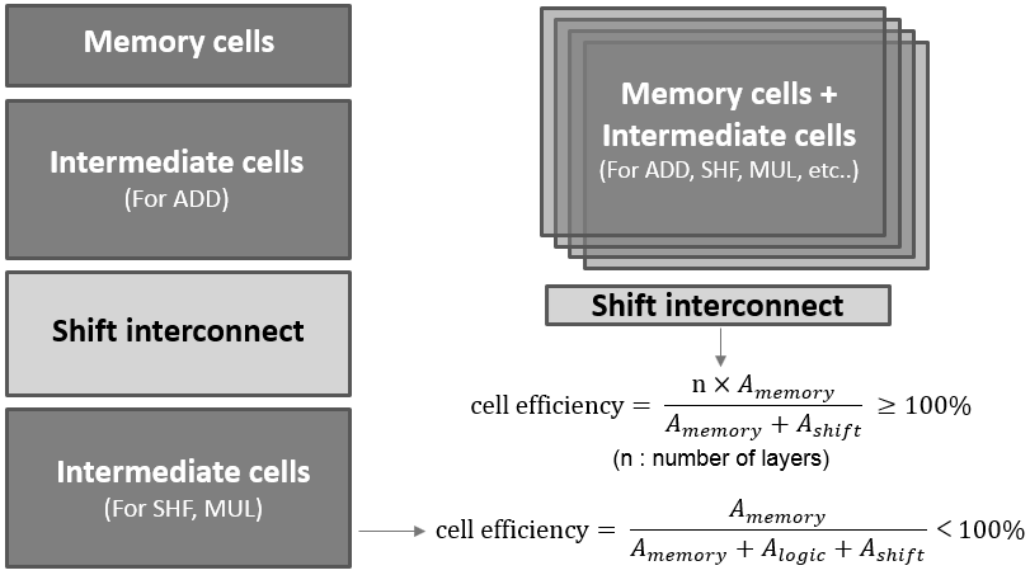


Figure 3.7. Diagram of prior 2D (left) and proposed 3D (right) PIM structures

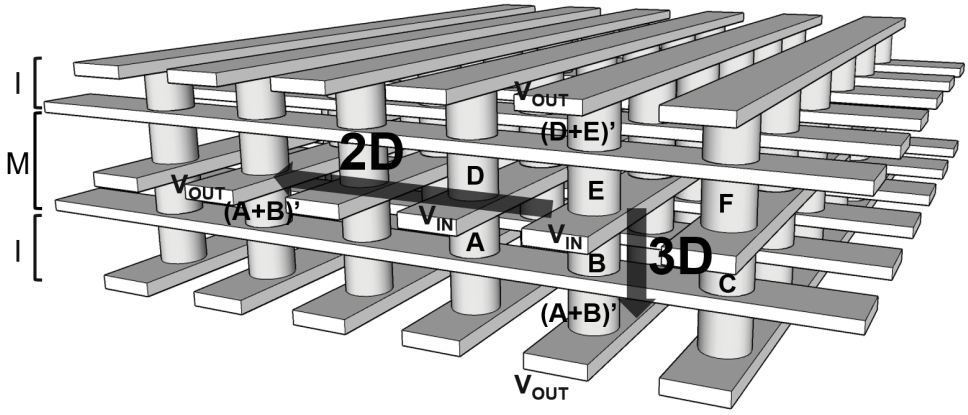


Figure 3.8. The integrated structure of 3D UPIM

the 2D PIM approaches.

3.4 Experimental Results

3.4.1 Experimental Setup

Performance and energy consumption of the proposed design have been obtained from circuit-level simulations in a 45nm CMOS process and design kits of Cadence Virtuoso and Spectre simulators. We use the memristor model [91] for our resistance-based memory design

and simulation 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 test the efficiency of UPIM on six general OpenCL applications including *Sobel, Robert, Fast Fourier transform (FFT), DwHaar1D, Sharpen* and *Quasi Random*. For image processing, we use random images from *Caltech 101* library [81], while for non-image processing applications inputs are generated randomly. We compare the efficiency of the proposed UPIM design with AMD R390 GPU and state-of-the-art PIM designs, [5, 6, 7]. We used Hioki 3334 power meter to measure the power consumption of GPU, while we use in-house cycle-accurate simulator to estimate UPIM energy and performance.

3.4.2 Energy and Performance

As discussed in Section 3.3.1 and 3.3.2, our unipolar-based logic is operated in the 1D1R structure, which shows lower static power consumption by reducing sneak current dissipation. Fig. 3.9 presents the static energy saving of 1D1R over 1R structure. The energy has been estimated with the total energy consumed by three input memristors in read condition (V_{DD}=1V) during the memristor switching time of 1.1ns. As array size increases, an energy saving of 1D1R over 1R structure is enhanced due to an increase of unselected cells, the sneak current path. Based on our experimental results in the range of 25∼10K cell array, our estimation shows that the 1D1R structure achieves up to 100× static energy saving at the 100Mb memory size compared to the 1R structure.

Table 3.1. Performance of proposed UPIM and other technologies

	IMPLY [5]	MAGIC [6]	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**

Fig. 3.10 shows the energy and energy-delay product (EDP) improvements of running applications on proposed UPIM and state-of-the-art PIM designs [5, 6, 7], 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 a 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 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. 3.3.2.

3.4.3 Process Variations

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.

This results in the resistance variation on UPIM, since a memristor resistance with a cylindrical shape has an inverse dependency with its diameter [92]. 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, which is an atomic access unit for a

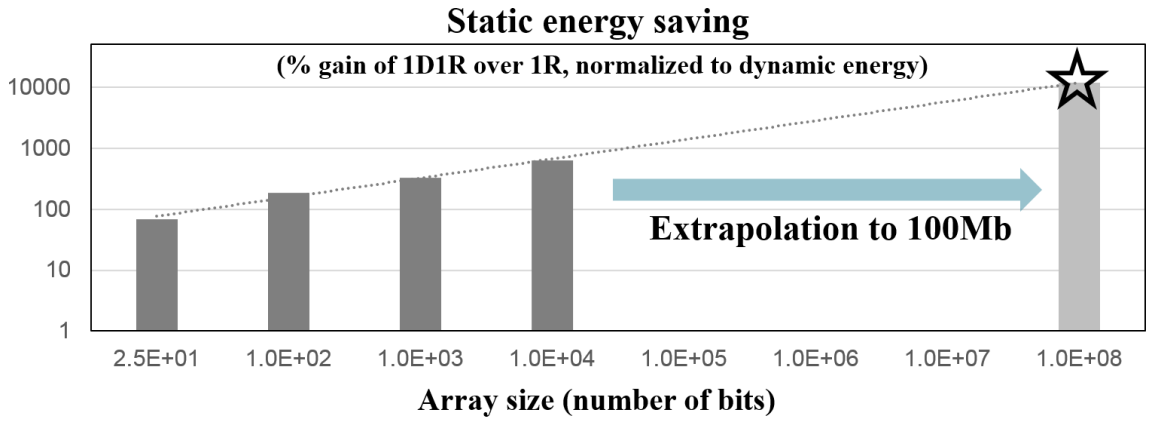


Figure 3.9. Static energy saving of 1D1R over 1R (normalized to dynamic energy)

single memory operation [93], as shown in Fig. 3.11(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} is different over near and far cells. For example, in the case of the far cells, the BL resistance is 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. 3.11(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. 3.11(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.

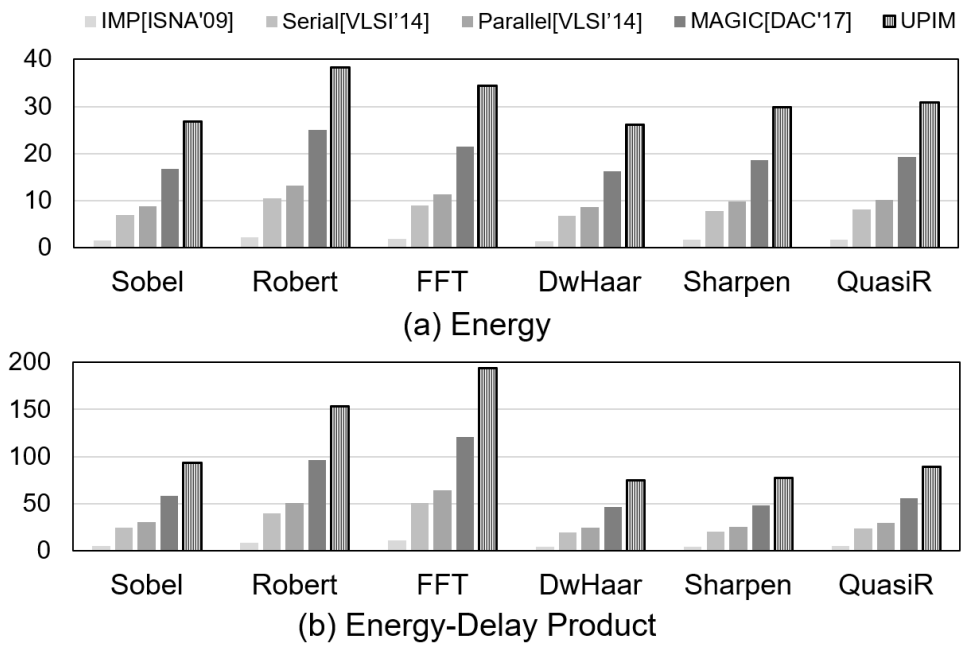


Figure 3.10. Energy and Energy-delay product improvement of proposed UPIM and state-of-the-art [5, 6, 7] for different applications.

3.4.4 Evaluation for Area Efficiency

We evaluated the area efficiency of our design as compared to the MAGIC [6], a state-of-the-art PIM design. The area efficiency of the PIM techniques is mainly dependent on two factors, i.e., the 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 3.12(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. 3.12(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 [94].

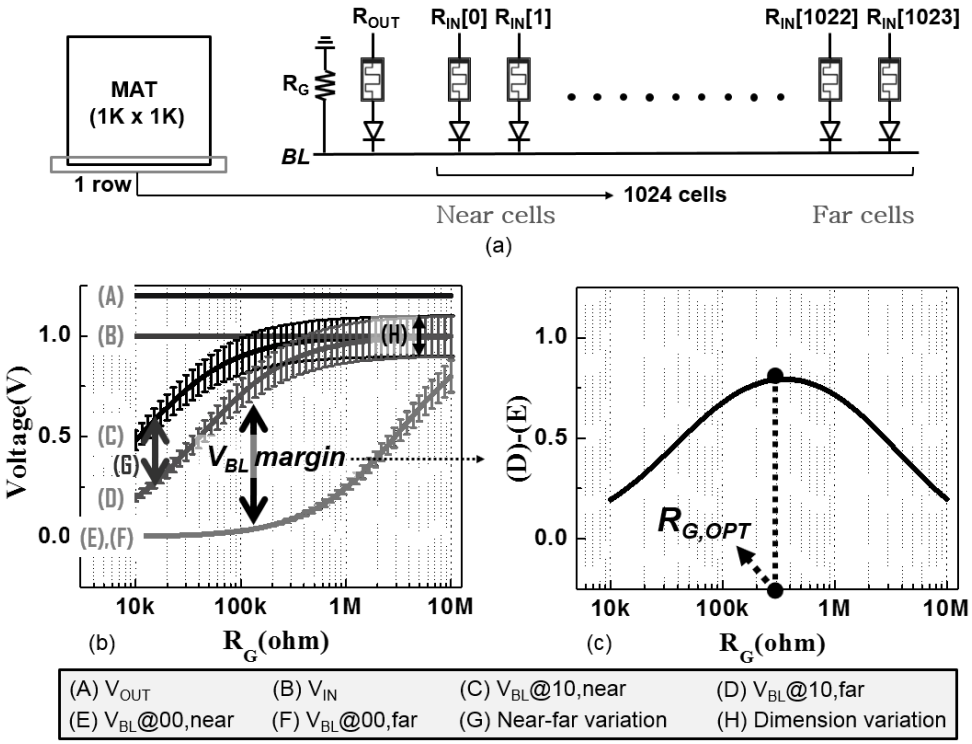


Figure 3.11. V_{BL} margin and R_G optimization considering process variation

Fig. 3.12(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 $4F^2$, i.e., the minimum cell size of any 2D-based memristor design [95].

3.5 Conclusion

In this chapter, we presented 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

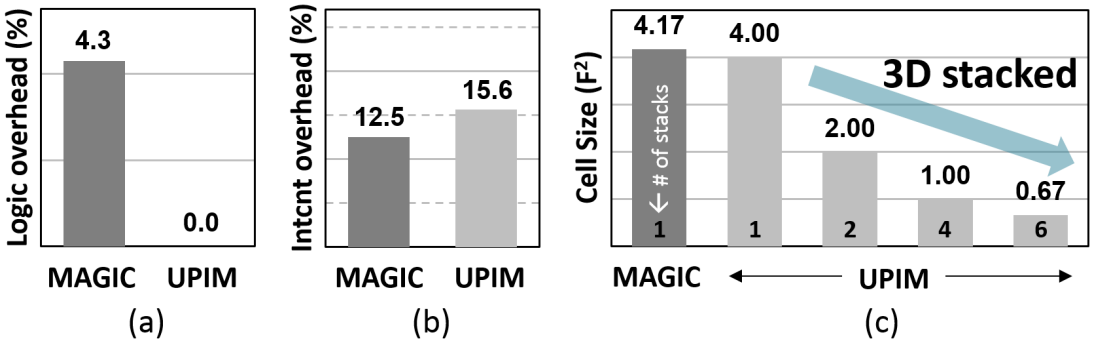


Figure 3.12. Overhead and cell size comparison between UPIM and MAGIC [6]

consumption compared to the state-of-the-art PIM designs [5, 7, 8] and AMD Radeon 390 GPU [36], respectively. In the following chapter, we address another approach to PIM design, identifying the existing problems with sense amplifier (SA)-based PIM technology, and present our technology to overcome this.

This chapter contains material from Joonseop Sim, Saransh Gupta, Mohsen Imani, Yeseong Kim, Tajana Rosing, " UPIM : Unipolar Switching Logic for High Density Processing-in-Memory Applications", ACM Great lakes symposium on VLSI (GLSVLSI), 2019. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Current-Sensing Adder for PIM Designs

PIM technologies either use a cell-based computation in which the result of the operation of multiple input cells is stored in the output cell as a resistance value, or implement the logic function by modifying the current sensing circuit of the NVM. In the previous chapter, we discussed the cell-based computation, addressing issues in current technology, and we presented our 'UPIM' design as an alternative that solves the power consumption problems of the technologies using the bipolar switching method. In this chapter, we discuss another PIM approach, sense amplifier (SA)-based PIM. We analyze existing PIM circuits, point out the limitations of performance due to their multi-cycle operation, and present our design that implements single cycle operation using novel device technology.

4.1 Introduction

Although many PIM techniques have been proposed so far, they support limited basic functionalities such as basic bitwise operations (AND, OR, and IMP), which are only applicable to specific applications. For example, the designs shown in [9, 29] support bitwise operations but cannot support arithmetic functions like the addition and multiplication. The techniques in [96] have been designed exclusively for accelerating neural network algorithms. Many applications including machine learning algorithms and image processing involve complex functions [97, 98, 13]. Hence, several techniques have been proposed to perform functions like

addition and multiplication in NVM architectures [5, 18, 8, 30]. However, they execute these functions by combining multiple boolean operations (IMP, NOT, NOR). Therefore, they are inherently slow due to their multi-cycle operation as well as slow processing speed.

In this chapter, we present a new PIM architecture, called *LUPIS*, which enables the addition and multiplication in an efficient manner. We design a new sensing circuit which uses the analog properties of NVM. We simplify computation by exploiting the latch-up effect of thyristor devices to directly generate the results from the input data without any intermediate logic. We further leverage the back-down effect at latch-up points of the thyristor to implement functions with minimal increase in the number of gates. Also, the proposed design performs the operations in a modified sensing circuit, which is compatible with the conventional current sense amplifier (CSA). It does not need additional cells to support calculations, thus requires negligible area overhead. We show that the proposed LUPIS can improve performance and energy efficiency of many popular applications such as machine learning and data analysis, which involve a large number of additions and multiplications.

The main contributions of this work are listed as follows:

- We present high performance and low-cost PIM architecture based on the latch-up effect of thyristors, enabling single-cycle addition (ADD), and significantly improving the performance of multiplication (MUL).
- Our design requires no additional cell array for processing, hence can be an excellent candidate for the storage class memory which has been considered as the main application of memristor-based products.
- Our experimental results show that LUPIS can provide 12.7× speedup and 20.9× energy efficiency as compared to the state-of-the-art PIM accelerator, APIM [8].

4.2 Related Work

The emerging nonvolatile memory (NVM) technologies such as phase change memory (PCM) and resistive RAM (ReRAM) are considered good candidates for PIM due to their high density, scalability, and low power consumption [28, 27]. However, the supported functionality in most of the PIM designs is limited to either bitwise operations or operations derived from basic bitwise operations which require multiple cycles. For example, [29, 9] proposed a sensing circuit to implement the basic bitwise operations such as AND, OR, and INV. However, they do not support addition and multiplication, which are the critical arithmetic functions involved in many applications such as machine learning algorithms and image processing [30]. Several designs presented in [5, 18, 8, 6] have designed the full adder function based on bitwise operations. Since these approaches implement the operation by combining multiple basic operations, e.g., NOR or IMP, they require tens of cycles. They include computing the intermediate outcomes until obtaining the final results. Thus, these designs pose inevitable timing overheads. In addition, the considerable area overhead due to the extra processing cell arrays makes them unsuitable for storage class memory, which demands high-density integration.

4.3 Background and Motivation

4.3.1 NVM Sensing Scheme

Emerging nonvolatile memories such as ReRAM, STT-RAM and PCRAM can be classified as the resistance-based memories. These technologies store and read the data by changing the cell resistance, e.g. its high and low resistance states are interpreted as logic 0 and 1 respectively. One of the major differences between NVM and DRAM is the sense amplifier design. While charge-based DRAM uses a voltage sense amplifier (VSA) which detects the electronic potential between the bitline (BL) and bitbar-line ($\overline{BL}$), NVM uses a different current sense amplifier (CSA) due to its better distinguish-ability of the resistance difference than the VSA. Fig. 4.1a shows the sensing scheme of the conventional CSA [99]. The data in a memory cell

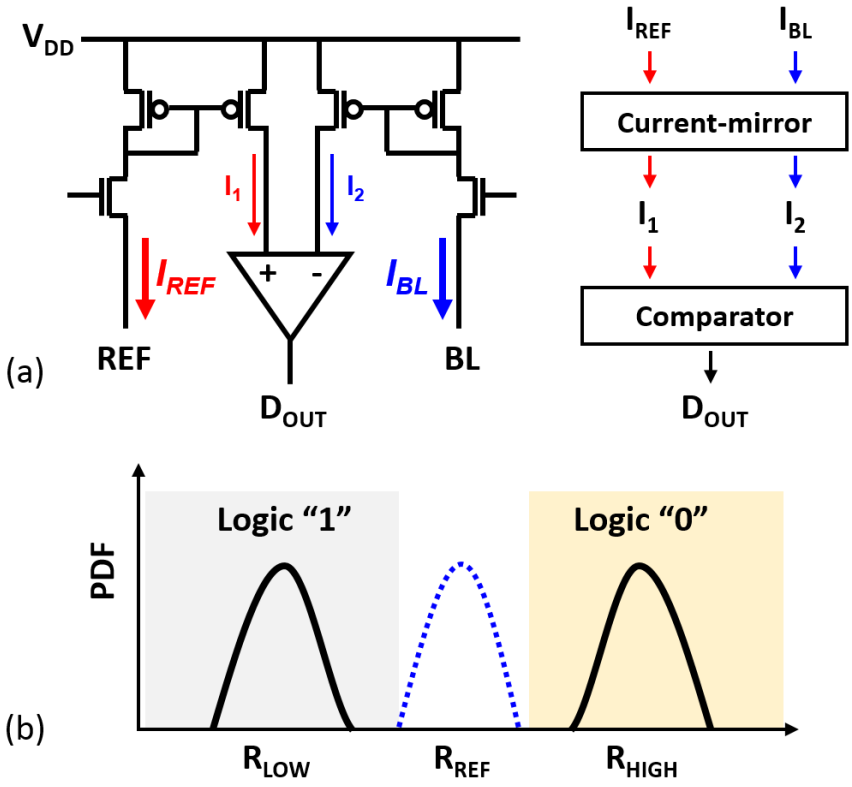


Figure 4.1. Conventional Sensing Scheme for NVM

is determined by assessing the current from the selected memory cell. When the current from the selected BL (I_{BL}) and the current from the reference cell (I_{REF}) are mirrored to (I_1) and (I_2) respectively, they are compared to each other and changed to voltage signals (D_{OUT}) [100]. The state of $R_{CELL} < R_{REF}$ is considered as logic "1" and the other case is considered as logic "0" as shown in Fig. 4.1b. The conventional CSA is capable of only judging the resistance from the selected cell higher and lower than reference resistance. In this chapter, we present a current sensing scheme, which also enables arithmetic functions, *i.e.*, addition and multiplication inside the memory module, compatible with the conventional sensing scheme.

4.3.2 Thyristor Latch-Up

We exploit a vertical PNPN structure commonly referred to a *thyristor* [101]. Fig. 4.2a shows the structure used in our design. The structure has three P-N junctions and is equivalent to two cross-coupled bipolar junction transistors (BJTs) as shown in Fig. 4.2a. This structure has a

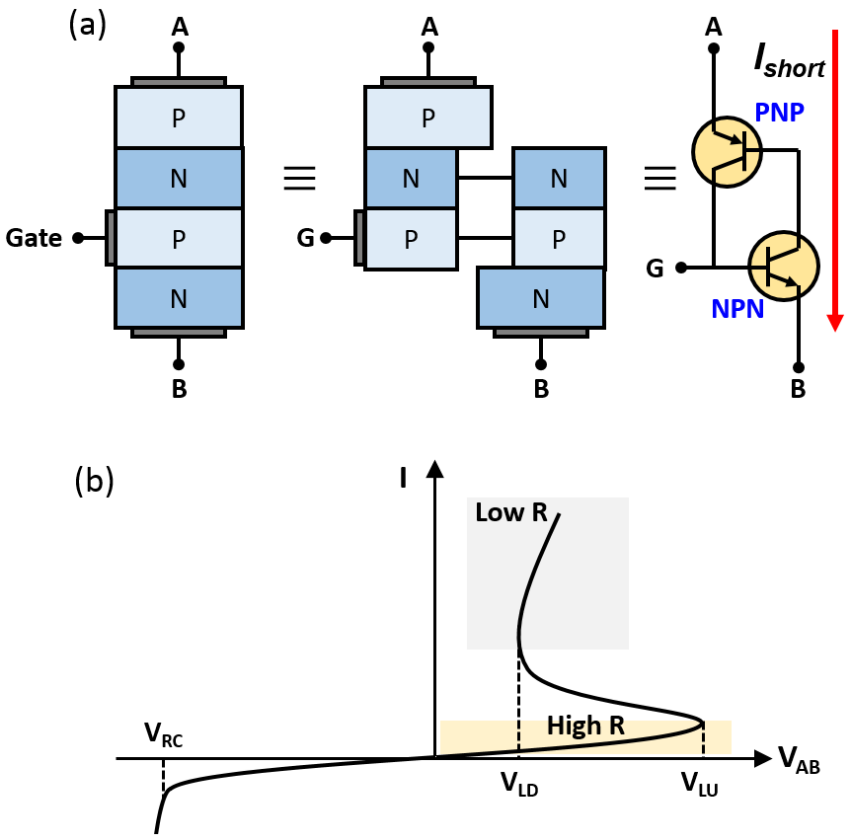


Figure 4.2. (a) Thyristor Structure and (b) Voltage-Current Behavior

short-circuit path, often referred to as *latch-up* in CMOS design. When one of the two BJTs gets forward biased, it feeds the base of the other BJT. This positive feedback increases the current until it saturates to I_{short}.

Fig. 4.2b shows the voltage and the current behaviors of the structure. In the initial state, a thyristor has high resistance (2MΩ in our experimental setup). When the voltage across the device (V_{AB}) is increased, the device keeps the high resistance state until V_{AB} reaches the latch-up voltage (V_{LU}). Latch-up occurs at V_{LU} and the current through the cell (i.e., from A to B in Fig. 4.2a) abruptly increases until the applied bias turns back to the latch-down voltage (V_{LD}). In order to restore the thyristor device resistance to the original state, a reverse bias, V_{RC}, should be applied to V_{AB}. It moves the minor carriers out from the base regions, and the device is set to the initial state again. In the rest of the paper, we call this recovery state as the *write-back* step.

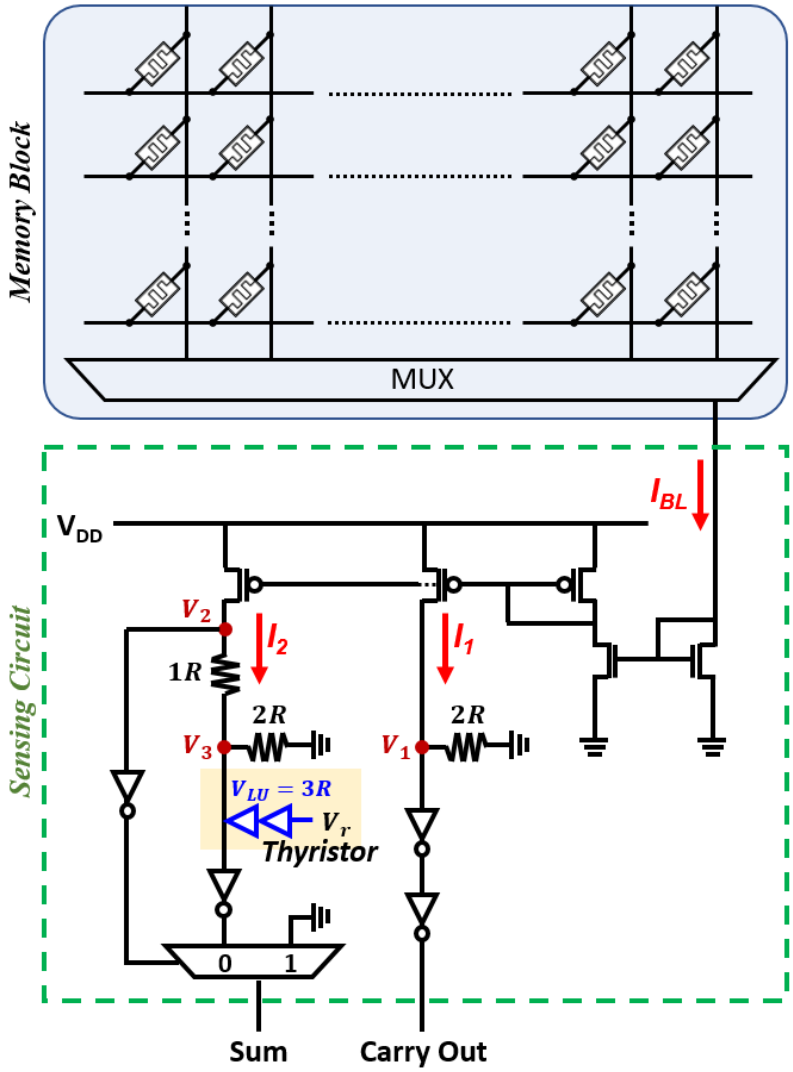


Figure 4.3. Proposed Sensing Circuit for Addition

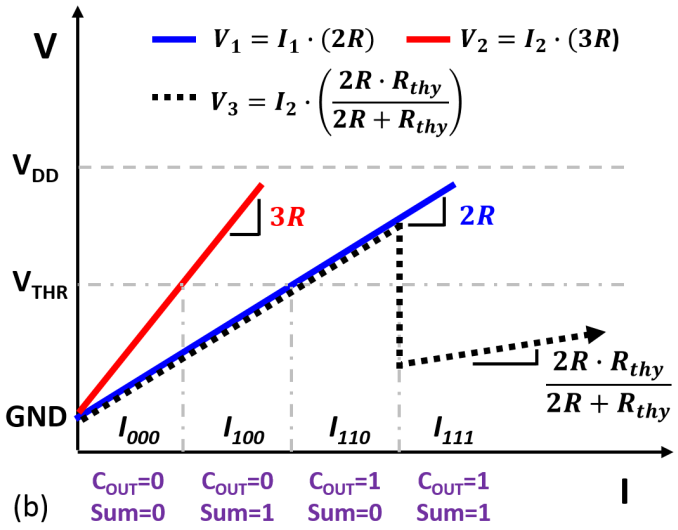
4.4 Proposed Design

4.4.1 Latchup-based Addition

we present a new sensing circuit, which exploits the thyristor latch-up effect, to enable ADD operation in a cycle. Fig. 4.3 is the schematic of the proposed design. The design consists of two parts: the current mirror and the adder. Once three rows of a memory block corresponding to the input values (A, B, C_{in}) are activated, the total current from the activated rows denoted as I_{BL} is delivered to the selected BL. The current mirror circuit in Fig. 4.3 copies the I_{BL} to I_1 and

A	B	C_{IN}	C_{OUT}	Sum	I_{BL}
0	0	0	0	0	I_{000}
0	0	1	0	1	I_{100}
0	1	0	0	1	
1	0	0	0	1	
0	1	1	1	0	I_{110}
1	0	1	1	0	
1	1	0	1	0	
1	1	1	1	1	I_{111}

(a)



(b)

Figure 4.4. Voltage Transfer as an Input Current (I_{BL})

I_2 and delivers them to *Carry Out* (C_{out}) and *Sum* branches, respectively. Our design computes the outputs by distinguishing the total current amount reached to the sensing circuit.

Fig. 4.4a presents the truth table of a full adder. The sum is the exclusive-or (XOR) result for the three inputs and the carry out is the "majority" function of the inputs. The three inputs are interchangeable in that the order of them does not affect the output. In the memristor devices, the amount of the bitline current is the combination of one R_{on} and two R_{off}. Based on this characteristic, there are four different cases depending on the current amount, I_{000}, I_{100}, I_{110}, and I_{111}, according to the number of high (0) and low (1) resistances in the memristors of activated rows.

Fig. 4.4b shows how the proposed circuit distinguishes the four current regions to create the desired C_{out} and *Sum*. In our circuit design, there are three major voltage nodes (i.e., V_1, V_2, and V_3 shown in Fig. 4.3) whose potential determine the final outputs of the *Sum* and C_{out} by the following digital logic gates. The voltage of each node is transferred as a function of the current in the selected bitline. V_{THR} is a threshold value which determines whether an input potential is interpreted as a logic 0 or 1 (i.e., any value less than V_{THR} is 0 and any value above V_{THR} is 1.) Let R_{thy} be the resistance of the thyristor explained in Section 4.3.2. Then,

the electric potentials at V_1, V_2, and V_3 are represented by $V_1 = I_1 \cdot (2R)$, $V_2 = I_2 \cdot (3R)$ and $V_3 = I_2 \cdot (2R \cdot R_{thy})/(2R + R_{thy})$, respectively.

As for the carry-out function, V_1 node has higher electric potential than V_{THR} in cases of I_{110}, I_{111} when it delivers a logic 1 to C_{out} through two inverters which strengthen the signal. For the I_{000} and I_{100} cases, V_1 node a has a lower potential than V_{THR} and C_{out} presents a logic 0.

The *Sum* function uses branches where V_2 and V_3 nodes are located. As shown in Fig. 4.4b, V_2 node has a lower potential than V_{THR} only in case of I_{000} and its inverted logic 1 is delivered to the MUX, pulling down the *Sum* potential to the ground, i.e., logic 0. In the opposite cases, i.e., I_{100}, I_{110} and I_{111}, V_2 has a logic 1, and the MUX delivers the data from the connection where V_3 is located, so that V_3 decides the outputs for the three cases. For the I_{100} and I_{110} case, the V_3 shows either logic 0 or logic 1 depending on the input current in a similar way of V_1, since the thyristor is not activated in this region. However, once the current increases and reaches the I_{111} range, the latch-up occurs in the thyristor device, and thus the electric potential of V_3 abruptly falls below V_{THR}, making *Sum* logic 1. With this logic, our design is able to complete all *Sum* and C_{out} results. Once the final outputs are generated, we reset the thyristor for the next cycle, by invoking the write-back procedure to turn the thyristor resistance back to the original state. In the case of N-bit addition, C_{out} is written back to the memory and is used to calculate the results for the next bit.

In our experimental setup, we assume $V_{THR} = V_{DD}/2$ and set R=20kΩ to yield the described voltage transfer functions. The latch-up also affects the potential of the V_2 node. However, the potential V_2 does not drop below V_{THR} in our design, since the thyristor resistance on the conductance state is very low compared to 2R, a higher portion of the supply voltage is applied between V_2 and V_3. This keeps the V_2 over the V_{THR} and V_3 below V_{THR} with a marginal window in the case of I_{111}.

4.4.2 Multiplier Design

Implementing multiplication in memory is challenging due to a large number of parallel computations and shift operations for each multiplicand bit. The multiplication is performed in three stages, partial product generation, fast addition, and final product generation. The partial product generation stage generates *n* partial products, where *n* represents the size of the multiplier. They are propagated to the next stage. The fast addition method used in [8], optimizes the latency involved in the addition of the generated partial products. They implement tree-based carry save addition shown in Fig. 4.5 to push carry propagation to the last stage, hence enabling faster operations. A carry save adder implements half-addition at each bit. It takes in three inputs and generates two outputs, sum and carry, resulting in a 3:2 reduction. Successive carry save additions reduce the initial *n* partial products into two numbers which are then added in the final stage.

Although we use similar techniques as those in [8] to support multiplication, our work is different from the implementation perspective. We design novel circuits using a thyristor device to enable single-cycle computations. For single-bit addition, which is the basic operation to implement MUL, the design in previous work has larger latency than the proposed design. Moreover, it uses interconnects to enable shift operations which require a large number of additional transistors. It induces significant area overhead, which grows exponentially as the block size increases. Our design generates intermediate results at the sensing circuits and writes them back for further computations. Hence, the shift operations are dealt with while writing the results back to the memory. This does not require the expensive interconnects used by the previous work. Furthermore, we can utilize the thyristor write-back, which happens at the end of the ADD operation, to hide the write latency.

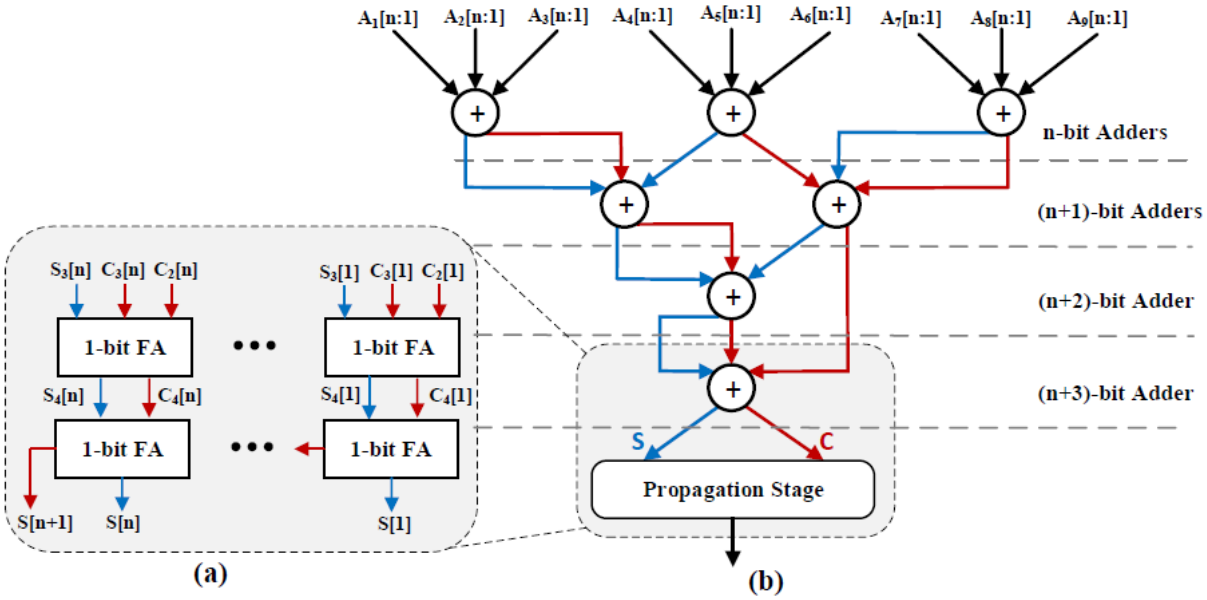


Figure 4.5. (a) Carry save addition (b) Tree structured addition of 9 n-bit numbers

4.5 Experimental Results

4.5.1 Experimental Setup

Performance and energy consumption of the proposed design have been obtained from circuit-level simulations in a 45nm CMOS process and design kits of Cadence Virtuoso and Spectre simulators. We use VTEAM memristor model [91] for our resistance-based memory design and simulation with R_{on} and R_{off} of 10KΩ and 10MΩ respectively. The thyristor device has been designed and simulated using Silvaco ATLAS TCAD software to investigate the latch-up effects to the PIM architecture and optimize the process conditions.

To evaluate the efficiency of LUPIS for practical applications by designing a cycle-accurate simulator which models the memory function. We compare the proposed design with AMD Southern Island GPU, Radeon HD 7970 device, which is one of the most recent GP-GPU architectures. We compared the efficiency of LUPIS to the GPU architecture for four OpenCL applications: *Sobel, Robert, Fast Fourier transform (FFT)* and *DwHaar1D*. For image processing, we used random images from *Caltech 101* [81] library, while for non-image

processing applications inputs were generated randomly. These applications involve many additions and multiplications, and we further approximated other common operations such as square root with the two operations. In the application level, we also modified the source code of the applications so that applications utilize PIM-based addition/multiplications as much as possible, e.g., using Taylor expansion.

4.5.2 Device Optimization

In order to optimize the process conditions of a thyristor used in our proposed design, a device simulation was performed using Silvaco Atlas. As shown in Fig. 4.6a, we used a lateral PNPN structure consisting of a p-type Si substrate, n+contact, n-well, and p+contact, with doping concentrations of $1 \times 10^{16} cm^{-3}$, $1 \times 10^{20} cm^{-3}$, $2 \times 10^{16} cm^{-3}$ and $1 \times 10^{20} cm^{-3}$. A width of 0.1 um was used for the 2-dimensional simulation. For the latch-up simulation, 1.0 us was used for Shockley-Read-Hall life times for both electrons and holes, and the Selberherr model was applied for the impact ionization. Fig. 4.6(b) illustrates the simulation result with d_1=d_2=0.2um. Based on this simulation result, we exploited a V_{LD} of 0.89 V, a V_{LU} of 0.98 V, an R_H of 1.9 MΩ, and an R_L of 1.7 kΩ, where R_L and R_H are the resistance of thyristor in the higher and lower conductance state respectively. As shown in Fig. 4.6c and Fig. 4.6d, R_H and V_{LU} can be easily controlled by changing either the device structure or the doping concentration of the p- and n- regions. This is because R_H and V_{LU} in the high resistance regime are dominated by the characteristics of the reverse biased PN junction at the central p- and n- regions, so we can tune them by varying the device structure and/or doping process conditions. Since there is a clear dependency between R_H and V_{LU}, i.e., V_{LU} increases as increasing R_H based on our simulation results, the thyristor ensure stable support of device characteristics that upper circuit and system design need with marginal process window.

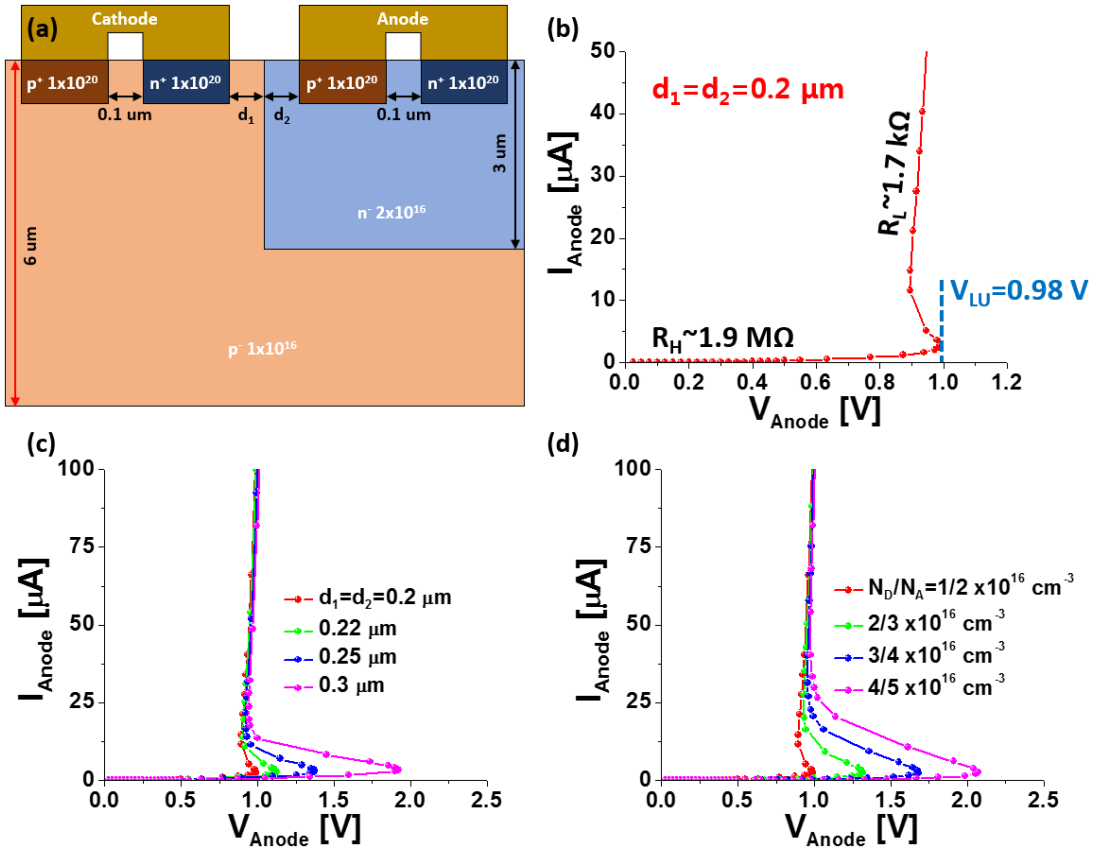


Figure 4.6. (a) A cross-sectional schematic of the lateral PNPN structure, I-V characteristics of (b) d_1=d_2=0.2 um, (c) various d_1=d_2 for 0.2, 0.22, 0.25, and 0.3 um, and (d) various N_D/N_A for $1/2\times10^{16}$, $2/3\times10^{16}$, $3/4\times10^{16}$, and $4/5\times10^{16} cm^{-3}$

4.5.3 Energy and Performance Comparison

Circuit level

Table 4.1 shows the 1-bit addition results of proposed LUPIS and other prior technologies. As explained in Section 4.2, most of the current PIM approaches including selected ones [8, 5, 7, 18] use bitwise-based logic (i.e. calculating IMP, NOR, NOT) to execute 1-bit addition. Thus, they require inevitable sub-cycle executions for the intermediate bitwise computations, creating huge latency bottleneck, e.g., the long latency of SET cycle [8]. In contrast, since our LUPIS design executes the ADD operation in the sensing circuit in a single cycle, the total latency is determined only by the sensing circuit delay, without any extra latency from the memristor. Thus, our proposed design can achieve higher speedup than all the other techniques. Furthermore, our

Table 4.1. performance of 1-bit adder for LUPIS and other technologies

	[5]	[18]	[8]	[7]	**This work**
No. Memristors	3N+5	3N+3	3N+8	N+2	**3N**
Cell efficiency	38%	50%	27%	33%	**100%**
Latency	149.6ns	31.9ns	14.3ns	9.9ns	**33.3ps**
Energy	3237fJ	690fJ	289fJ	214fJ	**7.9fJ**

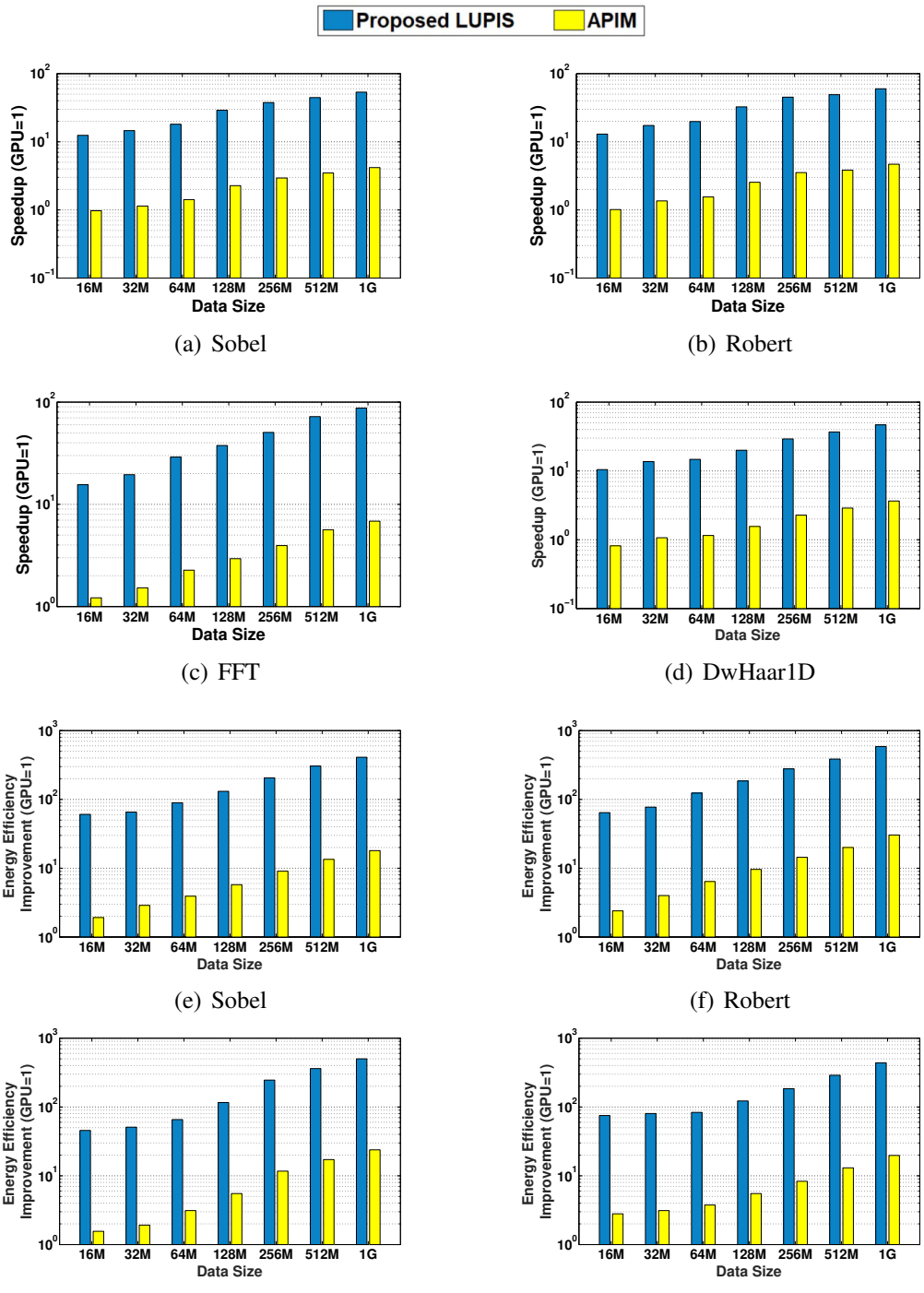
design shows superior cell efficiency since it does not require additional cells. This makes our design a good candidate for NVM-based PIM architectures, in particular, for the storage class memory which should handle a large amount of data.

Application level

There are several applications which can benefit from the PIM-based addition and multiplication. Fig. 4.7 shows the speedup and energy efficiency improvement of, i) the proposed design and ii) a state-of-the-art PIM technique, APIM [8], over the AMD GPU core. The results present that our proposed design can achieve significantly better energy and performance efficiency. Apart from the superior efficiency improvement, our evaluation also shows that LUPIS energy consumption increases linearly with the data set size since the PIM capability can highly hide the cost of data movements. In contrast, the energy and execution time of the GPU case do not scale linearly with the data size, as the larger dataset requires higher costs for the data movements before processing. To sum up, our design can achieve up to 62× speedup and 484× lower energy consumption than GPU architecture. As compared to APIM, the results present 12.7× and 20.9× higher efficiency for speedup and energy respectively.

4.5.4 Overhead

Fig. 4.8 shows the area and latency overhead of our design compared to the TC-adder [7], the most competitive design in cell efficiency, as described in Section 4.5.3. The area overhead has been estimated by the ratio of the additional number of cells and gates as compared to the conventional memory design. As shown in Fig. 4.8a, LUPIS has 21% area overhead,



(a) Sobel (b) Robert

(c) FFT (d) DwHaar1D

(e) Sobel (f) Robert

(g) FFT (h) DwHaar1D

Figure 4.7. Speedup and Energy Efficiency of Proposed LUPIS and APIM [8] over Different Applications.

which outperforms the TC-Adder by 10× since it just takes insignificant modifications to the conventional CSA circuit and no additional cells are required.

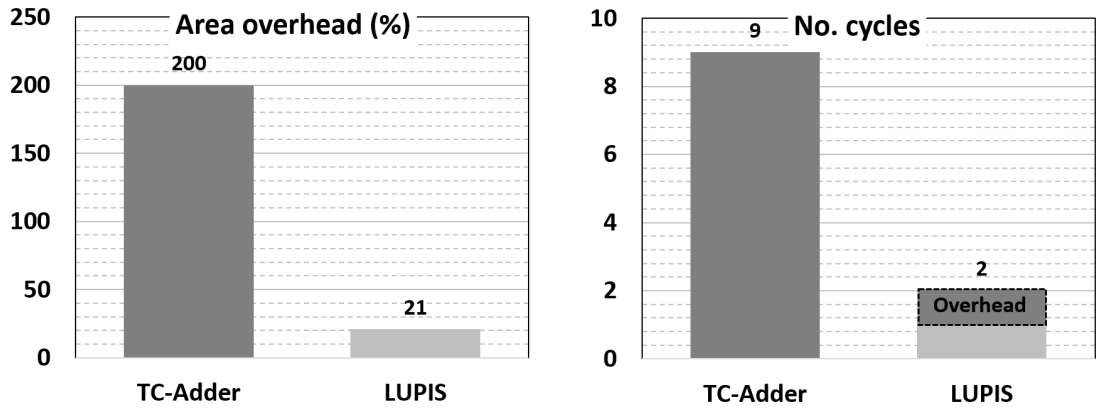


Figure 4.8. The Overhead of Area (a) and Latency (b)

As for the latency overhead, our design requires the write-back step after the sensing operation to initialize the state of thyristor for the next operations. As shown in Fig. 4.8b, the overhead caused by the write-back inclusion is one cycle. This is negligible compared to the latency in the TC-Adder requiring 9 cycles per operation [7]. Furthermore, the overhead due to the write-back step can be utilized in the MUL operation to hide the latency of shift operations as explained in Sec. 4.4.2.

4.6 Conclusion

We have presented an ultra-efficient PIM architecture which effectively enables addition and multiplication inside memory by utilizing the thyristor latch-up effect. The proposed design also addresses the low cell-efficiency issue of other PIM technologies due to redundant cell requirements for logic operations by executing the calculations in the sensing circuitry. The experimental results show that, compared to a state-of-the-art PIM accelerator [8], our design presents 12.7× and 20.9× improvement of latency and energy consumption. In the following chapter, we present the memory architecture techniques for maximizing performance through parallel operation using the PIM techniques covered in Chapter 3 and Chapter 4.

This chapter contains material from Joonseop Sim, Mohsen Imani, Woojin Choi, Yeseong Kim, and Tajana Rosing. “Current-Sensing Efficient Adder for Processing in Memory Design.”

Non-Volatile Memory Workshop (NVMW), 2019. The dissertation author was the primary investigator and author of this paper.

This chapter contains material from Joonseop Sim, Mohsen Imani, Woojin Choi, Yeseong Kim, and Tajana Rosing. "Current-Sensing Efficient Adder for Processing in Memory Design." Non-Volatile Memory Workshop (NVMW), 2019. The dissertation author was the primary investigator and author of this paper.

Chapter 5

High Performance Mat-parallelized PIM

In Chapter 3 and 4, we presented technologies that can improve the performance of NVM-based PIM designs through device and circuit level contributions. To implement these novel PIM designs in the existing NVM architecture, however, requires further performance enhancement factors since NVM memory leveraged DRAM or SRAM organization so far instead of designing architecture that maximizes its strengths. In this chapter and Chapter 6, we present technologies that can further accelerate PIM designs by optimizing the existing architecture to maximize NVM and PIM performance. This chapter presents a multi-bit parallel execution method for local bit acceleration within a mat, while Chapter 6 presents a multi-mat parallel operation for the very long word execution, maximizing performance improvement of our PIM designs.

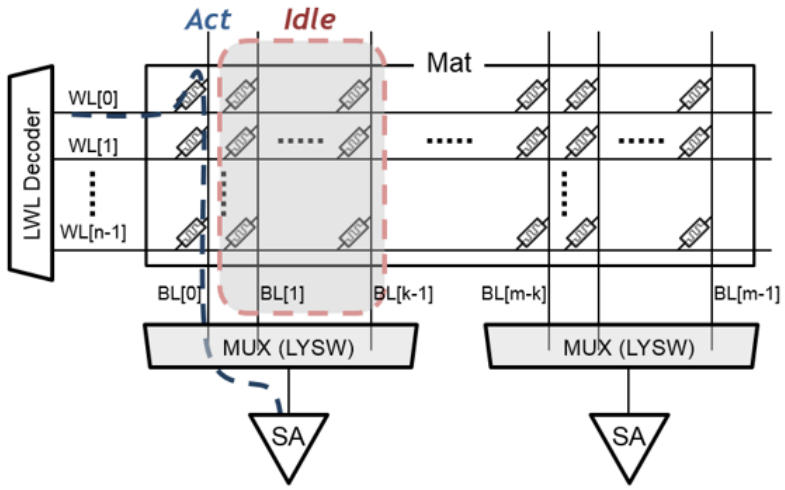
5.1 Introduction

Recent work modifies sense amplifiers (SA) to enable computing [9, 26, 87, 86, 29, 102] in PIM. Since the NVM uses the current-mode SA, the SA size of the NVM is much larger than that of the DRAM which senses a voltage difference between the bitlines [9, 103]. As a result, a SA of the NVM takes charge of multiple BLs, and multiple bits under a single SA cannot operate in parallel. In the conventional memory, this restriction has not been critical since the off-chip bandwidth is limited. The work in [104] showed that, since the commodity memory

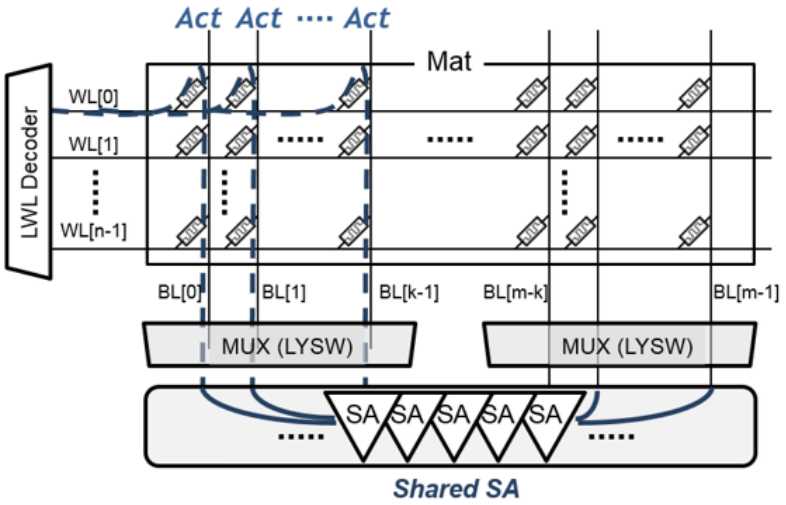
hierarchies were designed targeting the processor sector size, increasing atomic size leads to performance degradation [19]. In a DDR3 channel, for example, the 64-bit I/O interface operates at 800MHz, providing 12.8GB/s of memory bandwidth. To support this channel bandwidth, with a memory subsystem that has 8 chips-per-rank, 8 banks-per-chip, 32 subarrays-per-bank and 32 mats-per-subarray, a bank in the memory, which has a 200MHz internal frequency with an 8-bit prefetch, outputs 8 bits at each channel clock. This estimate implies that a mat atom size is less than 1-bit per single clock frequency, meaning mat-level parallelism does not effect on system performance due to narrow off-chip bandwidth. In contrast, PIM design needs to execute multiple BLs *in parallel* as much as possible since the channel bandwidth is not the bottleneck. However, the existing NVM-based PIM techniques have to execute each bit connected to a single MUX *sequentially* in the mat array due to limitations to the size of SA, resulting in much lower performance.

In this chapter, we present a high-performance PIM architecture which enables the simultaneous execution of the PIM operations for multiple BLs. Baseline NVM architecture is based on the work in [103, 105] whose $m \times n$ mat structure is shown in Fig. 5.1(a). A mat is an atomic access unit for a single memory operation [93]. It has its private SAs and decoders. The WLs and BLs are paired with the local wordline (LWL) decoder and SA to control which bits are selected from the array. Compared to DRAM, the most commercialized memory product, a big difference in NVM design is a SA structure. While in DRAM SA, each BL is connected to an individual SA circuit, NVM array structure places a multiplexer (MUX) in front of the SA, which are used to select a BL connecting to a single I/O line from the multiple BLs, denoted as k lines in Fig. 5.1(a). Therefore, the local BLs tied in a MUX (*i.e.* 8BLs or 16BLs) cannot be accessed in parallel but in bit by bit mode.

Our design, called `MAPIM` (Mat Parallelism for High-Performance Processing in Non-volatile Memory Architecture), is built with two novel components: multi-column/row latch (*MCRL*) and shared SA routing (*SSR*). Fig. 5.1(b) shows the sensing scheme of the proposed `MAPIM`. In baseline NVM structure, the number of SAs in a mat ranges from 32 to



(a) Conventional NVM sensing scheme



(b) Proposed ***MAPIM*** sensing scheme

Figure 5.1. Conventional NVM and proposed `MAPIM` sensing structures

64, which is similar to the number of bits in a word. `MAPIM` shares the SAs in a mat and thus enables word-size of multiple bits to be sensed in parallel. The *MCRL* activates multiple WLs/BLs at the same time, so that the PIM operations are requested in a row/column-parallel way. The *SSR* allows the requested multi-BLs to use multiple SAs across a MUX, thus fully utilizing the SAs of the mat for the parallel execution in the PIM operations.

Our contribution can be summarized as follows:

- We show that the SA size is a significant constraint to achieve a further performance of PIM operations.
- Our `MAPIM` design parallelizes multiple-BLs execution within a mat. We also show the detailed circuit-level design with the robustness and sensitivity analysis.
- The proposed *MCRL* design enables efficient multi-row/column activations. Unlike the state-of-the-art design, it holds the activation signals with a wide margin of μs order, allowing to select any target operands for the parallel PIM computations.
- With the *SSR* technology, the proposed `MAPIM` fully utilizes the limited number of SAs located in a mat, when the PIM operations are performed for a large number of rows and columns.

Our evaluation shows that the proposed `MAPIM` achieves up to 339× speedup and 221× energy saving compared to AMD Radeon R9 390 GPU architecture, and 16× speedup and 1.8× energy saving compared to the state-of-the-art PIM designs [9, 10].

5.2 Related Work

Several works modified the SA to enable the PIM function on the existing architecture [9, 26, 87, 86, 29, 5, 102]. However, the current-mode SA of NVM, which takes a larger area compared to the voltage-mode SA of DRAM, hinders the local parallelism since the multiple BLs assigned to a multiplexer should operate in sequence. Our work addresses this lack of parallelism by proposing a novel design utilizing address latching and SA sharing techniques. Parallelism at rank [106], bank [107], subarray [108] has been actively studied, but mat-level parallelism has not been deeply explored since it is not very effective in conventional off-chip memory due to the boundness of channel bandwidth. However, PIM design can fully utilize the mat parallelism to performance improvement. In this chapter, we enable mat-level parallelism with insignificant modification to state-of-the-art NVM architecture.

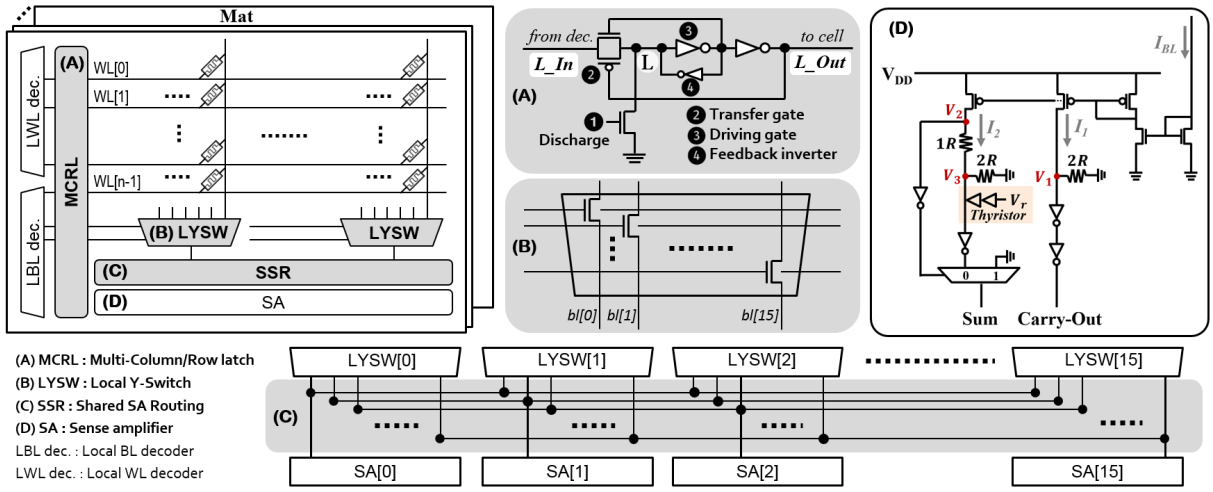


Figure 5.2. Mat structure of proposed `MAPIM` design

5.3 Proposed `MAPIM` design

5.3.1 Overview

Fig. 5.2 shows the mat structure of `MAPIM`. Mat is the building block of a *bank*, a fully-functional memory unit [1]. It consists of cell arrays, SA, and the local decoder. The figure does not show the upper bank level design, which remains the same as before [103]. We introduce two new structures to enable mat-level parallelism: multi-column/row latch (*MCRL*) and shared SA routing (*SSR*). A local bitline (LBL) decoder is placed under a local word-line (LBL) decoder to ease layout. The decoded signals from the LWL decoder and LBL decoder are transferred to the WLs and local Y-switch (LYSW), respectively, to activate the corresponding WLs and BLs as shown in Fig. 5.2. *MCRL*, denoted as (A) in Fig. 5.2, refers to both multi-column latch and multi-row latch. It serves as the buffer that keeps the sequentially activated WLs and signals to the LYSWs from the decoder and enables them simultaneously. *SSR*, marked as (C), enables multi-bit execution in parallel by sharing the limited number of SAs in a mat to the requested BLs. NVM cell is sensed by the resistance difference between a logical 0 and 1, so it is less sensitive to the BL capacitance as compared to the conventional DRAM. This allows NVM SAs to be delocalized within a mat and be flexible to associate with segmented BLs.

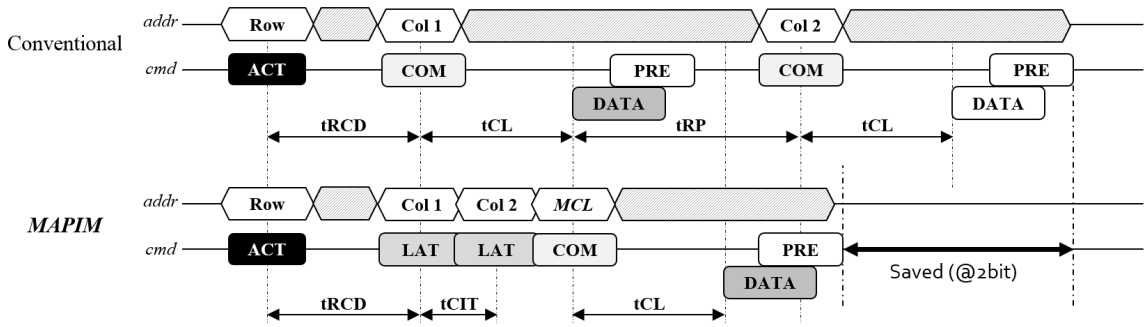


Figure 5.3. Timing graph of two requests from two different BLs

5.3.2 Multi-Column/Row Latch (*MCRL*)

We next discuss each of our design's components in detail. Fig. 5.2(A) shows a circuit unit of the proposed *MCRL* corresponding to a single WL as a representative. It consists of four components: ❶ a discharge transistor, ❷ a transfer gate, ❸ a driving gate, and ❹ a feedback inverter. First, the discharge transistor resets all WLs to ground. In the second step, the transfer gate passes the decoded signals from the local decoders to the latch. The transfer gate is made up of two transistors, an n-channel MOSFET (*NMOS*) and a p-channel MOSFET (*PMOS*), connected in parallel since they need to transfer both 0 and 1 without loss of strength. Following two inverters, a driving gate and a feedback inverter, drive the decoded signal to '*L_Out*' in a cell array. When the input signal from the decoder is low, a node '*L*' potential does not change since the initial status of the node is low. When the decoded signal is high, *i.e.*, a corresponding WL is *activated*, the node '*L*' potential flips to one and the inverted potential turns the transfer gate off, so the input signal is latched in the *MCRL*. The feedback inverter is used to prevent the node '*L*' from floating when the transfer gate is off. It needs to be weaker compared to the transfer gate for faster switching. In our experiments, the drivability of the feedback gate is set to 1/3 of transfer gate for easy signal transference. The circuit evaluation of our *MCRL* design is shown in Sec.5.4.2.

5.3.3 Shared SA Routing (*SSR*)

As mentioned in Sec.5.3.1, NVM is less sensitive to BL length compared to DRAM. NVM has a resistance-based operation, so its SA works by detecting current change across the NVM cells. The work in [1] showed that the current-mode SA has a sub-linear dependency on BL length, so cells can be sensed out of the cell array. Moreover, NVM read is not destructive while DRAM's is. Thus resistances such as MUX can be placed in front of SA. This gives us more room to increase parallelism with our *SSR* design. Fig. 5.2(C) shows the proposed *SSR* design. Compared to the conventional NVM SA design in which each SA corresponds to their own LYSW as shown in Fig. 5.1(a), the LYSWs of our design share the SAs within a mat as shown in Fig. 5.2(C). 16 BLs in an LYSW, described in Fig. 5.2(B), are allocated in SA[0]∼SA[15]. All other LYSWs in a mat, *i.e.*, LYSW[0]∼LYSW[15], share the SAs in the same sequence. Consequently, 16BL-sets of all LYSWs in a mat have the same SA allocation as shown in Fig. 5.2(C). In this work, we have 16 BLs in an LYSW and 16 LYSW in a mat for illustration purposes. However, the actual array size can be flexible with a typical range of 8 BLs and 8 LYSWs to 32 BLs and 32 LYSWs as a function of the memory granularity. The LBL decoder has two steps. Pre-decoding selects an LYSW in a mat. A subsequent decoding step selects a BL in that LYSW. At a given LYSW selected in the pre-decoding, proposed *MCRL* holds the consecutively activated BL addresses. Then the multiple signals buffered in the *MCRL* enable multiple-columns to activate by turning on NMOS transistors in an LYSW as shown in Fig.5.2(B). The selected BLs are read out simultaneously. The accessed data is assigned to the corresponding SAs by our shared-routing method.

The latency of *SSR* is shown in Fig. 5.3, where *tRCD* is the row activation time, *tRP* is the precharge time, and *tCL* is the column read time. A read cycle (tRC) is represented as $tRC = tRCD + tCL + tRP$. Data restoration time is zero since the NVM is inherently non-destructive in sensing operation. [109]. We define a new timing parameter, *tCIT*, the time interval between sending two consecutive column addresses. The top of Fig. 5.3 presents the timing graph of the

baseline in which the local BLs are read in sequential mode. The row address activates a WL and the first column address (*Col* 1) activates a BL. Then the selected bit is sent to an assigned SA. As our sensing circuit can perform a 1-bit addition, which will be explained in Sec. 5.3.4, a *carry-out* of 1-bit addition is written back to the memory cell in the next bit and a *sum* is sent to the cache. Once the *Col* 1 is activated, 1-bit operation and precharge have completed, during *tCL* and *tRP*, then the *Col* 2 is activated for the next bit operation.

`MAPIM` temporarily holds on multiple columns requests, *e.g. Col* 1 and *Col* 2 as shown on the bottom of Fig. 5.3, in the latch and they are activated simultaneously with the *MCRL* activation request. While the conventional operation uses two *tCL* and one *tRP* for the two-bit calculation as shown in Fig. 5.3, `MAPIM` uses one *tCL* for the same operation, which can accelerate the requested operation by saving *tCL*. Note that *tCIT* is negligible compared to *tCL* and *tRP* since multiple column latches occur inside the decoder logic whereas *tCL* and *tRP* are between the on-chip memory controller and the off-chip memory. In N-bit operation, conventional design takes N×*tCL* + (N-1)×*tRP*, while `MAPIM` takes *tCL* + (N-1)×*tCIT*. Since `MAPIM` consumes a fixed amount of time, *tCL*, as the number of bits increases, the overall speedup of `MAPIM` increases due to saving column read and precharge latency.

5.3.4 Sensing Circuits for Arithmetic Operation

Fig. 5.2(D) is the sensing circuit for the arithmetic operation in `MAPIM`. Our design modifies a conventional SA [10], which determines the logical 0 and 1 by reading the cell resistance, to perform 1-bit addition. We exploit 1-bit addition using our sensing circuit since the addition is a building block of the order of operations for our application test. The current mirror in Fig. 5.2(D) copies the I_{BL} to I_1 and I_2 and delivers them to *Carry-out* (C_{out}) and *Sum* nodes, respectively. I_{BL} is the read current in a BL, accumulated from the selected three bits when three rows are activated in a cell array. Fig. 5.4 shows how the circuit carries out 1-bit addition. Since each memristor has two states, 0 and 1, there are four equivalent combinations of I_{BL}: I_{000}, I_{100}, I_{110} and I_{111} in the case of three-bit calculation. There are three voltage nodes: V_1, V_2, and V_3,

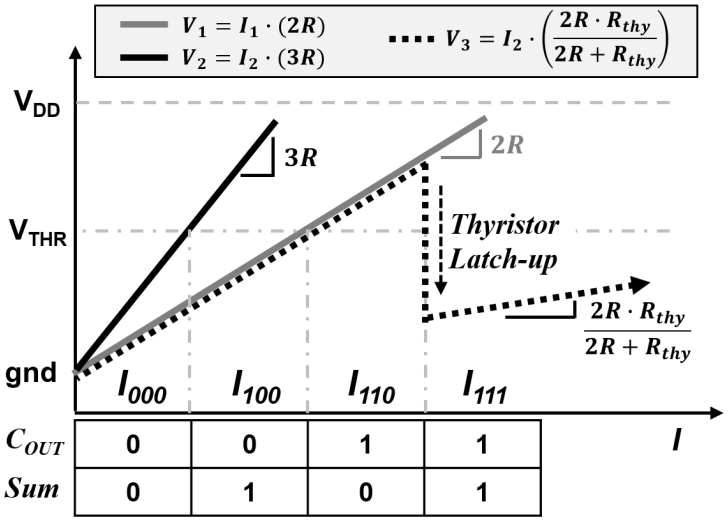


Figure 5.4. Sensing circuit for 1-bit addition

as shown in Fig. 5.2(D), whose potential determines the final outputs of the *Sum* and C_{out}. The voltage of each node is determined as a function of the current, C_{out} and *Sum* nodes exhibit the corresponding values in each case of I_{BL} and finally result in the execution of 1-bit addition. Our sensing circuit utilizes the thyristor latch-up effect [101] to execute the I_{111} region to create the desired C_{out}=1 and *Sum*=1.

5.4 Experimental Results

5.4.1 Experimental Setup

We evaluated the performance and energy consumption of the proposed design as compared to the work in [9, 10]. We used *VTEAM* memristor model [91] for our cell design with R_{on} and R_{off} of 10KΩ and 10MΩ respectively. Circuit-level simulations are performed with Cadence Virtuoso and Spectre based on a 45*nm* CMOS process. We compared the performance and energy of `MAPIM` with AMD Radeon R9 390 GPU with 8GB memory. We modified Multi2sim [110], a cycle-accurate CPU-GPU simulator, to evaluate the impact of different PIM designs when the parallelized instructions are performed with the PIM operations. We experimented with four applications, *Sobel*, *Robert*, *Fast Fourier transform (FFT)* and *DwHaar1D*.

5.4.2 *MCRL* Robustness

Fig. 5.5 presents the result of circuit evaluation of *MCRL* in MAPIM and comparison to the state-of-the-art design in [9]. For the latch design in [9], we observe that i) low signal drivability leads to large area overhead to boost the signal across gates, and ii) the node, which holds the signal (denoted as 'L' at Fig. 5.2(A)), is likely to be floating. These issues may make the latch unable to hold the signal long enough for the multiple activations due to the leakage-current dissipation. As shown in Fig. 5.5, the signal of the 'L' node in *Pinatubo* [9] is so weak as that only 48% of the total signal strength at *L_In* node is transferred. Moreover, the holding latency is too short as to keep the signal less than a few *ns*. Considering our experimental setup, tRC=tRP+tCL+tRP=28ns, the signal holding time in [9] is too low to keep the signal during only one cycle.

In contrast, MAPIM proposed a robust circuit design for the multiple-addressed activation. This benefit derives from the following novelties in our design: 1) the transfer gate can deliver both high(1) and low(0) signals from the decoder without the loss of strength. 2) the feedback inverter prevents the node inside the latch from floating, which suppresses the static power consumption. As seen in Fig. 5.5, *MCRL* holds the activated signal for over 600ns. Although we show the data for the range of 0$\sim$600ns for easy illustration, our experimental results show that *MCRL* retains the activated signal over a few μs without losing signal strength. This allows MAPIM to keep the signal long enough for the three-row activation which is common in PIM operation. Furthermore, the signal at the 'L' node is as strong as that at *L_In* node in *MCRL* design, thus ensuring robustness.

5.4.3 Performance Sensitivity to the Number of Bits

We compare the mat-parallelism in our design with two state-of-the-art PIM designs, *Pinatubo* [9] and *LUPIS* [10], which execute sequential operations in a MUX. Fig. 5.6 presents the latency improvement of MAPIM for *addition* and *multiplication*. The data in Fig. 5.6 is

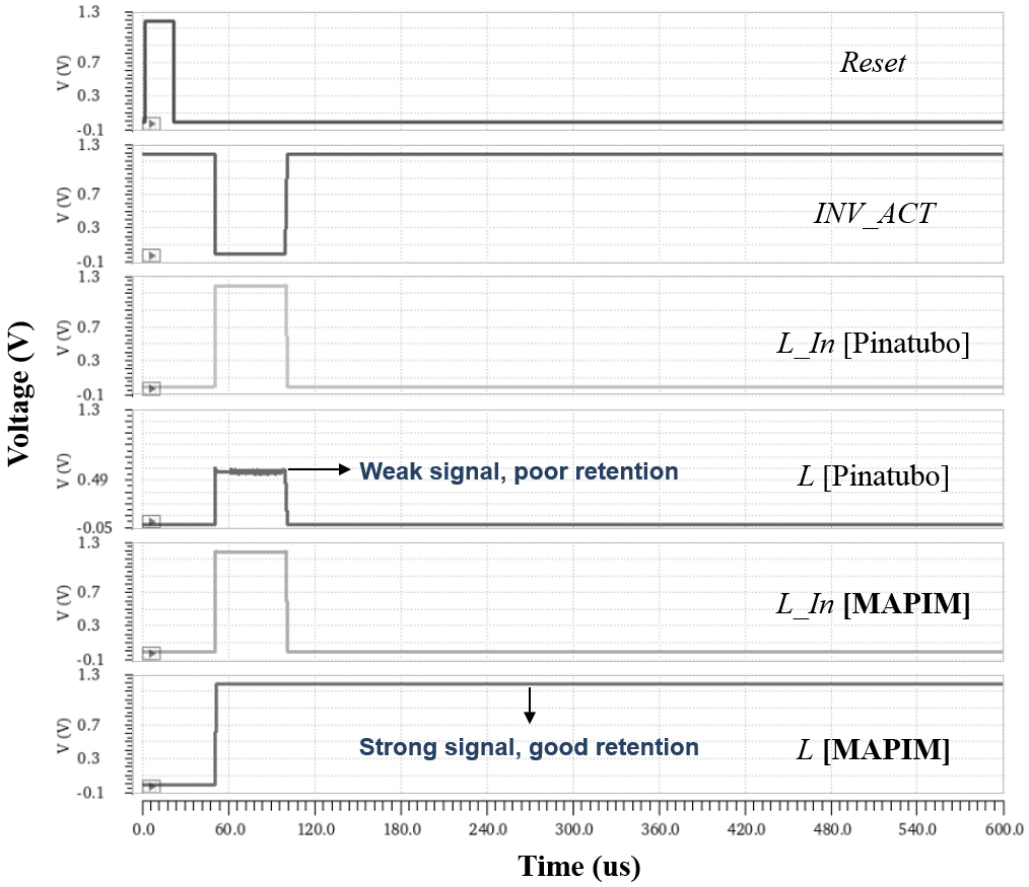


Figure 5.5. Circuit evaluation of the multi-activation latch between Pinatubo [9] and `MAPIM`

normalized to the latency of *Pinatubo*, set to 1. The latency of addition is evaluated from the design in Fig. 5.2 while considering the timing parameters shown in Fig. 5.3. The latency of multiplication is estimated by Eq. in [111],

$$t_{MUL} \propto [(M-1)+(N-2)] \cdot t_{carry} + (N-1) \cdot t_{sum} + (N-1) \cdot t_{and}$$

where M and N are the multiplicand and multiplier, and t_{carry}, t_{sum}, and t_{and} are the latencies for evaluating *carry*, *sum*, and *AND*, respectively. Although the latency improvement of `MAPIM` is around 4 times for 2-bit operations, the improvement grows as the size of operands increases, with 36× improvement for 32-bit operations. Both [9] and [10] use sequential mode in multi-bit operation, thus having much higher latency. On the contrary, `MAPIM` buffered all required

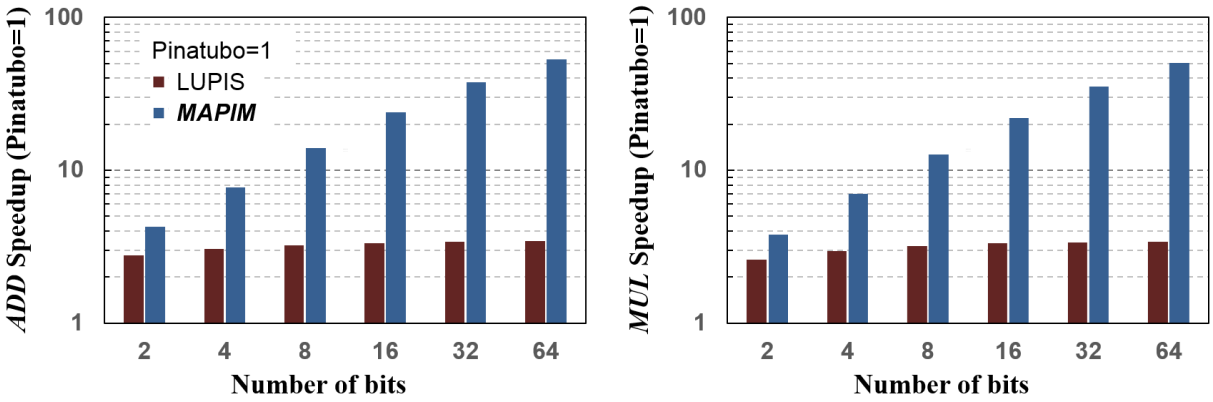


Figure 5.6. Latency improvement of MAPIM for arithmetic operations

addresses and activated them in a single request, so it takes one *tCL* regardless of the number of bits, hence much better performance as the number of bit increases.

5.4.4 Parallelism Efficiency in Applications

We evaluate the efficiency of MAPIM with four OpenCL applications. Fig. 5.7 shows the speedup and the energy efficiency of proposed MAPIM and two state-of-the-art PIM designs, *Pinatubo* [9] and *LUPIS* [10]. All results are normalized to the value of unmodified AMD GPU. The results show that the three PIM designs outperform the GPU-based computation for the wide range of the memory size. It is because PIM designs reduce data movement costs. Among the PIM designs, MAPIM achieves the best performance improvement as compared to the other PIM techniques [9, 10]. For example, the proposed design is 339× faster and 221× more energy efficient as compared to the GPU, and 16× faster as compared to the state-of-the-art PIM designs, *Pinatubo*[9] and *LUPIS*[10]. This suggests that utilizing mat-level parallelism is key to designing highly efficient PIM architectures.

5.4.5 Overhead

MAPIM adds a latch circuit and a routing block to each mat array. The area overhead has been estimated by the ratio of the additional latch and sharing routing area over the corresponding cell area. We assume 32 BLs-per-LYSW and 32 LYSWs-per-mat. The area estimates from

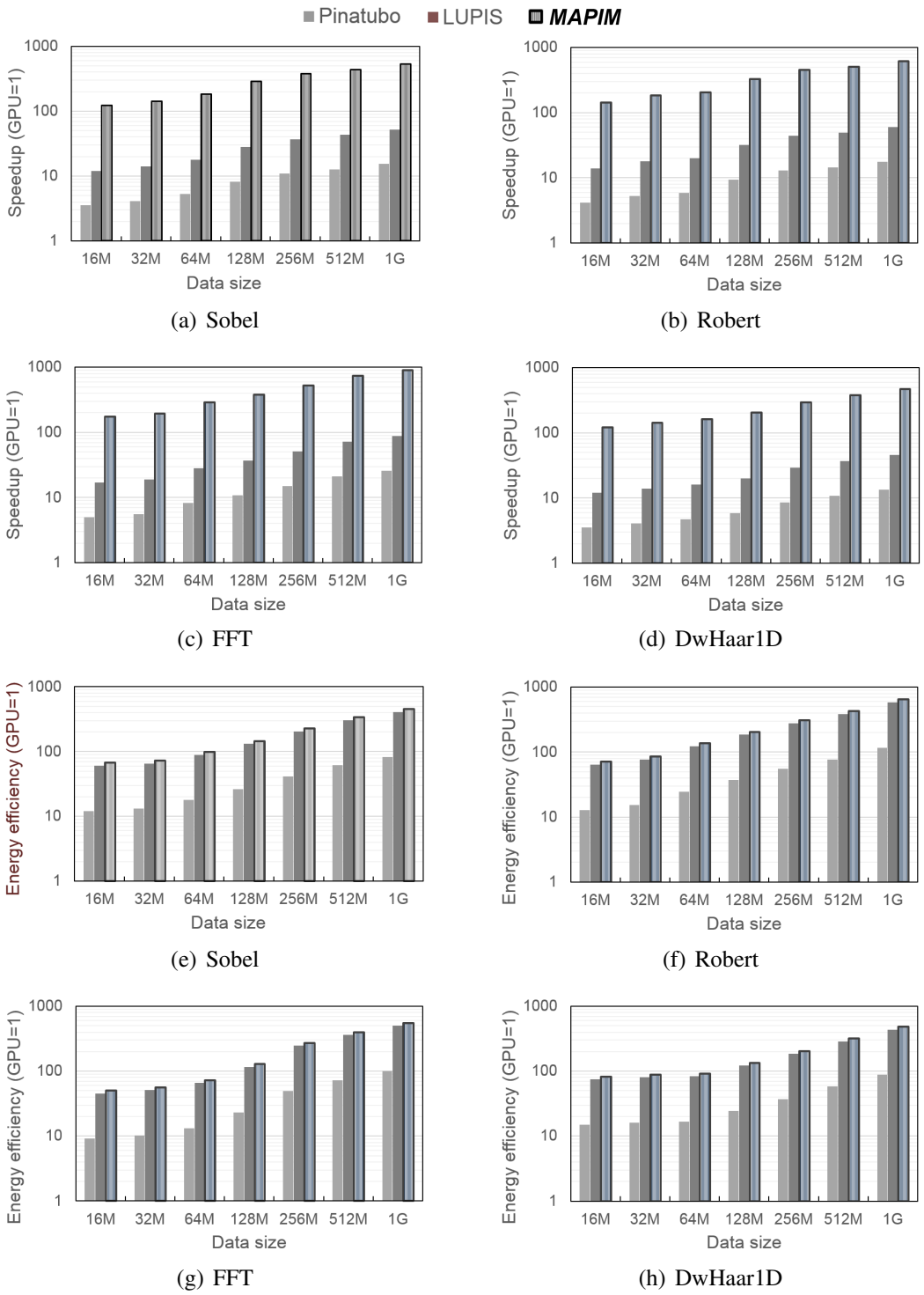


Figure 5.7. Speedup and energy efficiency of proposed MAPIM to other PIM architectures [9, 10] for different applications.

Cadence p-cell data with 45*nm* process technology, the overhead incurred in MAPIM occupies an area of $1.5um^2$, which is a 0.57% area overhead compared to a corresponding row of sub-array. This is a very similar value to the work in [9] which has 0.49% area overhead but works for better performance as shown in Sec. 5.4.2. MAPIM also shows outstanding area efficiency compared to the work in [108], a state-of-the-art latch design for DRAM, showing an area of $42.9um^2$.

5.5 Conclusion

In this chapter, we presented a high-performance PIM architecture which enables mat-level parallel operations. The proposed design accelerates the PIM applications whose performance is not restricted by the off-chip channel bandwidth. The experimental results show that our design presents 16× and 339× performance improvement compared to the state-of-the-art PIM designs [9, 10] and the GPU architecture [36], respectively. The following chapter will discuss a multi-mat parallel operation for long byte word processing common to many big-data algorithms.

This chapter contains material from Joonseop Sim, Minsu Kim, Yeseong Kim, Saransh Gupta, Behnam Khaleghi and Tajana Rosing, "MAPIM: Mat Parallelism for High Performance Processing in Non-volatile Memory Architecture", IEEE International Symposium on Quality Electronic Design (ISQED), 2019. The dissertation author was the primary investigator and author of this paper.

This chapter contains material from Joonseop Sim, Minsu Kim, Yeseong Kim, Saransh Gupta, Behnam Khaleghi, and Tajana Rosing. "Multi-bit Parallelized Sensing for Processing in Non-volatile Memory." Non-Volatile Memory Workshop (NVMW). 2019. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Multi-mat Parallelized Execution for Hyperdimensional Computing

Today's emerging computing techniques, such as deep neural network (DNN) graph processing and hyper dimensional (HD) computing, are increasingly requiring very large data operations. This inevitably requires the system to have higher performance and parallelism than ever before. As of Chapter 5, we are presenting fine-grained parallelism to further accelerate PIM design through architectural optimization of existing commodity memory. In the previous chapter, we covered intra-mat parallelism techniques for the acceleration of local bit processing. In this chapter, we present inter-mat parallelism method for long word processing that can maximize performance improvement. We present an example of a long word operation used in HD computing, and demonstrate how to make simple long word read processing possible by taking full advantage of the existing architecture for each logic operation and searching.

6.1 Introduction

Cognitive systems that have the ability to analyze, interpret, and reason like the human brain are gaining in popularity. The recent spike of neural networks (NNs) has accelerated the interest in bio-inspired computing [112]. However, conventional computing based on the binary number becomes increasingly inefficient due to the computation complexity and limited resources of embedded devices [113]. Hyperdimensional (HD) computing is a new paradigm of

brain-inspired computing that emulates the neuron's activity. Instead of using binary numbers, it represents information by allocating data points to vectors in a high-dimension space, called *hypervectors* (HVs). HD computing has a number of advantages over the existing machine learning algorithm such as deep neural networks: 1) fast and one-shot learning, 2) simplicity of calculation, e.g., no need of back propagation, and 3) robust against failure and noise [114].

The HD computing processes the high dimensionality of the hypervectors, e.g., at least $D = 10,000$. Thus, the ideal computing platform should offer i) high parallelism for arithmetic operations and ii) sufficient memory bandwidth. Prior works have implemented HD computing on various hardware (HW) platforms. For example, the works in [115, 114, 116] discuss the HD model based on the CPU and FPGA. The work in [117] designed a ReRAM-based HD accelerator with an error-resilient and energy-efficient architecture. However, high-efficient HD computing is still challenging even on these cutting-edge platforms. The CPU has a limited number of cores, thus significantly limiting potential parallelism that the HD computing can achieve. Although the GPU is an attractive platform in terms of the parallelism, the last-level cache (LLC) size is limited. For example, when $D = 10,000$, the cache memory can only store 120 hypervectors, which is insufficient for large size training datasets of practical classification problems. In FPGA, the cost of the memory accesses dominates the execution time, since it has to fetch a large amount of data from the main memory of the host systems.

Processing-in-memory (PIM) is a promising solution to address HD's data movement issues and offer high parallelism. It places the computation into the memory instead of loading all data to an on-chip cache side, hence reduces the costly data movement between processor and memory. Moreover, PIM can parallelize HD operations by utilizing a component-wise operation [118]. However, an existing memory architecture has restrictions that limit HD's efficiency. First, the size of a single mat, the atomic access unit for a single memory operation [93], is restricted to 1K$\times$1K array because of the physical limitations, parasitic resistance and capacitance [4, 119]. This clearly limits processing 10,000-dimensional bits of a single HV. HD inference with k classes and D=10,000 dimensions requires $k\times$D multiplications and addi-

tions concurrently. Therefore, as shown in Fig. 6.1(b), a single HV needs to be segmented into multiple mats and sequentially executed in traditional architecture. Second, since the DDR-based memory operations are designed targeting 64-byte cache-lines, the atomic size in a single mat array is no more than 2-bits [93]. For these reasons, the existing memory hierarchy is unsuitable for dealing with a high dimension of HD computing. A simple approach to address this is to reduce the number of dimensions in a single HV. However, this results in an unacceptable loss of accuracy [120].

In this chapter, we propose a novel PIM architecture, called Multi-mat Parallelized Execution for Hyperdimensional Computing (*MAPLE*), which enables concurrent activation of multiple mats corresponding to an HV. By enabling a bypass function to the existing decoder scheme, our design supports concurrent execution of HD computing with high dimensionality, resulting in no accuracy loss and latching delay. This provides an impression that multiple mats participate in executing an HV as if they were a single bank. Consequently, our design improves latency by up to 10$\times$ as compared to the work in [11].

Main contributions of *MAPLE* are summarized as follows:

- We present a novel design, *MAPLE*, which enables fast and low-overhead HD computing whose HVs having thousands of bits in non-volatile memory architecture.
- The proposed *global decoder bypass* (GDP), a design component of *MAPLE*, stores a vector segmented to multiple mats but **concurrently** executing them during HD training.
- We also propose a global wordline search technique, called *shunt via routing* (SVR), which performs a **one-shot** search across all the mats storing an HV with a simple via contact implant.
- Our design enables both computation and memory functions to be performed in the same cell array, unlike most of prior PIM accelerators which needs an extra array for PIM functions.

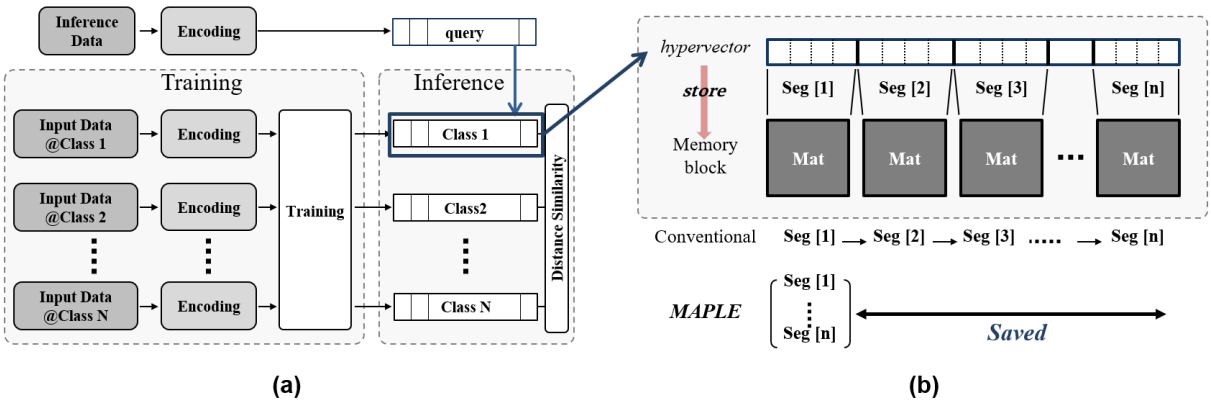


Figure 6.1. (a) HD overview and (b) conventional and proposed *MAPLE* comparison for HD PIM

Our evaluation shows that *MAPLE* achieves up to 10× speedup and 51× energy saving compared to the state-of-the-art HD accelerators [11].

6.2 Background and Related Work

In this section, we first describe the three step operation of HD computing: encoding, training, and inference. We next review the related work on the implementation of HD computing and the parallelism we are aiming at in this study.

6.2.1 Hyperdimensional Computing

Hyperdimensional computing uses high-dimensional vectors with ∼10,000-bits as a building block of representation [115]. HD computing model has three steps: encoding, training, and inference as shown in Fig. 6.1(a). The encoder maps input data to the HVs for each class in the high dimensional space. The encoded HVs are then combined with a training model to create a single HV standing for the class [114]. This association of multiple HVs involves arithmetic operations, *e.g.*, *binding* operation denoted with '×' and *bundling* operation denoted with '+'. All trained HVs for each class are then stored in the memory and ready for an associative search with the input query HVs. For inference, the same encoding step maps an unknown input data to a query HV. The associative memory is then enabled to search the similarity of the query

HV across all trained HVs. The class HV with the highest similarity to the query HV is the final result of the inference. The number of classes can vary depending on the application, which is determined by the user. TCAM is commonly used as a building block for the similarity check [121]. There are two searching methods used depending on the type of HV, Hamming distance for binary HVs and cosine similarity for non-binary HVs. In this work, we use binary HVs and Hamming distance metric for a similarity check to reduce the computational cost.

6.2.2 Related Work

Our research aims to maximize the performance of HD computing through memory optimization while reducing the cost of moving data by executing a very long byte operation in memory. Previous studies have successfully implemented HD computing based on various HW platforms. For example, the work in [122] fabricated an end-to-end HD computing solution by leveraging CNT-FET and Resistive RAM. They used RAM resistance and variation of the input current of the CNT-FET to transfer the features of the input data to the query HV. The work in [117] implements HD computing based on in-memory kernels that enable multiplication, addition, and permutation by analog operations using 3D vertical RRAM. The author in [123] implements HD computing with variable model precision, capable of meeting user constraints on different FPGA platforms. Although the state-of-the-art designs described above have effectively implemented HD computing using emerging memory and FPGA engines, it is difficult to overcome the massive data movement problem because computing locations follow traditional methods. *MAPLE* addresses this data access challenge of HD computing by implementing a computing unit inside the emerging NVM.

Many previous studies on memory parallelism have also been carried out. Rank [106], bank [107] level parallelism have been previously proposed. They placed their own row-buffers to the rank and bank levels, respectively, to allow for individual operation, thereby granting parallelism. The work in [103, 108] introduced subarray-level parallelism in the DRAM architecture. By inserting the row address latches to each sub-array, it can hold an active

WL and interleave sub-banks to hide memory access latency. To the best of our knowledge, sub-array parallelism has been the lowest level in the research on the memory parallelism, but mat-level parallelism has not explored so far. *MAPLE* is a study on parallelism at mat level, an atomic access unit in the memory architecture, which has excavated additional performance enhancements allowed by PIM characteristics. The work in [9] proposes the multiple WL activation, which possibly can be used to store an HV into a single mat. However, this requires at least 10 WLs activation, which requires significant circuit overhead for inserting latch to every mat. Besides, the design in [9] allows parallelism by overlapping a portion of execution to other executions, like *interleaving* [124]. In contrast, *MAPLE* enables the concurrent execution of multiple mats; hence it makes the system faster without latching delay.

Several works proposed TCAM (*ternary content addressable memory*)-based design for associative searching. CMOS-based TCAMs consisting of two SRAM cells show low area efficiency since their cost per bit is 8X more than SRAM [125]. NVMs are a promising device for substituting the CMOS design due to their high density and low static power [126, 11]. The work in [11] exploits *Resistive Random Access Memory* (ReRAM) for TCAM design. However, the design is used only for a searching engine, hence incurs additional area overhead to the memory cell. In contrast, *MAPLE* is comparable with existing memory design. Our design utilizes a memristor cell for both memory and TCAM cell by simply switching the operation mode, whose detail is explained in Sec. 6.3.2.

6.3 *MAPLE* Architecture

6.3.1 *MAPLE* overview

Fig. 6.2 shows the bank structure of our design. The figure does not include the upper bank level architecture which remains the same as the design in [4]. We assume an x8 ReRAM chip configuration with a DDR3-compatible interface and 64-bit internal prefetch width, which are distributed to 32 mats in a subarray of the banks. The mat size of our design is 1024 WLs $\times$

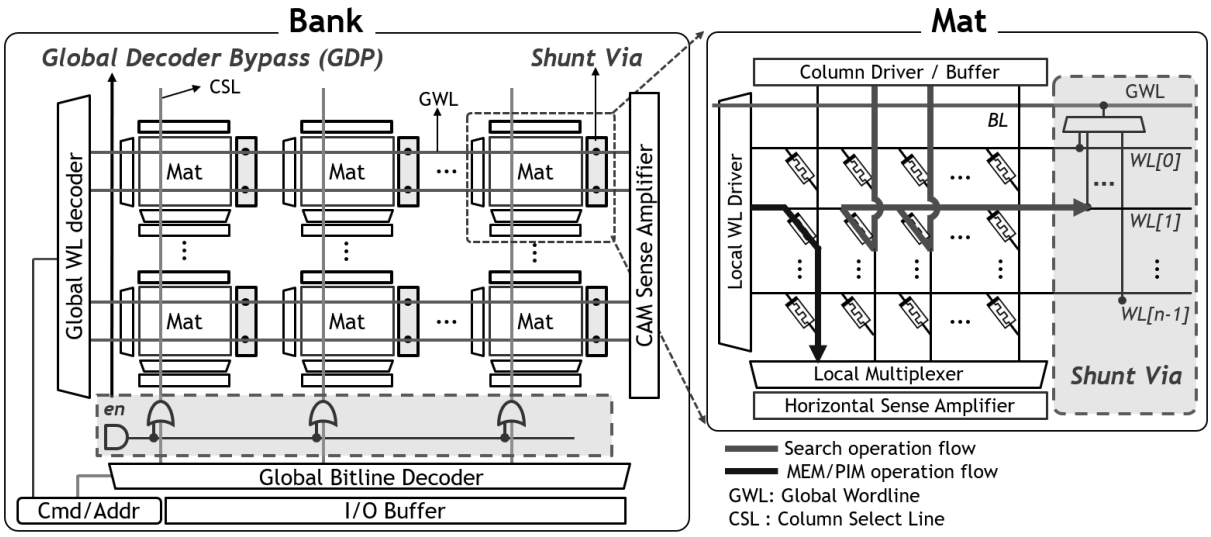


Figure 6.2. Schematic of the Bank Structure in MAPLE

1024 BLs, which is considered the maximum size due to parasitic resistance and capacitance in today's 28nm technology [119]. Hierarchical WL structure [31] addresses multiple mats where each mat has sub-wordline drivers and local bitline sense amplifiers (SAs). We use a memristor device as the building block of *MAPLE* based on the model in [91]. Our memory block can operate in dual mode: memory and computation. In memory mode, it executes read/write operations in the same way as the DDR-based memory [127]. For HD computation, we propose two novel design components: *global decoder bypass* (GDP) and *shunt via routing* (SVR). GDP enables simultaneous activation of multiple mats in a bank for HD training. Then, for speeding up HD inference, SVR parallelizes associative search of a single HV segmented into the multiple mats. In the following subsections, we discuss our design components in detail for each step of the HD computation.

6.3.2 Global Decoder Bypass (GDP) for HD training

Global Decoder Bypass (GDP)

HD training requires massive component-wise calculations, e.g., multiplication and permutation, to create the class HVs from the multiple HVs. In contrast to existing DDR-based NVM design, which accesses each mat in sequential order, we propose multi-mat activation

technique for executing HVs which have 10,000 bits each. Hierarchical WL structure, set as our baseline, uses two decoding steps to select a bit cell. First, the global decoder selects a mat in a bank using the *column select line* (CSL), and then local decoder selects a single bit in the selected mat. We place OR gate array between the global and local WL decoder, called *global decoder bypass* (GDP), as shown in Fig. 6.2. In memory mode, GDP is disabled (*en*=0). GWL decoder delivers the original decoded signal to each mat, and it accesses the requested bit to a particular mat according to the conventional method. In computation mode, however, GDP is enabled (*en*=1). All mats corresponding to one row in the bank are activated simultaneously regardless of the global decoded signals. In the baseline architecture, the address information of the local decoder is common to all mats corresponding to one subarray row, but when a specific mat is selected by the CSL, local addressing occurs only in the corresponding mat. Since the proposed GDP simply bypasses the address information already assigned to the CSL, it does not require additional address assignment while enabling simultaneous activation of the same WL in all mats. Then all the bits consisting of the HV can be trained in parallel.

The parallelism benefits of *MAPLE* are shown in Fig. 6.3, where tRCD is the row activation time and tRP is the precharge time. To compute an HV in a conventional way, the row decoder activates the WL only in a single mat. Multiple mats consisting of an HV are accordingly in a queue during single-mat processing. For example, in Fig. 6.3(a), once the computation is done in the Mat 1, consuming tRCD and tRP, then the Mat 2 is activated for the next partial bits operation. In this way, all other mats in charge of a single HV run in sequence. In contrast, as shown in Fig. 6.3(b), *MAPLE* executes an HV in a single cycle. By enabling *GDP*, a single row-address command activates all mats which have information of an HV. We assume tRCD includes component-wise execution time. Consequently, *MAPLE* only consumes a single tRCD for executing an HV while the conventional access takes $k \times tRCD + (k-1) \times tRP$, where k is the number of bits to be requested.

HD training uses arithmetic operations to combine multiple HVs and generate a single HV for each class. We use *MAGIC* [6] to implant logic operations. A memristive memory

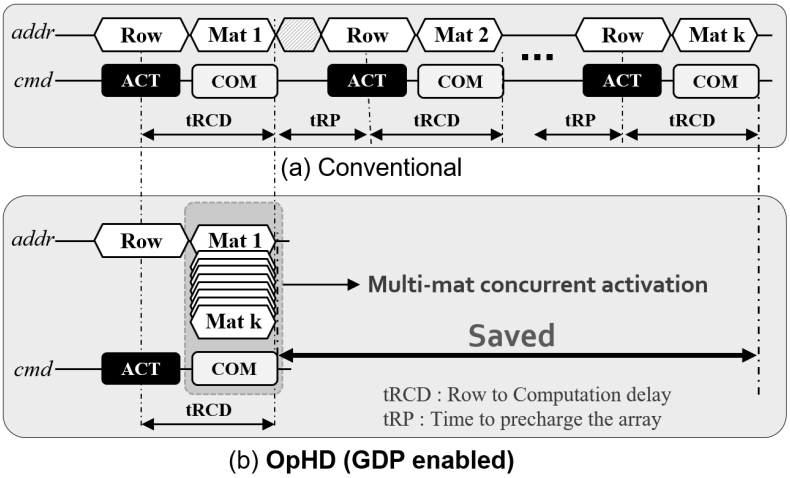
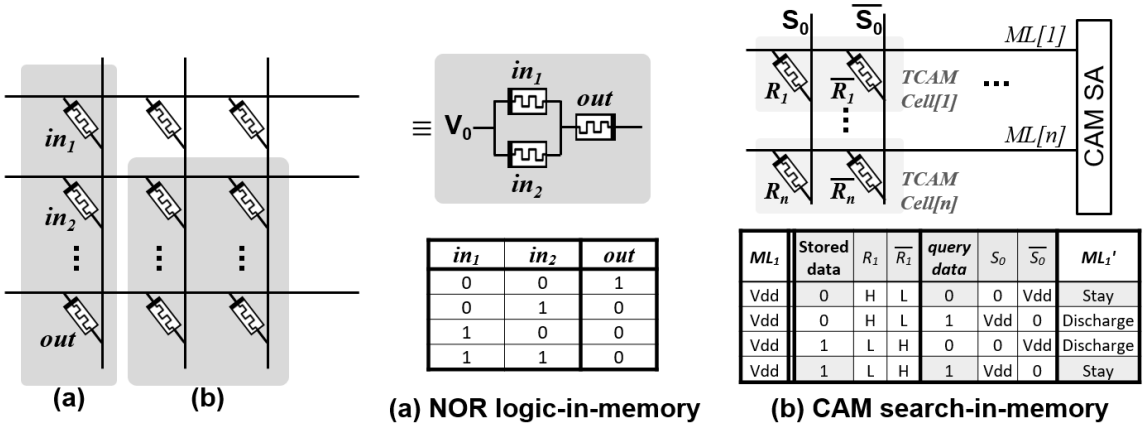


Figure 6.3. Timing graph of a single HV request with multiple mats

processing unit (mMPU) [118] performs logic operations by applying voltage to the input/output WLs. To perform the operation, two operands located in different rows and aligned in columns are executed, and their outputs are stored in a separate row. The arithmetic operation is built using NORs shown in Fig. 6.4(a). When both inputs have high resistance, *i.e.*, 00, the electrical potential of the shared BL is almost pulled down to ground. The *out* cell thus keeps the low resistance representing the logical 1. In all other cases, *i.e.*, 01, 10, and 11, the applied voltage across the *out* cell is over than threshold voltage and this incurs 1 to 0 switching. We used the crossbar adder using NOR logic from [34]. Using the proposed GDP, a concurrent logic operation is possible by applying voltage to all mat WLs corresponding to an HV. By enabling GDP, we can simultaneously activate all bits corresponding to a single HV by selecting SWLs in the same position in all mats. Then, we can operate a HV with a single bias input command, so we can implement addition and multiplication for HD training by combining NOR operation of multiple cycles.



in_1	in_2	out
0	0	1
0	1	0
1	0	0
1	1	0

ML_1	Stored data	R_1	$\overline{R_1}$	query data	S_0	$\overline{S_0}$	ML_1'
Vdd	0	H	L	0	0	Vdd	Stay
Vdd	0	H	L	1	Vdd	0	Discharge
Vdd	1	L	H	0	0	Vdd	Discharge
Vdd	1	L	H	1	Vdd	0	Stay

Figure 6.4. Building Block of In-memory Computation

6.3.3 Global WL searching

Shunt Via Routing (SVR) for HD inference

MAPLE exploits TCAM for the search operations in the crossbar array block. Fig. 6.4(b) shows the TCAM unit in the memory array. Unlike prior TCAM designs [11, 121, 128], which require additional cell blocks, the cell array of *MAPLE* can be used both as the memory and TCAM as shown in the figure. Data is prestored in two memristors, R_n and $\overline{R_n}$, according to their resistances. As shown in the table of Fig. 6.4(b), the case when R_n is high and $\overline{R_n}$ is low represents '0', and the opposite case represents '1'. During the search, a matching line (ML) is first initialized to V_{dd}. The query data is defined as '0' when S_0 is 0 and $\overline{S_0}$ is V_{dd}, while the opposite is defined as '1'. The matching line (ML) discharges current in case of a mismatch between the query data and the stored data. Then, CAM SA detects the discharge current and compares with that of adjacent WL to detect an HV which has higher similarity with the query data.

Although prior work utilizes the row-parallel execution using NVM devices [129, 11], their designs have not thoroughly considered the row-directional limitation. For example, the work in [11] places costly sensing circuits into every mat, and sums the output current at the end of the subarray to fulfill a single HV. In contrast, *MAPLE* proposes *shunt via routing* to enable global-WL-search, *i.e.,* executing the similarity check in a single cycle while using the

existing memory architecture. As shown in Fig.6.5, we utilize the existing global wordline (GWL) by placing a simple via-contact in each mat. SVR uses the same '*en*' switch with GDP. By enabling SVR (*en*=1), the SWLs located in parallel with the activated WL are shunted to the upper GWL. Then, all SWLs matching to an HV are connected and the total current can be read simultaneously. Hamming distance search block is then placed only at the end of a global WL instead of in every mat, resulting in negligible overhead.

The CAM SA in Fig. 6.4(b) performs Hamming distance search between the query data and the trained class vectors. This Hamming distance search block needs to 1) compare currents from two WLs to distinguish low current values, which means high similarity, and 2) trace the WL that has lower current value. The work in [11] used Loser Takes All (LTA) circuit to compare the Hamming distance of two WLs. Although it is energy efficient and benefits in the area based on its analog searching technique, we found the following problems. First, when it compares the Hamming distance between two WLs, LTA identifies a high discharge current, which is against the purpose of the design to find the lower Hamming distance. Second, LTA can compare currents between two WLs to distinguish low current values, but cannot identify from which WL the lower current came. Third, in the cascade structure that searches for the WL closest to the query data among all the WLs, all other stages are turned on even in one stage operation. In this work, we propose a new Hamming distance search block, called *Take Lower Current* (TLC), that can solve all of the LTA problems mentioned above. Our design compares the discharge currents on the two WLs to find the lower current and identifies which WL it is coming from. Also, it can operate independently for each stage, maximizing power efficiency. We next discuss the detail of TLC implementation.

Tracing a WL with the lowest current

Fig. 6.5 shows the proposed TLC blocks which consist of the following four key components: (a) analog search circuit to find a WL with lower current, which is equivalent to high similarity with the query data, (b) latch circuit based on a cross-coupled inverter, (c) sensing

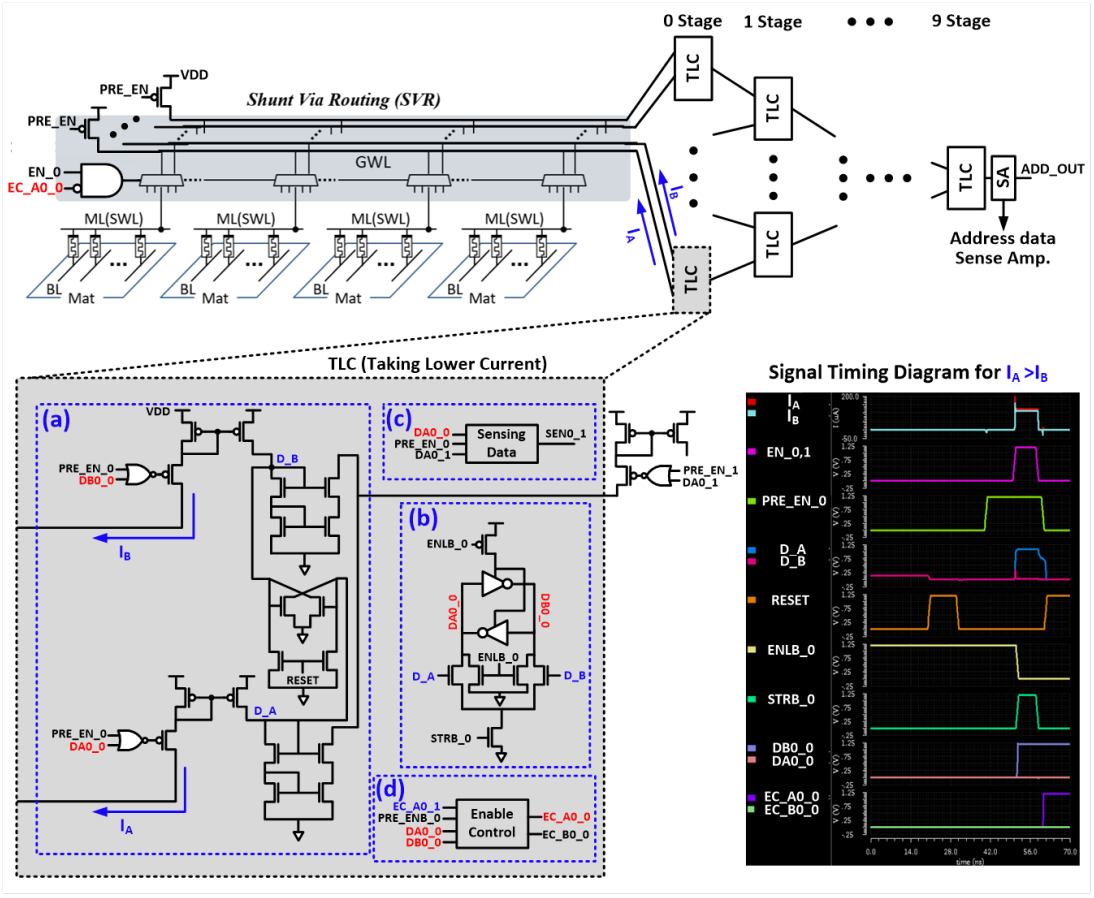


Figure 6.5. Global wordline search with shunt-via-routing (SVR) scheme

data block to check the latch status, (d) enable control block to enable the selected WLs. The searching circuit in Fig. 6.5(a) is responsible for judging similarity in two assigned WLs. Here, we insert the latching circuit of Fig. 6.5(b) and store each similarity comparison value. These stored values are used when performing stage isolation in the searching circuit and tracking the WL with the lowest current in the sensing data block of Fig. 6.5(c), respectively. To explain each design component, we first summarize the signal names of the figure as follows. The signal *EN* is a enable signal for discharging the GWL, where x of EN_x is the row number of GWL. In the signal *EC* of EC_Ax_y and EC_Bx_y, A and B is the lower and upper connected control area in TLC block, and x is the row number of TLC blocks at the assigned y stage and y is the stage number of TLC blocks. The DA/DB nodes of DAx_y and DBx_y follow the same notation as the signal EC. In the signal ENLB of ENLB_x, x is the stage number of TLC blocks. The signal

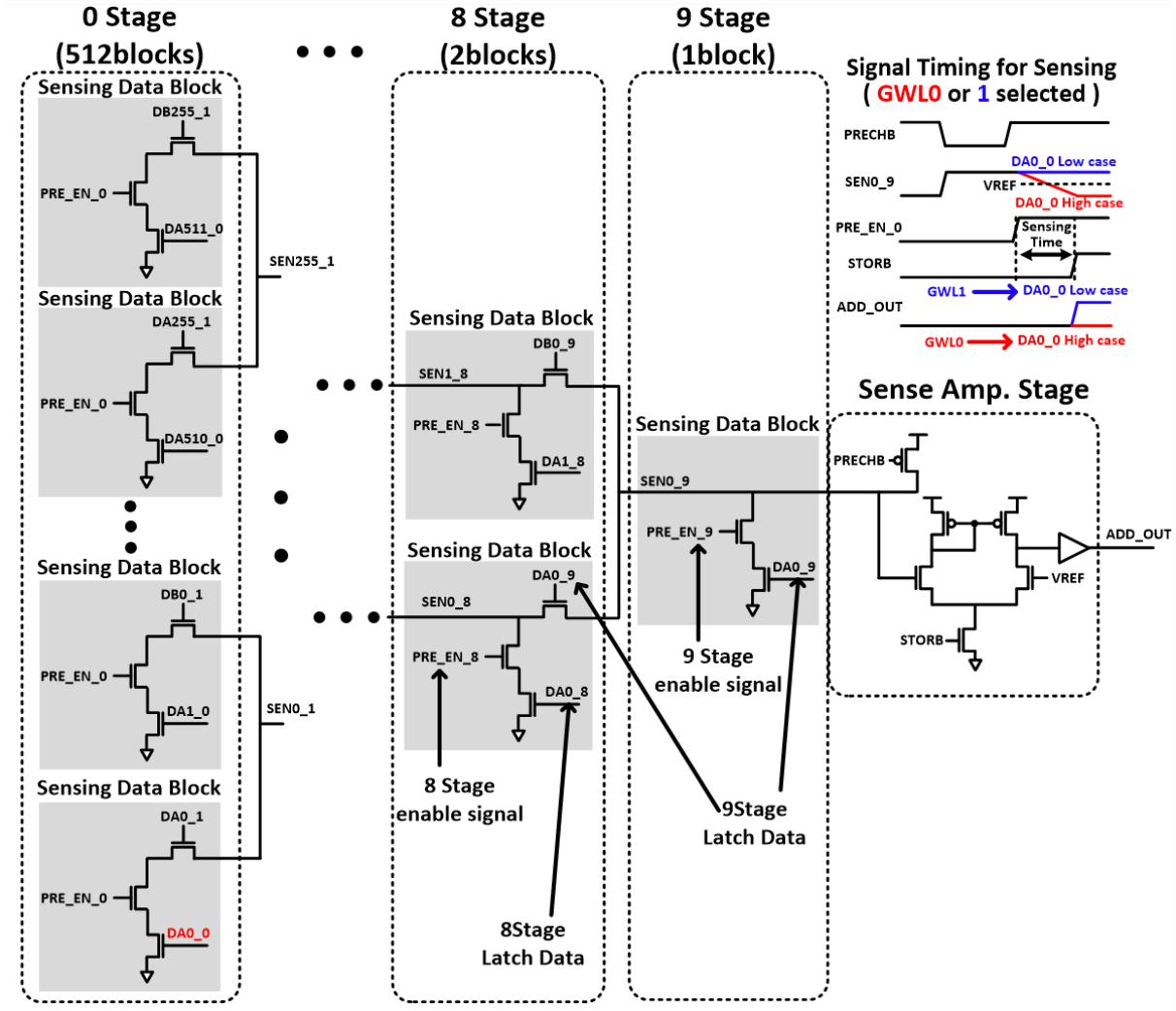


Figure 6.6. Sensing Data Blocks

STRB is used to store the latch data from D_A and D_B status, and the 'x' of STRB_x is the stage number of TLC blocks.

Analog searching circuit in Fig. 6.5(a) identifies the lower discharge current from two GWLs assigned. Consider the case where I_A is larger than I_B from the GWL0 and GWL1, as shown in the timing graph of Fig. 6.5. For comparison of I_A and I_B, D_A and D_B nodes are fully discharged by enabling RESET. Then, enabling EN_0,1 and PRE_EN_0, disabling EC_A0 and B0_0 make I_A and I_B identifiable to the analog searching circuit. Since D_A and D_B nodes are cross-connected to two NMOS transistors, the status of D_A node changes from low to high and

D_B node bounces a little but keeps low status. The two transistor sizes need to be optimized for proper operation. If the drivability of the transistors is not sufficient, the status of D_B and D_A nodes change from low to high simultaneously and the GWL with a higher current is not selected. Conversely, if the drivability of the transistors is excessive, the current through the turned-on NMOS causes excessive power consumption. When ENLB_0 disables and STRB_0 enables, the status of DA0_0 node on Fig. 6.5(b) keeps low and DB0_0 changes low to high due to the high/low status of D_A/D_B. Then PRE_EN_0 disables, and the status of EC_B0_0 keeps low and EC_A0_0 changes low to high due to the low/high status of DA0_0/DB0_0. Once the operation to search for the GWLs with the lower current is complete, all the addresses of lower current WL are stored on the latches of each TLC.

Based on the current comparison results of GWLs by the analog searching circuit, the sensing data block of Fig. 6.5(c) enables the TLC to find the lowest cell current WL. Fig. 6.6 shows the cascaded structure of the block in Fig. 6.5(c) in detail. The sensing data block consists of three NMOSs from the 0th to 8th stages, and the last 9th stage consists of two NMOSs. The upper NMOS connects the selected sensing data blocks to the next stage based on the latch status. The gate node of the NMOS connects to either DA or DB node of the next stage to decide the connection. For example, if the DA0_9 is low and DB0_9 is high at the 9th stage, the upper sensing data block of the 8th stage is connected by the high status of DB0_9. In the opposite case, the lower sensing data block of the 8th stage is connected by the high status of DB0_9. The middle NMOS serves to enable or disable the stage and its gate node is connected to PRE_EN signal to check the status of the selected sensing data block. The lower NMOS checks the data of the selected sensing data block and its gate node is connected to DA node of the TLC block.

The sense amplifier (SA) stage in Fig. 6.6 judges the data of the selected sensing block at each stage. Assuming that GWL0 or 1 is the lowest current GWL as shown at the signal timing graph in Fig. 6.6, the status of DA0_9 to DA0_1 are all high and DA0_0 is low or high. To check the DA0_0 status, the SEN nodes connected to the selected sensing data block first need to be precharged by lowering the PRECHB signal. Then, PRECHB signal changes from low to high

and PRE_EN_0 enables to check the selected sensing data block at 0th stage. In this case, since the DA0_1 to DA0_9 are high status, SEN0_1 node is connected to the lowest sensing data block. As shown at the signal timing graph, the status of the connected SEN node is determined by the DA0_0 node. During the sensing time from PRE_EN_0 rising to STORB rising, the status SEN0_9 is compared to VREF(VDD/2) at the SA stage. ADD_OUT is low when SEN0_9 is discharged below VREF and ADD_OUT is high when SEN0_9 keeps high status. As a result, we can determine the GWL with the lowest current from the status of DA0_0 and ADD_OUT. For example, the GWL0 is the lowest current GWL when DA0_0 is high and ADD_OUT is low, and the GWL1 is the lowest current GWL in the opposite case.

Reducing current dissipation

Although the proposed SVR technique increases the performance, power consumption issue caused by multi-mat execution still needs to be addressed. Assuming an array with C rows, TCAM searching blocks with $\log_2$C stages of binary tree type are required [11]. For example, In the case of a TCAM array with an array size of 1024x1024, the first stage is performed by comparing two WLs in each of 1024 WLs in a row and finding 512 lower current WLs. Then a WL with the smallest current is traced through total 10(=$\log_2$1024) stages. As mentioned in Sec. 6.3.3, LTA circuit in [11] is inefficient during multiple stage operation. When the searching operation is performed in one stage, the LTA blocks of all other stages are also turned on resulting in high current consumption. Based on our evaluation, when each cell has the minimum resistance and the maximum current flow, the current from the stage in operation and non-operation is at least over 500uA and 300uA, respectively. When all ten stages are turned on, the total current is estimated over 3.2mA even though we optimized the PMOS header size.

To address current dissipation issue shown in [11], we propose a method to perform Hamming distance search operations independently for each stage. Our design enables only the stage in which the operation is performed to be turned on and all other stages are turned off resulting in a much lower current. We designed a latch circuit with a cross-coupled inverter

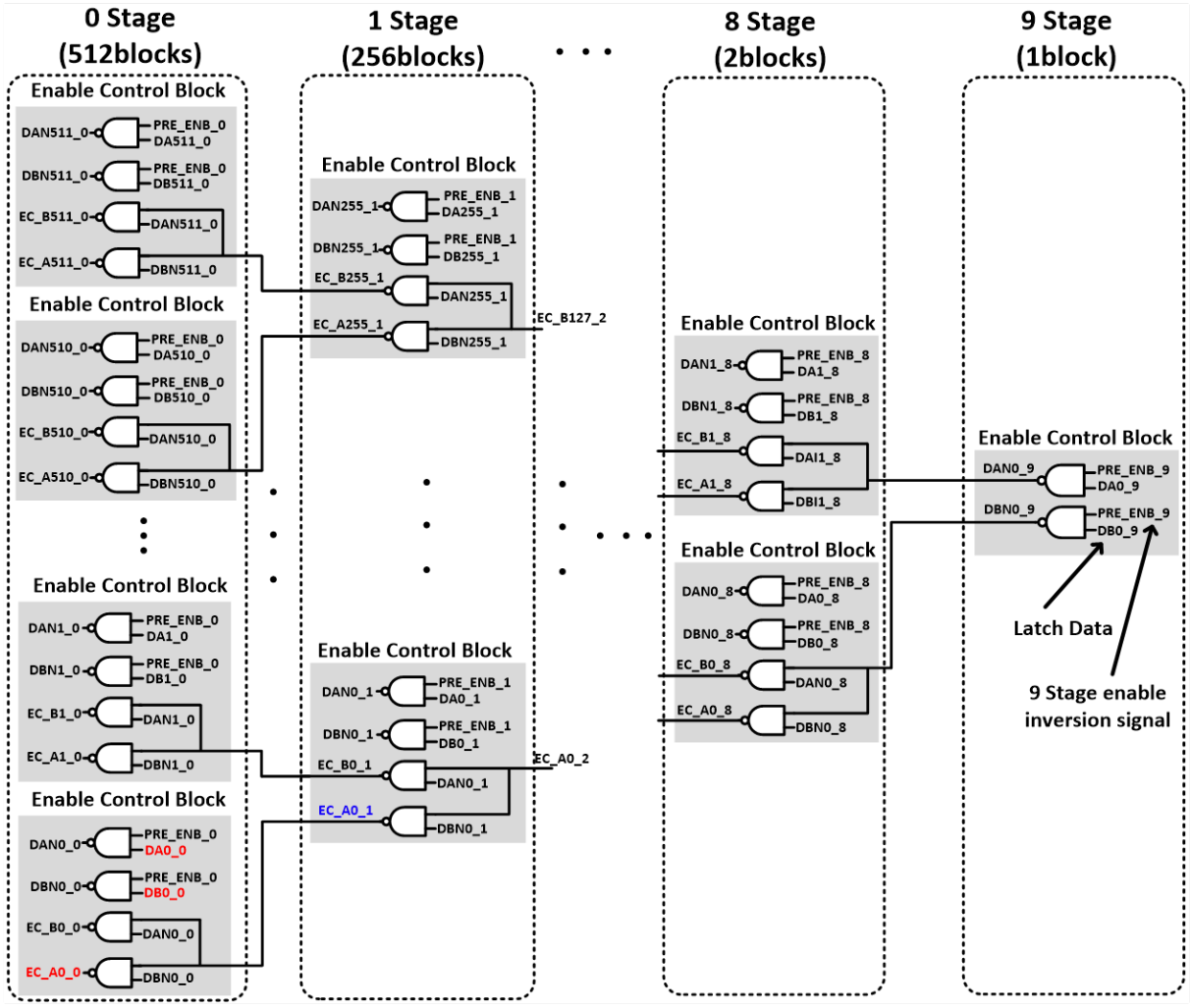


Figure 6.7. Enable Control Blocks

shown in Fig. 6.5(b) in order to enable isolated operation for each stage. As shown in the signal diagram of Fig. 6.5, DA0_0 and DB0_0 nodes are initially low status. When the PMOS connected to GWL is enabled by PRE_EN_0, the status of ENLB_0 node changes from high to low. Then, STRB_0 signal is enabled and DA0_0 and DB0_0 nodes store low or high status according to the status of D_A and D_B nodes. After storing the status of DA0_0 and DB0_0 nodes, PRE_EN0 is disabled. During the 0th stage operation mentioned above, the TLC blocks from the 1st to 9th stages are all disabled since both PRE_EN and DA/DB nodes are low. As a result, our design prevents unnecessary current consumption shown in prior LTA circuit [130]. Our evaluation shows that the proposed design has 62.5% energy savings compared to the LTA circuit [130].

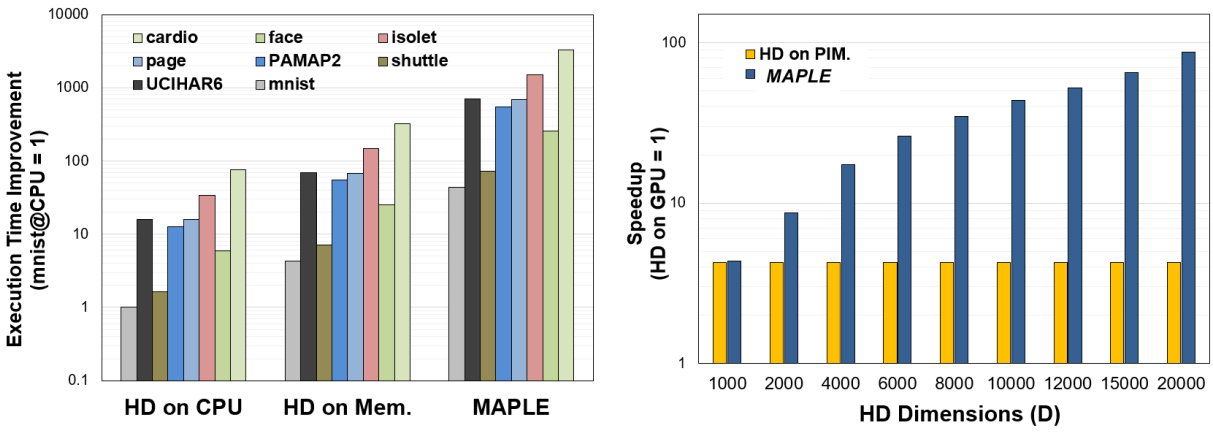


Figure 6.8. Performance Comparison for the different HW engines (left) and for the different HD dimensions (right)

We also insert a enable control block as shown in Fig. 6.5(d) to enable the only selected GWLs, hence maximizing power efficiency. It disables the unselected GWLs during the operation of tracing the lowest current GWL. Fig. 6.7 shows the entire structure including all the enable control blocks. The enable control block consists of 4 NAND gates from 0th to 8th stages, and the last 9th stage consists of 2 NAND gates. In the 0th stage, when PRE_ENB_0 is low, EC_A0_0 and EC_B0_0 nodes have the same status of EC_A0_1. In case that PRE_ENB_0 is high, if DA0_0 and DB0_0 nodes of the latch are high and low, EC_A0_0 node is high and EC_B0_0 node follows the output signal EC_A0_1 status. Conversely, if DA0_0 and DB0_0 nodes of the latch are low and high, EC_A0_0 node follows the output signal EC_A0_1 status and EC_B0_0 node is high. This is an example of the 0th stage, and the following 1st through 9th stages operate in the same way. As a result, the node connected to the first GWL is discharged by enabling EN_0 signal only when EC_A0_0 node is low, which can disable the unselected GWLs during the operation.

Table 6.1. Dataset summary (F: the number of features, K: the number of activity classes, N_{train}: the number of samples in the training data, N_{test}: the number of samples in the testing data)

Name	Data size (KB)	F	K	N_{train}	N_{test}		Name	Data size (KB)	F	K	N_{train}	N_{test}
cardio	721	21	3	2048	2048		**page**	482	10	5	4925	548
face	121483	608	2	22441	2494		**PAMAP2**	7340	27	5	16384	16384
isolet	38543	617	26	6237	1559		**shuttle**	4640	9	8	43500	14500
mnist	220080	392	10	60000	10000		**UCHHAR6**	34920	561	12	6213	1554

6.4 Experimental Results

6.4.1 Experimental Setup

To evaluate *MAPLE*, we execute the code implemented with Python 2.7 and Numpy which use C++ backend. We used *VTEAM* memristor model [91] for designing a memory cell with R_{on} and R_{off} of 10KΩ and 10MΩ, respectively. Circuit-level evaluations are performed with HSPICE simulator based on a 65nm CMOS process and VDD=1.2V. To verify how *MAPLE* works for general classification problems, we experimented with eight datasets, covering a wide range of applications: *mnist* for image processing, *shuttle* and *page* for canonical datasets commonly evaluated in literature, and practical datasets (*cardio* for fetal disease diagnosis, *face* for face recognition, *isolet* for voice recognition, and *PAMAP2, UCIHAR6* for human activity recognition) as shown in Table 6.1. We estimated the parallelism efficiency of *MAPLE* during HD training compared to the conventional CPU and PIM platforms. Also, we evaluated the performance and energy consumption of *MAPLE* in HD inference and compared to the work in [11] and [12].

6.4.2 Parallelism Efficiency in Training

We compare the efficiency of HD computing in the proposed design with other HW platforms. Fig. 6.8 (left) presents the log plot of speedup of *MAPLE* compared with two other references, *HD-on-CPU* and *HD-on-PIM*, respectively, based on the commodity CPU [131] and memory design [127]. The execution time considers HD training and inference. All results are normalized to the data of *MNIST* on CPU. As expected, the results show that HD computing is faster when using PIM than CPU because PIM reduces data movement costs. Among the PIM designs, *MAPLE* achieves better performance compared to the conventional PIM techniques. For example, in the case of *cardio*, *MAPLE* is 10× faster than using the conventional PIM design. Due to the parallelism effect of GDP, one of the key components of *MAPLE* explained in Sec. 6.3.2, *MAPLE* outperforms the conventional PIM design with insignificant modification

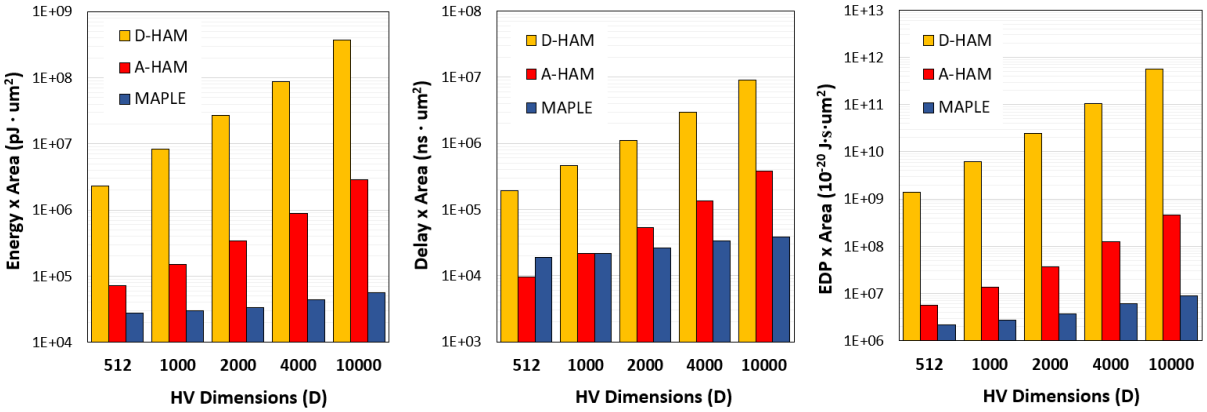


Figure 6.9. Area efficiency of associative search in energy consumption and searching delay

to existing memory architecture.

Fig. 6.8 (right) shows the speedup improvement of our design for different dimensions in HD. The result is obtained from the *MNIST* dataset as a representative. The data shown in this figure is normalized to HD on the conventional CPU. While the improvement in the case of HD on PIM almost remains constant as the increase of the HD dimension over the case of HD on CPU, *MAPLE* shows an exponential increase of speedup improvement compared to both HD on CPU and PIM cases. Although, in the low dimensional region, *MAPLE* does not show a prominent effect over other platforms, it does in the high dimensional region due to the increase of the number of mats to be executed in parallel by enabling GDP. As a result, while the latency improvement of *MAPLE* is around 4 times for the case of D=1,000, the improvement grows as the number of dimensions increases, with 87$\times$ improvement for the case of D=20,000. This result shows that *MAPLE* is more efficient in the dimensions of actual HD computing regions (D$\geq$10,000).

6.4.3 Inference Scalability

We compare the inference scalability of *MAPLE* with two state-of-the-art HD searching designs, D-HAM and A-HAM [11], where CMOS-based and analog-based searching techniques are used, respectively, as their Hamming distance metric. Fig. 6.9 shows the energy consumption and searching delay of *MAPLE* in HD inference as compared to D-HAM and A-HAM, when

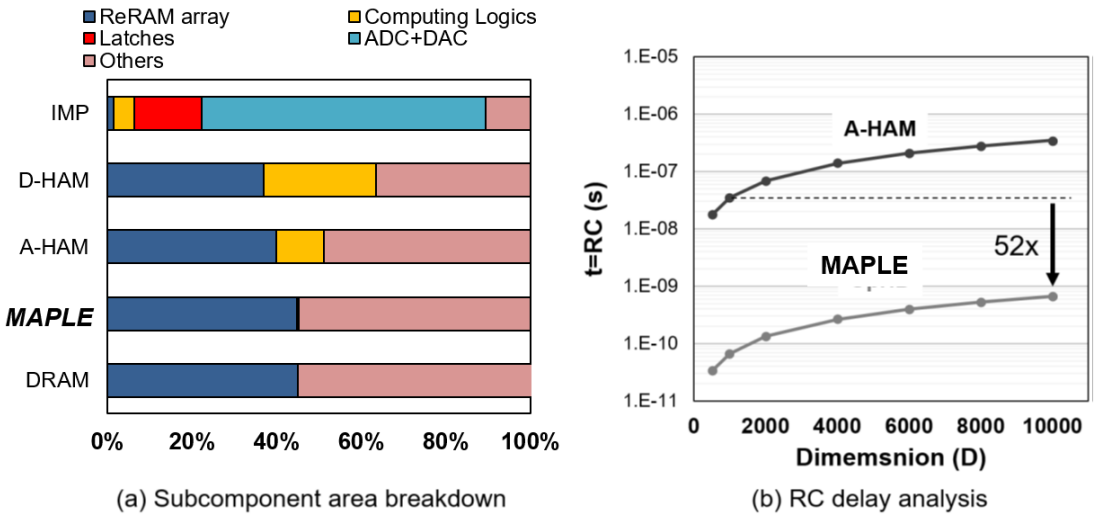


Figure 6.10. Overhead comparison of *MAPLE* with other PIM designs (D-HAM [11], A-HAM [11], IMP [12]) and DRAM

D changes from 512 to 10,000. To evaluate the physical scalability of our design, we show the energy and delay multiplied by area. Compared with D-HAM, A-HAM shows the slower rate of increase in energy and searching delay as the increase of HD dimensions. This is because A-HAM adjusts the searching accuracy by modifying the resolution of the LTA blocks, while D-HAM needs longer interconnection and additional circuits, *i.e.*, counters and comparators. *MAPLE* takes on LTA circuits that used in A-HAM design but outperforms that in the high dimensional region as shown in fig. 6.9. In conventional memory architecture, each mat of A-HAM requires the LTA circuit block individually. However, the proposed SVR technique in *MAPLE* enables having an LTA block only at the end of each subarray row by implementing simple via-contact in each mat instead. This results in significant area savings, aside from the energy and performance benefits of A-HAM design. For example, in the D=10,000 region, *MAPLE* is 51× more energy efficient and 10× faster compared to A-HAM design under the same area constraint.

6.4.4 Overhead

1) Area: Fig. 6.10(a) shows the area breakdown for each sub-component of *MAPLE* and other architectures to implement HD computing. We assume 32 mats-per-subarray and

32 subarrays-per-bank. The area is estimated from Cadence p-cell data with 45nm process technology. In this work, we use a *cell efficiency*, defined by the ratio of the cell area over the total area of the memory array, as the index to investigate the area overhead. The results show that prior state-of-the-art work still requires significant area cost. For example, IMP [12] consumes 67% of the total area for the analog from/to digital converters (ADC, DAC). D-HAM and A-HAM [11], state-of-the-art HD accelerators, also require over 60% of peripheral circuits in addition to the cell array. However, *MAPLE* has a superb cell efficiency of 45% over the total area, mainly due to a significant savings in the computing logic, 11% less than A-HAM, by using SVR method. Notably, our design has the same level of cell efficiency as the commodity DRAM as shown in Fig. 6.10(a).

2) Parasitic Effect: The parasitic effect is inevitable but undesirable in integrated circuit design. This is mainly caused by parasitic resistance and capacitance, which reduces system latency and causes performance degradation. In this work, we analyzed the parasitic effect with the RC delay parameter. We compared *MAPLE* with A-HAM [11], which basically use the same search block, to exclude elements not related to the parasitic effect. As shown in Fig. 6.10(b), MAPLE outperforms A-HAM [11], for the RC delay even in the high dimensional region. For example, searching for a 10,000-dimensional HV in *MAPLE* shows 52× less of the RC delay than only a single mat search of a 1,000 dimension in A-HAM [11],. The GWL, typically run in the first-level metal layer in commodity memory, uses wide-pitch copper (Cu) as an electrical wire, while the SWL uses narrow-pitch polysilicon [31]. Since *MAPLE* utilizes the GWL, which has much less sheet-resistance (R_S) compared to the SWL, it is less affected by the parasitic components to A-HAM although it has a longer current path during the search.

6.5 Conclusion

We present a novel PIM architecture which enables multi-mat parallel operations for HD computing with *Global-Bypass-Decoding* (GDP) and *Shunt-Via-Routing* (SVR) techniques. The

proposed design accelerates HD computing, by making minor changes to NVM architecture while keeping it compatible with existing standards and interfaces. The experimental results show that *MAPLE* offers 87× speedup over the conventional PIM design by enabling GDP. Moreover, the proposed SVR presents 10× performance improvement and 51× energy saving in HD inference compared to the state-of-the-art HD accelerators [11] with a superior cell-efficiency.

This chapter contains material from Joonseop Sim, Minsu Kim, Mohsen Imani, Yeseong Kim, Chris Kim and Tajana Rosing, "MAPLE: Multi-mat Parallelized Execution for Hyperdimensional Computing", which was submitted to ACM Journal of Emerging Technologies in Computing. The dissertation author was the primary investigator and author of this paper.

Chapter 7

Summary and Future Work

7.1 Thesis Summary

The advent of the Internet of Things has increased the amount of data, which has seriously challenged limited device resources. Sending all data to the cloud server is not scalable, does not guarantee a real-time response, and is not preferable due to causing private concerns and security. This increases the need to perform operations on a certain amount of data in an edge device. However, running data-intensive workloads with large datasets in traditional cores leads to massive data movement between memory and processing units, resulting in high power consumption and slow processing speed. Processing-in-Memory is a promising solution to address this bandwidth bottleneck by performing a portion of computation inside the memory. Previous work has proposed ways to address the issue of data movement between the processor and memory by supporting basic functionality inside the memory module [5, 8, 9, 13, 18, 29, 30, 96, 97, 98]. The emerging nonvolatile memory technologies are considered as good candidates for PIM due to their high density, scalability, and low power consumption. However, compared to the commodity DRAM and SRAM, that is a lack of performance due to device limitations and immature architecture. To address this, in this thesis, we proposed novel PIM designs and memory optimization methods for PIM acceleration.

7.1.1 Overcoming device limitation

Although many PIM techniques have been proposed so far, they support limited basic functionality such as basic bitwise operations (AND, OR, and IMP). Several techniques have been proposed to perform functions like addition and multiplication in NVM architectures. However, they execute these functions by combining multiple boolean operations. Therefore, they are inherently slow due to their multi-cycle operation as well as slow processing speed. This thesis proposes a new PIM architecture, which enables the addition and multiplication in an efficient manner.

In Chapter 2, we first presented a novel access device technique that enables the implementation of the $4F^2$ structure. This makes a variety of PIM technologies utilizing the crossbar structure realizing the minimum cell size in a two terminal access approach. We presented vertical nanowire field effect transistors (VNFETs) directly in current mode logic to improve their performance and energy compared to conventional planar-FETs, as well as their area efficiency. In Chapter 3, we presented a unipolar-switching logic for high-density PIM applications. 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, it takes advantages of a 3D vertical crossbar array structure to increase memory utilization per unit area for high-density applications. In Chapter 4, we presented a new sensing circuit which uses the analog properties of NVM. We simplify computation by exploiting the latch-up effect of thyristor devices to directly generate the results from the input data without any intermediate logic. We further leverage the back-down effect at latch-up points of the thyristor to implement functions with minimal increase in the number of gates. In addition, the proposed design performs the operations in a modified sensing circuit which is compatible with the conventional current sense amplifier. It does not need additional cells to support calculations, thus requires negligible area overhead. This thesis also addresses power consumption issues that previous PIM designs have. 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.

7.1.2 Memory architecture optimization

Today's research on NVM has been actively studied from the viewpoint of device and material, whereas the architecture that constitutes memory cell has brought the existing platform of DRAM and NAND Flash. Therefore, there are some points that they do not fully utilize the potential performance of NVM. This thesis suggests novel designs that can fully exploit the potential of NVM and PIM in the application of existing memory architectures.

Many prior studies have enabled various PIM operations on nonvolatile memory (NVM) by modifying sense amplifiers (SA). They exploit a single sense amplifier to handle multiple bitlines with a multiplexer (MUX) since a single SA circuit takes larger area than an NVM 1-bit cell. This limits potential parallelism that the PIM techniques can ideally achieve. In Chapter 5, we presented a multi-bit parallelism technique for high-performance processing in non-volatile memory architecture. Our design carries out multiple bit-lines requests under a MUX in parallel with two novel design components, multi-column/row latch (MCRL) and shared SA routing (SSR). The MCRL allows the address decoder to activate multiple addresses in both column and row directions by buffering the consecutively-requested addresses. The activated bits are then simultaneously sensed by the multiple SAs across a MUX by the proposed SSR.

We also propose a way to efficiently implement long byte processing required by today's emerging computing. Technologies such as deep learning vector operations, graph processing and brain-inspired computing require the computation of very long words. Due to the requirement of massive data access, they need a memory-centric architecture, which motivates us to utilize processing-in-memory (PIM) architecture. However, the existing memory architecture has physical limitations to fully support the PIM execution in applications such as the HD computing. To compute over 10,000-bit words corresponding to that word, tens of memory mats consisting of only 1K bit-rows need to be simultaneously activated. This conflicts with the operation in

conventional memory architecture, which accesses 8 to 64-bit words. In Chapter 6, we presented an optimization of processing-in-memory architecture for very long word processing. Our design works as both memory and PIM within the same cell array. In memory mode, our design follows the conventional read/write access pattern, with consecutive requests to each mat. In PIM mode, it enables concurrent activation of all mats in a bank by using our proposed global decoder bypass, enabling a single cycle operation of over 10,000-bit word. We also propose a global wordline search for nearest Hamming distance, to facilitate for the search over bits that are located across multiple mats.

7.2 Future Work

In this thesis, we have developed PIM logic using device physics and proposed a fine-grained parallelism technique at the sub-bank level for PIM acceleration. However, in order for this thesis to have more practical utility, more work is needed to combine our technology with the latest memory products. In this section, we provide future directions for further improvements on the two topics based on the work of this thesis: 1) Optimizing HBM architecture for extreme bandwidth system and 2) Locality-aware PIM architecture design.

7.2.1 Optimizing HBM architecture for extreme bandwidth system

Memory products, which have been developed to maximize bandwidth rather than minimize cost, have brought the bandwidth of the system to its limits. High bandwidth memory (HBM) [132] has been a key driving force of the continuous performance scaling of commodity processors, e.g., CPU and GPU, and other parallel processors. HBM has a small form factor compared to DDR4 or GDDR5 and can consume less energy while achieving higher bandwidth. It is a three-dimensional structure achieved by stacking several DRAM dies using through-silicon vias (TSVs) and microbumps. Although HBM2, the next-generation solution of HBM1, provides world-class performance through higher bandwidth (Up to 256GB/s) and lower power consumption in comparison to other conventional DRAMs, future memory products are expected

to demand multiple TB/s of memory bandwidth requiring more than the limit of bandwidth that current DRAM devices can provide. More research is needed in how to apply the parallelism technique proposed by this thesis to HBM architecture. There are four data transfer paths on the HBM architecture: 1) data transfer within a sensing block, 2) data transfer within a chip, 3) data transfer within a stack and 4) data transfer on an interposer. Prior approaches for maximizing the bandwidth of HBM have focused on the 3) and 4) direction since most researchers and companies have been hesitant to hurt the sub-chip level because of concerns to adding overhead to the architecture that has been optimized over a long period. However, considering the current situation that limits the development of technology to improve the bandwidth, we are approaching the point where it is necessary to parallel the technology development at the level below the chip. The work in Chapter 5 and 6 can be extended to perform parallel operations at sub-chip levels of HBM since the commodity HBM follows the basic bank design of the existing DDR-based memory products.

7.2.2 Locality-aware PIM architecture design

Utilizing data locality in the memory is essential for the efficient use of existing cache and commodity memory hierarchy. Although modern programming models and compiler designs take into account the physical locality of data, it is difficult for them to approach the level of locality proposed in this thesis. Many previous studies have been carried out under the assumption that the distribution of the bits constituting a word in a memory cell is continuous, but in practice, there may be cases of having a discontinuous distribution. For example, NVM architecture places a multiplexer (MUX) in front of the SA, which are used to select a BL connecting to a single I/O line from the multiple BLs [4]. Since it operates with a 64-bit I/O interface in a typical DDR-based memory system, 8 to 16 bits are read per internal clock on a single mat [19]. Therefore, instead of reading all the bits in the MUX, only some of the bits in each MUX are read. Our research on fine-grained parallelism in this thesis needs to be extended to the study of locality-aware parallelism.

Bibliography

[1] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 994–1007, 2012.

[2] G. B. Beneventi, *Characterization and modeling of phase-change memories*. PhD thesis, Université de Grenoble, 2011.

[3] M. Imani, S. Patil, and T. Š. Rosing, "Approximate computing using multiple-access single-charge associative memory," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 3, pp. 305–316, 2018.

[4] C. Xu, D. Niu, N. Muralimanohar, R. Balasubramonian, T. Zhang, S. Yu, and Y. Xie, "Overcoming the challenges of crossbar resistive memory architectures," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pp. 476–488, IEEE, 2015.

[5] E. Lehtonen and M. Laiho, "Stateful implication logic with memristors," in *Proceedings of the 2009 IEEE/ACM International Symposium on Nanoscale Architectures*, pp. 33–36, IEEE Computer Society, 2009.

[6] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magicmemristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014.

[7] 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.

[8] 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.

[9] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, "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*, pp. 1–6, IEEE, 2016.

[10] J. Sim, M. Imani, W. Choi, Y. Kim, and T. Rosing, "Lupis: latch-up based ultra efficient processing in-memory system," in *2018 19th International Symposium on Quality Electronic Design (ISQED)*, pp. 55–60, IEEE, 2018.

[11] 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.

[12] D. Fujiki, S. Mahlke, and R. Das, "In-memory data parallel processor," in *ACM SIGPLAN Notices*, vol. 53, pp. 1–14, ACM, 2018.

[13] J. Sim, M. Imani, Y. Kim, and T. Rosing, "Enabling efficient system design using vertical nanowire transistor current mode logic," in *2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 1–6, IEEE, 2017.

[14] S. A. Dayeh, R. Chen, Y. G. Ro, and J. Sim, "Progress in doping semiconductor nanowires during growth," *Materials Science in Semiconductor Processing*, vol. 62, pp. 135–155, 2017.

[15] J. Sim, S. Gupta, M. Imani, Y. Kim, and T. Rosing, "Upim: Unipolar switching logic for high density processing-in-memory applications," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, pp. 255–258, ACM, 2019.

[16] J. Sim, M. Imani, W. Choi, Y. Kim, and T. Rosing, "Current-sensing efficient adder for processing-in-memory design," in *10th Annual Non-Volatile Memories Workshop*, 2019.

[17] J. Sim, M. Kim, Y. Kim, S. Gupta, B. Khaleghi, and T. Rosing, "Mapim: Mat parallelism for high performance processing in non-volatile memory architecture," in *20th International Symposium on Quality Electronic Design (ISQED)*, pp. 145–150, IEEE, 2019.

[18] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based material implication (imply) logic: Design principles and methodologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2054–2066, 2014.

[19] M. O'Connor, N. Chatterjee, D. Lee, J. Wilson, A. Agrawal, S. W. Keckler, and W. J. Dally, "Fine-grained dram: energy-efficient dram for extreme bandwidth systems," in *Proceeding of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 41–54, ACM, 2017.

[20] E. Kültürsay, M. Kandemir, A. Sivasubramaniam, and O. Mutlu, "Evaluating stt-ram as an energy-efficient main memory alternative," in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 256–267, IEEE, 2013.

[21] T. Perez and C. A. De Rose, "Non-volatile memory: Emerging technologies and their impacts on memory systems," *Porto Alegre*, 2010.

[22] R. Waser and M. Aono, "Nanoionics-based resistive switching memories," in *Nanoscience And Technology: A Collection of Reviews from Nature Journals*, pp. 158–165, World Scientific, 2010.

[23] S. Yu, "Resistive random access memory (rram)," *Synthesis Lectures on Emerging Engineering Technologies*, vol. 2, no. 5, pp. 1–79, 2016.

[24] X. Dong, N. P. Jouppi, and Y. Xie, "Pcramsim: System-level performance, energy, and area modeling for phase-change ram," in *Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, pp. 269–275, IEEE, 2009.

[25] R. Nair, S. F. Antao, C. Bertolli, P. Bose, J. R. Brunheroto, T. Chen, C.-Y. Cher, C. H. Costa, J. Doi, C. Evangelinos, *et al.*, "Active memory cube: A processing-in-memory architecture for exascale systems," *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 17–1, 2015.

[26] V. Seshadri, K. Hsieh, A. Boroum, D. Lee, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, "Fast bulk bitwise and and or in dram," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 127–131, 2015.

[27] J. Ahn, S. Yoo, O. Mutlu, and K. Choi, "Pim-enabled instructions: A low-overhead, locality-aware processing-in-memory architecture," in *Computer Architecture (ISCA), 2015 ACM/IEEE 42nd Annual International Symposium on*, pp. 336–348, IEEE, 2015.

[28] Y. Wang, Y. Han, L. Zhang, H. Li, and X. Li, "Propram: exploiting the transparent logic resources in non-volatile memory for near data computing," in *Proceedings of the 52nd Annual Design Automation Conference*, p. 47, ACM, 2015.

[29] M. Imani, Y. Kim, and T. Rosing, "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.

[30] M. Imani, A. Rahimi, and T. S. Rosing, "Resistive configurable associative memory for approximate computing," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*, pp. 1327–1332, IEEE, 2016.

[31] A. N. Udipi, N. Muralimanohar, N. Chatterjee, R. Balasubramonian, A. Davis, and N. P. Jouppi, "Rethinking dram design and organization for energy-constrained multi-cores," in *ACM SIGARCH Computer Architecture News*, vol. 38, pp. 175–186, ACM, 2010.

[32] S. Gupta, M. Imani, and T. Rosing, "Felix: Fast and energy-efficient logic in memory," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–7, IEEE, 2018.

[33] J. Y. Seok, S. J. Song, J. H. Yoon, K. J. Yoon, T. H. Park, D. E. Kwon, H. Lim, G. H. Kim, D. S. Jeong, and C. S. Hwang, "A review of three-dimensional resistive switching

cross-bar array memories from the integration and materials property points of view," *Advanced Functional Materials*, vol. 24, no. 34, pp. 5316–5339, 2014.

[34] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky, "Logic design within memristive memories using memristor-aided logic (magic)," *IEEE Transactions on Nanotechnology*, vol. 15, no. 4, pp. 635–650, 2016.

[35] M. Zhou, M. Imani, S. Gupta, and T. Rosing, "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.

[36] "AMD Radeon R9 390." https://www.techpowerup.com/gpu-specs/radeon-r9-390.c2664.

[37] "AMD Radeon HD 7970." https://www.techpowerup.com/gpu-specs/radeon-hd-7970.c296.

[38] X. Li, Z. Chen, Z. Fang, A. Kamath, X. Wang, N. Singh, G.-Q. Lo, and D.-L. Kwong, "Integration of resistive switching memory cell with vertical nanowire transistor," in *Proceedings of World Academy of Science, Engineering and Technology*, World Academy of Science, Engineering and Technology, 2012.

[39] C.-C. Hsieh, Y.-F. Chang, Y.-C. Chen, D. Shahrjerdi, and S. K. Banerjee, "Highly non-linear and reliable amorphous silicon based back-to-back schottky diode as selector device for large scale rram arrays," *ECS Journal of Solid State Science and Technology*, vol. 6, no. 9, pp. N143–N147, 2017.

[40] C.-H. Wang, Y.-H. Tsai, K.-C. Lin, M.-F. Chang, Y.-C. King, C.-J. Lin, S.-S. Sheu, Y.-S. Chen, H.-Y. Lee, F. T. Chen, *et al.*, "Three-dimensional 4f 2 reram cell with cmos logic compatible process," in *2010 International Electron Devices Meeting*, pp. 29–6, IEEE, 2010.

[41] V. Moroz, N. Strecker, X. Xu, L. Smith, and I. Bork, "Modeling the impact of stress on silicon processes and devices," *Materials Science in Semiconductor Processing*, vol. 6, no. 1-3, pp. 27–36, 2003.

[42] B. Yu, L. Chang, S. Ahmed, H. Wang, S. Bell, C.-Y. Yang, C. Tabery, C. Ho, Q. Xiang, T.-J. King, *et al.*, "Finfet scaling to 10 nm gate length," in *Digest. International Electron Devices Meeting,*, pp. 251–254, IEEE, 2002.

[43] M. Imani, S. Patil, and T. S. Rosing, "Hierarchical design of robust and low data dependent finfet based sram array," in *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH´ 15)*, pp. 63–68, IEEE, 2015.

[44] S. Bangsaruntip, G. M. Cohen, A. Majumdar, Y. Zhang, S. Engelmann, N. Fuller, L. Gignac, S. Mittal, J. Newbury, M. Guillorn, *et al.*, "High performance and highly uniform gate-all-around silicon nanowire mosfets with wire size dependent scaling," in *2009 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4, IEEE, 2009.

[45] C. Pan, P. Raghavan, D. Yakimets, P. Debacker, F. Catthoor, N. Collaert, Z. Tokei, D. Verkest, A. V.-Y. Thean, and A. Naeemi, "Technology/system codesign and benchmarking for lateral and vertical gaa nanowire fets at 5-nm technology node," *IEEE Transactions on Electron Devices*, vol. 62, no. 10, pp. 3125–3132, 2015.

[46] U. K. Das, M. G. Bardon, D. Jang, G. Eneman, P. Schuddinck, D. Yakimets, P. Raghavan, and G. Groeseneken, "Limitations on lateral nanowire scaling beyond 7-nm node," *IEEE Electron Device Letters*, vol. 38, no. 1, pp. 9–11, 2017.

[47] B.-H. Lee, J. Hur, M.-H. Kang, T. Bang, D.-C. Ahn, D. Lee, K.-H. Kim, and Y.-K. Choi, "A vertically integrated junctionless nanowire transistor," *Nano letters*, vol. 16, no. 3, pp. 1840–1847, 2016.

[48] M. Yamashina and H. Yamada, "An mos current mode logic (mcml) circuit for low-power sub-ghz processors," *IEICE Transactions on Electronics*, vol. 75, no. 10, pp. 1181–1187, 1992.

[49] R. Wagner and W. Ellis, "Vapor-liquid-solid mechanism of single crystal growth," *Applied physics letters*, vol. 4, no. 5, pp. 89–90, 1964.

[50] E. Givargizov, "Fundamental aspects of vls growth," in *Vapour Growth and Epitaxy*, pp. 20–30, Elsevier, 1975.

[51] G. Bootsma and H. Gassen, "A quantitative study on the growth of silicon whiskers from silane and germanium whiskers from germane," *Journal of Crystal Growth*, vol. 10, no. 3, pp. 223–234, 1971.

[52] E. Kaldis, "Current topics in material science: Vol. 1," *North-Holland Publishing Co,*, p. 770, 1978.

[53] J. Kasahara, K. Kajiwara, and T. Yamada, "Gaas whiskers grown by a thermal decomposition method," *Journal of Crystal Growth*, vol. 38, no. 1, pp. 23–28, 1977.

[54] J. Noras and M. Ryall, "Whisker growth on nickel-coated gallium phosphide," *Journal of Physics D: Applied Physics*, vol. 12, no. 2, p. 277, 1979.

[55] K. Haraguchi, T. Katsuyama, K. Hiruma, and K. Ogawa, "Gaas p-n junction formed in quantum wire crystals," *Applied Physics Letters*, vol. 60, no. 6, pp. 745–747, 1992.

[56] A. M. Morales and C. M. Lieber, "A laser ablation method for the synthesis of crystalline semiconductor nanowires," *Science*, vol. 279, no. 5348, pp. 208–211, 1998.

[57] Y. Wu, R. Fan, and P. Yang, "Block-by-block growth of single-crystalline si/sige superlattice nanowires," *Nano Letters*, vol. 2, no. 2, pp. 83–86, 2002.

[58] L. J. Lauhon, M. S. Gudiksen, D. Wang, and C. M. Lieber, "Epitaxial core–shell and core–multishell nanowire heterostructures," *Nature*, vol. 420, no. 6911, p. 57, 2002.

[59] M. S. Gudiksen, J. Wang, and C. M. Lieber, "Size-dependent photoluminescence from single indium phosphide nanowires," *The Journal of Physical Chemistry B*, vol. 106, no. 16, pp. 4036–4039, 2002.

[60] J. Trägårdh, A. Persson, J. Wagner, D. Hessman, and L. Samuelson, "Measurements of the band gap of wurtzite inas 1- x p x nanowires using photocurrent spectroscopy," *Journal of applied physics*, vol. 101, no. 12, p. 123701, 2007.

[61] H. J. Joyce, J. Wong-Leung, Q. Gao, H. H. Tan, and C. Jagadish, "Phase perfection in zinc blende and wurtzite iii- v nanowires using basic growth parameters," *Nano letters*, vol. 10, no. 3, pp. 908–915, 2010.

[62] M. Yazawa, M. Koguchi, A. Muto, and K. Hiruma, "Semiconductor nanowhiskers," *Advanced materials*, vol. 5, no. 7-8, pp. 577–580, 1993.

[63] J. Motohisa, J. Noborisaka, J. Takeda, M. Inari, and T. Fukui, "Catalyst-free selective-area movpe of semiconductor nanowires on (111) b oriented substrates," *Journal of crystal growth*, vol. 272, no. 1-4, pp. 180–185, 2004.

[64] L. Fröberg, W. Seifert, and J. Johansson, "Diameter-dependent growth rate of inas nanowires," *Physical Review B*, vol. 76, no. 15, p. 153401, 2007.

[65] E. A. Sutter and P. W. Sutter, "Size-dependent phase diagram of nanoscale alloy drops used in vapor- liquid- solid growth of semiconductor nanowires," *ACS nano*, vol. 4, no. 8, pp. 4943–4947, 2010.

[66] V. Schmidt, S. Senz, and U. Gösele, "The shape of epitaxially grown silicon nanowires and the influence of line tension," *Applied Physics A*, vol. 80, no. 3, pp. 445–450, 2005.

[67] E. Sutter and P. Sutter, "Phase diagram of nanoscale alloy particles used for vapor- liquid-solid growth of semiconductor nanowires," *Nano letters*, vol. 8, no. 2, pp. 411–414, 2008.

[68] T. Park, H. Cho, J. Choe, S. Han, S.-M. Jung, J. Jeong, B. Nam, O. Kwon, J. Han, H. Kang, *et al.*, "Static noise margin of the full dg-cmos sram cell using bulk finfets (omega mosfets)," in *IEEE International Electron Devices Meeting 2003*, pp. 2–2, IEEE, 2003.

[69] G. Pasandi, M. Jafari, and M. Imani, "A new low-power 10t sram cell with improved read snm," *International Journal of Electronics*, vol. 102, no. 10, pp. 1621–1633, 2015.

[70] B. Yu, J. Song, Y. Yuan, W.-Y. Lu, and Y. Taur, "A unified analytic drain–current model for multiple-gate mosfets," *IEEE Transactions on Electron Devices*, vol. 55, no. 8, pp. 2157–2163, 2008.

[71] S. Kim, G. Klimeck, S. Damodaran, and B. P. Haley, "Mugfet," 2008.

[72] K.-S. Im, C.-H. Won, S. Vodapally, R. Caulmilone, S. Cristoloveanu, Y.-T. Kim, and J.-H. Lee, "Fabrication of normally-off gan nanowire gate-all-around fet with top-down approach," *Applied Physics Letters*, vol. 109, no. 14, p. 143106, 2016.

[73] A. o. Shapiro, "Mos current mode logic near threshold circuits," *Journal of Low Power Electronics and Applications*, 2014.

[74] C. Chuang and S. Cheng, "Fabrication and properties of well-ordered arrays of single-crystalline nisi 2 nanowires and epitaxial nisi 2/si heterostructures," *Nano Research*, vol. 7, no. 11, pp. 1592–1603, 2014.

[75] J. Kedzierski, P. Xuan, E. H. Anderson, J. Bokor, T.-J. King, and C. Hu, "Complementary silicide source/drain thin-body mosfets for the 20 nm gate length regime," in *International Electron Devices Meeting 2000. Technical Digest. IEDM (Cat. No. 00CH37138)*, pp. 57–60, IEEE, 2000.

[76] M. Imani, A. Rahimi, Y. Kim, and T. Rosing, "A low-power hybrid magnetic cache architecture exploiting narrow-width values," in *Non-Volatile Memory Systems and Applications Symposium (NVMSA), 2016 5th*, pp. 1–6, IEEE, 2016.

[77] M. Imani, Y. Kim, A. Rahimi, and T. Rosing, "Acam: Approximate computing based on adaptive associative memory with online learning," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, pp. 162–167, ACM, 2016.

[78] "BSIMCMG." http://bsim.berkeley.edu/models/bsimcmg/.

[79] "GPDK45." https://support.cadence.com/apex/ArticleAttachmentPortal?id\=a1Od000000051TqEAI&pageName=ArticleContent.

[80] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "Mosaic: Mask optimizing solution with process window aware inverse correction," in *Proceedings of the 51st Annual Design Automation Conference*, pp. 1–6, ACM, 2014.

[81] "Caltech 101." http://www.vision.caltech.edu/ImageDatasets/Caltech101/.

[82] E. Amrani, A. Drori, and S. Kvatinsky, "Logic design with unipolar memristors," in *Very Large Scale Integration (VLSI-SoC), 2016 IFIP/IEEE International Conference on*, pp. 1–5, IEEE, 2016.

[83] L. Chang, Z. Wang, Y. Zhang, and W. Zhao, "Reconfigurable processing in memory architecture based on spin orbit torque," in *2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp. 95–96, IEEE, 2017.

[84] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3d ics: The pros and cons of going vertical," *IEEE Design & Test of Computers*, vol. 22, no. 6, pp. 498–510, 2005.

[85] J. Jeddeloh and B. Keeth, "Hybrid memory cube new dram architecture increases density and performance," in *VLSI Technology (VLSIT), 2012 Symposium on*, pp. 87–88, IEEE, 2012.

[86] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "Drisa: A dram-based reconfigurable in-situ accelerator," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 288–301, ACM, 2017.

[87] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, "Ambit: In-memory accelerator for bulk bitwise operations using commodity dram technology," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 273–287, ACM, 2017.

[88] T. Finkbeiner, G. Hush, T. Larsen, P. Lea, J. Leidel, and T. Manning, "In-memory intelligence," *IEEE Micro*, vol. 37, no. 4, pp. 30–38, 2017.

[89] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, "Buddy-ram: Improving the performance and efficiency of bulk bitwise operations using dram," *arXiv preprint arXiv:1611.09988*, 2016.

[90] T.-C. Chang, K.-C. Chang, T.-M. Tsai, T.-J. Chu, and S. M. Sze, "Resistance random access memory," *Materials Today*, vol. 19, no. 5, pp. 254–264, 2016.

[91] S. Kvatinsky, 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.

[92] F. W. Sears, M. W. Zemansky, and H. D. Young, *College physics*. Addison Wesley Publishing Company, 1974.

[93] T. Zhang, K. Chen, C. Xu, G. Sun, T. Wang, and Y. Xie, "Half-dram: a high-bandwidth and low-power dram architecture from the rethinking of fine-grained activation," in *Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on*, pp. 349–360, IEEE, 2014.

[94] Y. Guerfi and G. Larrieu, "Vertical silicon nanowire field effect transistors with nanoscale gate-all-around," *Nanoscale research letters*, vol. 11, no. 1, p. 210, 2016.

[95] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie, "Design implications of memristor-based rram cross-point structures," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, pp. 1–6, IEEE, 2011.

[96] 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," in *Proceedings of the 43rd International Symposium on Computer Architecture*, pp. 14–26, IEEE Press, 2016.

[97] M. Imani, D. Peroni, and T. Rosing, "Cfpu: Configurable floating point multiplier for energy-efficient computing," in *Proceedings of the 54th Annual Design Automation Conference 2017*, p. 76, ACM, 2017.

[98] M. Imani, D. Peroni, Y. Kim, A. Rahimi, and T. Rosing, "Efficient neural network acceleration on gpgpu using content addressable memory," in *Proceedings of the Conference on Design, Automation & Test in Europe*, pp. 1026–1031, European Design and Automation Association, 2017.

[99] M.-F. Chang, S.-J. Shen, C.-C. Liu, C.-W. Wu, Y.-F. Lin, Y.-C. King, C.-J. Lin, H.-J. Liao, Y.-D. Chih, and H. Yamauchi, "An offset-tolerant fast-random-read current-sampling-based sense amplifier for small-cell-current nonvolatile memory," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 864–877, 2013.

[100] X. Dong, C. Xu, N. Jouppi, and Y. Xie, "Nvsim: A circuit-level performance, energy, and area model for emerging non-volatile memory," in *Emerging Memory Technologies*, pp. 15–50, Springer, 2014.

[101] X. Tong, J. Luo, H. Wu, Q. Liang, H. Zhong, H. Zhu, and C. Zhao, "Two-terminal vertical memory cell for cross-point static random access memory applications," *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena*, vol. 32, no. 2, p. 021205, 2014.

[102] 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.

[103] M. Poremba, T. Zhang, and Y. Xie, "Fine-granularity tile-level parallelism in non-volatile memory architecture with two-dimensional bank subdivision," in *Proceedings of the 53rd Annual Design Automation Conference*, p. 168, ACM, 2016.

[104] M. Rhu, M. Sullivan, J. Leng, and M. Erez, "A locality-aware memory hierarchy for energy-efficient gpu architectures," in *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 86–98, IEEE, 2013.

[105] Y. Choi, I. Song, M.-H. Park, H. Chung, S. Chang, B. Cho, J. Kim, Y. Oh, D. Kwon, J. Sunwoo, *et al.*, "A 20nm 1.8 v 8gb pram with 40mb/s program bandwidth," in *2012 IEEE International Solid-State Circuits Conference*, pp. 46–48, IEEE, 2012.

[106] W. Shin, J. Jang, J. Choi, J. Suh, Y. Kwon, Y. Moon, and L.-S. Kim, "Rank-level parallelism in dram," *IEEE Transactions on Computers*, vol. 66, no. 7, pp. 1274–1280, 2017.

[107] C. J. Lee, V. Narasiman, O. Mutlu, and Y. N. Patt, "Improving memory bank-level parallelism in the presence of prefetching," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 327–336, IEEE, 2009.

[108] Y. Kim, V. Seshadri, D. Lee, J. Liu, and O. Mutlu, "A case for exploiting subarray-level parallelism (salp) in dram," *ACM SIGARCH Computer Architecture News*, vol. 40, no. 3, pp. 368–379, 2012.

[109] M. Poremba and Y. Xie, "Nvmain: An architectural-level main memory simulator for emerging non-volatile memories," in *2012 IEEE Computer Society Annual Symposium on VLSI*, pp. 392–397, IEEE, 2012.

[110] R. Ubal, B. Jang, P. Mistry, D. Schaa, and D. Kaeli, "Multi2sim: a simulation framework for cpu-gpu computing," in *2012 21st International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pp. 335–344, IEEE, 2012.

[111] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2010.

[112] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[113] M. Shafique, R. Hafiz, M. U. Javed, S. Abbas, L. Sekanina, Z. Vasicek, and V. Mrazek, "Adaptive and energy-efficient architectures for machine learning: Challenges, opportunities, and research roadmap," in *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 627–632, IEEE, 2017.

[114] A. Rahimi *et al.*, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ACM, 2016.

[115] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, 2009.

[116] M. Imani, S. Salamat, S. Gupta, J. Huang, and T. Rosing, "Fach: Fpga-based acceleration of hyperdimensional computing by reducing computational complexity," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pp. 493–498, ACM, 2019.

[117] H. Li, T. F. Wu, A. Rahimi, K.-S. Li, M. Rusch, C.-H. Lin, J.-L. Hsu, M. M. Sabry, S. B. Eryilmaz, J. Sohn, *et al.*, "Hyperdimensional computing with 3d vrram in-memory kernels: Device-architecture co-design for energy-efficient, error-resilient language recognition," in *2016 IEEE International Electron Devices Meeting (IEDM)*, pp. 16–1, IEEE, 2016.

[118] N. Talati, A. H. Ali, R. B. Hur, N. Wald, R. Ronen, P.-E. Gaillardon, and S. Kvatinsky, "Practical challenges in delivering the promises of real processing-in-memory machines," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1628–1633, IEEE, 2018.

[119] Y. H. Son, O. Seongil, Y. Ro, J. W. Lee, and J. H. Ahn, "Reducing memory access latency with asymmetric dram bank organizations," in *ACM SIGARCH Computer Architecture News*, vol. 41, pp. 380–391, ACM, 2013.

[120] M. Imani, D. Kong, A. Rahimi, and T. Rosing, "Voicehd: Hyperdimensional computing for efficient speech recognition," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–8, IEEE, 2017.

[121] D. Peroni, M. Imani, and T. Rosing, "Alook: adaptive lookup for gpgpu acceleration," in *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pp. 739–746, ACM, 2019.

[122] T. F. Wu, H. Li, P.-C. Huang, A. Rahimi, J. M. Rabaey, H.-S. P. Wong, M. M. Shulaker, and S. Mitra, "Brain-inspired computing exploiting carbon nanotube fets and resistive ram: Hyperdimensional computing case study," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 492–494, IEEE, 2018.

[123] S. Salamat, M. Imani, B. Khaleghi, and T. Rosing, "F5-hd: Fast flexible fpga-based framework for refreshing hyperdimensional computing," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 53–62, ACM, 2019.

[124] M. Dubois, C. Scheurich, and F. Briggs, "Memory access buffering in multiprocessors," in *ACM SIGARCH computer architecture news*, vol. 14, pp. 434–442, IEEE Computer Society Press, 1986.

[125] A. Goel and P. Gupta, "Small subset queries and bloom filters using ternary associative memories, with applications," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1, pp. 143–154, 2010.

[126] C. S. Hwang, "Prospective of semiconductor memory devices: from memory system to materials," *Advanced Electronic Materials*, vol. 1, no. 6, 2015.

[127] B. Jacob, S. Ng, and D. Wang, *Memory systems: cache, DRAM, disk*. Morgan Kaufmann, 2010.

[128] A. Rahimi, A. Ghofrani, K.-T. Cheng, L. Benini, and R. K. Gupta, "Approximate associative memristive memory for energy-efficient gpus," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pp. 1497–1502, EDA Consortium, 2015.

[129] C. Carlo, "del mundo, vincent t. lee, luis ceze, mark oskin, ncam: Near-data processing for nearest neighbor search," in *Proceedings of the 2015 International Symposium on Memory Systems*, 2015.

[130] R. Dlugosz, A. Rydlewski, and T. Talaska, "Low power nonlinear min/max filters implemented in the cmos technology," in *2014 29th International Conference on Microelectronics Proceedings-MIEL 2014*, pp. 397–400, IEEE, 2014.

[131] "Xeon Platinum 818x." https://ark.intel.com/content/www/us/en/ark/products/120498/intel-xeon-platinum-8180m-processor-38-5m-cache-2-50-ghz.html.

[132] Y. Jeon, H. Kim, J. Kim, and M. Je, "Design of an on-silicon-interposer passive equalizer for next generation high bandwidth memory with data rate up to 8 gb/s," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 7, pp. 2293–2303, 2018.