The Future of Computing: An Energy-Efficient In-Memory Computing Architectures with Emerging VGSOT MRAM Technology

Md Rubel Sarkar

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Masters of Science

in

Computer Engineering

Cindy Yang Yi, Chair

Jeffrey Sean Walling

Creed F. Jones

April 19^{th}, 2024

Blacksburg, Virginia

Keywords: In-Memory Computing, MRAM, VGSOT, BNN

The Future of Computing: An Energy-Efficient In-Memory Computing Architectures with Emerging VGSOT MRAM Technology

Md Rubel Sarkar

(ABSTRACT)

This thesis work presents an unique architecture with a capacity of 1.57-Mb storage including in-memory compuitng capability, leveraging state-of-the-art gate voltage assisted spin-orbit torque (VGSOT) magnetic random-access memory (MRAM) technology. Beyond its role as a non-volatile storage solution, this architecture facilitates a diverse array of In-Memory Computing (IMC) operations, inclusive of logic-inside-memory (LinM/LiM), in-memory-dot-product multiplication tailored for binary-neural-networks, and content-accessable memory (CAM). Our designed bit-cell proposed in this architecture occupies a compact area of 0.195 μm^2 and exhibits remarkable performance metrics. It achieves impressive writing speeds of 200 MHz and reading speeds of 1.5 GHz, applicable to non-volatile storage tasks and LinM operations. Notably, the LinM functionality supports a wide range of logical operations such as AND, NAND, OR, NOR, and MAJ, while the CAM feature enables efficient data searches of up to 1024 bits. Furthermore, in performance evaluations conducted using the MNIST and FMNIST datasets with a BNN model structured as 512-512-10 (input layer - hidden layer - output layer), the proposed VGSOT MRAM demonstrates exceptional inference accuracy. Specifically, it achieves a high accuracy rate of 97.40% for the MNIST dataset and 84.15% for the FMNIST dataset. In comparison to the 2T1R SOT-MRAM technology, the proposed VGSOT MRAM showcases significant advancements in read performance and reliability metrics. Notably, it features a 65.74% reduction in bit-cell area, alongside 84.78% and 33.4% lower read-write power consumption and 54.11% and 30.57% reduced LinM power consumption, respectively.

The Future of Computing: An Energy-Efficient In-Memory Computing Architectures with Emerging VGSOT MRAM Technology

Md Rubel Sarkar

(GENERAL AUDIENCE ABSTRACT)

This work brings forth towards a new technology called VGSOT MRAM, which is a type of memory device that can store information without using extensive power. Its part of a larger architecture called IMC, which has many useful features. One of the main advantages of this technology is that it can perform different operations while storing data. For example, it can do calculations, search for specific information, and perform tasks for artificial intelligence networks. The design of the memory cells is also very efficient, taking up a small amount of space. In terms of performance, this technology is quite impressive. It can write data very quickly, at a speed of 250 million times per second, and read data even faster, at 1.67 billion times per second. It can also perform different logical operations, like AND, OR, and NAND, which are important for many computing tasks when tested with real-world tasks, such as recognizing images, this technology showed excellent accuracy. It achieved a recognition accuracy of 97.40% for the MNIST dataset and 84.15% for the FMNIST dataset, which is quite good. Compared to other similar technologies, this VGSOT MRAM has some advantages. It takes up less space, uses less power when reading and writing data, and consumes less power when performing calculations. These improvements make it a promising option for future devices.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the realm of modern computing, where data is growing at an astonishing rate and real-time performance is the key to success, traditional disk-based storage and processing approaches often fall short [1]. It is in this context that in-memory computing emerges as a game-changing technology, revolutionizing the way we handle and process data [2].

In-memory computing, as the name suggests, revolves around the concept of storing and processing data in the main memory (RAM) of a computer or a distributed cluster of machines, instead of relying on slower, disk-based storage systems [3]. By leveraging the immense speed and random access capabilities of memory, in-memory computing unlocks unprecedented performance gains, enabling organizations to tackle complex problems, process massive datasets, and deliver real-time insights with remarkable speed and efficiency [4].

The fundamental principle behind in-memory computing is simple yet powerful: keep as much data as possible in memory, making it instantly accessible to applications and analytics engines [5]. This approach eliminates the need for costly and time-consuming disk I/O operations, which have traditionally been a major bottleneck in data processing pipelines.

With data residing in memory, operations such as querying, aggregating, and analyzing can be performed at lightning-fast speeds, accelerating decision-making processes and enabling businesses to respond rapidly to changing market conditions [6].

In-memory computing has a broad range of applications that span multiple industries and domains. Its utilization extends beyond any specific field, finding relevance in diverse sectors such as object detection, weather prediction, stock market data analysis, e-commerce, telecommunications, bio-medical fields, and more [7]. For example, within the realm of finance, in-memory computing plays a crucial role in facilitating real-time risk analysis and fraud detection [8]. By enabling rapid processing of extensive transactional data, it empowers financial institutions to swiftly identify and mitigate potential risks. Similarly, in the e-commerce industry, the speed and responsiveness of in-memory computing are harnessed to provide personalized recommendations and enable real-time inventory management, enhancing the overall customer experience [9]. The healthcare sector also benefits from in-memory computing, as it enables real-time analysis of patient data, facilitating timely diagnoses and aiding in treatment decisions. By leveraging the power of in-memory computing, various fields and industries can unlock new levels of efficiency, accuracy, and responsiveness in their operations [10].

In-memory computing (IMC) holds immense potential for revolutionizing the performance and efficiency of deep neural networks (DNNs) [11]. By leveraging the proximity of data storage and processing within memory units, IMC drastically reduces data movement and latency, overcoming bottlenecks inherent in traditional computing architectures. This proximity enables parallel processing of vast datasets directly within memory, facilitating faster inference and training of DNNs [12, 13]. Additionally, IMC architectures can exploit the inherent parallelism and high bandwidth of memory systems, leading to significant improvements in energy efficiency and computational throughput.

In-memory computing (IMC) promises to revolutionize 6G wireless applications by embedding computation directly into memory units [14]. This approach reduces data movement and latency, enhancing the efficiency and responsiveness of communication networks [15]. IMC enables real-time signal processing tasks like channel equalization and beamforming, as well as intelligent resource allocation through machine learning algorithms [16, 17, 18]. Leveraging non-volatile memory (NVM), IMC facilitates efficient data storage and analysis, optimizing network performance and reliability.

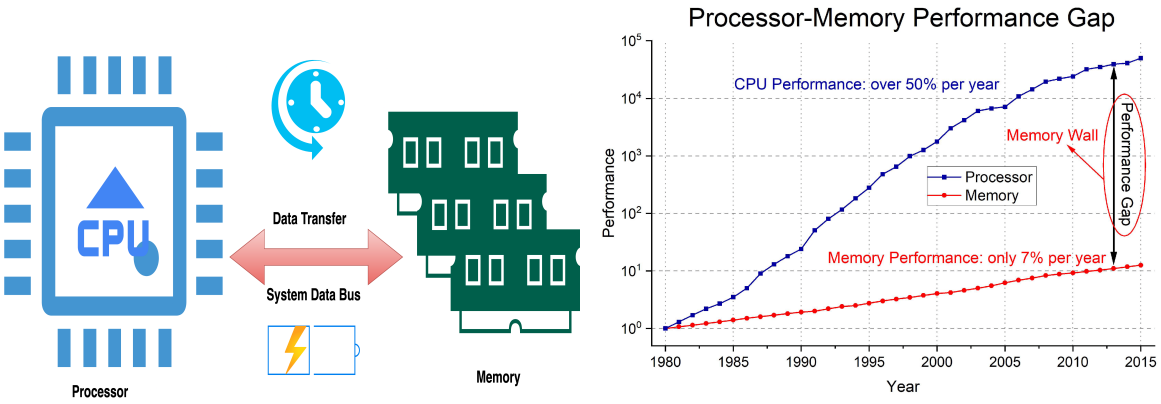
However, in-memory computing is not without its challenges. The cost of memory is typically higher than disk-based storage, and organizations must carefully balance the size of the dataset they can afford to keep in memory. Additionally, ensuring data consistency and durability in the event of power failures or system crashes requires sophisticated techniques, such as replication and check pointing.

Despite these challenges, the benefits of in-memory computing are compelling, and its adoption is on the rise. As memory prices continue to decline and technologies evolve, the potential for in-memory computing to transform the way we process and analyze data is immense. With its ability to deliver real-time insights, accelerate application performance, and drive innovation, in-memory computing holds the promise of unlocking new possibilities and reshaping the future of computing.

1.1 Von Neumann Processor-Memory Bottleneck

In conventional Von Neumann computing architectures, the processing and memory components operate as separate entities interconnected through either a system bus or a Network-on-Chip (NoC) [19], as illustrated in Figure 1.1a. While these architectures have long served as the cornerstone of computing systems and will likely continue to do so, they exhibit in-

efficiencies when confronted with modern computing workloads such as data analytics and machine learning [20]. In Von Neumann systems, the execution of these workloads necessitates frequent data transfers between the processor and memory subsystems [1.1a], resulting in a notable consumption of system energy and time [21]. This energy and time expenditure during data transfers represents an inefficiency, as it does not contribute to productive computations.



(a) Separate Processor and Memory (b) Processor-Memory Performance Gap

Figure 1.1: Von-Neumann Computing System.

Another inefficiency inherent in Von Neumann systems stems from what is commonly referred to as the processor-memory gap, as depicted in Figure 1.1b. This gap signifies the performance mismatch between the processor and memory components [22]. In contemporary computing workloads, there's a prevalent need to access substantial volumes of relatively slow memory and execute comparatively straightforward computations on the fetched data [23]. As a result, bottleneck scenarios emerge at the processor-memory interface, leading to a degradation in overall system performance.

1.2 Near-Memory Computing (NMC)

Near Memory Computing (NMC) is transforming the way data is processed and workloads are accelerated in modern computing systems [2]. Traditionally, data movement between the main memory and the processor has been a significant bottleneck. However, NMC challenges this paradigm by integrating processing units, such as accelerators and specialized logic, within or near the memory subsystem [24]. This proximity allows computations to be performed directly on the data, reducing operational delay and energy consumption. NMC leverages advancements in memory technologies like High-Bandwidth Memory (HBM) and 3D XPoint to enable this tight integration.

The advantages of NMC are substantial. Firstly, it enhances performance by reducing data movement and minimizing latency [25]. Tasks that require intensive computation, such as AI inference and graph analytics, benefit greatly from the proximity of computation to data. Real-time processing of large datasets becomes possible, improving system responsiveness. Secondly, NMC improves energy efficiency by eliminating the energy overhead associated with data movement. This is particularly valuable in power-constrained environments, leading to longer battery life, reduced cooling requirements, and lower operating costs.

Scalability and memory bandwidth are also enhanced by NMC architectures. Leveraging high memory bandwidth offered by advanced memory technologies, NMC enables efficient processing of large datasets, parallel execution of tasks, and improved scalability. Memory-level parallelism becomes achievable, where multiple memory banks or channels can be accessed simultaneously, further boosting performance and throughput.

NMC finds applications in various domains. NMC (Neuromorphic Computing) architectures have emerged as efficient solutions in the realm of artificial intelligence, including machine learning, particularly for tasks such as data preprocessing, model training, and inference.

[26, 27]. Real-time AI applications at the edge become feasible due to the proximity of computation to memory. In big data analytics, NMC accelerates data-intensive tasks such as data filtering, aggregation, and complex queries. Real-time insights and faster decision-making are enabled, allowing businesses to extract value from their data more efficiently. In high-performance computing, NMC transforms simulations, scientific computations, and computational fluid dynamics by performing computations directly on massive datasets residing in memory. Complex simulations and scientific discoveries become more accessible and time-efficient [28].

Despite its potential, NMC faces challenges that need to be addressed. Efficient NMC architectures, programming models, and software frameworks need to be developed. Algorithms also require optimization for data-centric computation. Considerations such as data security, reliability, and cost-effectiveness must be carefully evaluated [29]. However, the future of NMC is promising. Ongoing research and innovation in NMC designs, emerging memory technologies, and programming paradigms will drive the field forward. As NMC continues to evolve, we can expect transformative applications and exciting opportunities in data-intensive computing. Near Memory Computing is poised to shape the future of computing and drive innovation across industries.

1.3 In-Memory Computing (IMC)

To address the limitations of the Von Neumann computing architecture as discussed in Section 1.1, researchers are exploring alternative approaches. Various strategies have been proposed to enhance the efficiency of traditional computing systems [30].

To begin with, improving the memory hierarchy involves integrating diverse memory types (such as cache memory, main memory, and non-volatile memory) and employing techniques

like prefetching, caching, and optimizing memory access. These measures aim to diminish memory access latencies and mitigate the gap between the processor and memory. To optimize performance and tackle complex computational tasks, several approaches have been proposed in the field of computer architecture. These include enhancing parallelism at various levels, such as instruction-level parallelism (ILP), thread-level parallelism (TLP), and data-level parallelism (DLP). Additionally, the development of specialized hardware accelerators, such as graphics processing units (GPUs) for parallel processing and tensor processing units (TPUs) for machine learning tasks, has been proposed. These approaches aim to leverage parallelism and harness the power of specialized hardware to achieve efficient and high-performance computing for a wide range of applications. [31].

Other notable solutions include bringing memory devices closer to computing units, known as near-memory computing, and exploring in-memory computing techniques. These approaches seek to minimize data movement and optimize computational efficiency.

1.3.1 What is In-Memory Computing?

In the realm of computing, a groundbreaking concept known as in-memory computing (IMC) has emerged to revolutionize traditional architectures. Unlike conventional methods that involve transferring data between memory and processing units, IMC executes data processing directly on memory chips. This paradigm shift eliminates the bottleneck associated with data transfer, resulting in significant reductions in logic execution delay and energy expenditure, ultimately resulting in highly energy efficiency and rapid computation [32].

However, despite its numerous advantages, IMC faces certain challenges, particularly when it comes to CMOS-based memory macros. These challenges encompass a range of issues, including sub-threshold leakage, single event upsets (SEUs) vulnerability, susceptibility to

noise, and variations in process, voltage, and temperature (PVT) [33, 34].

To overcome these hurdles and further enhance memory performance, spintronics-based magnetic random-access memory (MRAM) devices have emerged as promising solutions [35]. MRAM encompasses a variety of technologies, such as spin hall effect (SHE) MRAM, spin transfer torque (STT) MRAM, differential spin hall effect MRAM, racetrack (RT) MRAM, spin-orbit torque (SOT) MRAM, and gate voltage assisted spin-orbit torque (VGSOT) MRAM [36, 37].

Compared to traditional memory technologies, MRAM devices offer several distinct advantages [38, 39]. They are compatible with CMOS technology and boast a straightforward design. Additionally, they exhibit non-volatility, minimal leakage, minimal energy utilization, rapid performance, robustness or longevity, thermal steadiness, long-term data retention, and high integration density [40, 41].

Moreover, the integration of MRAM into computing systems holds immense potential to revolutionize various applications, including but not limited to artificial intelligence, Internet of Things (IoT) devices, edge computing, and high-performance computing (HPC). Given its unique combination of characteristics, MRAM stands as a promising candidate for next-generation memory solutions, paving the way for more efficient and powerful computing architectures.

1.3.2 Current Advancement of In-Memory Computing

Current advancements in processing in memory (PIM) or in-memory computing (IMC) are driving significant innovations across various domains, from artificial intelligence to high-performance computing [42, 43]. One key area of progress is the integration of specialized processing elements directly into memory units, enabling efficient and parallelized compu-

tation on massive datasets with minimal data movement. Emerging technologies such as resistive random-access memory (RRAM) and magnetic random-access memory (MRAM) are being explored for their potential to perform logic operations within memory cells, offering unprecedented levels of energy efficiency and computational density [42, 43]. Moreover, advancements in hardware-software co-design are enabling the seamless integration of PIM capabilities with existing computing architectures, enabling applications such as deep learning inference, database querying, and scientific simulations to benefit from the speed and efficiency gains of in-memory computation [44]. Additionally, research efforts are focusing on developing novel algorithms and programming models optimized for PIM architectures, further enhancing their scalability and applicability across diverse workloads [45]. Overall, the current advancements in PIM and IMC are poised to unlock new levels of performance, efficiency, and scalability, paving the way for transformative advancements in computing and data processing.

Memory-centric computer architectures have been introduced and developed intensively to overcome the limitations of current technology and existing architecture performance. Several classes of memory-centric architectures have been developed so far. Some of them are DRISA-3T1C, CRS, PLiM, ReVAMP [46, 47].

1.3.3 Advantages and Challenges of In-Memory Computing

In-memory computing offers numerous advantages that have generated substantial interest among researchers and practitioners. One key advantage is the significantly faster data access and processing speeds enabled by storing data in the main memory [33]. With reduced latency compared to disk-based storage, in-memory computing facilitates real-time analytics, faster transaction processing, and improved response times for interactive applications [48].

Complex analytical queries and algorithms can be executed directly on the data residing in memory, eliminating the need for expensive and time-consuming data transfers between disk and memory [49]. This direct access to data in memory also enables efficient data caching, mitigating performance bottlenecks in data-intensive applications [50]. Overall, in-memory computing empowers organizations to achieve enhanced performance, accelerated data processing, and streamlined real-time decision-making [51].

Although in-memory computing provides significant benefits, it also poses specific challenges that require attention and resolution [52]. One significant challenge is the cost associated with acquiring and maintaining large amounts of memory to accommodate the entire dataset. The expense of high-speed memory can be prohibitive, particularly for organizations dealing with massive datasets [24]. Additionally, the volatility of main memory necessitates robust data backup and recovery mechanisms to safeguard against data loss in the event of system failures or power outages [50]. Furthermore, the scalability of in-memory computing systems may be limited due to memory constraints and the requirement for specialized hardware. These challenges require careful consideration and innovative solutions to optimize the cost-effectiveness, reliability, and scalability of in-memory computing systems [48]. Despite these hurdles, the advantages of in-memory computing, including improved performance, real-time analytics, and simplified data processing, make it a compelling approach with the potential to revolutionize various domains of computing.

1.3.4 Impact of IMC in Neural Network

In-memory computing (IMC) has emerged as a game-changer in the field of neural networks, revolutionizing their speed, efficiency, and scalability [53, 54]. By integrating processing units directly within or near the memory subsystem, IMC enables computations to be performed

directly on the data, eliminating the need for data movement and reducing latency. This proximity has a profound impact on the training process of neural networks. Traditionally, training large-scale models with extensive datasets has been a time-consuming and computationally intensive task [55]. However, with IMC, training times are significantly reduced as computations are performed directly on the data stored in memory. This breakthrough in speed opens up new possibilities for researchers and developers, allowing them to experiment with larger and more complex models and accelerate the pace of innovation in the field.

Real-time inference is another area where IMC has made a remarkable impact. In applications such as autonomous vehicles, natural language processing, and real-time analytics, quick and accurate responses are crucial [56]. IMC enables high-speed inference by performing computations directly on the data residing in memory. This eliminates the need to retrieve data from external storage, resulting in near-instantaneous inference times. Real-time decision-making becomes feasible, enabling applications that require immediate responses to operate efficiently.

Energy efficiency is a crucial consideration in modern computing systems, and IMC addresses this concern effectively [57]. Traditional computing architectures often involve significant data movement, which incurs energy overhead. By performing computations directly on data in memory, IMC minimizes data movement and reduces the associated energy consumption [58]. This energy efficiency is particularly valuable in power-constrained environments, such as mobile devices and edge computing, where optimizing energy consumption is essential. IMC enables longer battery life, reduces cooling requirements, and lowers operating costs, making neural networks more sustainable and economical.

Scalability is a fundamental requirement for neural networks, especially as models and datasets continue to grow in size and complexity [59]. IMC excels in this aspect by leveraging the high memory bandwidth offered by advanced memory technologies. The proximity of

computation to memory allows for efficient processing of large neural networks and parallel execution of tasks. Multiple memory banks or channels can be accessed simultaneously, enabling memory-level parallelism and boosting overall performance and throughput. This scalability is critical for handling massive datasets and enables neural networks to tackle increasingly challenging problems.

In conclusion, in-memory computing has had a profound impact on neural networks, transforming their speed, efficiency, and scalability [60]. By eliminating data movement and performing computations directly on data in memory, IMC accelerates training times, enables real-time inference, enhances energy efficiency, and improves scalability [61]. Ongoing research and innovation in IMC designs, programming models, and hardware accelerators will continue to push the boundaries of neural network capabilities. As IMC continues to evolve, we can expect even faster training times, real-time inference on resource-constrained devices, and the ability to process massive neural networks more efficiently. The future of neural networks powered by in-memory computing is promising, with far-reaching applications and potential for transformative advancements in artificial intelligence.

1.4 Non-Volatile Emerging Memory Devices

The pursuit of faster, smaller, and more energy-efficient electronic devices has driven extensive research and development in innovative memory technologies [62]. Among these advancements, non-volatile emerging memory devices have garnered significant attention for their potential to revolutionize various sectors, from consumer electronics to data storage and beyond [63]. These novel memory solutions promise enhanced performance, durability, and efficiency, paving the way for a new era of computing capabilities [45].

This overview provides a glimpse into the landscape of non-volatile emerging memory devices,

delving into their fundamental principles, key characteristics, and promising applications. From Resistive Random-Access Memory (RRAM) to Phase-Change Memory (PCM) and Ferroelectric RAM (FeRAM), Magnetic Random Access Memory (MRAM), each technology brings unique advantages and challenges, shaping the trajectory of future computing architectures [64, 65]. By gaining an understanding of the underlying mechanisms and current state-of-the-art developments, stakeholders in academia, industry, and beyond can grasp the transformative potential of these cutting-edge memory solutions. Through this exploration, we aim to shed light on the evolving memory landscape and its implications for the future of electronics and information technology.

1.4.1 Magnetic Random Access Memory (MRAM)

MRAM is a NVM technology that uses the magnetic properties of materials to cache data [66]. It employs magnetic tunnel junctions (MTJs) consisting of two magnetic layers separated by an insulating layer [67]. The relative orientation of the magnetization in the free layer represents data. MRAM, or Magnetoresistive Random-Access Memory, boasts impressive characteristics including rapid read/write speeds, minimal power consumption, remarkable endurance, and resistance to radiation [68]. It finds applications in cache memory, embedded systems, and automotive electronics. Ongoing research aims to improve storage density and reduce costs to unlock its full potential.

MRAM encompasses various types of memory technologies that utilize magnetic properties for data storage. One prominent variant is Spin-Transfer Torque MRAM (STT-MRAM), which employs a spin-polarized current to switch the magnetization direction in the free layer of the magnetic tunnel junction (MTJ) [69]. STT-MRAM offers high endurance, fast switching speeds, and scalability. Another type is Spin-Orbit Torque MRAM (SOT-MRAM), which

utilizes spin-orbit torque to manipulate the magnetization in the free layer. This technology has potential advantages in terms of lower power consumption and improved scalability [70]. Domain Wall MRAM (DW-MRAM) relies on the movement of magnetic domain walls to store and retrieve data, while Spin Hall Effect MRAM (SHE-MRAM) utilizes the spin Hall effect for magnetization switching. Each MRAM variant presents unique characteristics and challenges, and ongoing research seeks to optimize them for various applications. The choice of MRAM type relies on the specific requirements, and performance considerations of the target application.

Spin Transfer Torque (STT) MRAM

STT-MRAM, or Spin-Transfer Torque Magnetoresistive Random-Access Memory, is a type of non-volatile memory that utilizes the magnetic properties of electrons to store data [71]. In STT-MRAM, data is stored by controlling the direction of electron spins within a magnetic tunnel junction (MTJ) [68]. This is achieved by passing a current through the MTJ, which creates a torque on the magnetic moments of the electrons, flipping their spins to represent either a "0" or a "1." This allows for fast read and write operations, low power consumption, and high endurance compared to traditional memory technologies like DRAM or NAND flash [72].

Unlike conventional memory technologies that rely on the storage and movement of electrical charge, STT-MRAM leverages the spin of electrons, providing several advantages including lower power consumption, faster operation, and improved scalability [73]. By manipulating the spins of electrons with a current, STT-MRAM can achieve high-density data storage with excellent retention properties [66]. Additionally, its non-volatile nature ensures that data remains intact even when power is removed, making it suitable for a wide range of applications, from consumer electronics to data center storage solutions.

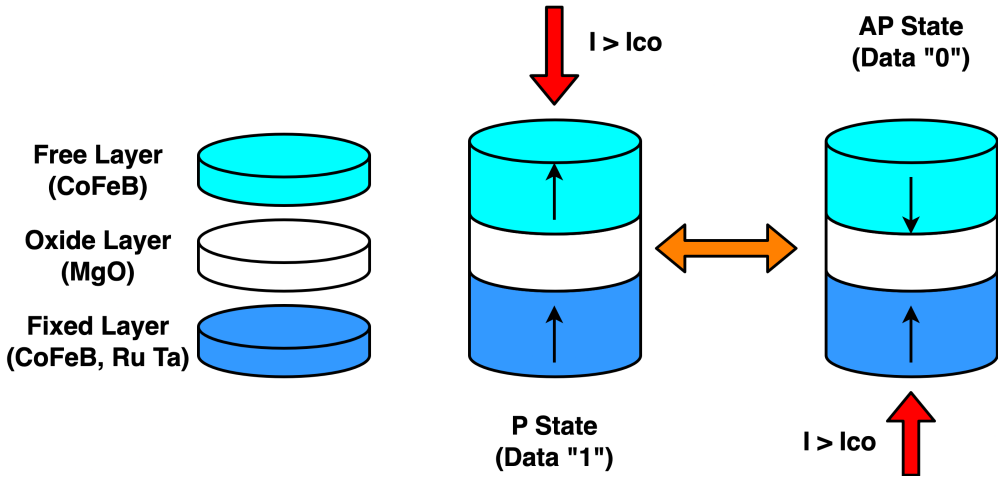


Figure 1.2: STT-MRAM Device Structure.

Spin Orbit Torque MRAM

Spin Orbit Torque MRAM (SOT-MRAM) is an emerging type of magnetoresistive random-access memory that relies on spin-orbit coupling to control the magnetic state of the memory cells. Unlike conventional STT-MRAM, which uses electrical current to manipulate the spins of electrons, SOT-MRAM utilizes spin-orbit coupling to generate a torque on the magnetic moments within the memory cells. This torque is induced by passing an electrical current through heavy metal layers with strong spin-orbit coupling, which in turn affects the magnetization direction of the magnetic layer in the memory cell.

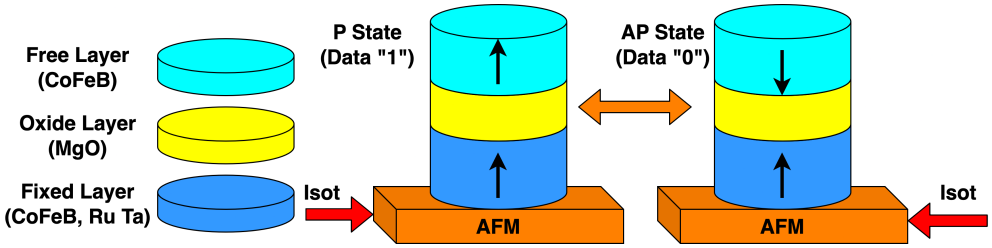


Figure 1.3: SOT-MRAM Device Structure.

SOT-MRAM offers advantages such as lower write currents, improved scalability, and reduced susceptibility to thermal stability issues compared to STT-MRAM, making it a promis-

ing candidate for next-generation non-volatile memory technologies.

1.4.2 Resistive Random Access Memory (RRAM)

Resistive Random Access Memory (RRAM) is a type of non-volatile memory that stores data by varying the resistance of a solid-state material [62, 74]. In RRAM, data is stored as different resistance states, typically high resistance (representing "0") and low resistance (representing "1") [75]. This resistance switching is achieved by applying voltage to the RRAM cell, which causes the movement of oxygen vacancies or metal ions within the material, altering its conductivity [76]. RRAM offers advantages such as fast read and write speeds, low power consumption, high endurance, and scalability [77]. These properties make RRAM promising for various applications, including embedded systems, storage-class memory, and neuromorphic computing, where its non-volatile nature and high-density storage capabilities are highly desirable [78, 79].

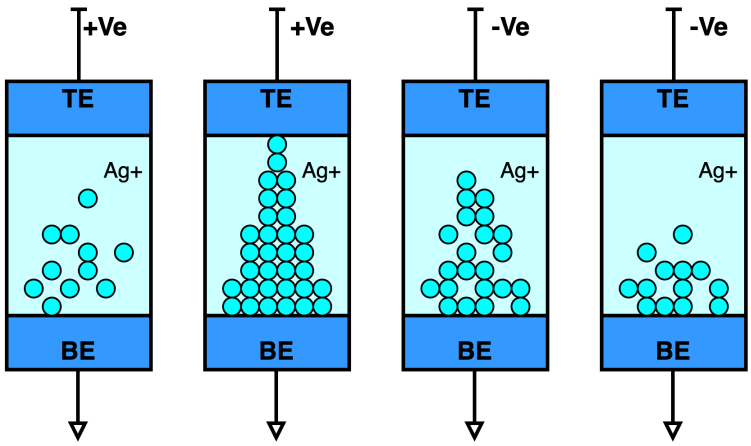


Figure 1.4: RRAM Device Structure.

Additionally, RRAM has the potential to complement or even replace existing memory technologies like NAND flash and DRAM due to its superior performance characteristics and compatibility with advanced manufacturing processes.

1.4.3 Phase Change Random Access Memory (PCRAM)

Phase Change Random Access Memory (PCRAM) is a type of non-volatile memory that utilizes the unique properties of chalcogenide glass materials to store data [80, 81]. In PCRAM, data is stored by exploiting the reversible phase transition between amorphous and crystalline states in these materials [82]. By applying electrical pulses, the chalcogenide material can be switched between these states, representing binary data as distinct resistance levels [83, 84]. PCRAM offers several advantages, including fast read and write speeds, high endurance, low power consumption, and scalability [62]. These properties make it suitable for various applications, including embedded systems, storage-class memory, and high-speed cache memory in computing devices [85, 86].

Moreover, PCRAM's non-volatile nature ensures data retention even when power is removed, making it a reliable choice for data storage solutions. As a result, PCRAM holds significant potential as a next-generation memory technology capable of meeting the increasing demands of modern computing systems.

1.4.4 Ferroelectric Random-Access Memory (FeRAM)

Ferroelectric Random-Access Memory (FeRAM) is a type of non-volatile memory that utilizes ferroelectric materials to store data [87]. Unlike traditional volatile memory like DRAM, FeRAM retains data even when power is turned off [88]. FeRAM stores data by polarizing the ferroelectric material, typically a lead zirconate titanate (PZT) compound, in one of two stable states representing binary "0" or "1." Data can be written by applying an electric field to the ferroelectric material, which causes polarization, and read by measuring the resulting electrical polarization [89].

FeRAM offers advantages such as fast read and write speeds, low power consumption, high

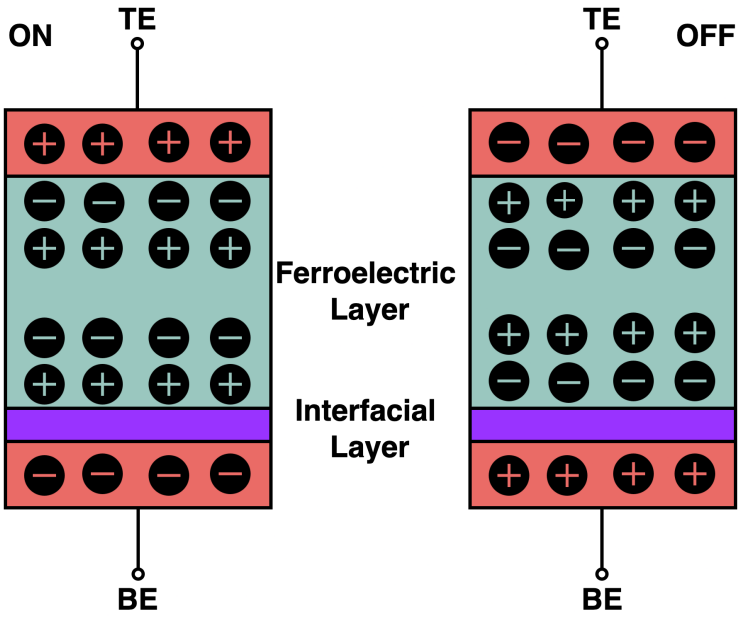


Figure 1.5: FeFET Device Structure.

endurance, and resistance to radiation, making it suitable for applications requiring reliable and high-performance memory, such as embedded systems, automotive electronics, and smart cards [90, 91, 92]. However, FeRAM's relatively high production costs and lower storage density compared to other memory technologies like NAND flash have limited its widespread adoption in consumer electronics but it remains a promising technology for specific applications requiring fast, non-volatile memory.

Chapter 2

VGSOT MRAM based In-Memory Computing Architecture

2.1 VGSOT IMC Architecture Specifications

In our research project, we present a novel In-Memory Computing (IMC) architecture that utilizes an advanced gate voltage assisted spin-orbit torque (VGSOT) magnetic random-access memory (MRAM) device. This MRAM device has a substantial capacity of 1.57 Mb. Apart from functioning as a non-volatile memory (NVM) storage system, this design offers a diverse set of IMC operations. These operations include performing logic operations within the memory array (LiM), enabling content-addressable-memory (CAM) functionality, and conducting in-memory dot product operations. These capabilities allow for the construction of binary neural networks (BNN) for image dataset classification. The bit-cell designed in this architecture boasts a remarkably compressed area of 0.195 μm^2 while delivering impressive performance metrics. This IMC architecture can write data in the memory array with a speed of 250 MHz and a reads data with a speed of 1.67 GHz, applicable to both NVM

and LiM operations. Notably, the LiM functionality supports a diverse array of logical operations, encompassing AND, NAND, OR, NOR, and majority (MAJ) operations, while CAM enables efficient data searches spanning up to 1024 bits. Furthermore, in performance evaluations conducted using the MNIST and FMNIST datasets with a BNN model structured as 512-512-10(input or first layer - hidden layer - final or output layer), remarkable inference accuracies of 97.51% and 84.03% have been achieved, respectively. The proposed VGSOT MRAM device exhibits improvements in read performance and reliability over conventional 2T1R SOT-MRAM technology. Specifically, it features a 65.74% reduction in bit-cell area, along with 84.78% and 33.40% lower read-write power consumption and 54.11% and 30.57% reduced LiM power consumption, respectively.

More specifically, the innovations encompass the following key aspects -

- The design highlights an unique 4-Transistor-1-MTJ 4T1M bit-cell architecture, which integrates the cutting-edge VGSOT MRAM device, promising advancements in memory technology. This configuration offers enhanced performance and reliability, catering to the evolving demands of modern computing systems.

- Employing an intricately devised pipelined structure, the clock signal gated triple word line decoder (TWLDR) is augmented with multi-bit flip-flop (MBFF) functionality. This design optimization ensures streamlined operation and efficient utilization of resources, contributing to overall system efficiency and speed.

- A sophisticated self-referenced separately pre-charged sense amplifier (SPCSA) is introduced, enhancing in-memory logic operations. By utilizing self-reference techniques, this amplifier ensures robust and accurate data sensing, vital for critical computing tasks in various applications.

- The seamless integration of CAM (content-Addressable Memory) enriches network de-

vices with accelerated content-based search capabilities. This integration empowers networking solutions with swift and efficient data retrieval, enhancing overall system responsiveness and performance.

- The implementation of lightweight advanced Binary Neural Network (BNN) circuitry revolutionizes image detection capabilities. Leveraging innovative design principles, this circuitry enables swift and accurate image processing, facilitating tasks ranging from object recognition to pattern analysis in diverse applications.

2.2 VGSOT MRAM Device Overview

The VGSOT MRAM device stands out for its innovative three-terminal architecture, featuring distinct read and write pathways. Unlike basic or regular spin-orbit-torque MRAM devices, it is structured by integrating an anti-ferromagnetic (AFM) metal layer with an oxide layer (e.g. MgO). The oxide layer is sandwitched between a free layer (FL) and, a pinned layer (PL)[93, 94]. The arrangement of AFM-Oxide-FM in the perpendicular magnetic tunnel junction (p-MTJ) allows for a field-free mechanism of SOT switching, supported by a delicate exchange bias (H_{EX}) serving as an in-plane magnetic field, thus enabling switching without the need for an external field [95, 96].

The concept of voltage-controlled magnetic anisotropy (VCMA) plays a pivotal role in enhancing SOT switching efficiency, denoted as gate voltage assisted SOT (VGSOT). VCMA lowers the energy barrier temporarily at the time of MTJ switching by providing a bias voltage across the p-MTJ. This manipulation, quantified by the VCMA coefficient β, introduces a transformative approach to MRAM technology, as illustrated in Eq. 2.1. This innovative combination of field-free SOT, H_{EX}, and VCMA not only ensures practical and energy-efficient MRAM devices but also promises super-low power utilization, rapid speed,

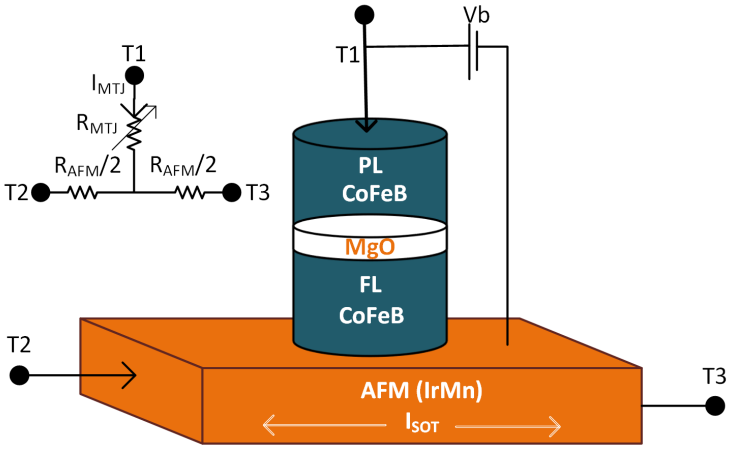


Figure 2.1: VGSOT MRAM Device Structure.

highly densed system, and minimal switching errors.

$$\vec{H}_{EX} = \mathrm{H_{EX}}\vec{e}_y, \overrightarrow{\boldsymbol{H}}_{VCMA} = -\frac{2\beta V_{\mathrm{MTJ}}}{\mu_0 M_{\mathrm{S}} t_{\mathrm{ox}} t_{\mathrm{f}}} m_z \overrightarrow{\boldsymbol{e}}_z \tag{2.1}$$

$$\mathrm{TMR_{real}}\ (\mathrm{V_{MTJ}}) = \frac{\mathrm{TMR}}{1 + \mathrm{V_{MTJ}^2}/\mathrm{V_h^2}} \tag{2.2}$$

$$R_{\mathrm{MTJ}}(V_{\mathrm{MTJ}}) = \frac{R_{\mathrm{P}}\left[1 + \left(\frac{V_{\mathrm{MTJ}}}{V_{\mathrm{h}}}\right)^2 + \mathrm{TMR}\right]}{1 + \frac{V_{\mathrm{MTJ}}^2}{V_{\mathrm{h}}^2} + [0.5(1+\cos\theta)]\,\mathrm{TMR}} \tag{2.3}$$

Unlike traditional MRAM devices, the VGSOT MRAM (Voltage-Generated Spin-Orbit Torque MRAM) adopts an in-plane current for the writing process. This approach circumvents the breakdown of the oxide barrier and mitigates errors that may arise from high tunneling currents. In order to address read disturbance issues, the retrieval of data from the Magnetic Tunnel Junction (MTJ) employs an out-of-plane current. The VGSOT MRAM thus represents a significant advancement in memory technology, offering unparalleled performance

and reliability in various computing applications.

The VGSOT MRAM device parameters are summarized in Table 2.1. These parameters play crucial roles in defining the performance and characteristics of the device. Notably, the MgO barrier thickness (t_{ox}) and the free layer thickness (t_{sl}) influence tunnel magnetoresistance ratio (TMR) and switching characteristics. Exchange bias field (H_{y}) and AFM strip dimensions (thickness d, width w, and length l) are critical for controlling magnetic properties and stability. Additionally, the VCMA coefficient (β), anisotropy energy (k_i), and resistivity of the AFM strip (ρ) significantly impact energy efficiency and performance.

Table 2.1: VGSOT MTJ Device Parameters

Parameters	**Descriptions**	**Default Value**
t_{ox}	MgO Barrier	1.4 nm
t_{sl}	Free Layer Thickness	1.1 nm
TMR (%)	VGSOT Tunnel Magnetoresistance Ratio	100
H_{y} (Oe)	MTJ Exchange Bias Field	-50
d (nm)	AFM Strip Thickness	3 nm
w (nm)	AFM Strip Width	50 nm
l (nm)	AFM Strip Length	60 nm
a (nm)	MTJ Surface Length	25 nm
b (nm)	MTJ Surface Width	50 nm
r (nm)	MTJ Surface Radius	25 nm
ρ (Ω-m)	Resistivity of AFM Strip	2.78×10^{-8}
η	Spin Hall Effect	0.25
k_i (J/m^2)	Anisotropy Energy	0.32×10^{-3}

The interplay of these parameters allows for the fine-tuning of the VGSOT MRAM device, ensuring optimal functionality and reliability in various operational scenarios. By meticulously adjusting these parameters, researchers can tailor the device to meet the specific requirements of diverse applications, from embedded systems to high-performance computing platforms.

These parameters collectively contribute to the performance and functionality of the VGSOT

MRAM device, making it a versatile and adaptable solution for diverse memory applications. The comprehensive understanding and optimization of these parameters are essential for unlocking the full potential of VGSOT MRAM technology.

2.3 VGSOT MRAM IMC Architecture Sub-Blocks

2.3.1 4T1M Bit-Cell

A pioneering 4T1M bit-cell design has been engineered (as depicted in Figure 2.2), wherein access transistors N1 and N3 facilitate the read and write operations. Notably, transistor N2, serving as an n-type diode, administers the write bias voltage (V_b) through net T1 while safeguarding against data overwrite during read operations. Meanwhile, transistor N0 plays a pivotal role while executing CAM operation. Notably, transistors MN0 and MN1 share their source (RT) and drain (T1) terminals, yet possess distinct gate terminals (TCL, RWL) for CAM control and read operations, respectively.

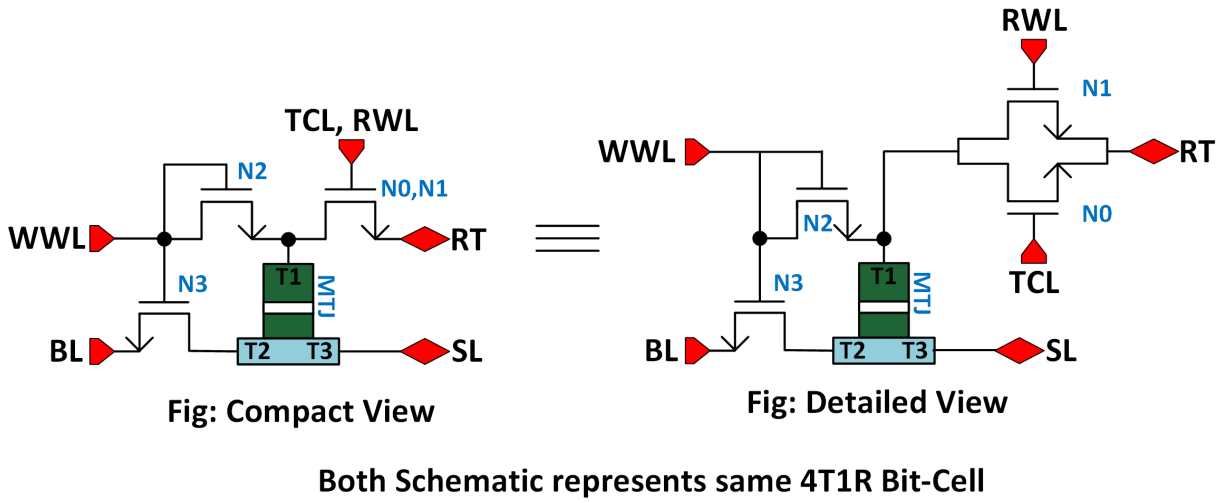


Figure 2.2: 4T1M Bit-Cell, Compact and Detailed Schematic View.

The write process involves setting BL to 300 mV and SL to 0 for data '1', prompting current flow from bit-line (BL) to source-line (SL). On the contrary, when the data is '0', the source line (SL) is adjusted to 300 mV while the bit line (BL) is set to 0, directing the current from SL to BL. This procedure consequently prompts a shift in the MTJ state to either anti-parallel (AP) (resulting in high resistance) or parallel (P) (resulting in low resistance) configurations, as depicted in Figure 2.2, causing a corresponding rise (662.755 kΩ) or drop (340.296 kΩ) in MTJ resistance. Remarkably, the device exhibits exceptional energy efficiency, requiring merely a low voltage (300 mV) for data writing, distinguishing it from STT or other SOT MTJ devices [94]. Moreover, the compact 0.195 μm^2 bit cell layout has been meticulously crafted based on the parameters of the VGSOT MTJ device detailed in Table 2.1.

2.3.2 Triple Word-Line Decoder (TWLDR)

Our research introduces a novel multibit flip-flop (MBFF) employed, pipelined, clock-gated (CKG) configured TWLDR capable of executing Non-Volatile Memory (NVM) and Logic-in-Memory (LiM) operations by activating single, dual, or triple WL addresses. Figure 2.3 illustrates a 3-to-8 bit TWLDR blockdiagram, incorporating multi-bit positive edge triggered registers (MBPR) and multi-bit negative edge triggered registers (MBNR). These registers allocate clock circuitry, leading to a dense flip-flop design that significantly minimizes the overall area footprint. Specifically, employing this approach in a 1024-bit TWLDR yields a notable 15.1% area savings compared to regular register-based designs using the 22FDX 7-track (7T) standard cell library.

The C1C0 terminals play a crucial role in determining the WL addresses for NVM, LiM, and Majority (MAJ) operations. During NVM operation (C1C0=00), only address A<2:0>

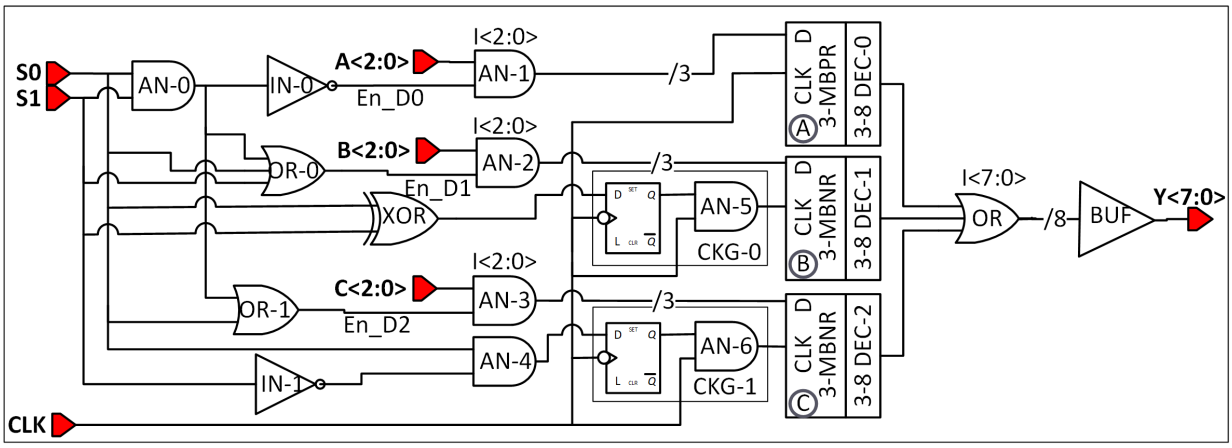


Figure 2.3: TWLDR Schematic.

is sampled by block-A and passes through the BUF, thereby deactivating CKG-0, 1, and DEC-1, 2 to conserve power when LiM operation is inactive. In the case of AND/OR LiM operation (C1C0=01), addresses A<2:0> and B<2:0> are sampled by block-A and B, while CKG-1 and DEC-2 remain disabled. During MAJ operation (C1C0=10), CKG-0, 1, and DEC-0, 1, 2 become active, allowing all three addresses to be decoded and passed through BUF. Depending on the activation of single, dual, or triple word lines, the power consumption varies, consuming approximately 10.38 μW, 20.76 μW, and 30.85 μW, respectively. Thus, the CKG scheme effectively retains approximately 51.36 μW of power.

As shown in figure-2.4, when C1C0=00, only address A<2:0> will be sampled, at this time address B<2:0>, and C<2:0> will not be sampled as block-B, C is disabled by CKG-0, and CKG-1, also DEC-0, 1 are disabled. When C1C0=01, address A<2:0>, and B<2:0> will be selected. When C1C0=10 all addresses A<2:0>, B<2:0>, and C<2:0> will be sampled. Decoders used in block-A, B, C only become enable when IN-0, OR-0, OR-1 become logic high. If LiM operation is not performed, registers, decoders block remain disabled, and save power consumption.

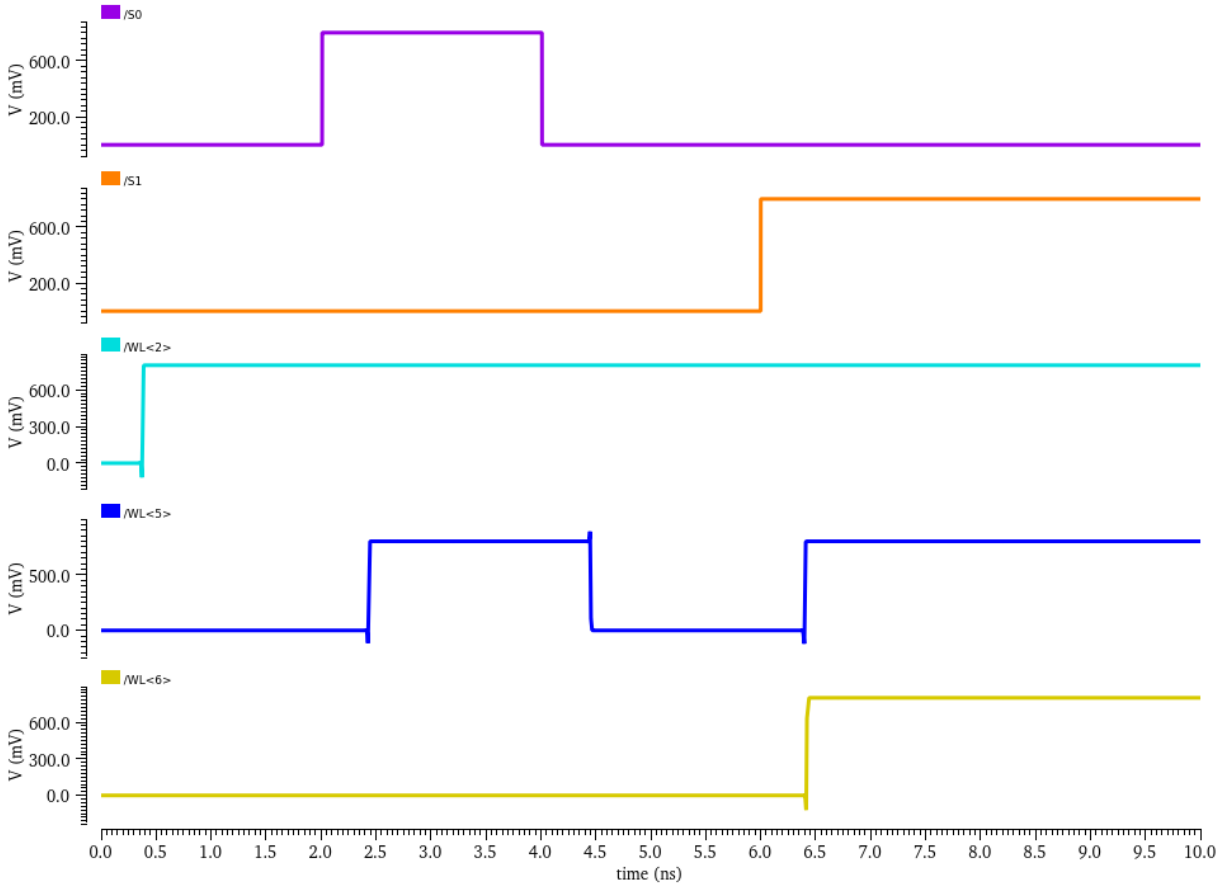


Figure 2.4: Triple Word Line Decoder (TWLDR) Simulation.

2.3.3 Separately Pre-Charged Sense Amplifier (SPCSA)

A novel Self-Referenced Separately Pre-Charged Sense Amplifier (SPCSA) has been developed to facilitate Non-Volatile Memory (NVM) and Logic-in-Memory (LiM) operations, as illustrated in Figure 2.5. The SPCSA comprises three key modules: pre-charge, evaluation, and reference generation. In the context of NVM or LiM operations, Magnetoresistive Tunnel Junctions (MTJs) from distinct rows (BC-P, BC-Q, BC-R) are interconnected to the BC_CNET in a parallel configuration within the SPCSA. This configuration undergoes precharging to ensure uniform initial conditions for both the Memory Under Test (BCUT) and the Base MTJ (BMTJ) during the subsequent evaluation phase.

During the pre-charge stage, the SEN (Select Enable) signal is set to logic low, thereby initially maintaining the VOUT and VOUTB signals at logic high. To generate the reference signals necessary for subsequent operations, the design employs various combinations of Magnetoresistive Tunnel Junction (MTJ) devices. By adopting this approach, the requirement for a separate current or voltage reference CMOS circuit is eliminated. This not only conserves silicon area but also addresses MTJ resistance drift issues. It is worth noting that any variations affecting the MTJ devices have a uniform impact on both the memory and Sense Amplifier (SA) blocks. The design incorporates two inverters (INV-01 and INV-02) along with NMOS transistors (MN2, MN3) to enhance the limited voltage or current difference between the Memory Under Test (BCUT) and the Base MTJ (BMTJ) during the discharge phase. By slightly increasing the width of the N3 transistor (50nm) compared to N2, a small voltage slew rate (SR) difference is created, facilitating logic evaluation when BCUT and RMTJ possess identical resistance values.

During the subsequent evaluation phase, if MN2 switches on first due to its higher slew rate, it pulls the VOUT net to zero, resulting in a logic low at VOUT and a logic high at

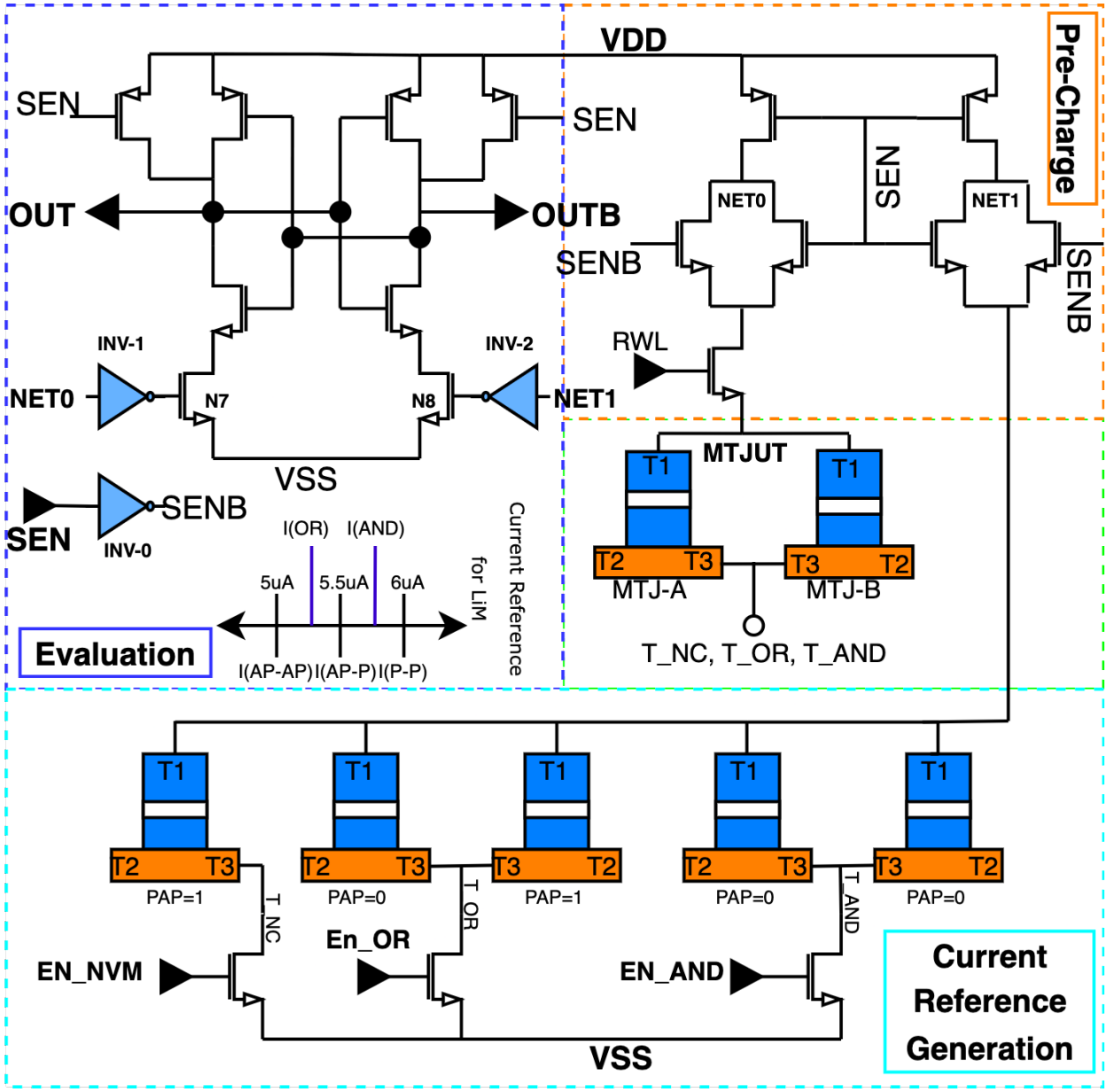


Figure 2.5: SPCSA for IMC Operations.

VOUTB. Conversely, the reverse scenario unfolds if MN3 activates first. Section **??** provides an illustrative example of AND logic evaluation, inclusive of simulation results demonstrating the slew rate values of transistors MN2 and MN3.

2.4 Working Principle of Different IMC Operation

2.4.1 Non-Volatile Memory Read-Write Opearation

To access data stored in the memory array, the Read Word Line (RWL) corresponding to the desired row must be activated. Following this, the Sense Pulse Current Sensing Amplifier (SPCSA) initiates a pre-charging process for the bit-cells. The activation of the EN_NVM terminal then enables the SPCSA to analyze and determine the stored data by measuring the current discharge rate between the main and reference bit-cells.

As depicted in Figure 2.6, when the EN_NVM signal is raised to logic high twice, occurring at 9ns and 23ns respectively, the bit-cell-1 (MS<1>) stores both data 0 and 1 within that timeframe. Consequently, precise read data is obtained from the VOUT<1> terminal of the SPCSA. It is noted that the SPCSA demonstrates the capability to execute read operations at a speed of 1.67-GHz.

2.4.2 Logic In-Memory Operation

LiM represents a breakthrough by allowing computations directly within the memory, eliminating the necessity of data transfer to and from the processor. This advancement promises substantial improvements in performance, energy efficiency, and scalability.

In Fig. 2.6, the LiM operation between bit-cell-1 (MS<1>) and bit-cell-9 (MS<9>) across

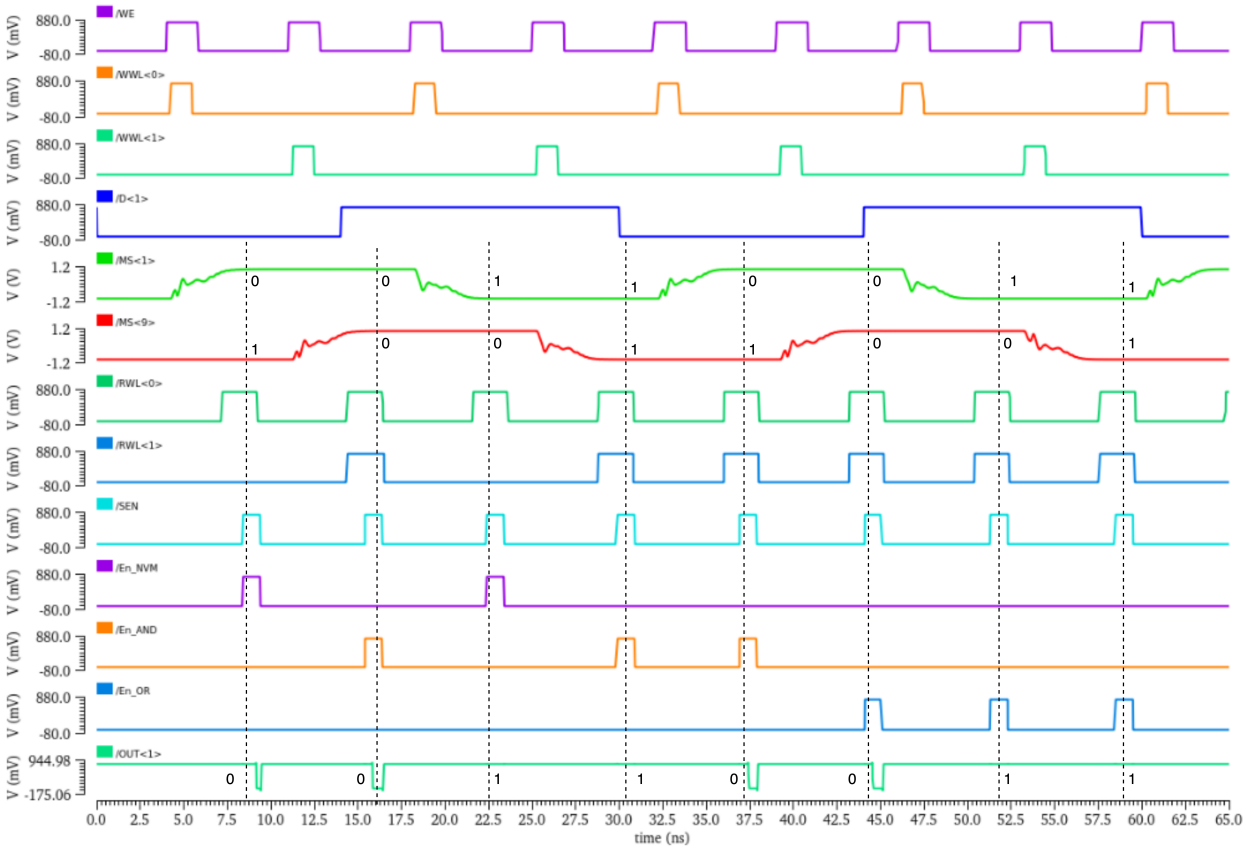


Figure 2.6: NVM and LiM Operation.

row-1 and row-2 is depicted. Upon activating RWL<0> and RWL<1> while setting the En_AND signal to logic high (at 16ns, 31ns, and 38ns), both bit-cells store data sequences of 00, 11, and 01. The resulting computed AND logic data from VOUT<1> is 010. Similarly, with the En_OR signal set to logic high (at 45ns, 51ns, and 59ns), both bit-cells store data sequences of 00, 10, and 01, leading to a computed OR logic data of 011 from VOUT<1>. Notably, the VOUTput VOUT<1> exhibits both NAND and NOR behavior when considering its complementary output.

For the SPCSA, it yields a high output for the OR operation only when both bit-cells (BC-X and BC-Y) store the data 0, while producing a logic high output for the AND operation when both bit-cells store the data 1.

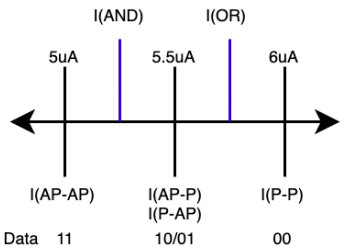


Figure 2.7: Current Reference for LiM Operation.

Recent advancements in ReRAM, STT, and SOT IMC architectures have embraced a dual bit-cell approach to store data in a complementary manner, facilitating LiM operations and improving the sense margin for the sense amplifier (SA). However, this approach necessitates 50% more bit-cells, increased write driver area, and higher write power. Alternatively, some architectures focus on the reconfigurability of the memory array. Although efficient for functioning solely as NVM storage, performing LiM operations requires data to be stored in a complementary configuration, akin to other architectures. In contrast, the proposed IMC architecture enables LiM operations without the need for a complementary bit-cell configuration, significantly saving system area and reducing power consumption.

Fig. 2.7 illustrates the Current Reference for LiM Operation. Using different of VGSOT MRAM device in the SA thesr reference currents are generated.

2.4.3 Content Addressable Memory Operation

To elaborate further, let's delve into a specific scenario within a content-Addressable Memory (CAM) setup. Suppose we have a data stream stored within the memory cells, represented as 001 (as depicted in Fig. 2.8). Now, let's say we're searching for a particular pattern, denoted as the search data stream (SeD), which in this case is 101.

To determine if there's a match between the stored data and the search pattern, we utilize an

XOR operation. This operation compares each corresponding bit of the stored data stream and the search data stream. In our example, performing the XOR operation yields 101, indicating the bits where the stored and search data streams differ.

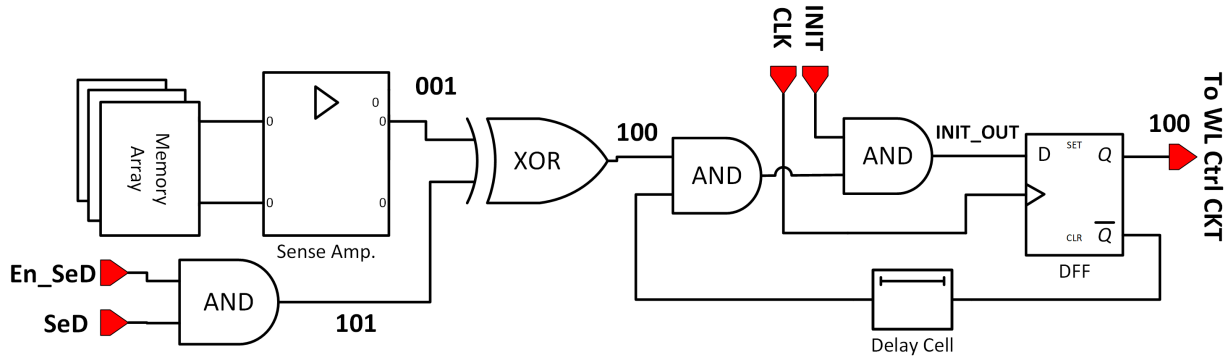


Figure 2.8: CAM Circuitry to Search Data.

However, in CAM systems, there are often situations where certain bits in the search pattern are not crucial for the matching process. In such cases, we employ a "don't care" condition, denoted by En_Sed being set to low. This condition enables the CAM circuitry to systematically compare against the stored data within the bit-cell. It ensures a continuous search operation, where VOUT is constrained by specific bits in the search pattern.

Now, regarding the power consumption aspect, it's essential to quantify the energy expended during these search operations. The power consumption attributed to conducting a bit search in each Content Addressable Memory (CAM) block is quantified at 4.17 μW. This metric gives us insight into the energy efficiency of the CAM system and its operational characteristics, aiding in optimizations and performance evaluations.

2.4.4 Binary Neural Network Operation for Image Classification

In our work, we present a lightweight hardware approach designed to facilitate Binary Neural Network (BNN) computations. Unlike traditional neural networks that employ floating-point

precision, BNNs utilize binary values (i.e., 0s and 1s) for both weights and activations. This binary representation offers several advantages, including low memory, super fast classification speeds, less energy utilization of the device, and improved abstraction or generalization capabilities [97, 98].

In the context of BNN operation, the process typically starts with the retrieval and binarization of input and weight data. These data undergo preprocessing steps, including normalization and quantization, to convert them into binary format. Subsequently, a software-level simulation is performed, where the network undergoes training through iterative epochs. In this simulation, a tanh activation function is applied to the network. This training phase aims to optimize the binary weights and thresholds to achieve desired performance metrics, such as accuracy and convergence speed.

$$H_1 = \sum_{i=1}^{n} (In_1.W_{11} + In_2.W_{12} + ... + In_n.W_{1n}) \tag{2.4}$$

$$H_m = \sum_{i=1}^{n} (In_n.W_{m1} + In_2.W_{m2} + ... + In_n.W_{mn}) \tag{2.5}$$

Once the training process is complete, the binary weights are mapped or stored within dedicated memory arrays, while the input data is prepared for inference. During inference, the input data is propagated through the network, and binary operations are performed at each layer to compute the output. This often involves binary convolutions, binary activations, and binary pooling operations, all of which are tailored to exploit the binary nature of the network's parameters.

In our hardware-based approach, we focus on optimizing the inference process by implementing efficient hardware architectures capable of performing binary operations in real-

time. Specifically, we leverage memory arrays and specialized circuitry to enable fast and energy-efficient computation of binary neural network models.

During the operation of our proposed hardware architecture, input data is fed into the system through Read Word Line (RWL) terminals. When the incoming input bit streams come as logic 1, the corresponding RWL signal activates, connecting the relevant bit-cell Magnetoresistive Tunnel Junctions (MTJs) to the SPCSA for XNOR operation. In contrast, when the input bit stream is 0, both the read-word line and an accompanying counter are deactivated, bypassing the counting. This dynamic behavior of the counter ensures efficient processing tailored to the specific input conditions, leading to optimized performance and energy consumption.

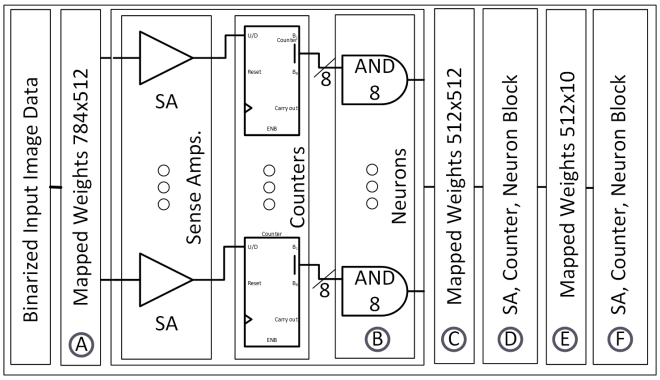


Figure 2.9: BNN Hardware Block Level Diagram.

Block-B, D, F are composed of the same set of functional components, namely Sense Amplifiers (SAs), Counters, and Neurons. As a neuron cell or block we used AND-8 (having 8 input) gate. This configuration allows for efficient processing within the neural network architecture.

During operation, when the counter count value fills its maximum counting capacity, the neurons contained within them produce spikes. These spikes then act as inputs for the subsequent layers, enabling passing of information across the network.

In addition, to ensure the proper functioning of the system in subsequent iterations, the counter undergoes a reset process. This reset mechanism is crucial for maintaining the integrity and accuracy of the computation performed by the neural network.

Overall, our hardware-based approach offers a promising avenue for accelerating BNN computations, paving the way for efficient deployment of binary neural network models in resource-constrained environments.

Chapter 3

Analysis of Power Consumption, Performance Metrics, Area Utilization, and Reliability Characteristics

Table 3.1 presents a comprehensive comparison of three MRAM-based IMC architectures: STT (Spin Transfer Torque), SOT (Spin-Orbit Torque), and VGSOT (gate voltage assisted Spin-Orbit Torque). In terms of both performance and power efficiency, our architecture surpasses other alternatives. Although the VGSOT device does have a longer write time, it ensures an extremely low switching error rate, nearly approaching zero. The proposed architecture utilizes bit-cell designed in a non-complementary (e.g. 2T2M for single bit data storage) configuration and utilizes MuBit registers, effectively reducing silicon area and lowering production costs.

In terms of architectural specifications, all three IMC architectures are digital. The bit-cell configurations differ, with STT adopting a 1T1R (one transistor-one resistor) structure, SOT using a 2T1R configuration, and the proposed VGSOT employing a 4T1M (four transistor-one resistor) design. The bit-cell area is significantly smaller in the proposed VGSOT architecture compared to the other two. The supply voltage is 1V for STT and SOT, while the VGSOT architecture operates at 0.8V.

Table 3.1: Evaluation with Latest IMC Architecture.

Architecture Details	**STT 28nm [99]**	**SOT 45nm [100]**	**VGSOT (Our Work)**
Architecture Type	Digital	Digital	Digital
Bit-Cell Type	1T1R	2T1R	4T1M
Measusred Bit-Cell Area (μm^2)	0.074	0.5589	0.195
System Operating Voltage (V)	1	1	0.8
Bit-Cell Delay (ns)	-	0.25, 0.15	0.3, 3
Bit-Cell Power (Write-Read) (μW)	-	19.73, 10.91	13.10, 1.65
Bit-Cell Fault Rate of Switching	-	-	≈ 0
Logic(NOR/OR) (μW)	-	79.43	55.2
Logic(AND/NAND) (μW)	-	76.93	35.30
Majority) (μW)	-	-	68.95
Tera-OPS/Watt	58.69	-	199.10
MNIST Inference (%)	92.12	-	97.40
FMNIST Inference (%)	-	-	84.15

Regarding read-write latency, the values for STT are not provided, while SOT achieves latencies of 0.25ns for read and 0.15ns for write operations. The VGSOT architecture demonstrates slightly higher latencies of 0.3ns for read and 3ns for write operations. In terms of read-write power, STT values are not available, while SOT consumes 10.91μW for read and 19.73μW for write operations. The VGSOT architecture exhibits lower power consumption, with 1.65μW for read and 13.10μW for write operations.

One crucial aspect is the switching error rate, which is approximately 0 for the VGSOT architecture, indicating a high level of reliability. Additionally, the power consumption of

various logic-in-memory (LiM) operations is provided. While the LiM power consumption for STT is not specified, SOT consumes 76.93μW for NAND/AND operations and 79.43μW for OR/NOR operations. The proposed VGSOT architecture achieves lower power consumption, with 35.30μW for NAND/AND operations and 55.2μW for OR/NOR operations.

The table also includes performance metrics such as TOPS/W (tera-operations per second per watt) and accuracy results for MNIST and FMNIST datasets. The TOPS/W values for STT and SOT are not available, while the proposed VGSOT architecture achieves a significantly higher value of 199.10 TOPS/W. In terms of accuracy, STT achieves 92.12% accuracy for the MNIST dataset, while SOT and VGSOT accuracy values are not provided for MNIST. However, for the FMNIST dataset, the proposed VGSOT architecture achieves an accuracy of 84.15

In summary, the comparison highlights the superior performance and power efficiency of the proposed VGSOT MRAM-based IMC architecture. It offers advantages such as reduced chip area, lower power consumption, and a high resistance to switching errors. These properties make it a promising candidate for next-generation computing systems. The bit-cell we've developed is only 34.53% of the size of a 2T1R cell. That means it's much smaller and takes up a lot less space. In fact, it's a reduction of about 65.74% in the area needed for the bit-cell. We did the calculations based on the equations we've shown below.

$$\begin{aligned}&\text{BC Area Percentage(\%) of 4T1M to 2T1R}\\&= \left(\frac{\text{4T1M Bit-cell area}}{\text{2T1R Bit-cell area}}\right) \times 100\%\\&= \left(\frac{0.195}{0.5589}\right) \times 100\% \approx 34.53\%\end{aligned}$$

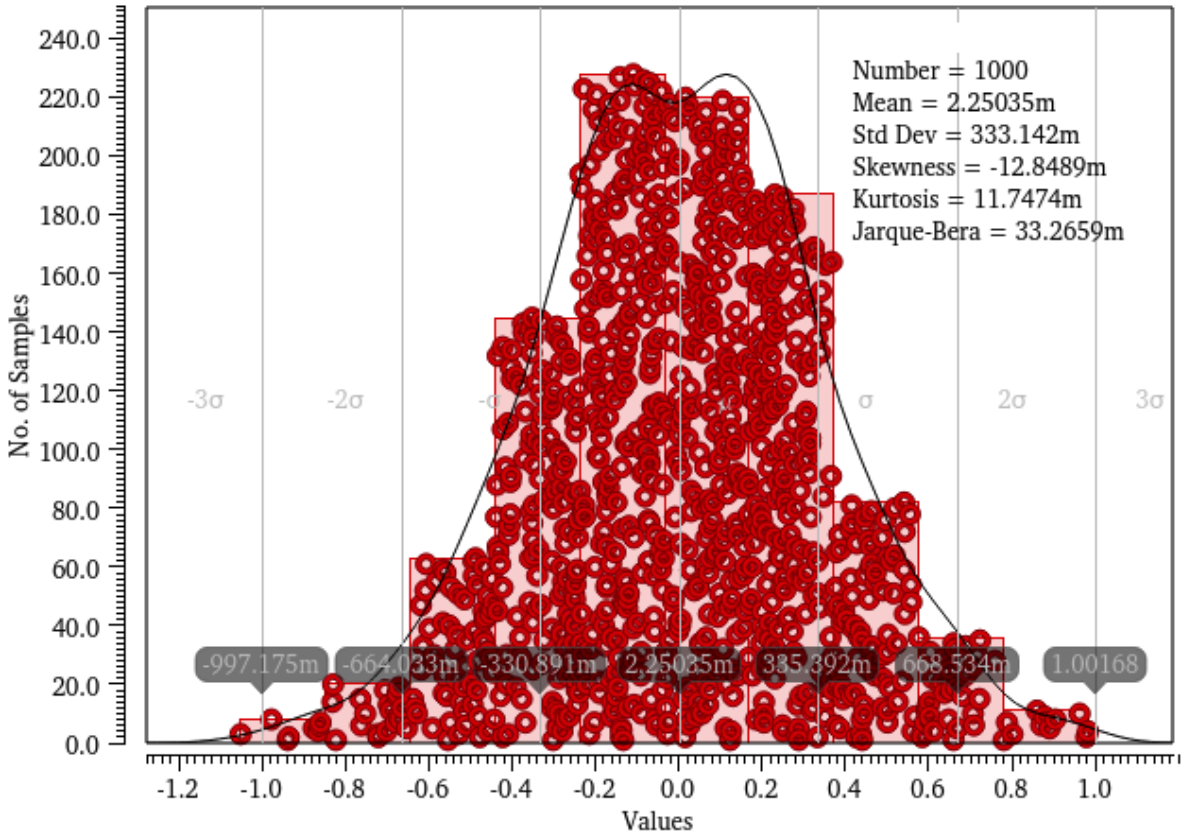


Figure 3.1: Monte Carlo Yield Simulation.

$$
\begin{aligned}
&\% \text{ Decrease of 4T1M BC} \\
&= \left(\frac{\text{2T1R BC Area} - \text{4T1M BC Area}}{\text{2T1R BC Area}} \times 100 \right) \% \\
&= \left(\frac{0.5589 - 0.195}{0.5589} \times 100 \right) \% = 65.47\%
\end{aligned}
$$

Resistance drift refers to the change in resistance values over time. In MRAM devices, resistance drift can occur due to various factors such as temperature variations, stress, and aging effects. Resistance drift highly affected by the device architecture, and materials used to construct. For example, a higher TMR ratio can be beneficial. This is because a larger resistance difference between the parallel and antiparallel states provides a larger margin for detecting and distinguishing the stored data. It helps to mitigate the impact of resistance variations caused by drift, resulting in improved reliability and data retention. The VGSOT MRAM device we are utilizing has a TMR ratio of 100%. This is one of the device level solution.

As we are working on circuit level, in our IMC architecture, we have implemented a solution to address the variability in device resistance so that we can maintain our system performance. To achieve this, we designed our sense amplifier blocks with the VGSOT device serving as a reference cell. By doing so, any resistance variability in the VGSOT device will impact both the memory macro and sense amplifier blocks. This approach effectively balances resistance drift between these components.

Chapter 4

Conclusions

In the pursuit of highly energy-efficient, and next-generation advance faster computing, the article introduces a novel IMC architecture that combines several key components. This architecture includes non-volatile memory (NVM) storage, LiM (Logic-in-Memory) operation, CAM (content-Addressable Memory) finding operation, and BNN (Binary Neural Network) for large data based AI applications. By integrating these elements, the architecture achieves significant operational power efficiency and silicon area savings, making a significant contribution to the field of computing.

The incorporation of NVM storage within the IMC architecture allows for persistent data storage, eliminating the need for constant power supply and reducing energy consumption. This feature is particularly beneficial for applications that require data retention even during power-off states. Furthermore, the LiM operation, which involves performing logic operations directly on memory, leverages the parallelism and proximity of data in memory chips, leading to enhanced performance and reduced data transfer overhead.

The CAM search operation plays a crucial role in applications that involve high-speed data

retrieval and matching, such as routing tables and database systems. By integrating CAM functionality within the IMC architecture, the system achieves efficient and fast search capabilities, contributing to overall performance improvements. Lastly, the utilization of BNN in big data AI applications enables efficient processing of binary neural networks, which are well-suited for certain AI tasks. This integration enhances the architecture's ability to handle complex AI workloads while minimizing computational requirements and energy consumption.

Overall, the introduced IMC architecture, incorporating NVM storage, LiM operation, CAM search operation, and BNN, represents a significant advancement in the field of energy-efficient and advanced computing. By leveraging these components, the architecture achieves notable power and chip area savings, paving the way for the development of more efficient and powerful next-generation computing systems.

Appendices

Appendix A

First Appendix

A.1 Section one

A.1.1 What is majority (MAJ) logic?

Majority logic refers to a decision-making process or a computational method in which an output or decision is determined based on the majority of individual inputs. In a binary system, majority logic involves comparing the number of "1" (true) and "0" (false) inputs and making a decision based on which value appears more frequently.

For example, in a three-input majority logic system, the output would be determined by the majority value among the three inputs. If at least two inputs are "1," the majority logic output would be "1." Similarly, if at least two inputs are "0," the majority logic output would be "0."

A.1.2 What is mult-bit flip-flop (MBFF)?

A multi-bit flip-flop, also known as a multi-bit register or multi-bit latch, is a sequential logic circuit capable of storing and manipulating multiple bits of digital information. Unlike a single-bit flip-flop that can store only a single binary value, a multi-bit flip-flop can store multiple bits simultaneously. It consists of multiple flip-flops interconnected to create a larger storage unit.

The most common type of multi-bit flip-flop is the D flip-flop, which stands for "data flip-flop." In a multi-bit D flip-flop, each bit has its own D input, clock input, and output. The D input determines the value to be stored, and the clock input controls when the data is transferred to the output. When the clock signal transitions from low to high (or high to low, depending on the specific flip-flop implementation), the values at the D inputs are latched and stored in the flip-flop. This allows for synchronous operation and ensures that all bits within the multi-bit flip-flop are updated simultaneously.

Multi-bit flip-flops are widely used in digital systems that require the storage and manipulation of multiple bits of information. They are essential building blocks in many applications, including microprocessors, memory units, and data storage systems. By using multi-bit flip-flops, designers can efficiently store and process data in parallel, allowing for faster and more complex operations. The size of a multi-bit flip-flop is typically specified by the number of bits it can store, such as 4-bit, 8-bit, or 16-bit flip-flops. These larger storage units provide the necessary capacity for handling multi-bit data and enable the creation of more sophisticated digital circuits.

A.1.3 What is clock-gating(CKG)?

Clock gating is a power-saving technique used in Very Large Scale Integration (VLSI) design to reduce dynamic power consumption in digital circuits. It involves selectively disabling or enabling clock signals to specific circuit elements or functional blocks based on their operational requirements. By gating the clock signal, unnecessary clock transitions and associated power consumption can be eliminated, resulting in significant power savings.

In a typical digital circuit, the clock signal is distributed to all the components, ensuring synchronized operation. However, not all components within the circuit require clock pulses at all times. Clock gating exploits this fact by dynamically controlling the clock signal to specific areas of the circuit that are inactive or idle.

Clock gating is typically implemented using clock gating cells or clock gating logic. These cells are inserted into the clock path of the circuit, allowing the clock signal to be enabled or disabled based on certain conditions. The gating conditions are usually derived from the circuit's control signals, activity level, or other relevant factors.

When a clock gating cell receives a signal indicating that the associated circuit block is inactive or not required to operate, it disables the clock signal, preventing unnecessary clock transitions and reducing power consumption. Conversely, when the circuit block becomes active, the clock gating cell enables the clock signal, allowing the block to operate normally.

The benefits of clock gating extend beyond power savings. By reducing unnecessary clock transitions, clock gating can also improve circuit performance by reducing overall power supply noise, minimizing clock skew, and mitigating potential timing issues.

Designers employ clock gating techniques at various levels of abstraction, from individual registers and flip-flops to larger functional units or even entire blocks. Automatic tools

and methodologies are available to assist in the identification and insertion of clock gating circuitry, optimizing power consumption without compromising functionality.

In summary, clock gating is a power-saving technique used in VLSI design to reduce dynamic power consumption by selectively disabling or enabling clock signals to specific circuit elements. By eliminating unnecessary clock transitions, clock gating reduces power consumption, improves circuit performance, and contributes to overall power efficiency in digital circuits.

A.2 Section two

Appendix B

Second Appendix

Bibliography

[1] , , and , "Survey on in-memory computing technology," *Journal of Software*, vol. 27, no. 8, pp. 2147–2167, 2016.

[2] G. Singh, L. Chelini, S. Corda, A. J. Awan, S. Stuijk, R. Jordans, H. Corporaal, and A.-J. Boonstra, "A review of near-memory computing architectures: Opportunities and challenges," in *2018 21st Euromicro Conference on Digital System Design (DSD)*. IEEE, 2018, pp. 608–617.

[3] S. Mittal, G. Verma, B. Kaushik, and F. A. Khanday, "A survey of sram-based in-memory computing techniques and applications," *Journal of Systems Architecture*, vol. 119, p. 102276, 2021.

[4] P. Siegl, R. Buchty, and M. Berekovic, "Data-centric computing frontiers: A survey on processing-in-memory," in *Proceedings of the Second International Symposium on Memory Systems*, 2016, pp. 295–308.

[5] H. Zhang, G. Chen, B. C. Ooi, K.-L. Tan, and M. Zhang, "In-memory big data management and processing: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1920–1948, 2015.

[6] S. Yin, Z. Jiang, J.-S. Seo, and M. Seok, "Xnor-sram: In-memory computing sram

macro for binary/ternary deep neural networks," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 6, pp. 1733–1743, 2020.

[7] S. Ghose, A. Boroumand, J. S. Kim, J. Gómez-Luna, and O. Mutlu, "Processing-in-memory: A workload-driven perspective," *IBM Journal of Research and Development*, vol. 63, no. 6, pp. 3–1, 2019.

[8] M. Gokhale, B. Holmes, and K. Iobst, "Processing in memory: The terasys massively parallel pim array," *Computer*, vol. 28, no. 4, pp. 23–31, 1995.

[9] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 27–39, 2016.

[10] J. Ahn, S. Hong, S. Yoo, O. Mutlu, and K. Choi, "A scalable processing-in-memory accelerator for parallel graph processing," in *Proceedings of the 42nd Annual International Symposium on Computer Architecture*, 2015, pp. 105–117.

[11] R. Gaurav, B. Tripp, and A. Narayan, "Spiking approximations of the maxpooling operation in deep snns," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[12] R. Gaurav, T. C. Stewart, and Y. C. Yi, "Spiking reservoir computing for temporal edge intelligence on loihi," in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*. IEEE, 2022, pp. 526–530.

[13] R. Gaurav, T. C. Stewart, and Y. Yi, "Reservoir based spiking models for univariate time series classification," *Frontiers in Computational Neuroscience*, vol. 17, p. 1148284, 2023.

[14] C. Lin, M. F. Azmine, Y. Liang, and Y. Yi, "Leveraging neuro-inspired ai accelerator for high-speed computing in 6g networks," *Frontiers in Computational Neuroscience*, vol. 18, p. 1345644, 2024.

[15] C. Lin, Y. Liang, and Y. Yi, "Fpga-based reservoir computing with optimized reservoir node architecture," in *2022 23rd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2022, pp. 1–6.

[16] C. Lin, M. F. Azmine, and Y. Yi, "Accelerating next-g wireless communications with fpga-based ai accelerators," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023, pp. 1–8.

[17] A. M. Asif Khan, T. Islam, M. W. Absar, M. R. Sarker, and M. S. Islam, "Design of a high voltage closed loop charge pumping system with a self-regulation mechanism," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, 2019, pp. 1–6.

[18] M. M. Abir Bappy, M. R. Sarkar, S. I. Hasan, M. M. Azmir, and D. M. Rashid, "Design process and performance analysis of two stage differential op-amp by varying the body biasing in fully depleted silicon on insulator technology," in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2021, pp. 0726–0730.

[19] I. Arikpo, F. Ogban, and I. Eteng, "Von neumann architecture and modern computers," *Global Journal of Mathematical Sciences*, vol. 6, no. 2, pp. 97–103, 2007.

[20] R. Eigenmann and D. J. Lilja, "Von neumann computers," *Wiley Encyclopedia of Electrical and Electronics Engineering*, vol. 23, pp. 387–400, 1998.

[21] M. Shaafiee, R. Logeswaran, and A. Seddon, "Overcoming the limitations of von neumann architecture in big data systems," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*. IEEE, 2017, pp. 199–203.

[22] S. Petrenko and S. Petrenko, "Limitations of von neumann architecture," *Big Data Technologies for Monitoring of Computer Security: A Case Study of the Russian Federation*, pp. 115–173, 2018.

[23] E. Peláez, "Parallelism and the crisis of von neumann computing," *Technology in Society*, vol. 12, no. 1, pp. 65–77, 1990.

[24] S. Khoram, Y. Zha, J. Zhang, and J. Li, "Challenges and opportunities: From near-memory computing to in-memory computing," in *Proceedings of the 2017 ACM on International Symposium on Physical Design*, 2017, pp. 43–46.

[25] G. Singh, L. Chelini, S. Corda, A. J. Awan, S. Stuijk, R. Jordans, H. Corporaal, and A.-J. Boonstra, "Near-memory computing: Past, present, and future," *Microprocessors and Microsystems*, vol. 71, p. 102868, 2019.

[26] D. Fujiki, X. Wang, A. Subramaniyan, and R. Das, *In-/near-memory Computing*. Springer, 2021.

[27] V. Iskandar, M. A. A. E. Ghany, and D. Goehringer, "Near-memory computing on fpgas with 3d-stacked memories: applications, architectures, and optimizations," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 16, no. 1, pp. 1–32, 2022.

[28] Y. Li, T. Bai, X. Xu, Y. Zhang, B. Wu, H. Cai, B. Pan, and W. Zhao, "A survey of mram-centric computing: From near memory to in memory," *IEEE Transactions on Emerging Topics in Computing*, 2022.

[29] K. S. Mohamed and K. S. Mohamed, "Near-memory/in-memory computing: pillars and ladders," *Neuromorphic Computing and Beyond: Parallel, Approximation, Near Memory, and Quantum*, pp. 167–186, 2020.

[30] M. Di Ventra, *MemComputing: fundamentals and applications*. Oxford University Press, 2022.

[31] R. Nair, "Evolution of memory architecture," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1331–1345, 2015.

[32] W. Xiao and Y. Shi, "A survey for realizing in-memory computing," in *2022 14th International Conference on Computer Research and Development (ICCRD)*, 2022, pp. 345–348.

[33] C.-J. Jhang, C.-X. Xue, J.-M. Hung, F.-C. Chang, and M.-F. Chang, "Challenges and trends of sram-based computing-in-memory for ai edge devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 5, pp. 1773–1786, 2021.

[34] M. Shaafiee, R. Logeswaran, and A. Seddon, "Overcoming the limitations of von neumann architecture in big data systems," pp. 199–203, 2017. [Online]. Available: http://dx.doi.org/10.1109/CONFLUENCE.2017.7943149

[35] M. R. Sarkar, M. M. A. Bappy, M. M. Azmir, D. M. Rashid, and S. I. Hasan, "Vg-sot mram design and performance analysis," in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2021, pp. 0715–0719.

[36] T. Endoh, H. Honjo, K. Nishioka, and S. Ikeda, "Recent progresses in stt-mram and sot-mram for next generation mram," in *2020 IEEE Symposium on VLSI Technology*, 2020, pp. 1–2.

[37] P. Jangra and M. Duhan, "A review on emerging spintronic devices: Cmos counterparts," in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 2022, pp. 90–99.

[38] M. R. Sarkar and C. Y. Yi, "An in-memory computing architecture utilizing energy-efficient vgsot mram device," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2024.

[39] F. Nowshin, Y. Huang, M. R. Sarkar, Q. Xia, and Y. Yi, "Merrc: A memristor-enabled reconfigurable low-power reservoir computing architecture at the edge," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 1, pp. 174–186, 2024.

[40] K. Garello, F. Yasin, H. Hody, S. Couet, L. Souriau, S. H. Sharifi, J. Swerts, R. Carpenter, S. Rao, W. Kim, J. Wu, K. Sethu, M. Pak, N. Jossart, D. Crotti, A. Furnémont, and G. S. Kar, "Manufacturable 300mm platform solution for field-free switching sot-mram," in *2019 Symposium on VLSI Technology*, 2019, pp. T194–T195.

[41] P. Barla, V. K. Joshi, and S. Bhat, "Spintronic devices: a promising alternative to cmos devices," in *Journal of Computational Electronics*, vol. 20, no. 2, 2021, pp. 805–837. [Online]. Available: https://doi.org/10.1007/s10825-020-01648-6

[42] M. E. Pereira, R. F. de Piva Martins, E. Fortunato, P. Barquinha, and A. Kiazadeh, "Recent progress in optoelectronic memristors for neuromorphic and in-memory computation," *Neuromorphic Computing and Engineering*, 2023.

[43] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, and G. W. Burr, "Recent progress in analog memory-based accelerators for deep learning," *Journal of Physics D: Applied Physics*, vol. 51, no. 28, p. 283001, 2018.

[44] C. Wang, G. Shi, F. Qiao, R. Lin, S. Wu, and Z. Hu, "Research progress in architecture and application of rram with computing-in-memory," *Nanoscale Advances*, vol. 5, no. 6, pp. 1559–1573, 2023.

[45] S. Yu and P.-Y. Chen, "Emerging memory technologies: Recent trends and prospects," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 2, pp. 43–56, 2016.

[46] A. Gebregiorgis, H. A. Du Nguyen, J. Yu, R. Bishnoi, M. Taouil, F. Catthoor, and S. Hamdioui, "A survey on memory-centric computer architectures," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 18, no. 4, pp. 1–50, 2022.

[47] S. Yu, H. Jiang, S. Huang, X. Peng, and A. Lu, "Compute-in-memory chips for deep learning: Recent trends and prospects," *IEEE Circuits and Systems Magazine*, vol. 21, no. 3, pp. 31–56, 2021.

[48] R. Rizk, D. Rizk, A. Kumar, and M. Bayoumi, "Demystifying emerging nonvolatile memory technologies: understanding advantages, challenges, trends, and novel applications," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[49] W. Tang, J. Liu, H. Li, D. Chen, C. Jiang, X. Li, and H. Yang, "Computing-in-memory with thin-filmtransistors: challenges and opportunities," *Flexible and Printed Electronics*, vol. 7, no. 2, p. 024001, 2022.

[50] S. Channamadhavuni, S. Thijssen, S. K. Jha, and R. Ewetz, "Accelerating ai applications using analog in-memory computing: Challenges and opportunities," in *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, 2021, pp. 379–384.

[51] J.-M. Hung, C.-J. Jhang, P.-C. Wu, Y.-C. Chiu, and M.-F. Chang, "Challenges and

trends of nonvolatile in-memory-computation circuits for ai edge devices," *IEEE Open Journal of the Solid-State Circuits Society*, vol. 1, pp. 171–183, 2021.

[52] S.-T. Wei, B. Gao, D. Wu, J.-S. Tang, H. Qian, and H.-Q. Wu, "Trends and challenges in the circuit and macro of rram-based computing-in-memory systems," *Chip*, vol. 1, no. 1, p. 100004, 2022.

[53] H. Jia, M. Ozatay, Y. Tang, H. Valavi, R. Pathak, J. Lee, and N. Verma, "15.1 a programmable neural-network inference accelerator based on scalable in-memory computing," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 236–238.

[54] S. Yin, Z. Jiang, M. Kim, T. Gupta, M. Seok, and J.-S. Seo, "Vesti: Energy-efficient in-memory computing accelerator for deep neural networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 1, pp. 48–61, 2019.

[55] S. Angizi, Z. He, D. Reis, X. S. Hu, W. Tsai, S. J. Lin, and D. Fan, "Accelerating deep neural networks in processing-in-memory platforms: Analog or digital approach?" in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2019, pp. 197–202.

[56] S. Gupta, M. Imani, H. Kaur, and T. S. Rosing, "Nnpim: A processing in-memory architecture for neural network acceleration," *IEEE Transactions on Computers*, vol. 68, no. 9, pp. 1325–1337, 2019.

[57] M. Imani, M. Samragh, Y. Kim, S. Gupta, F. Koushanfar, and T. Rosing, "Rapidnn: In-memory deep neural network acceleration framework," *arXiv preprint arXiv:1806.05794*, 2018.

[58] S. Woźniak, A. Pantazi, T. Bohnstingl, and E. Eleftheriou, "Deep learning incorporating biologically inspired neural dynamics and in-memory computing," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 325–336, 2020.

[59] M. Imani, S. Gupta, Y. Kim, and T. Rosing, "Floatpim: In-memory acceleration of deep neural network training with high precision," in *Proceedings of the 46th International Symposium on Computer Architecture*, 2019, pp. 802–815.

[60] T.-N. Pham, Q.-K. Trinh, I.-J. Chang, and M. Alioto, "Stt-bnn: A novel stt-mram in-memory computing macro for binary neural networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 12, no. 2, pp. 569–579, 2022.

[61] F. Schuiki, M. Schaffner, F. K. Gürkaynak, and L. Benini, "A scalable near-memory architecture for training deep neural networks on large in-memory datasets," *IEEE Transactions on Computers*, vol. 68, no. 4, pp. 484–497, 2018.

[62] A. Chen, "A review of emerging non-volatile memory (nvm) technologies and applications," *Solid-State Electronics*, vol. 125, pp. 25–38, 2016.

[63] P. Mannocci, M. Farronato, N. Lepri, L. Cattaneo, A. Glukhov, Z. Sun, and D. Ielmini, "In-memory computing with emerging memory devices: Status and outlook," *APL Machine Learning*, vol. 1, no. 1, 2023.

[64] D. S. Jeong, R. Thomas, R. Katiyar, J. Scott, H. Kohlstedt, A. Petraru, and C. S. Hwang, "Emerging memories: resistive switching mechanisms and current status," *Reports on progress in physics*, vol. 75, no. 7, p. 076502, 2012.

[65] J. Boukhobza, S. Rubini, R. Chen, and Z. Shao, "Emerging nvm: A survey on architectural integration and research challenges," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 23, no. 2, pp. 1–32, 2017.

[66] "Opportunities and challenges for spintronics in the microelectronics industry," vol. 3, no. 8, pp. 446–459, 2020.

[67] H. Kallinatha, S. Rai, and B. Talawar, "A detailed study of sot-mram as an alternative to dram primary memory in multi-core environment," *IEEE Access*, 2024.

[68] L. Wu, M. Taouil, S. Rao, E. J. Marinissen, and S. Hamdioui, "Survey on stt-mram testing: Failure mechanisms, fault models, and tests," *arXiv preprint arXiv:2001.05463*, 2020.

[69] Z. Guo, J. Yin, Y. Bai, D. Zhu, K. Shi, G. Wang, K. Cao, and W. Zhao, "Spintronics for energy-efficient computing: An overview and outlook," *Proceedings of the IEEE*, vol. 109, no. 8, pp. 1398–1417, 2021.

[70] Z. He, Y. Zhang, S. Angizi, B. Gong, and D. Fan, "Exploring a sot-mram based in-memory computing for data processing," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 676–685, 2018.

[71] H. Koike, T. Tanigawa, T. Watanabe, T. Nasuno, Y. Noguchi, M. Yasuhira, T. Yoshiduka, M. Yitao, H. Honjo, K. Nishioka *et al.*, "Review of stt-mram circuit design strategies, and a 40-nm 1t-1mtj 128mb stt-mram design practice," in *2020 IEEE 31st Magnetic Recording Conference (TMRC)*. IEEE, 2020, pp. 1–2.

[72] D. Apalkov, A. Khvalkovskiy, S. Watts, V. Nikitin, X. Tang, D. Lottis, K. Moon, X. Luo, E. Chen, A. Ong *et al.*, "Spin-transfer torque magnetic random access memory (stt-mram)," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 9, no. 2, pp. 1–35, 2013.

[73] Y. Huai *et al.*, "Spin-transfer torque mram (stt-mram): Challenges and prospects," *AAPPS bulletin*, vol. 18, no. 6, pp. 33–40, 2008.

[74] Y. Chen, "Reram: History, status, and future," *IEEE Transactions on Electron Devices*, vol. 67, no. 4, pp. 1420–1433, 2020.

[75] H. Akinaga and H. Shima, "Reram technology; challenges and prospects," *IEICE Electronics Express*, vol. 9, no. 8, pp. 795–807, 2012.

[76] G. Indiveri, E. Linn, and S. Ambrogio, "Reram-based neuromorphic computing," *Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications*, pp. 715–736, 2016.

[77] W. Chen, Z. Qi, Z. Akhtar, and K. Siddique, "Resistive-ram-based in-memory computing for neural network: A review," *Electronics*, vol. 11, no. 22, p. 3667, 2022.

[78] S. Mittal, "A survey of reram-based architectures for processing-in-memory and neural networks," *Machine learning and knowledge extraction*, vol. 1, no. 1, pp. 75–114, 2018.

[79] S. Sahoo and S. Prabaharan, "Nano-ionic solid state resistive memories (re-ram): A review," *Journal of nanoscience and nanotechnology*, vol. 17, no. 1, pp. 72–86, 2017.

[80] M. Shen, T. Lill, N. Altieri, J. Hoang, S. Chiou, J. Sims, A. McKerrow, R. Dylewicz, E. Chen, H. Razavi *et al.*, "Review on recent progress in patterning phase change materials," *Journal of Vacuum Science & Technology A*, vol. 38, no. 6, 2020.

[81] Q. Wang, G. Niu, W. Ren, R. Wang, X. Chen, X. Li, Z.-G. Ye, Y.-H. Xie, S. Song, and Z. Song, "Phase change random access memory for neuro-inspired computing," *Advanced Electronic Materials*, vol. 7, no. 6, p. 2001241, 2021.

[82] Y. Xie and J. Zhao, "Emerging memory technologies." *IEEE Micro*, vol. 39, no. 1, pp. 6–7, 2019.

[83] S. R. Elliott, "Chalcogenide phase-change materials: Past and future," *International Journal of Applied Glass Science*, vol. 6, no. 1, pp. 15–18, 2015.

[84] G. W. Burr, M. J. Brightsky, A. Sebastian, H.-Y. Cheng, J.-Y. Wu, S. Kim, N. E. Sosa, N. Papandreou, H.-L. Lung, H. Pozidis *et al.*, "Recent progress in phase-change memory technology," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 2, pp. 146–162, 2016.

[85] P. Noé, C. Vallée, F. Hippert, F. Fillot, and J.-Y. Raty, "Phase-change materials for non-volatile memory devices: from technological challenges to materials science issues," *Semiconductor Science and Technology*, vol. 33, no. 1, p. 013002, 2017.

[86] S. G. Sarwat, "Materials science and engineering of phase change random access memory," *Materials science and technology*, vol. 33, no. 16, pp. 1890–1906, 2017.

[87] T. Eshita, T. Tamura, and Y. Arimoto, "Ferroelectric random access memory (fram) devices," in *Advances in non-volatile memory and storage technology*. Elsevier, 2014, pp. 434–454.

[88] S. Kawashima and J. S. Cross, "Feram," in *Embedded Memories for Nano-Scale VLSIs*. Springer, 2009, pp. 279–328.

[89] D. Takashima, "Overview and trend of chain feram architecture," *IEICE transactions on electronics*, vol. 84, no. 6, pp. 747–756, 2001.

[90] G. R. Fox, R. Bailey, W. B. Kraus, F. Chu, S. Sun, and T. Davenport, "The current status of feram," in *Ferroelectric Random Access Memories: Fundamentals and Applications*. Springer, 2004, pp. 139–148.

[91] H. Ishiwara, "Ferroelectric random access memories," *Journal of nanoscience and nanotechnology*, vol. 12, no. 10, pp. 7619–7627, 2012.

[92] M. Suzuki, "Review on future ferroelectric nonvolatile memory: Feram from the point

of view of epitaxial oxide thin films," *Journal of the Ceramic Society of Japan*, vol. 103, no. 1203, pp. 1099–1111, 1995.

[93] K. Zhang, D. Zhang, C. Wang, L. Zeng, Y. Wang, and W. Zhao, "Compact modeling and analysis of voltage-gated spin-orbit torque magnetic tunnel junction," *IEEE Access*, vol. 8, pp. 50 792–50 800, 2020.

[94] S. Alla, V. K. Joshi, and S. Bhat, "Voltage-gated spin-orbit torque magnetic tunnel junction model analysis," in *2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 2022, pp. 96–101.

[95] S. Z. Peng, J. Q. Lu, W. X. Li, L. Z. Wang, H. Zhang, X. Li, K. L. Wang, and W. S. Zhao, "Field-free switching of perpendicular magnetization through voltage-gated spin-orbit torque," in *2019 IEEE International Electron Devices Meeting (IEDM)*, 2019, pp. 28.6.1–28.6.4.

[96] W. Li, S. Peng, J. Lu, H. Wu, X. Li, D. Xiong, Y. Zhang, Y. Zhang, K. L. Wang, and W. Zhao, "Experimental demonstration of voltage-gated spin-orbit torque switching in an antiferromagnet/ferromagnet structure," *Phys. Rev. B*, vol. 103, p. 094436, Mar 2021.

[97] R. Sayed, H. Azmi, H. Shawkey, A. H. Khalil, and M. Refky, "A systematic literature review on binary neural networks," *IEEE Access*, vol. 11, pp. 27 546–27 578, 2023.

[98] W. Zhao, T. Ma, X. Gong, B. Zhang, and D. Doermann, "A review of recent advances of binary neural networks for edge computing," *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 2, no. 1, pp. 25–35, 2021.

[99] T. Na, "Ternary output binary neural network with zero-skipping for mram-based

digital in-memory computing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 7, pp. 2655–2659, 2023.

[100] B. Wu, H. Zhu, K. Chen, C. Yan, and W. Liu, "Mlim: High-performance magnetic logic in-memory scheme with unipolar switching sot-mram," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 6, pp. 2412–2424, 2023.