Artificial Neural Network (ANN) design using Compute-in-Memory

Tatiana Moposita

Sorbonne Université

Università della Calabria

Doctoral School
LISITE-Institut supérieur d'électronique de Paris (ISEP)

Artificial Neural Network (ANN) design using Compute-in-Memory

By

Tatiana Moposita

Headed by Lionel Trojman, Felice Crupi

Presented on December, 2023

ARTIFICIAL NEURAL NETWORK (ANN) DESIGN USING COMPUTE-IN-MEMORY

Thesis committee:

Prof. Lionel Trojman
Institut supérieur d'électronique de Paris (ISEP)
Director

Prof. Andrei Vladimirescu
University of California at Berkeley/Delft University of Technology
Co-Director

Prof. Felice Crupi
Università della Calabria
Director

Prof. Marco Lanuzza
Università della Calabria
Co-Director

Prof. Alix Munier
Sorbonne University
President of the Jury

Prof. Dimitri Galyko
Sorbonne University
Jury

Prof. Gilles Jacquemond
Université Côte d'Azur
Rapporteur

Dr. Fernando Redondo
Interuniversity Microelectronics Centre (IMEC)
Rapporteur

Date approved: December, 2023

ACKNOWLEDGMENTS

First and foremost, I would like to extend my deepest gratitude to my family for their unwavering support throughout this arduous journey. In particular, I am profoundly thankful to my mother, whose constant presence and unconditional love have been a pillar of strength for me.

I would like to thank the members of my thesis committee for their invaluable assistance in preparing this work. Prof. Lionel Trojman, Prof. Andrei Vladimirescu, Prof. Felice Crupi, and Prof. Marco Lanuzza, I am truly thankful for your patience and for the constructive feedback you provided throughout these years of study.

I would like to extend special thanks to my friends and colleagues to support me day by day. Your encouragement and presence have meant a great deal to me.

I am grateful to the members of the research group, particularly Raffaele De Rose and Esteban Garzón, for generously sharing their knowledge and prior works. Their contributions have been instrumental in refining and advancing the ideas presented in this work.

TABLE OF CONTENTS

Abstract

Nowadays, the era of ”More than Moore” has arisen as a significant influence in light of the limitations anticipated by Moore's law. The computing systems are exploring alternative technologies to sustain and enhance performance improvements. The idea of alternative innovative technologies has emerged in solving challenges to overcome the development of electronic systems inspired by biological neural networks, commonly referred to as Artificial Neural Networks (ANN).

An ANN is based on interconnected units (nodes or neurons) which contains the synapses in order to transmit and process the information to other neurons. Accordingly ANN are able to determine patters and make predictions base on the input data. Therefore, ANNs are considered computational models designed to process information, recognize patterns, and make decisions with the goal of replicate the human brain functionality. However, as ANNs become larger and more complex, processing them with traditional Von Neumann computing systems is limited by the need to shuttle massive amounts of data between processing and memory units resulting in significant cost in latency and power consumption. Hence, hardware solutions have been developed to mitigate this bottleneck and improve the efficiency of ANNs. Different design solutions can be utilized to build these hardware ANNs, ranging from traditional digital circuits to the development of new approaches such as Compute-in-Memory (CIM) and Neuro-Inspired neuromorphic computing , which seek to improve the speed and efficiency of ANN computation.

In this context, memory technologies and their integration with computing elements in hardware implementations of ANNs have recently achieved significant

impact. Therefore, the use of emerging non-volatile memory (eNVM) technologies (i.e., resistive memory, magnetic memory, and memristors) are being explored as promising alternatives. These technologies offer several advantages over traditional CMOS technology, such as increased speed, higher densities, and lower power consumption. As a result, CIM employs eNVMs to perform computation within the memory itself, hence increasing memory capacity and processing speed.

The objective of this thesis focuses on the research of Artificial Neural Networks design using Compute in Memory, by employing efficient hardware solutions for ANNs at both circuit- and architecture-level. Recent research work in this context has proposed very efficient circuit designs to optimize the enormous computational needs required by data processing by ANNs.

To explore the capabilities of an ANN at the output node, the design of activation function (AF) are proposed. An AF is utilized within ANN to introduce nonlinearity into the output of a neuron. The selection of an AF is significant as it determines the power and capabilities of the neural network. The accuracy of predictions is primarily dependent on this choice. To assess the effectiveness of an activation function designed for analog implementation, the sigmoid and the softmax activation function are proposed.

Besides, this project explores the integration of emerging memory devices like Spin-Transfer-Torque Magnetic Random Access Memory (STT-MRAM) with CMOS technology. This combined approach aims to leverage the intrinsic capability of in-memory computing offered by these devices. STT-MRAMs based on state-of-the-art perpendicular magnetic tunneling junction (MTJ) and FinFETs has been considered for this study.

Single-barrier magnetic tunnel junction (SMTJ) and double-barrier magnetic tunnel junction (DMTJ) devices are considered to evaluate the impact of STT-MRAM cell based on DMTJ against the conventional SMTJ counterpart on the

performance of a two-layer multilayer perceptron (MLP) neural network. The considered MLP is a fully connected neural network where the standard MNIST benchmark handwritten dataset is used. The assessment is carried out through a customized simulation framework from device and bitcell levels to memory architecture and algorithm levels. The SMTJ- and DMTJ-based 2-layer MLP neural network performance is evaluated in terms of learning accuracy versus latency and energy consumption, calculated at the run-time.

Moreover, to improve the energy-efficiency of a Logic-in-Memory (LIM) architecture based on STT-MTJ devices, a new architecture (SIMPLY+) from the Smart Material Implication (SIMPLY) logic and perpendicular MTJ based STT-MRAM technologies was developed. The SIMPLY+ scheme is a promising solution for the development of energy-efficient and reliable in-memory computing architectures. The proposed architecture is benchmarked against its conventional counterpart. Overall, the results prove that the SIMPLY+ scheme is an outstanding solution for the development of energy-efficient and reliable in-memory computing architectures.

All circuit solutions are evaluated using commercial circuit simulators (e.g. Cadence Virtuoso). Circuit design activity involving emerging memory devices also required the use and calibration of Verilog-A based compact models to integrate the behavior of such devices into the circuit design tool. The solutions presented in this thesis involve techniques that offer significant advancements for future applications. From a design perspective, the integration of logic modules with STT-MRAM memory is highly feasible due to the seamless compatibility between STT-MRAMs and CMOS circuits. This approach not only proves advantageous for standard CMOS technology but also leverages the potential of emerging technologies.

Résumé

De nos jours, l'ère du "More than Moore" se profile comme une influence significative à la lumière des limitations anticipées par la loi de Moore. Les systèmes informatiques explorent des technologies alternatives pour maintenir et améliorer les performances. L'idée de technologies innovantes alternatives a émergé pour résoudre les défis liés au développement de systèmes électroniques inspirés par les réseaux neuronaux biologiques, communément appelés Réseau Neurones Artificiels (ANN).

Un ANN est basé sur des unités interconnectées (noeuds ou neurones) qui contiennent les synapses afin de transmettre et traiter l'information vers d'autres neurones. Ainsi, les ANN sont capables de déterminer des modèles et de faire des prédictions basées sur les données en entrée. Par conséquent, les ANN sont considérés comme des modèles computationnels conçus pour traiter l'information, reconnaître des modèles et prendre des décisions dans le but de reproduire la fonctionnalité du cerveau humain. Cependant, à mesure que les ANN deviennent plus grands et plus complexes, les traiter avec des systèmes informatiques traditionnels de type Von Neumann est limité par la nécessité de faire circuler d'énormes quantités de données entre les unités de traitement et de mémoire, entraînant des coûts importants en termes de latence et de consommation d'énergie. Par conséquent, des solutions matérielles ont été développées pour atténuer ce goulot d'étranglement et améliorer l'efficacité des ANN. Différentes solutions de conception peuvent être utilisées pour construire ces ANN matériels, allant des

circuits numériques traditionnels au développement de nouvelles approches telles que le Compute-in-Memory (CIM) et Neuro-Inspired neuromorphic computing, qui visent à améliorer la vitesse et l'efficacité du calcul des ANN.

Dans ce contexte, les technologies de mémoire et leur intégration avec les éléments de calcul dans les implémentations matérielles des ANN ont récemment eu un impact significatif. Ainsi, l'utilisation de technologies émergentes de mémoire non volatile (comme les mémoires résistives, magnétiques et les memristors) est explorée comme des alternatives prometteuses. Ces technologies offrent plusieurs avantages par rapport à la technologie CMOS traditionnelle, tels qu'une vitesse accrue, des densités plus élevées et une consommation d'énergie moindre. Par conséquent, le CIM utilise des eNVM pour effectuer le calcul directement dans la mémoire, augmentant ainsi la capacité de la mémoire et la vitesse de traitement.

L'objectif de cette thèse se concentre sur la recherche de la conception de Réseau Neurones Artificiels en utilisant le Compute-in-Memory, en employant des solutions matérielles efficaces pour les ANN à la fois au niveau du circuit et de l'architecture. Des travaux de recherche récents dans ce contexte ont proposé des conceptions de circuits très efficaces pour optimiser les besoins de calcul énormes requis pour le traitement des données par les ANN.

Pour explorer les capacités d'un ANN au niveau de la sortie, la conception de la fonction d'activation (AF) est proposée. Une AF est utilisée dans les ANN pour introduire de la non-linéarité dans la sortie d'un neurone. Le choix d'une AF est significatif car il détermine la puissance et les capacités du réseau neuronal.

L'exactitude des prédictions dépend principalement de ce choix. Pour évaluer l'efficacité d'une fonction d'activation conçue pour une implémentation analogique, les fonctions d'activation sigmoïde et softmax sont proposées.

En outre, ce projet explore l'intégration de dispositifs mémoires émergents comme la Spin-Transfer-Torque Magnetic Random Access Memory (STT-MRAM) avec la technologie CMOS. Cette approche combinée vise à exploiter la capacité intrinsèque de calcul en mémoire offerte par ces dispositifs. Les STT-MRAM basées sur la technologie de pointe des perpendicular magnetic tunneling junction (MTJ) et les transistors FinFET ont été pris en considération pour cette étude.

Single-barrier magnetic tunnel junction (SMTJ) et double-barrier magnetic tunnel junction (DMTJ) sont considérés pour évaluer l'impact des cellules STT-MRAM basées sur DMTJ par rapport à leur homologue conventionnel SMTJ sur les performances d'un réseau neuronal à multilayer perceptron (MLP) à deux couches. Le MLP considéré est un réseau neuronal entièrement connecté où l'ensemble de données standard MNIST de chiffres manuscrits est utilisé. L'évaluation est réalisée grâce à un cadre de simulation personnalisé, de niveaux de dispositif et de cellule binaire à niveaux d'architecture de mémoire et d'algorithme. Les performances des réseau neurones MLP à 2 couches basés sur SMTJ et DMTJ sont évaluées en termes de précision d'apprentissage par rapport à la latence et à la consommation d'énergie.

De plus, pour améliorer l'efficacité énergétique d'une architecture Logic-in-Memory (LIM) basée sur les dispositifs STT-MTJ, une nouvelle

architecture (SIMPLY+) issue de la logique Smart Material Implication (SIMPLY) et des technologies STT-MRAM basées sur les MTJ perpendiculaires a été développée. Le schéma SIMPLY+ représente une solution prometteuse pour le développement d'architectures de calcul en mémoire économes en énergie et fiables. L'architecture proposée est comparée à sa contrepartie conventionnelle.

Dans l'ensemble, les résultats prouvent que le schéma SIMPLY+ constitue une solution exceptionnelle pour le développement d'architectures de calcul en mémoire économes en énergie et fiables.

Toutes les solutions de circuit sont évaluées à l'aide de simulateurs de circuits commerciaux (par exemple, Cadence Virtuoso). L'activité de conception de circuits impliquant des dispositifs mémoires émergents a également nécessité l'utilisation et l'étalonnage de modèles compacts basés sur Verilog-A pour intégrer le comportement de ces dispositifs dans l'outil de conception de circuits. Les solutions présentées dans cette thèse impliquent des techniques qui offrent des avancées significatives pour les applications futures. D'un point de vue de la conception, l'intégration de modules logiques avec la mémoire STT-MRAM est parfaitement réalisable en raison de la compatibilité sans faille entre les STT-MRAM et les circuits CMOS. Cette approche est non seulement avantageuse pour la technologie CMOS standard, mais elle exploite également le potentiel des technologies émergentes.

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AF activation function

ANN Artificial Neural Networks

AP anti-parallel

BNN Binarized Neural Network

DL deep learning

DMTJ double-barrier magnetic tunnel junction

DNN deep neural networks

eNVM emerging non-volatile memory

FeFET ferroelectric field-effect transistor

FL free layer

i-MTJ in-plane MTJ

LIM Logic-in-Memory

MAC multiply-and-accumulate

MLP multilayer perceptron

MRAM magnetoresistive random access memory

MTJ magnetic tunneling junction

P parallel

p-MTJ perpendicular plane MTJ

PCM phase change memory

RDR read disturbance rate

RL reference layer

RRAM resistive random access memory

SIMPLY Smart Material Implication

SMTJ Single-barrier magnetic tunnel junction

STT spin-transfer torque

STT-MRAM Spin-Transfer-Torque Magnetic Random Access Memory

TMR tunneling magnetoresistance ratio

VMM vector-matrix multiplier

Publication List

Journal publications

1. M. Vatalaro, **T. Moposita**, S. Strangio, L. Trojman, A. Vladimirescu, M. Lanuzza, and F. Crupi, "A low-voltage, low-power reconfigurable current-mode softmax circuit for analog neural networks," *Electronics*, vol. 10, no. 9, p. 1004, 2021

2. **T. Moposita**, E. Garzón, F. Crupi, L. Trojman, A. Vladimirescu, and M. Lanuzza, "Efficiency of Double-Barrier Magnetic Tunnel Junction-Based Digital eNVM Array for Neuro-Inspired Computing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 3, pp. 1254–1258, 2023

Conference publications

1. **T. Moposita**, L. Trojman, F. Crupi, M. Lanuzza, and A. Vladimirescu, "Voltage-to-voltage sigmoid neuron activation function design for artificial neural networks," in *2022 IEEE 13th Latin America Symposium on Circuits and System (LASCAS)*, 2022, pp. 1–4

2. **T. Moposita**, E. Garzón, F. Crupi, L. Trojman, A. Vladimirescu, and M. Lanuzza, "Efficiency of Double-Barrier Magnetic Tunnel Junction-Based Digital eNVM Array for Neuro-Inspired Computing," in *2023 IEEE 14th Latin America Symposium on Circuits and System (LASCAS)*, IEEE, 2023 *(*Best Student Paper Award)*

3. E. Garzón, B. Zambrano, **T. Moposita**, R. Taco, L.M. Prócel, and L. Trojman, "Reconfigurable CMOS/STT-MTJ non-volatile circuit for

logic-in-memory applications," in *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, 2020, pp. 1–4

Honor & Awards

- *Best Student Paper Award* in IEEE 14th Latin America Symposium on Circuits and System (LASCAS 2023) Quito-Ecuador

CHAPTER 1

INTRODUCTION

Today's computing architectures and device technologies face challenges in meeting the growing demands on low-power capabilities and high-performance. Therefore, alternative architectures leverages the novel post-CMOS device technologies to provide a promising solution to overcome these limitations. The eNVM technologies such as magnetoresistive random access memory (MRAM), resistive random access memory (RRAM), phase change memory (PCM), and ferroelectric field-effect transistor (FeFET) could greatly benefit the exploration of alternative computing architectures.

These technologies offer several advantages over traditional CMOS technology, such as increased speed, higher densities, and lower power consumption. The research in the area of memory technologies has focused on exploring new materials and device architectures that can offer improved performance, energy efficiency, and reliability.

Computation-In-Memory architectures based on eNVM technologies, perform computation within the memory units, reducing data transfer and hence, energy consumption. It stands out as potential alternatives capable of satisfying the compute and memory requirements of high-performance applications. CIM introduces a groundbreaking computing paradigm with the goal of addressing the long-standing difficulty faced by the memory bottleneck in traditional Von Neumann architectures. This paradigm change has the potential to transform the way we approach computing tasks, allowing us to achieve new levels of efficiency

and performance.

In neuromorphic computing eNVM technologies are employed for synaptic weight storage, which are crucial for neural network operations and accelerating machine learning tasks. By utilizing eNVM technologies, neuromorphic computing systems can efficiently store and manipulate these synaptic weights, enabling efficient and parallel computation.

As eNVM technology has undergone substantial evolution, the potential for its applications in neuromorphic computing and compute-in-memory is expanding, paving the way for next-generation computing architectures.

The objective of this thesis is research on Artificial Neural Networks design using Compute in Memory, by employing efficient hardware solutions for ANNs at both circuit- and architecture-level. Due to the increasing interest in developing new memory technologies, this project studies not only the capabilities about classical models and computational algorithms based on neural networks but also the integration of emerging memory devices such as STT-MRAM with CMOS technology to exploit their intrinsic capability of in-memory computing.

In the following, we propose a detailed overview of compute in memory technologies, describing the promising nonvolatile memory candidates in the Beyond CMOS paradigm followed by a discussion on the potential of STT-MRAMs, highlighting the benefits and applications. Section section 1.2 provides the purpose of the thesis research and the main contributions to the thesis objectives.

1.1 Overview on Compute in memory technologies and challenges for Neural Networks

In this section, we present a comprehensive overview of the current and emerging memory technologies that enable performing deep learning computations directly within the memory units. Furthermore, we provide a detailed comparison of these memory technologies, allowing us to gain valuable insights into the future prospects of memory development. We specifically emphasize the significance of STT-MRAM in this context.

1.1.1 Review of Deep Learning

Artificial Intelligence is a broad term for a programming methodology and techniques that aims to enable computers to perform tasks that mimic human intelligence. One of the primary goals of the field of AI is to produce fully autonomous agents that interact with their environments to learn optimal behaviors, improving over time through trial and error[6].

ML is a subset of AI that focuses on enabling machines to learn from data and make predictions or decisions without being explicitly programmed to perform the specific task. ML uses a variety of algorithms and techniques to train models. Then, deep learning (DL) is a specialized form of ML that involves the use of artificial neural networks, also referred to as deep neural networks (DNN). In short, DL is a specific technique within the broader field of ML. The relationship of deep learning and machine learning to the whole of artificial intelligence is illustrated in Figure 1.1.

In essence, deep learning employs a cascade of multiple layers of nonlinear processing units for analyzing and extracting features and patterns from large and complex datasets, making it particularly effective in tasks like image and speech recognition [7].

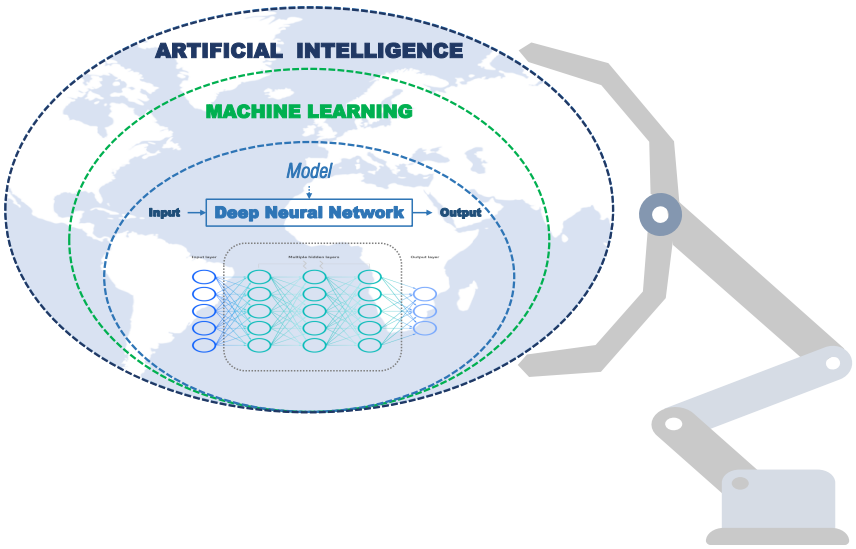


Figure 1.1: Relationship between Artificial Intelligence, Machine Learning and Deep Learning.

The field of deep learning is constantly evolving and there is a still lot of scope for digging deep into.

The popularity of deep learning nowadays can be attributed to several key advancements in the fields of machine learning, signal processing, and the substantial improvement in processing capabilities of computer chips, particularly those equipped with Graphics Processing Units (GPUs) [8]. They are currently widely used for many AI applications, including computer vision, speech recognition, and robotics, and are often delivering better than human accuracy. DNNs can offer outstanding accuracy at the cost of high computational complexity. Therefore, to expand the deployment of DNNs in both existing and new domains, strategies that enable efficient processing of DNNs to improve energy efficiency and throughput without sacrificing accuracy with cost-effective hardware are significant [9].

Today's the grand challenge for deep learning acceleration is the frequent data transfer between compute units and memory units [9]. CIM technologies have the potential to perform any computational tasks within the memory units, reducing the

need for data transfer between the memory and processing units, which can be a time-consuming and energy-intensive operation. As well as alleviating the costs in latency and energy associated with data transfer, CIM also has the potential to significantly improve the computational time complexity associated with certain computational tasks by minimizing data transfer overheads, enabling massive parallelism, handling large datasets efficiently, and optimizing for data-intensive operations. [10].

The operation which takes the most part of DNN processing is vector-matrix multiplication between the input vector and weight matrix, which essentially performs multiply-and-accumulate (MAC) operation. To this end, compute-in-memory is proposed as a promising paradigm since it emerges computation directly into memory sub-arrays [11]. As schematically illustrated in Figure 1.2.

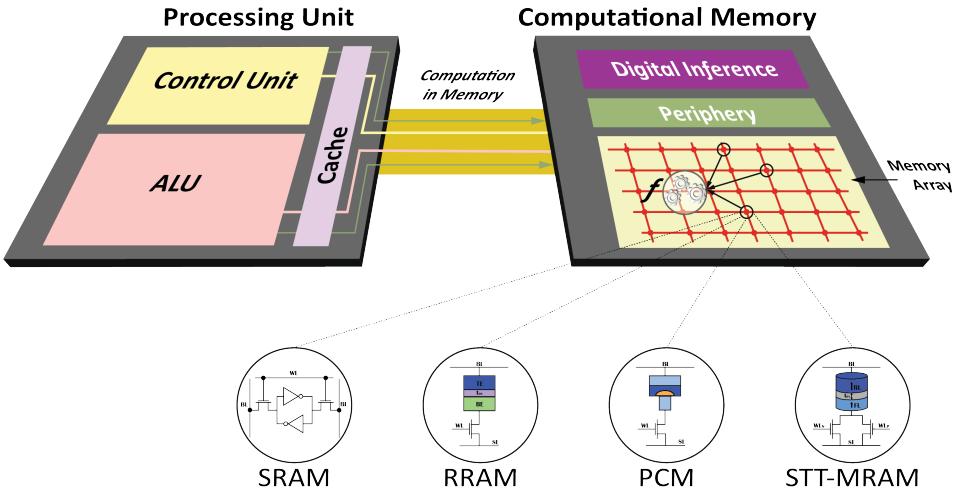


Figure 1.2: Compute-in-memory framework, the computational tasks are performed within the confines of the memory array. Static Random Access Memory (SRAM), Resistive Random Access Memory (RRAM), Phase-change Memory (PCM) and Magnetic Random-Access Memory (MRAM) technologies, can serve as elements of such a computational memory unit.

The weights of a DNN model could be represented as the conductance of memory cells in sub-array, while the input vector is supplied in parallel as voltage to the rows, then the multiplication is performed in analog way (i.e. input voltage multiplied by

weight conductance), and current summation along columns is used to generate the output vector. Here we use a black-box to conceptually represent the memory cell that could have multilevel states (for multi-bit weight), but actual implementation is often done with a series transistor to form a 1-transistor-1-resistor (1T1R) structure [11]. The block diagram of the memory array is shown in Figure 1.3. .

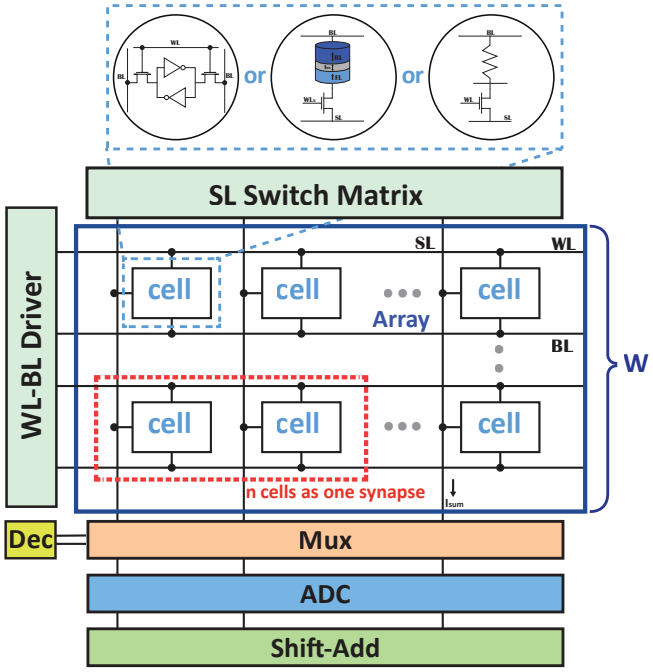


Figure 1.3: Compute-in-memory paradigm, crossbar nature of memory sub-array with perpendicular input rows and output columns.

To implement CIM, mature SRAM technologies (possibly with modified bit cell) have been proposed [12–16]. Even though SRAM is a fast and efficient memory technology, it has the drawback of being inherently volatile. This means that the data stored in SRAM is lost when the power supply is turned off. Besides that, it consumes significant standby leakage power, especially for the dynamic power gating often used in the edge devices[11].

With this perspective, eNVM are well-suited for use in memory applications. eNVM technologies [17–20] outstanding features involves non-volatility (save data even when power is turned off), energy efficiency (less power consumption compared

to MOSFET devices), high density (design at much smaller scales than traditional memory technologies) and high read/write speeds [21, 22]. Further, eNVM are more appropriate for the area/power constraint platforms, as they could be turned on and off instantly without losing the stored weights [11]. Therefore the increasing interest in exploiting eNVMs consider RRAM [23–25], phase change memory (PCM) [26–28], STT-MRAM [29–31], and ferroelectric field-effect transistor (FeFET) [32–34].

One of the promising candidates over the eNVM is STT-MRAM because of its fast read/write operation, very low standby power and high endurance. Furthermore, data is stored as a resistance value which is a function of the magnetization angle of the MTJ. Due to the limited resistance difference between the distinct resistance states of MTJ, multi-bit storage in STT-MRAM cells is difficult to achieve. As a consequence, the majority of STT-MRAM-based in-memory processing systems concentrate on bit-wise operations (i.e., operate individual bits within a binary representation of data) [21].

1.1.2 Review of Emerging Non-Volatile memory technologies

In recent years, eNVMs have emerged as promising candidates for future trends, thanks to their excellent scaling, high density, energy-efficient analog computing capabilities, and near-zero leakage power[21, 35]. Table 1.1 provides a comparison between traditional SRAM, and popular eNVM STT-MRAM, PCM and RRAM. Among these eNVM technologies, STT-MRAM stands out for its exceptional access speed and minimal energy consumption. However, it is worth noting that STT-MRAM requires a relatively larger cell area [36–38]. Nevertheless, the cell area required for STT-MRAM is significantly lower when compared to SRAM. Both PCMs and RRAMs have demonstrated the ability to store multiple logic bits in a single memory cell, showcasing superior density as technology scaling progresses [39–41].

Moreover, a qualitative comparison of STT-MRAM, RRAM, and PCM technologies is presented in Table 1.2, considering metrics such as scalability, speed, power, and reliability for evaluating the performance of memory devices.

Table 1.1: EMERGING NON-VOLATILE MEMORY COMPARISON WITH SRAM [42, 43]

	SRAM	**STT-RAM**	**RRAM**	**PCM**
Non-volatility	No	Yes	Yes	Yes
Cell size (F^2)	120-200	6-50	4-10	4-19
Multibit	1	1	> 2	2-7
Write Endurance	10^{16}	10^{12}-10^{15}	10^{8}-10^{11}	10^{8}-10^{9}
Read Latency	$\sim$0.2-2 ns	2-35 ns	$\sim$10 ns	20-60 ns
Write Latency	$\sim$0.2-2 ns	3-50 ns	$\sim$50 ns	20-150 ns

Table 1.2: KEY METRICS FOR MEMORY PERFORMANCE ASSESSMENT AND A QUALITATIVE COMPARISON OF STT-MRAM, PCM, AND RRAM TECHNOLOGIES[44].

Memory performance metrics		**STT-MRAM**	**RRAM**	**PMC**
Scalability	Size and proximity limit	high	high	medium
	MLC (multi-level cell)	medium	medium	high
	3D capability	worst	high	high
	Cell structure(1T1R, crossbar,...)	medium	high	high
Speed	Writing: switching mechanism	high	medium	medium
	Reading: on/off ratio, variability, sensing scheme, array layout	high	high	high
Power	Writing: switching I/V	medium	high	worst
	Reading: sensing schemes	high	high	high
Reliability and Variability	Retention	high	medium	high
	Endurance	high	worst	medium
	Variability	high	worst	medium

This comparison highlights the diverse strengths and characteristics of these eNVM technologies, each offering unique advantages for specific applications such as brain-inspired neuromorphic computing, hardware security, storage class memory, electronic synapses, neuromorphic architectures, and the Internet of Things (IoT).

Phase-Change Materials offer a range of useful properties, including high

energy storage density, fast response time, and long-term stability of stored data, making them valuable in applications such as data storage, energy storage, and thermal management, making them ideal candidates for crossbars [45]. PCMs have the ability to absorb, store, and release large amounts of latent heat during phase transitions between solid and liquid states or between solid and amorphous states. PCMs can be switched from one state to the other by heat, by applying a series of low- and highamplitude voltage pulses [35]. The relative volumes of the switching domain (amorphous/crystalline) allow multi-level conductance states to be stored [46].

Resistive Random Access Memory has attracted interest for its low-power consumption, high-density storage, and fast switching speed. The device is based on a typical metal-insulator-metal (MIM) structure. Once a sequence of voltage pulses is applied, the dielectric undergoes a soft breakdown, leading to the creation of conductance levels at multiple tiers [35]. RRAM offers just two states. Despite suffering from low endurance, power consumption is better along with the latency, see Table 1.2 and Table 1.1.

1.1.3 Magnetoresistive Random Acces Memory

Spin-based memory – MRAM have gained attention as a potential platform for logic circuit design. The discovery of Giant Magnetoresistance (GMR) [47–49] led to the development of spintronics [50]. Spintronic devices are based on the MTJ structure, which is the physics phenomenon involved in the MRAM behavior. MTJ is a multilayer magnetic nano-pillar structure, as shown in Figure 1.4. The MTJ can be categorized as an in-plane MTJ (i-MTJ) (see Figure 1.4(a-b)) if both the pinned and free ferromagnetic layers have their magnetic orientation in the plane of the MTJ. Conversely, if the ferromagnetic layer's magnetic orientation is perpendicular to the MTJ plane, then it is a perpendicular plane MTJ (p-MTJ)(see Figure 1.4(c-d)).

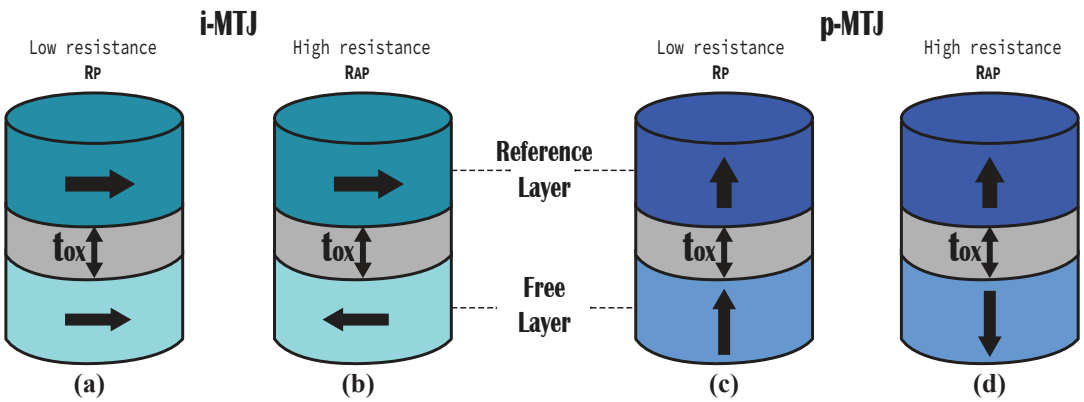


Figure 1.4: in-plane MTJ (i-MTJ) at **(a)** low resistance state, **(b)** high resistance state versus perpendicular MTJ (p-MTJ) at **(c)** low resistance state and **(d)** high resistance state .

The MTJ consists of two ferromagnetic CoFeB layers, i.e. one layer with a fixed magnetization called reference layer (RL) and the other with a variable magnetization orientation called free layer (FL). Both layers are separated by a non-magnetic insulating layer, i.e. thin MgO oxide barrier. The magnetization orientation can be changed by applying a switching current greater than the critical switching current of the device [38, 51]. When the magnetization orientation of the reference layer and free layer are parallel (P), the resulting resistance is lower (R_P), consequently this state is considered as a stored bit of ”1”. In contrast if the the magnetization orientation of the reference layer and free layer are anti-parallel (AP), it corresponds to high resistance state (R_{AP}) and the state is considered as a stored bit of ”0” (see Figure 1.4).

The ratio between the two resistance values is named Tunneling Magneto resistance Ratio (TMR) and is expressed as $TMR=(R_{AP}-R_P)/R_{AP}$, when the magnetizations of the two electrodes are aligned in parallel and antiparallel, respectively. The higher TMR, the better can be distinguished the cell state. Thus, TMR is adopted as the main performance metric of MRAM cell, regardless of the materials and structures used and remarkable improvement of the memory array.

Conventional MTJs, exibit a TMR up to 70% with amorphous Al_2O_3 with crystalline MgO as barrier material at room temperature [52]. In 2001, two separate studies conducted by Mathon *et al* and Butler *et al* exhibited a remarkably theory of high TMR ratios, surpassing 1000% in Fe/MgO/Fe sandwiched structures [53, 54]. In 2004, Yuasa *et al* reported a TMR value of 180% in an epitaxial Fe/MgO/Fe MTJ, and Parkin *et al* obtained a TMR value of 220% in a sputtered CoFe/MgO/CoFe MTJ [55, 56]. The successive invention [57] of the CoFeB/MgO/CoFeB MTJ structure reported in 2005 was decisive in the mass production of MgO based MTJs, it has been observed with a TMR ratio up to 230% at room temperature. From other recent reports a TMR of 213% at room temperature in Fe_3GaTe_2-based MTJs with a bias current down to 10 nA was reported in [58].

MRAM is a type of non-volatile memory technology that uses magnetic properties to store data. In MRAM, data is stored in magnetic elements, often referred to as magnetic tunnel junctions MTJs. The development of MRAMs has emerged as a exceptional technology in the field of non-volatile memory, promising high-speed, high Reliability, low-power consumption, and robust data storage solutions. The invention of spin-transfer torque (STT) magnetization switching and the enhancement of the TMR effect through the use of MgO barriers have led to commercial production of MRAMs [59]. Among the various types of MRAM devices, two prominent contenders have emerged: Spin-Orbit Torque MRAM (SOT-MRAM) and STT-MRAM. In SOT-MRAM, data is stored by applying an electric current that induces a spin-orbit torque on the magnetic layer of a memory cell. This torque effectively switches the magnetic orientation, allowing for writing and reading of data. SOT-MRAM offers advantages such as lower write currents, faster switching speeds, and potentially higher endurance.

STT-MRAMs differ from their predecessor MRAMs in terms of the magnetic reversal process. While MRAMs use an external magnetic field to reverse the

magnetic polarization, STT-MRAMs use a direct current for this purpose. One significant advantage of STT-MRAMs over MRAMs is that the magnitude of the required switching current reduces proportionately with the size of MTJ. This reduction in switching current allows STT-MRAMs to be scaled down further in size. Further the discovery of STT provided a perfect solution to the poor scalability of MRAM [60–62]. As a result, STT-MRAMs can be implemented in smaller technology nodes, making them viable for use in devices with limited space [63].

The scaling down of the MTJ size in STT-MRAMs results in a proportional reduction in the required switching current. This reduction enables STT-MRAMs to further scale down in size, making their implementation feasible in smaller technology nodes [63].

These magnetoelectronic circuits offer non-volatility and the potential for lower power consumption compared to traditional electronics. Various proposals have been made for using magnetic devices to construct logic circuits, with some independent of CMOS technology and others tightly integrated with it [64]. MTJs have the potential to compete with and replace conventional SRAM-based memory technologies since they provide a low write latency and high endurance.

Hence, STT-MRAMs are actually considered as a strong competitor for non-volatile cache applications due to their promising features. These include high integration density, high speed, almost zero standby power, long data retention time, and full compatibility with CMOS process, [65–68].

As a result, the technology quickly gained recognition as an appealing choice for persistent memory in embedded applications [69–73]. STT-MRAMs garnered significant attention from academia and attracted substantial investments from the industry.

1.2 Objective, scope and framework of the thesis

Within the above context, the research in this thesis presents efficient hardware solutions for ANNs at both circuit- and architecture-level. In particular, the activity is focusing on the integration of emerging memory devices (e.g. spintronic memories) with CMOS technology to exploit their intrinsic capability of in-memory computing. As a result of the significant advantages of spintronic devices such as, their non-volatile nature, consuming less power, high read and write speeds and good endurance, the spintronic devices possess unique features that make them attractive for applications that require high-density, non-volatile, and energy-efficient memory technologies. Hence, the thesis focuses in finding out efficient and optimized solutions for compute in memory. Circuit design activity involving emerging memory devices requires the use and calibration of Verilog-A based compact models [74, 75] to integrate the behavior of such devices into the circuit design tool.

Likewise the thesis project evaluates and improves circuit solutions to implement the building blocks of an ANN (i.e. artificial synapses and artificial neurons) using commercial circuit simulators (e.g. Cadence Virtuoso).

The detailed thesis organization is as follows.

Chapter 2 provides a comprehensive description of the technological aspects, devices, and topology used throughout the research. Additionally, the methodology employed for simulation and evaluation, with a specific emphasis on integrating spintronic memories with CMOS technology is analyzed.

Chapter 3 explores the architecture of an ANN with a particular focus on the activation function used at the output node. The analog implementation of sigmoid and softmax activation function is described.

Chapter 4 presents the efficiency of STT-MRAM cell based on double-barrier

MTJ on the performance of a two-layer MPL neural network. The DMTJ-based cell is benchmarked against the conventional single-barrier MTJ (SMTJ) counterpart by means of a comprehensive evaluation carried out through a state-of-the-art device-to-algorithm simulation framework.

Chapter 5 introduces an advanced Logic-in-Memory (LIM) architecture developed from the smart material implication (SIMPLY) logic scheme. SIMPLY+ scheme is proposed and benchmarked against its conventional counterpart, both implemented using STT-MRAM devices.

Chapter 6 concludes the dissertation and proposes future work.

CHAPTER 2

MRAM TECHNOLOGY FOR ANN METHOD AND IC STRATEGY

This chapter have been written using the publication reported by T. Moposita *et al* [29]. The combination of MRAM technology and IC strategy has gained attention in the context of ANN methods as it contributes to the advancement of memory technologies and the development of more efficient and powerful electronic devices. MRAM technology offers advantages that can be leveraged in the design and implementation of integrated circuit. The non-volatile nature of MRAM allows for the preservation of trained neural network models and data, even when power is turned off, simplifying circuit design and reducing power consumption.

This chapter describes the methodology for simulation and evaluation, focusing on the integration of emerging memory devices, such as spintronic memories, with CMOS technology. Starting with an overview from the algorithm level, focusing on an overview of ANNs and their importance in hardware implementations, to the device level, where MTJ devices and their switching mechanisms are detailed due to their impact on the performance of STT-MRAM. The goal is to take advantage of the inherent capacity of these memory devices for in-memory computing.

2.1 Overview Framework

Biological neural systems are incredibly complex and efficient machines that are capable of solving a wide range of problems with high speed and energy efficiency.

This has led researchers to try and develop electronic systems that are inspired by these biological neural networks, known as artificial neural networks (ANNs). ANNs are able to ”learn” from input data and examples, and then make predictions or classifications on new input data without being explicitly programmed with a set of predefined rules. ANNs have emerged as powerful tools in various fields, including pattern recognition, machine learning, and artificial intelligence. Their ability to mimic the human brain's functionality has led to significant advancements in solving complex problems.

However, as the demand for more efficient and powerful computing systems continues to grow, traditional computing architectures still face limitations in terms of speed, energy efficiency and scalability. To address these challenges, researchers have been exploring alternative computing paradigms that can harness the unique properties of emerging technologies. One such promising avenue is the integration of spintronic devices with ANNs. Spintronics, which exploits the intrinsic spin of electrons in addition to their charge, offers several advantages such as non-volatility, low power consumption, and high-speed operation. This methodology aims to develop a comprehensive framework for the design and implementation of spintronic-based ANNs, spanning from algorithmic formulation to device-level integration. As shown in Figure 2.1, the methodology encompasses four main levels: device, bitcell, architecture and algorithm.

At the device level (see Figure 2.1(a)), the design and characterization of spintronics devices such as MTJs (perpendicular SMTJ and DMTJ devices featuring circular geometries) are considered. Then, moving to bitcell- and architecture level (see Figure 2.1(b)-(c)), the analysis involves the design of specialized hardware architectures optimized for spintronic-based ANNs integrated with the CMOS technology to realize fully functional neural network accelerators. Finally, at the algorithm level (see Figure 2.1(d)), the focus lies on the study of neural network

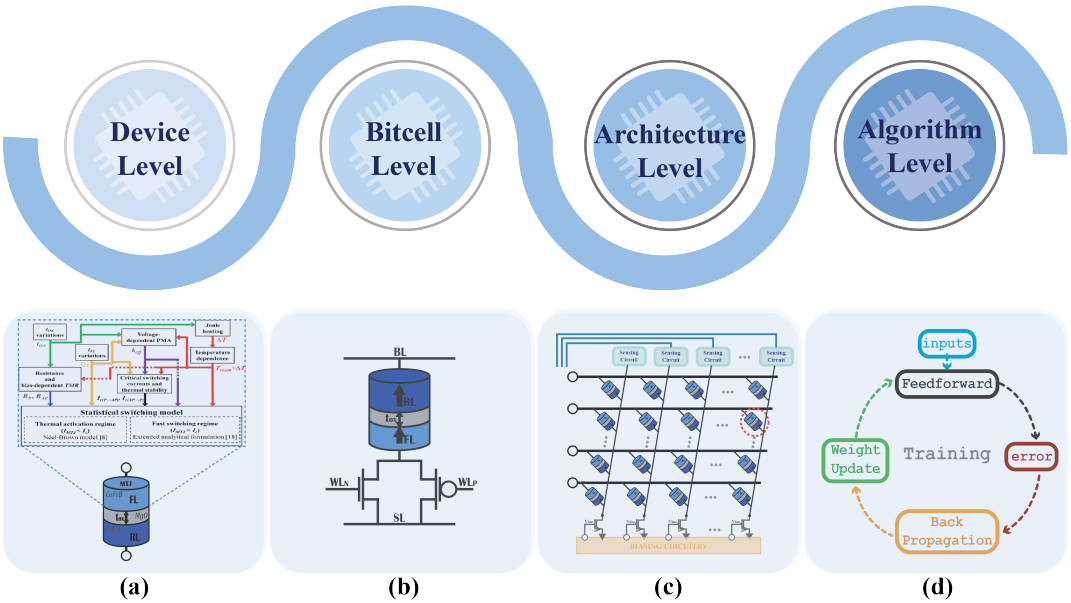


Figure 2.1: General framework including four levels of abstraction: (a) algorithm-, (b) architecture-, (c) bitcell-, and (d) device-level.

models, training techniques and the computation unit which involves the activation function.

In the following subsections, each aspect is further explored.

2.1.1 Spintronics: Material and Device-to-Circuit analysis

Basis of MTJ Device

MTJ, a basic unit of MRAM, is a multilayer magnetic nano-pillar structure, as shown in Figure 2.2. The perpendicular magnetic orientation of the ferromagnetic layer in the p-MTJ, as depicted in Figure 2.2(a), offers advantages in terms of potentially requiring smaller write current while maintaining equivalent thermal stability.

Figure 2.2(b) describes the switching process during the AP $\rightarrow$ P transition compared with the the P $\rightarrow$ AP transition. One common method to accomplish this is through spin-transfer-torque switching, which involves using spin-polarized current to invert the magnetization direction. Hence the switching process refers to

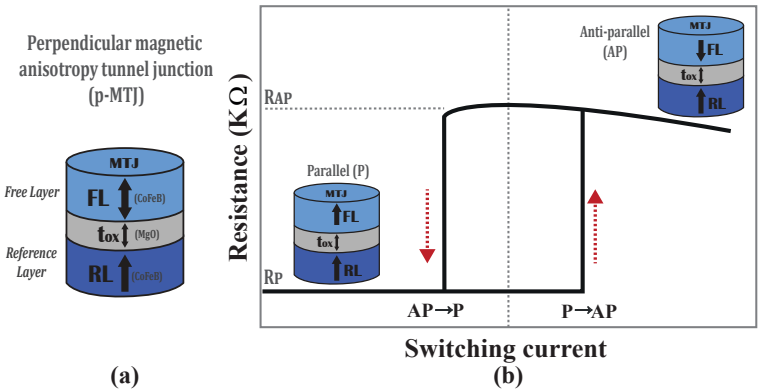


Figure 2.2: Schematic views of MTJ device. (a) Structure of p-MTJ. (b) Resistance-voltage characteristics of the MTJ

the transition that takes place when the magnetization orientations of the reference layer and free layer align either in parallel (resulting in lower resistance and a stored bit of ”1”) or in antiparallel (resulting in higher resistance and a stored bit of ”0”) due to a switching current applied to the MTJ.

The efficiency and stability of the switching process can be influenced by several factors, including the thermal-induced statistical magnetization process, the perpendicular magnetic anisotropy constant, and the TMR. Parameters such as the write current and external factors like heating can also have an impact on the effectiveness of the switching process [76].

Magnetic anisotropy

Magnetic anisotropy is a critical parameter that determines the orientation of the magnetic moment of the ferromagnetic layers in the device. It refers to the directional dependence of the magnetic material’s properties, where the magnetic moment of the material tends to align itself along its easy axis [43] in the absence of an external magnetic field or voltage.

The effective magnetic anisotropy (k_{eff}) refers to the effective energy associated with the preferred direction of magnetization in a magnetic material and is

represented in Equation 2.1

$$k_{eff} = k_v + \frac{2k_s}{t} \tag{2.1}$$

Where, k_v is the volume contribution, k_s the interface contribution and t the thickness of the magnetic layer. This relationship depicts the weighted average of magnetic anisotropy, in bulk materials, anisotropy is dominated by the volume component, whereas, in the thin films, the surface term is dominant. Critical thickness (t_{CO}) is represented in Equation 2.2. In thin-film material, the thickness is below the t_{CO}, and for bulk materials, the thickness exceeds t_{CO}.

$$t_{CO} = \frac{-2k_s}{k_v} \tag{2.2}$$

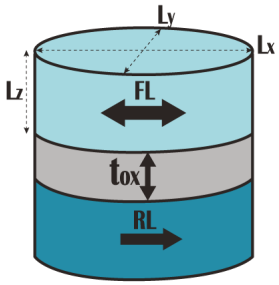
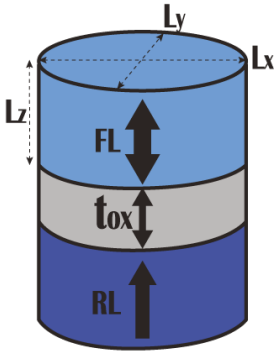
The correlation between in-plane magnetic anisotropy and perpendicular plane magnetic anisotropy is dependent on both the effective thickness and effective anisotropy constant. Additionally, the easy axis for the i-MTJ lies on the plane of the FL and its shape is elliptical, while the p-MTJ is circular. This comparison between i-MTJ and p-MTJ is briefly outlined in Table 2.1.

Spin transfer torque (STT)

STT devices are a type of spintronics-based electronics that use a spin-polarized current to flip the spin of electrons in a thin magnetic layer. The applied current generates a spin-polarized current that transfers an angular momentum to the magnetic layer, resulting in a change in the spin of the electrons.

The spin-transfer effect can be described by a simple s-d model as shown in Figure 2.3(a). The s-electrons flow among the localized d-electrons and contribute to a charge and spin current, while d electrons create a single large local magnetic moment because of strong d–d exchange interaction. s–d exchange interaction

Table 2.1: COMPARISON BETWEEN I-MTJ AND P-MTJ IN TERMS OF ITS BASIC MAGNETIC ANISOTROPY PROPERTIES

Characteristics	**In-plane magnetic**	**Perpendicular plane magnetic**
		
Free layer dimension	$L_z \ll L_y < L_x$	$L_z \ll L_y \sim L_x$
Dominan anisotropy	Shape	Volume
Switching current	Larger	Smaller
Scaling	Difficult / not desirable	Easy / desirable
Thermal stability	Low	High
Shape	Elliptical	Circular

causes a precession of s- and d-electrons. Since d-electrons create a single large spin moment, the precession angle of the d-electron system is considerably smaller than that of s-electrons.

Figure 2.3(b) describes the electrons flow from the RL to the FL, s-electrons are spin-polarized and aligned in the magnetic direction of the RL. Then the spin angular momentum is transferred to the d-electrons of the FL to conserve the total spin angular momentum. A large torque called STT is applied, causing the magnetic orientation of the FL to align with the RL. Hence, if MTJ was in AP configuration, it switches to P configuration. The opposite occurs when the electrons flow from the FL to the RL (see Figure 2.3(c)), causing the magnetic orientation of the FL to align with the RL and switching the MTJ from P to AP configuration.

As a result, bidirectional STT is applied in the form of a write current (I_W) to switch the MTJ between P and AP configurations. The magnetization state of the FL can only change if the applied torque is strong enough, which is determined by

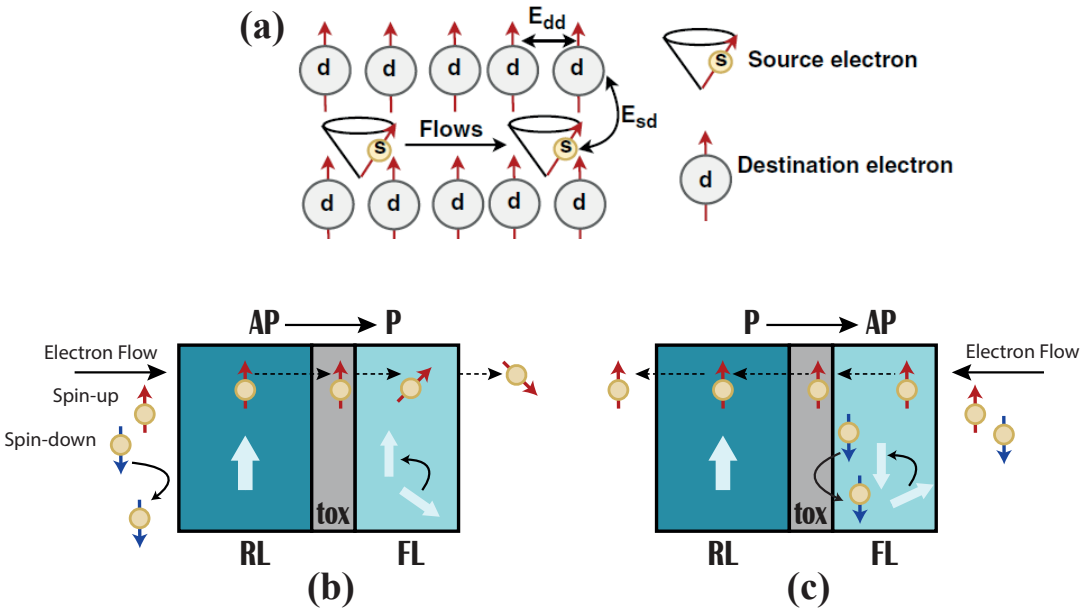


Figure 2.3: Schematic representation of STT switching mechanism **(a)** A simple s–d model to describe the spin-transfer effect, **(b)** s-electrons flow from RL to FL to change the MTJ resistance from R_{AP} to R_P, and **(c)** s-electrons flow from FL to RL to change the MTJ resistance from R_P to R_{AP} [43].

the critical current density (J_c). Although large currents significantly above J_c can quickly switch the magnetization of the FL, this also results in significant power dissipation. J_c can be defined through Equation 2.3 [77].

$$J_c = \left(\frac{\alpha}{\eta}\right)\left(\frac{2e}{\hbar}\right) M_s\, t_F\, H_{eff} + 2\pi M_s \tag{2.3}$$

where α is Gilbert damping constant, η is the STT efficiency parameter, e is electron charge, $\hbar$ is reduced Plank constant, M_s is the saturation magnetization, t_F is thickness of the FL and H_{eff} is effective magnetic field.

The STT switching mechanism is heavily influenced by magnetic anisotropy. While the p-MTJ is relatively problem-free when used in circuit applications, the i-MTJ suffers from a number of issues, including a shorter data retention time, lower thermal stability, and a higher critical current (I_{c0}) required to switch the

magnetic orientation of the FL [43].

STT effect enables bidirectional switching of the state of an MTJ by applying a current greater than a critical current, denoted as I_{c0}. This effect offers advantages such as high density and low power consumption , making it widely used in memory, logic, and hybrid circuit designs. This has enabled the launch of commercial products based on MRAM, and it also promises the scaling of circuits for even higher density [78, 79].

The MTJ behavior model i-MTJ and p-MTJ is given by the following equations,

$$I_{c0\,\perp} = \alpha\ \frac{\gamma e}{\mu_B g}\ (\mu_0 M_s)\ H_k\ V \tag{2.4}$$

$$I_{c0\,\parallel} = \alpha\ \frac{\gamma e}{\mu_B g}\ (\mu_0 M_s)\ V\ \left(H_k + \frac{M_s}{2}\right) \tag{2.5}$$

$$E = \frac{\mu_0\ M_s\ H_k\ V}{2} \tag{2.6}$$

Where α is the magnetic damping constant, γ is the gyromagnetic ratio, e is the elementary charge, μ_B is the Bohr magneton, g is the spin polarization efficiency factor, μ_0 is the permeability of free space, M_s is the saturation magnetization, H_k is the effective anisotropy field and V is the volume of the free layer.

Moreover Equation 2.7 gives the average MTJ state switching delay time;

$$\tau = \tau_0\ exp\left(\frac{E}{k_B T}\left(1 - \frac{I}{I_{c0}}\right)\right) \qquad when\ I < I_{c0} \tag{2.7}$$

Where τ_0 is the attempt period, k_B is the Boltzman constant and T is the temperature. Besides, the MTJ thermal stability is represented in the following expression, Equation 2.8.

$$\Delta = \frac{H_k \, M_s}{2 \, k_B \, T} \, V \tag{2.8}$$

The higher Δ (and I_{c0}) is, the more stable against the thermal fluctuation noise is the MTJ state.

Comparison between Equation 2.4 and Equation 2.5 elicits that STT must overcome additional $M_s/2$ factor in i-MTJ for satisfactory performance. Therefore p-MTJ requires lower write current than i-MTJ. Some studies have reported the CoFeB/MgO-based STT p-MTJ as high performance, with a TMR of 150% at room temperature, a small size of $\sim$ 20 nm diameter, a good thermal stability factor of $\Delta = 40$, and low I_{c0} of $\sim 9\,\mu$A [37, 38].

Device-level analysis

STT-MRAMs have emerged as top contenders for replacing conventional semiconductor-based cache memories at smaller technology nodes. Nonetheless, a significant hurdle in the widespread adoption of STT-MRAMs is the need to reduce their write currents to achieve energy and area savings[68]. In this regard, one effective strategy concerns the use of DMTJs with two reference pinned layers[80–83] instead of conventional SMTJs [38].

For Single-barrier MTJ, see Figure 2.4, based on the relative magnetization direction of the FL and RL, the SMTJ resides in one of two stable states: parallel or antiparallel. If two FM layers have the same magnetization directions, i.e., RL and FL in P, the resistance of the MTJ is low (R_0), indicating a "0" state. Conversely, if the two layers have different magnetization directions, i.e., RL and FL in AP, the resistance of the MTJ is high (R_1), indicating a "1" state [38].

For Double-barrier MTJ, see Figure 2.4, the FL is sandwiched between two MgO oxide barriers, each of them interfaced with one RL. The low resistance state

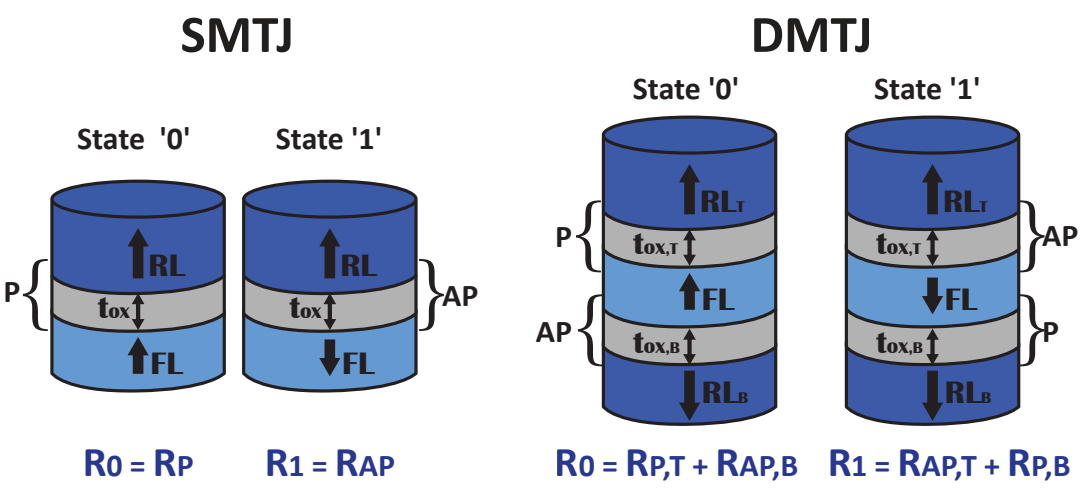


Figure 2.4: Single-barrier MTJ and Double-barrier MTJ devices

("0") corresponds to FL in P and AP with respect to the RL top and RL bottom, respectively. As for the high resistance state ("1"), the FL is in AP and P with respect to RL bottom and RL top, respectively. Accordingly, the DMTJ resistances in states '0' and '1' can be calculated as R_0=$R_{P,T}$+$R_{AP,B}$ and R_1=$R_{AP,T}$+$R_{P,B}$, respectively, [38]. Due to the presence of the second reference layer, the spin-transfer torque is enhanced. Thanks to the reduced switching current in DMTJs, the DMTJ-based non-volatile flip-flop demonstrates a $3\times$ reduction in backup time and a $6\times$ decrease in energy compared to its SMTJ-based counterpart. Such benefits are obtained along with smaller area occupation and better performance in the flip-flop active operation mode [75].

In this thesis, we explore STT-SMTJ/DMTJ devices with circular PMA geometry, which were developed by our group [74, 75], this devices have been validated against full micromagnetic and experimental results. Our group has developed Verilog-A based compact models for perpendicular STT MTJs that describe these devices. These models calculate the MTJ resistance in both states accounting for bias voltage dependence of the TMR ratio, and the critical switching current (I_{c0}) for $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions. The main physical device parameters

Table 2.2: MAGNETIC TUNNEL JUNCTION PARAMETERS

Parameter	Description	Value	Units
M_s	Saturation magnetization (300 K)	1000×10^3	A/m
α	Gilbert damping factor	0.03	–
r	MTJ radius	15	nm
$t_{OX}(\sigma/\mu)$	Oxide thickness (variability)	0.85 (1%)	nm
$t_{FL}(\sigma/\mu)$	FL thichness (variability)	1.0 (1%)	nm
φ_B	Oxide energy barrier	0.4	eV
RA	Resistance-area product	7.0	$\Omega\cdot$ µm^2
RP	MTJ resistance in P state	9.9	kΩ
TMR	TMR ratio (300 K and 0 V)	65%	–
V_H	Bias voltage for $TMR = 0.5 \times TMR(0)$	0.5	V
η	Spin-polarization factor	0.66	–
N_{xy}	In-plane demagnetizing factor	0.0439	–
N_z	Perpendicular demagnetizing factor	0.9122	–
k_{eff}	Effective anisotropy (300 K and 0 V)	0.405	–
$J_{c(P\rightarrow AP)}$	P $\rightarrow$ AP critical current density	~ 2.5	MA/ cm^2
$J_{c(AP\rightarrow P)}$	AP $\rightarrow$ P critical current density	~ 1.0	MA/ cm^2
ξ	Magnetoelectric coefficient (300 K)	40	fJ/V·m
T_{room}	Room temperature	300	K
λ	Thermal conductivity	20	W/m·K
C_v	Heat capacity per unit volume	3.5×10^6	J/ m^3 · K

for FinFETs and MTJs at 28- nm, 24- nm, and 20- nm technology nodes are presented in Table 2.2. The table also includes the impact of the MTJ process variability on parameters such as t_{OX}, t_{FL}, cross-section area, and TMR.

The implemented model devices take into account the effects of voltage-dependent perpendicular magnetic anisotropy, temperature-dependent parameters, thermal heating/cooling, MTJ process variations, and the spin-torque asymetry of the Slonczewski spin-polarization function in the switching process [77, 84].

2.1.2 STT-MRAM Bitcell-level analysis

The physical phenomena described in the section above can be utilized in the design of non-volatile storage devices for on-chip memory applications. Studies have

demonstrated that STT-MRAM exhibits significant potential as a future on-chip memory technology thanks to its nonvolatile nature [85, 86], compatibility with CMOS fabrication processes [87, 88], high endurance [89, 90], and scalability [91, 92].

The bit-cell assessment was developed by E.Garzón *et al* [37, 38], which involved conducting analyses at various levels of abstraction, from device-level to circuit-level analyses for the single memory bitcell. The device models were imported into the Cadence Virtuoso environment for a circuit-level analysis, using Verilo-A. This analysis was carried out to evaluate the performance of the single bitcell with respect to writing and reading operations under the impact of scaling and variation effects. In Figure 2.5, a typical STT-MRAM bit-cell is presented, which comprises an access transistor and a MTJ, commonly referred to as the 1T-1MTJ configuration. Similarly, another bit-cell configuration known as 2T-1MTJ is used, which utilizes complementary CMOS transistors.

MTJ can be connected to the access transistor(s) of an STT-MRAM bitcell in two ways, the standard connection (SC) and the reversed connection (RC), see Figure 2.5 [93, 94]. When using reversed connection, the access transistor(s) is connected to the free layer of the MTJ. Conversely, in standard connection, the access transistor is connected to the reference layer of the MTJ. Owing to their inherent asymmetry, the SC and RC configurations exhibit distinct switching behaviors and are subject to differing degrees of the source degeneration effect. This phenomenon refers to the reduction in write current resulting from the gate-source voltage droop when the MTJ is driven by the transistor source rather than its drain terminal [93].

These asymmetries adversely affect both performance and energy consumption. Specifically, SC bit-cells exhibit source degeneration during the P-AP transition, while RC bit-cells experience it during the AP-P transition. To address this issue, STT-MRAM bit-cells with two transistors (2T-1MTJ) have been developed, as

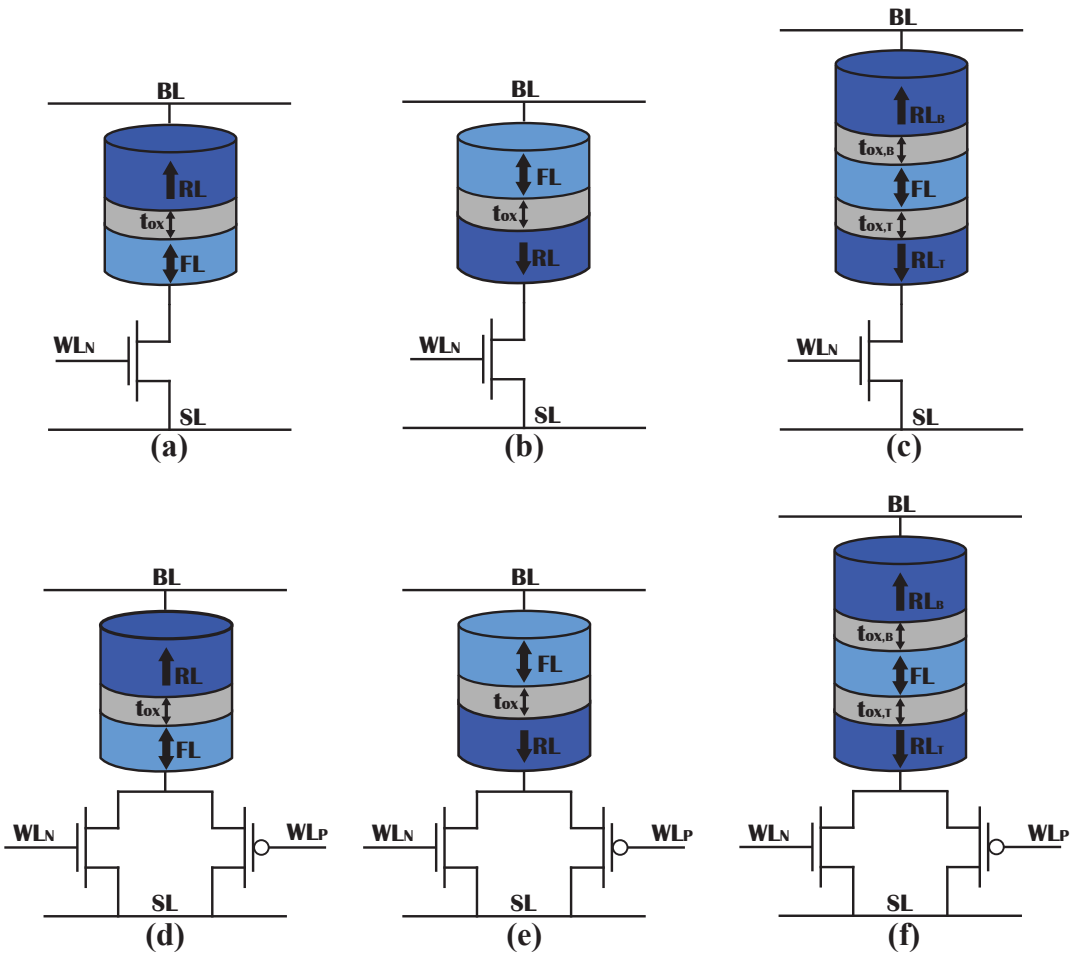


Figure 2.5: Connection of bottom-pinned STT-MRAM bit-cell (a) 1T-1MTJ RC, (b) 1T-1MTJ SC (c) 1T-1DMTJ (d) 2T-1MTJ RC (e) 2T-1MTJ SC (f) 2T-1DMTJ

shown in Figure 2.5(d-f) [63, 94]. These are based on complementary CMOS architectures, only one transistor is subject to source degeneration at a time, while the other delivers its maximum current, thus mitigating such asymmetries.

2.1.3 Architecture Level

STT-MRAM-based Binarized Neural Network (BNN) in-memory accelerators are a novel computing architecture that promises advantages in terms of density and leakage power [71, 95]. These architectures have the potential to greatly improve the efficiency and performance of computing systems [96].

Employing the MTJ based pseudo crossbar array architecture, it is possible to transform the current to operate as massively parallel computational units, transforming it into a standard STT-MRAM memory array capable of functioning as both nonvolatile memory and in-memory logic [71, 72, 97–103].

As we have demonstrated in earlier works referenced in [5], the proposed logic circuit based on hybrid CMOS and STT-MTJ has shown promising potential for designing efficient non-volatile logic-in-memory (NV-LIM) architectures, ensuring low power consumption and increased speed.

In light of the context presented, one segment of this thesis is devoted to analyzing the performance of conventional BNNs when implemented on STT-MRAM array architectures, with particular emphasis on single bit XNOR bit-cells [71] utilized for conducting the MAC operation.

STT-XNOR Architecture

This strategy has been explored in SRAM, emerging nonvolatile memories such as resistive random-access memory (XNOR-RRAM), and magnetic tunnel junction-based magnetic random-access memories (MRAMs) [97]. We use the MTJ physic model described in Verilog-A in Cadence Virtuoso to describe the XNOR architecture proposed by [104].

The process of computing the scalar product of weights and inputs in-memory has been identified as an XNOR operation, leading to the reference of designs using this method as XNOR-BNNs.

Through the usage of modified sense amplifiers, MTJ devices can activate two or more memory rows that store weights and inputs, and execute parallel X(N)OR operations on the Bit-lines that are essential in BNNs with a reduced cycle. Nevertheless, an increase in write power may occur when storing input feature maps of BNNs on-chip, but this can be prevented by utilizing modern design approaches

such as [105, 106].

The STT-XNOR architecture was introduced by [71] to perform unrestricted accumulation across rows for full utilization of array and BNN model scalability.

Figure 2.6 displays the conventional architectures for in-memory BNN acceleration considering the traditional column-level accumulation. The wordlines of the array are feed with the inputs of a given layer, and the corresponding weights are stored within a column are multiplied bit-wise and subsequently summed up to produce a cumulative current, which is featured in the corresponding bitline.

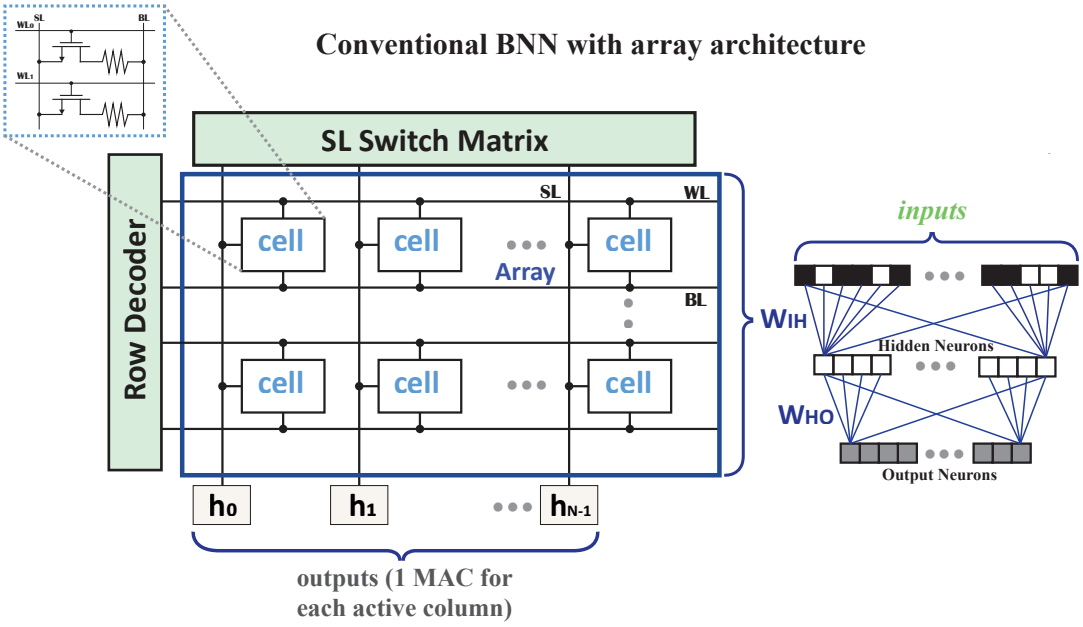


Figure 2.6: Generic BNN model and mapping onto conventional in-memory STT-MRAM architecture.

The adopted bitcell of the memory array is based on the popular 2T-2MTJ as shown in Figure 2.7, the access transistors M1 and M2 are configured such that their gates are connected to the wordline (WL), while the source is linked to the select line (SL). The magnetic junction, MTJ_0, is established with either high resistance (anti-parallel magnetization resistance, R_{AP}) or low resistance (parallel magnetization resistance, R_P) to store weights of -1 or +1, respectively. The other

magnetic junction, MTJ_1, is arranged in a complementary state. The two bitlines BL_0 and BL_1 connected to the bitcell and its associated column are used to feed the input features on a column basis.

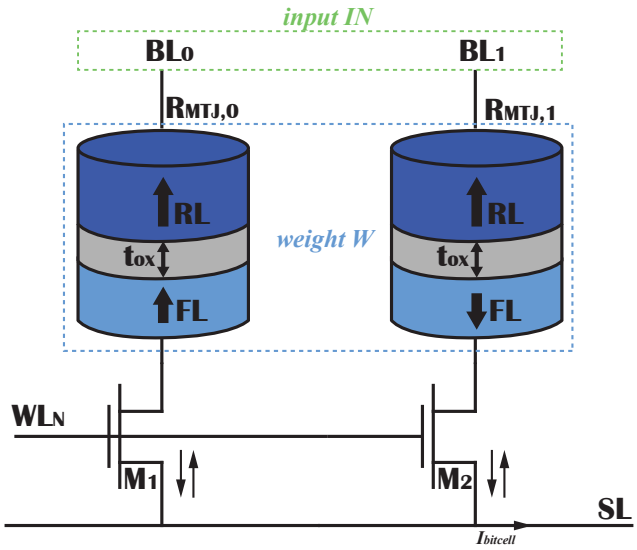


Figure 2.7: 2T-2MTJ STT-MRAM bitcell for single-bit XNOR (proposal of [71]).

To feed +1, the voltages are setted up to a proper $V_{BL,0} > 0$ and $V_{BL,1} = 0$ against to $V_{BL,1} > 0$ and $V_{BL,0} = 0$ to feed -1. The non-zero bitline voltage determines the current pushed by the bitcell onto the select line, according to the resistance of the MTJ, it is high (i.e., +1 output) in the MTJ has a low resistance (parallel), conversely, it is low (i.e., -1 output) under high resistance (antiparallel), as presented in Table 2.3.

Table 2.3: IN XNOR W

$\mathbf{MTJ_0, MTJ_1}$	**AP, P**	**P, AP**
$\mathbf{BL_0, BL_1}$	**(-1)**	**(+1)**
$\mathbf{V_{BL0} > 0, V_{BL1} = 0}$	Low	High
(+1)	(-1)	(+1)
$\mathbf{V_{BL0} = 0, V_{BL1} > 0}$	High	Low
(-1)	(+1)	(-1)

Hence, the bitcell's current that is applied to the select line corresponds to the XNOR of the input registered in the bitline and the stored weight that is retained

within the cell, see Equation 2.9.

$$I_{bitcell} = \frac{V_{BL}}{R_{MTJ} + R_{access}} \tag{2.9}$$

For each case the $I_{bitcell}$ is expressed as

$$I_{bitcell}(IN = +1) = \begin{cases} \frac{V_{BL}}{R_P + R_{access,0}} & if\ W = +1 \\ \frac{V_{BL}}{R_{AP} + R_{access,0}} & if\ W = -1 \end{cases} \tag{2.10}$$

$$I_{bitcell}(IN = -1) = \begin{cases} \frac{V_{BL}}{R_{AP} + R_{access,1}} & if\ W = +1 \\ \frac{V_{BL}}{R_P + R_{access,1}} & if\ W = -1 \end{cases} \tag{2.11}$$

The overall resistance called $R_{bitcell}$ is calculated as follows;

$$R_{bitcell} = (R_{MTJ,0} + R_{access}) \parallel (R_{MTJ,1} + R_{access}) \tag{2.12}$$

The resulting SL voltage for a single bit-cell is expressed as

$$V_{SL} = R_{bitcell} \cdot I_{bitcell} = \begin{cases} \frac{V_{BL} \cdot R_{bitcell}}{R_{AP} + R_{access}} & if\ \overline{W_{ij} \oplus IN_j} = -1 \\ \frac{V_{BL} \cdot R_{bitcell}}{R_P + R_{access}} & if\ \overline{W_{ij} \oplus IN_j} = 1 \end{cases} \tag{2.13}$$

Finally, in BNNs, the output OUT_i of the accumulator is a digital value that corresponds to the binarized select line voltage $V_{SL,i}$. The logic output of the accumulator OUT_i at the generic row i is equal to the sum of the XNOR computations across bitcells:

$$OUT_i = \sum_{j=0}^{N-1} \overline{W_{ij} \oplus IN_j} \tag{2.14}$$

Based on this novel architecture [71], we decided to evaluate the proposed bit

cell considering the double barrier magnetic tunnel junction device (see Figure 2.8) by means of the device-to-system level simulation framework.

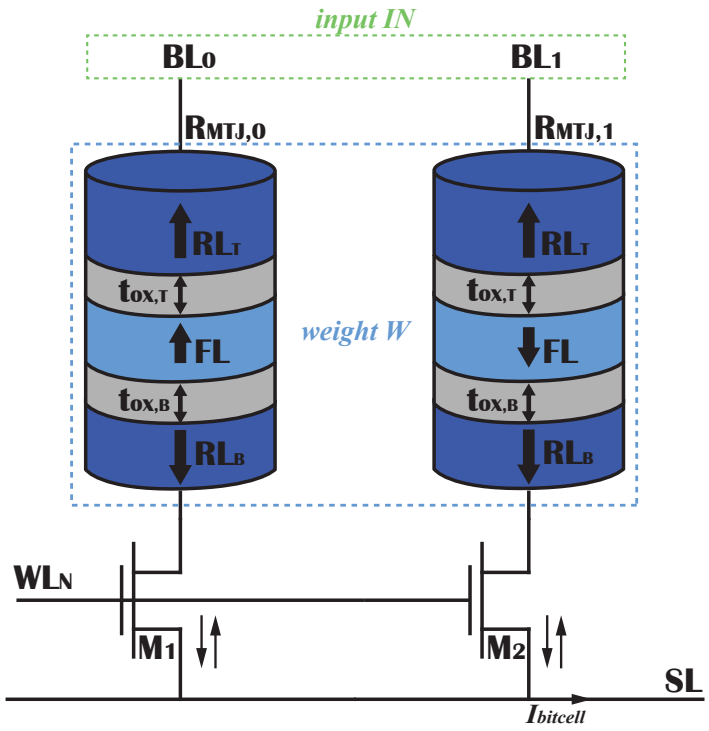


Figure 2.8: 2T-2DMTJ STT-MRAM bitcell for single-bit XNOR (modified proposal of [71]).

This simulation provides a validation for both Single and Double-barrier applied in XNOR cell. During the read operation, the WL is enabled, the SL is set to ground. The voltage at the bitline is then determined by the effective resistance of the bitcell, which is established by the MTJ state. In order to determine the state of the resistance, (i.e., high for AP state o low for P state), the senseamp is applied to compare the bitline voltage to an intermediate reference voltage V_{REF}.

The sensing margin at AP and P state, determine the robustness of the design. It is defines as the maximum deviation of the bitline voltage from the nominal value that can be tolerated (performed during the read operation) [104]. All simulation results reported below were obtained by means of electrical simulations within Cadence Virtuoso environment. The simulation include the effect of MTJ process

variations obtained from Monte Carlo simulations with 1000 runs. Based on the schemes shown in Figure 2.7 and Figure 2.8, the performed simulations include the MTJ device model developed by our group [74, 75] and 65-nm CMOS technology.

Accordingly, Figure 2.9 and Figure 2.10 shows the statistical distribution of sensing voltage in P and AP estate under process variations concerning to the read operation. The nominal Read Margin is defined as the difference between mean values of V_{SL} distributions for the states at P-AP and AP-P in each case (see Table 2.3). The extracted results indicate the variability of the bitcell (i.e., ratio of the standard variation (σ) and mean value (μ)). Besides, its reliability is determined by considering the BER, i.e., the failure probability in distinguishing one state from the other.

As a result, the corresponding estimated values for the RM, BER and reference voltage (V_{REF}) are presented in Figure 2.9 and Figure 2.10 regarding to 2T-2MTJ STT-MRAM bitcell when using SMTJ and DMTJ, respectively.

From Figure 2.9(a) and Figure 2.10(a), the nominal Read Margin obtained for Single and Double-barrier MTJ device results of about 95.9 mV and 78 mV, respectively. Likewise the corresponding Read Margin determined at 3σ are 60.35 mV and 48.2 mV.

Furthermore, Equation 2.15 and Equation 2.16 display the calculation of BER set up using the Cumulative Distribution Function (CDF_{normal}) and the sense variability expressed as the ratio of the standard deviation and the mean value during each configuration state AP_{MTJO}, P_{MTJ1} or P_{MTJO}, AP_{MTJ1}.

$$BER_{AP,P} = 1 + CDF_{normal}\left(-\frac{1}{\frac{\sigma_{AP,P}}{\mu_{AP,P}}}\right) \tag{2.15}$$

$$BER_{P,AP} = CDF_{normal}\left(\frac{1}{\frac{\sigma_{P,AP}}{\mu_{P,AP}}}\right) \tag{2.16}$$

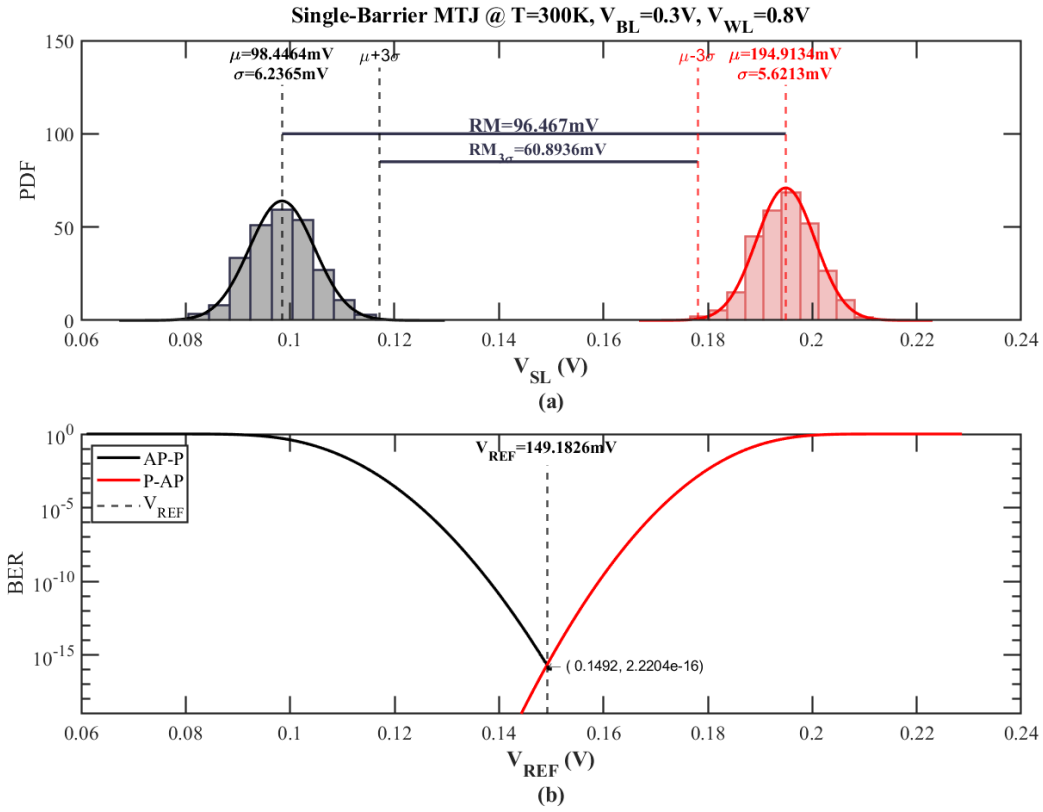


Figure 2.9: Statistical Distribution with SMTJ device.

The overall BER due to inadequate sensing margin is defined by the worst-case value (i.e., the largest) amongst $BER_{AP,P}$ and $BER_{P,AP}$ in Equation 2.15,Equation 2.16 [104].

After all, the minimum error rate is achieved when V_{REF} is selected once $BER_{AP,P}$ and $BER_{P,AP}$ are matched. The optimal value is closest to the state where its standard deviation is smaller compared to the other configuration state. Accordingly, the optimum V_{REF} is calculated as follows.

$$V_{REF,opt} = \mu_{AP,P} + \sigma_{AP,P} \cdot \frac{\mu_{P,AP} - \mu_{AP,P}}{\sigma_{AP,P} + \sigma_{P,AP}} \tag{2.17}$$

Then, the expression to obtain the optimum overall sensing margin variability is presented in Equation 2.18.

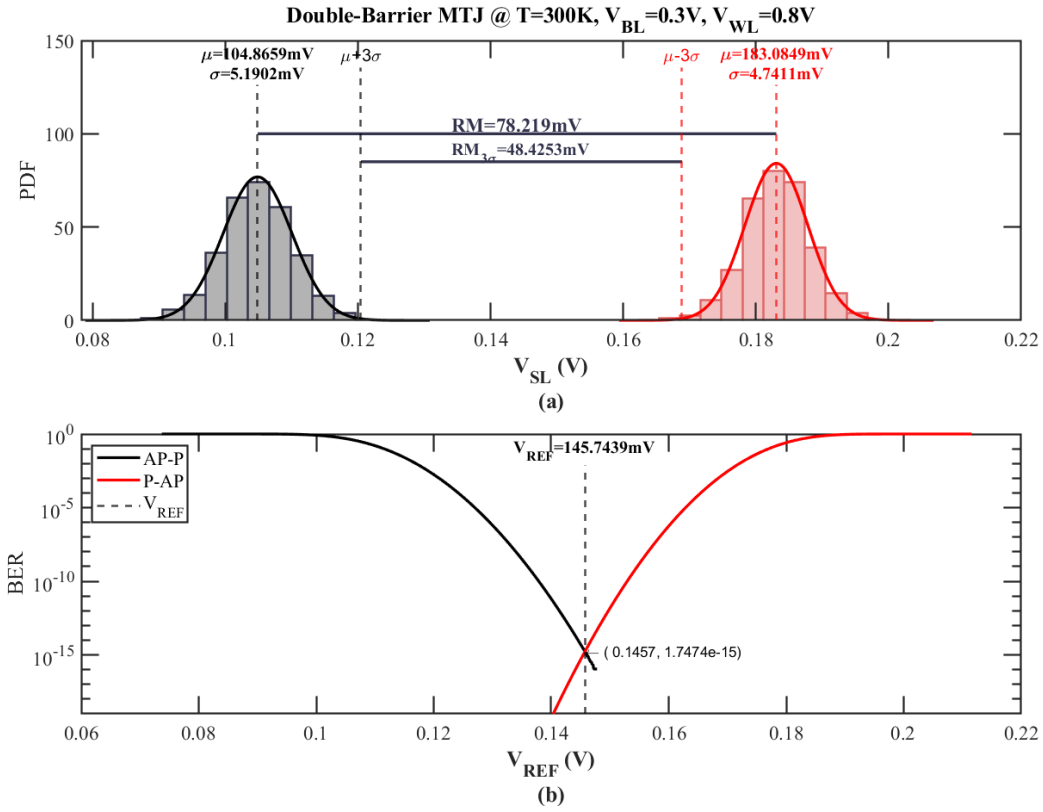


Figure 2.10: Statistical Distribution with Double-Barrier MTJ device.

$$\left(\frac{\sigma}{\mu}\right)_{opt} = \frac{\sigma_{AP,P} + \sigma_{P,AP}}{\mu_{P,AP} - \mu_{AP,P}} \tag{2.18}$$

Correspondingly, from Figure 2.9(b), it shows the sensing margin variability for $AP_{MTJO} - P_{MTJ1}$ and $P_{MTJO} - AP_{MTJ1}$ state versus V_{REF} under $V_{WL} = 0.3V$, $\sigma_{AP,P} = 6.2\,\mathrm{mV}$, $\sigma_{P,AP} = 5.65\,\mathrm{mV}$ and considering $\mu_{AP,P} = 98.44\,\mathrm{mV}$ and $\mu_{P,AP} = 194.91\,\mathrm{mV}$. As a consequence the optimal V_{REF} that minimizes BER results to 149.18 mV and the corresponding variability of sensing margin is 0.122.

Likewise, from Figure 2.10(b), the mean values are $\mu_{AP,P} = 104.86\,\mathrm{mV}$ and $\mu_{P,AP} = 183.085\,\mathrm{mV}$, respectively. Hence, the optimal V_{REF} results to 145.74 mV, leading a BER of 1.747e-15. The optimal variability of sensing margin is 0.127. Therefore, from the above results we can observe that the variability sensing margin

is sensible to small changes in V_{REF} producing BER degradation.

Table 2.4: COMPARISON SMTJ AND DMTJ WITH THE REFERENCE

	Unit	SMTJ		DMTJ		Comparison SMT vs DMTJ	Reference [104]	
		AP,P	P,AP	AP,P	P,AP	%	AP,P	P,AP
μ	mV	98.45	194.91	104.86	183.08	-	85	165
σ	mV	6.24	5.62	5.19	4.74	-	3.6	8.5
RM	mV	96.467		78.219		23.32	80	
BER	-	2.22E-16		1.747E-15		-87.29	$\sim$1E-10	
V_{REF}	mV	149.18		145.74		2.36	-	
σ/μ	-	0.122		0.127		-3.93	0.152	

Regardless the results when performing the comparative study between SMTJ- and DMTJ-based solutions (see Table 2.4) the SMTJ-based cell shows an improvement in terms of Read Margin of 23.32% , a low BER under read operation (-87.29%) and a standard deviation and mean value ratio of less than 4%, compared with the DMTJ-based cell.

As a result, the cell implemented with the compact models developed by our working group is an excellent option to use in-memory computing for BNN. Our results display a better behaviour at bit-cell level, compared to the studies carried out in reference [104].

2.1.4 Algorithm-Level

Background of Artificial Neural Networks

ANN have been based on the human brain and its biological neural networks, it consists of a network of interconnected units or nodes, also known as artificial neurons, which replicates the behavior of neurons in a biological neural network. These neurons receive signals from other neurons and produce an output signal using an activation function (AF), see Figure 2.11. The connections between the neurons in an ANN mimic the synapses in biological neural networks. Each

connection in an ANN is assigned a specific weight that represents its relative strength, the weights associated with these connections have to be properly adjusted to improve the accuracy of the network.

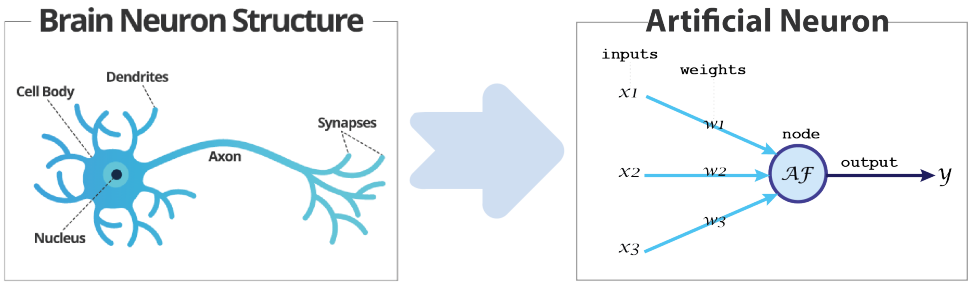


Figure 2.11: Model of a biological neuron and model of an artificial neuron. (As the brain is composed of connections of numerous neurons, the neural network is constructed with connections of nodes, which are elements that correspond to the neurons of the brain [107]).

The architecture of a neural network is structured by multiple interconnected artificial neurons, called units or nodes, into a sequence of layers. Each single neuron as shown in Figure 2.12 contains the synapses, which are the connections between neurons and a weight associated with them. Likewise, the weights can be adjusted through a process called training to improve the accuracy of the network.

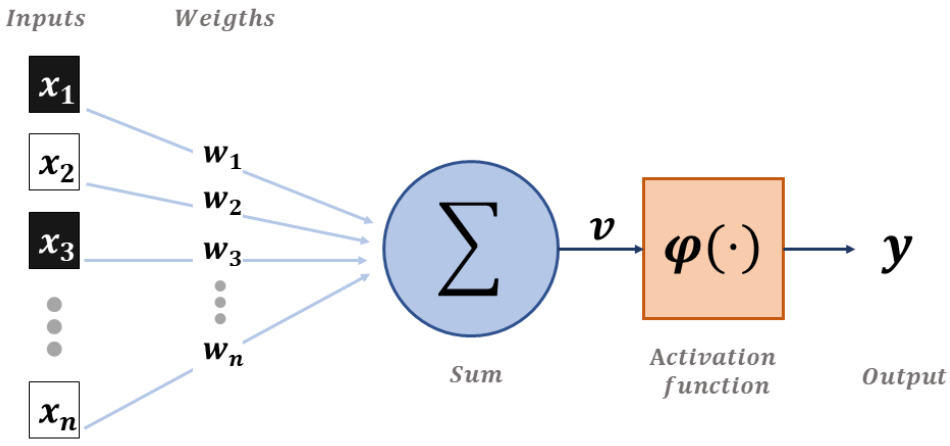


Figure 2.12: Representation of a single neuron.

The input signals x_i from the outside is multiplied by the associated weight w_i

before it reaches the node. Equation 2.19 shows the expression of the weighted sum of the input signals (the expression can even be written using matrices).

$$v = (w_1 \times x_1) + (w_2 \times x_2) + ... + (w_n \times x_n) \tag{2.19}$$

Once the weighted signals are collected at the node, these values are added to be the weighted sum. After summation the neural node applies the weighted sum into the activation function φ resulting in output *y*, see Equation 2.20.

$$y = \varphi(v) = \varphi\left(\sum_{i=1}^{n} (w_i \cdot x_i)\right) \tag{2.20}$$

Finally, the node enters the weighted sum into the activation function and yields its output. The activation function validates the behavior of the node and determines when the neurons are activated or not.

Most neural networks are constructed with the layered nodes. The layers in between the input and output layers are called hidden layers. For the layered neural network, the signal enters through the input layer, passes through the hidden layer, and exits through the output layer. A multi-layer neural network that contains two or more hidden layers is called a DNN, as depicted in Figure 2.13.

In Figure 2.13, the square nodes represent the input data, for instance, when using MNIST dataset, it contains handwritten black and white images and each pixel represents one input. The input layer receives the input data and passes it on to the next layer, the hidden layer performs the computation required for the network and the output layer predict the output data.

The neural network has undergone significant development in architecture, from a simple structure to a more complex one. Initially, neural network pioneers employed a basic layout enclosing only input and output layers, resulting in a single-layer neural network. A neural network that has a single hidden layer is

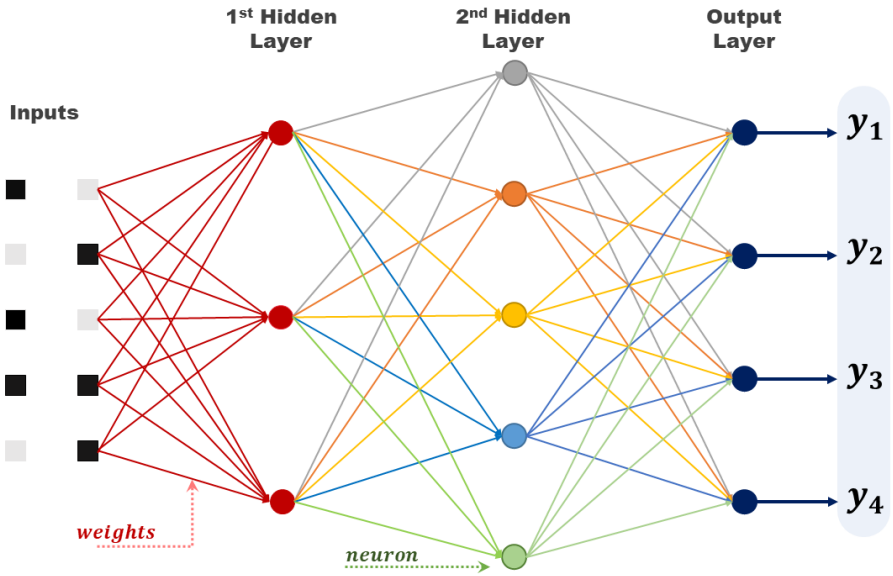


Figure 2.13: A layered structure of nodes .

referred to as a shallow neural network or a vanilla neural network. The addition of hidden layers to the single-layer network produces a multi-layer neural network, known as a deep neural network. In practical applications, deep neural networks are more commonly employed than shallow ones [107].

ANN are able to learn how to perform tasks without being explicitly programmed with task-specific rules (e.g. it determines patterns and make predictions based on the input data). Hence, ANNs are typically ”trained” on a set of data using a process called ”supervised learning”. For the neural network, supervised learning implements the process to adjust the weights to reduce the discrepancies between the correct output and output of the neural network [107].

Most applications in an artificial neural networks often requires several hundred neurons, and the number of weights are proportional to the square of the number of neurons. An essential aspect of implementing neural hardware is the mechanism used to represent and manipulate the weights. In ANN, online learning and offline learning, also known as batch learning, represent two distinct approaches to training

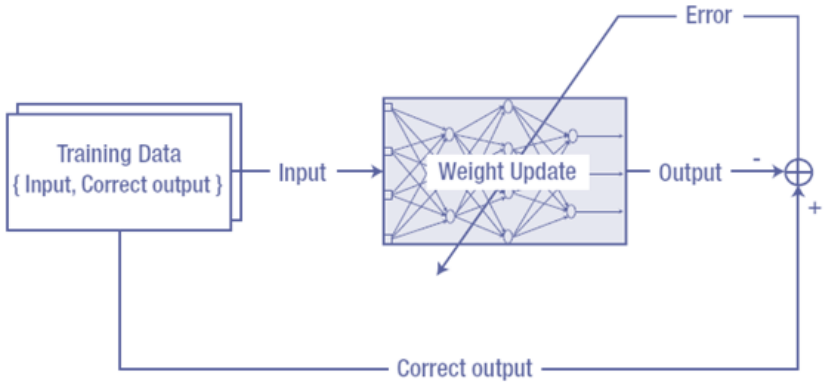


Figure 2.14: Training process for supervised learning [107]).

the network. Online learning updates the model's parameters after each individual training example, making adjustments in real-time as data is processed. In contrast, batch learning, updates the parameters after processing the entire training dataset.

Online learning is indeed more challenging to implement compared to straightforward weight adjustment. It is considered a vital component for the majority of neural network applications.

Although some approaches rely on off-line learning to train artificial neural systems, being able to learn on the chip as more data becomes available to the system is an invaluable feature. This is because it enables the model to adapt and learn dynamically from new data, making it a crucial ingredient for successful deployment in real-world scenarios. However, creating an artificial neural-system chip with on-chip, on-line learning capability is a challenging task. Even for limited learning capacity, the network design becomes highly intricate.

Traditional implementations of ANNs using Von Neumann machines are limited by the separation of computing and memory.

In Von Neumann architecture, which is the basis for most modern computers, there are separate units for processing data (CPU) and storing data (memory). When the computation is performed, data must be moved back and forth between the

processor and memory. For ANNs, which involve a large number of computations on vast datasets, this constant movement of data can be a significant bottleneck. This back-and-forth data transfer consumes time and energy, limiting the speed and efficiency of ANN computations.

Therefore, hardware solutions have been developed to mitigate this bottleneck and improve the efficiency of ANNs. Different design solutions can be utilized to build these hardware ANNs, ranging from traditional digital circuits to more recent compute-in-memory or neuromorphic computing approaches. Analog and digital approaches have been proposed for the implementation regarding ANNs, analog processing seems much more efficient than purely digital computation since response times for the inherently parallel analog hardware are significantly smaller than for digital hardware. Still, some efficient implementations of neural processors can be made as combinations of analog and digital hardware. In this context, memory technologies and their integration with computing elements in hardware implementations of ANNs have recently garnered significant attention. This integration could potentially overcome the Von Neumann bottleneck and lead to significant improvements in the performance and efficiency of ANNs.

Overall, ANNs have become an effective and mandatory tool for applications which include image recognition [108], speech recognition, object detention [109], pattern recognition, natural language processing and predictive analysis. [110–112]. Nowadays, ANNs are mostly used for universal function approximation in numerical paradigms because of their excellent properties of self-learning, adaptivity, fault tolerance, nonlinearity, and advancement in input to an output mapping [113]. Despite the extensive applications of ANNs, there is an increasing need to address the problem of adopting a systematic approach in ANNs development phase to improve its performance.

2.2 Overview of the Simulation Environment

This research activity focuses on integrating this emerging memory device, such as spintronic memories, with CMOS technology to leverage their inherent capability of compute-in-memory. We use a specific environment for circuit simulation.

We utilize the Cadence Virtuoso commercial circuit simulator to design circuits that employ MOS technologies. This allows us to generate netlists using Verilog-A models, which in turn helps us solve electrical property equations. To incorporate the behavior of these devices into our circuit designs, we rely on the use and calibration of Verilog-A based compact models. The simulation environment provided by Cadence Virtuoso enables us to create test benches, providing detailed information on circuit performance.

Moreover, we used the emulator NeuroSim to benchmark synaptic devices and array architectures in terms of system-level learning accuracy and hardware performance metrics. NeuroSim is an integrated simulation framework used to support a 2-layer MLP neural network to benchmark the DNN architecture, relied to digital synapse devices, in online learning and offline classification with MNIST handwritten dataset. The simulator was performed by Pai-Yu Chen, Xiaochen Peng and Yandong Luo and developed in C++ to emulate the online learning/offline classification scenario with MNIST handwritten dataset in a 2-layer multilayer perceptron (MLP) neural network based on SRAM, eNVM and FeFET array architectures [114, 115].

For benchmarking neuro-inspired architectures, NeuroSim used to assessment area, latency, dynamic energy, and leakage power of neuromorphic hardware accelerators to simplify the design space exploration. [115]

The target for this simulator is to estimate the system-level performance using the user-derived analog synaptic device data. The parameters are adjusted at device,

circuit and algorithm level. Figure 2.15 shows an overview of NeuroSim framework from Device to Algorithm-level. The input MNIST images are cropped and encoded into black/white data for simplification and the weights input-hidden and hidden-output are mapped to synaptic cores.

The MNIST dataset is a widely used database of handwritten digits ranging from the numbers 0 to 9. It consists of 60,000 training images and 10,000 test images. The dataset is commonly used for image classification tasks and is often used for training deep learning models, such as convolutional neural networks (CNNs) and MLPs [116].

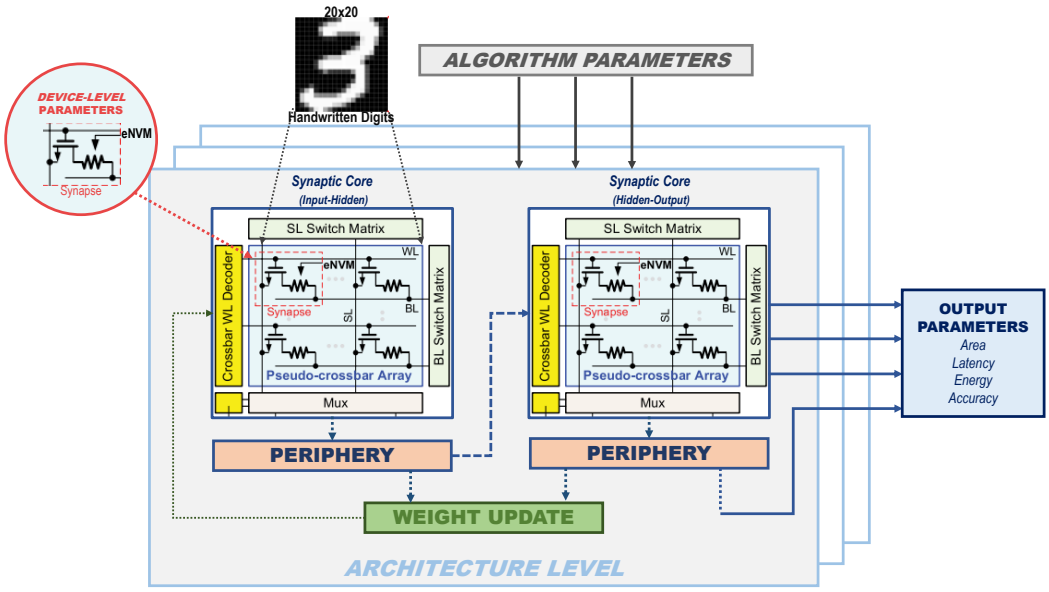


Figure 2.15: NeuroSim framework.

At the circuit level, several design options are available, such as the analog synaptic array architecture (eNVM crossbar, eNVM pseudo-crossbar or FeFET), or digital synaptic array architecture (eNVM crossbar, eNVM 1T1R, or SRAM). At the algorithm level, a simple 2-layer MLP neural network is provided for evaluation, thus only limited options are available to the users to modify, such as the size of

each layer and the size of weight matrices.

In this thesis, a variety of software tools were utilized including MATLAB, Cadence-Virtuoso, and the NeuroSim emulator. The Cadence-Virtuoso environment is leveraged for circuit-level design, while the NeuroSim+ emulator is used to support the MLP neural network. The objective is to propose techniques and designs to improve the performance of the circuits and evaluate their efficacy through the implementation of digital non-volatile memory technology.

Having described the simulators with its design tools details, Figure 2.16 illustrates the simulation environment used to evaluate the performance of each stage.

The design process begins at the device level and the performance varies depending on the category of proposal chosen. In certain instances, Cadence-Virtuoso is employed to develop Verilog-A code, such as in the case of SMTJ and DMTJ devices. After creating the schematic design at the circuit level, simulations are performed considering test criteria (i.e optimization or integration). To ensure the accuracy of Monte Carlo simulations, it is crucial to take into account the impact of the standard deviation, as it plays a significant role in determining the dependency on the Bit Error Rate. It is a key parameter used to measure the performance of a data.

The NeuroSim+ emulator approaches simulation stages differently, considering device, cell design, and network training parameters. This allows for customization of each design to better approximate the real architecture. Once each parameter is configured, the relevant data can be extracted. During the simulation, execution time is critical as it is influenced by the number of training and test images, as well as the number of epochs utilized.

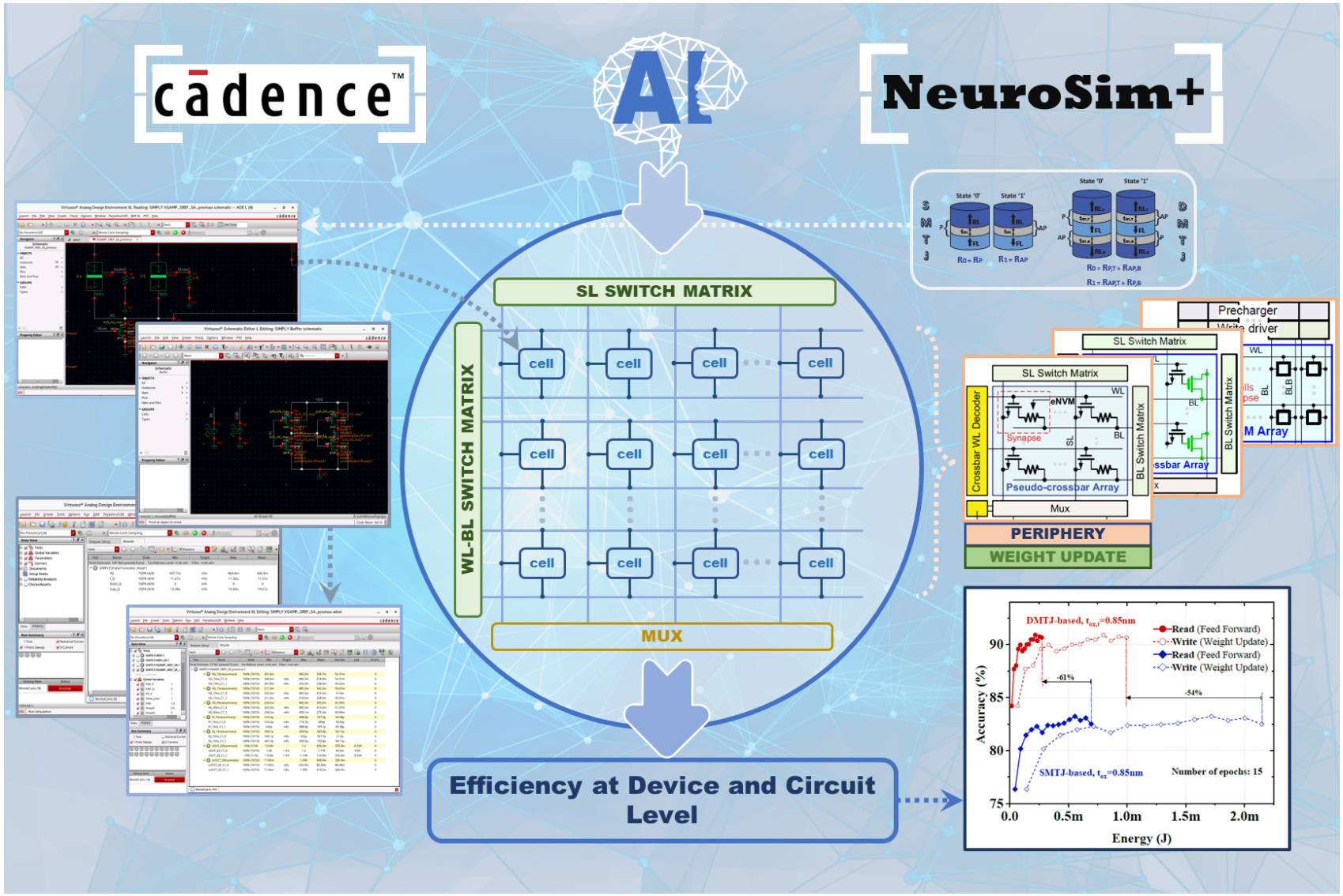


Figure 2.16: Schematic of the Simulation Framework using Cadence-Virtuoso and NeuroSim+ emulator

CHAPTER 3

ACTIVATION FUNCTIONS FOR ANN

ANNs are designed to process information, recognize patterns, and make decisions in a way that is similar to the human brain. ANNs consist of multiple layers of interconnected nodes or ”neurons”, and the strength of these connections (i.e., weights) is adjusted through a process known as training, which allows the network to improve its accuracy over time. The neurons work all together to perform complex tasks. Activation functions are used to calculate the output of the ANN once processed the input signals with their corresponding weights.

This chapter provides a comprehensive study on the AF used in ANN to transform input signals into and output signal. Then the Sigmoid and Softmax AF circuit implementation for analog Neural networks are discussed.

3.1 State of Art

As we mention in the previous chapter, the architecture of a neural network is structured with interconnected neurons called nodes. Each neuron in a layer receives inputs from the neurons in the previous layer and produces an output that is passed on to the next layer. For each layer, input data are first processed by a linear vector-matrix multiplier (VMM), then they pass through a nonlinear AF, which emulates the behavior of a biological neuron [117].

An activation function is a mathematical function that is utilized within artificial neural networks to introduce nonlinearity into the output of a neuron. Its significance lies in determining the power and the capabilities of the neural network, bearing in

mind that the accuracy of predictions is primarily dependent on the selection of the activation function.

The hardware implementation of ANN is a key area of research [118], with a focus on developing efficient and effective computational methods.

The main challenges regarding this field is the realization of activation function of neurons, which is a critical component of ANNs because it introduces non-linearity into the model. Activation function is essential for the network to effectively model and learn from complex data.

Some widely used activation functions include the sigmoid function [3, 118, 119], Softmax function [1, 120, 121], hyperbolic tangent function[122–124], and rectified linear unit (ReLU) function [125], among others, as depicted in Figure 3.1. Each activation function has its own strengths and weaknesses, and the selection of an AF often depends on the specific task and the architecture of the neural network.

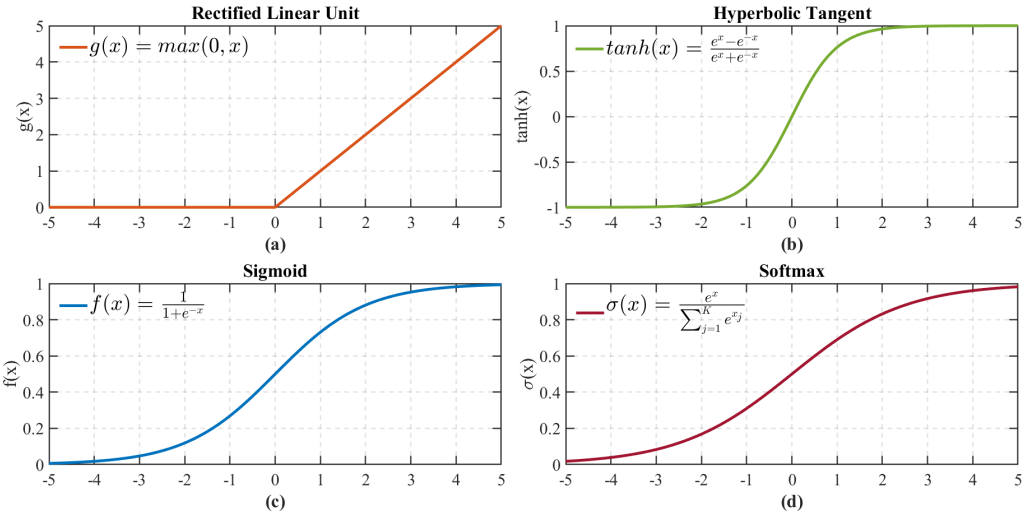


Figure 3.1: Types of activation functions; Sigmoid, Softmax, Hyperbolic tangent and ReLU.

3.2 Overview of Activation Function Types

The ReLU (rectified linear unit) activation function, see Figure 3.1 (a), is a popular non-linear function used in neural networks. It is considered more efficient than other activation functions because it only activates a certain number of neurons at a given time, unlike other functions where all neurons may be activated simultaneously. Specifically, ReLU sets all negative inputs to zero, while retaining all positive inputs. This allows neural networks using ReLU to learn faster and avoid the vanishing gradient problem that can occur with other activation functions.

The Tanh (hyperbolic tangent) function, see Figure 3.1 (b), has similarities with the sigmoid function. However, Tanh is symmetric about the origin, and this results in negative inputs being mapped to strong negative values, while zero inputs are mapped near zero in the Tanh graph. Both the Tanh and Sigmoid functions are used in feed-forward neural networks, but Tanh is preferred over Sigmoid because it enables the calculus of a gradients that are not restricted to vary in a certain direction, and its output is centered around zero. Additionally, the gradient of the Tanh function is steeper than the Sigmoid function.

The sigmoid function, see Figure 3.1 (c), is a non-linear, bounded function that maps a real-valued input to an output between 0 and 1. It is widely used in artificial neural networks for binary classification in logistic regression models, where the output represents a decision between two options. One of the limitations of the sigmoid function is that the probabilities of its outputs do not necessarily add up to 1. Despite this, sigmoid remains a popular activation function in neural networks due to its simple structure and ease of use.

Unlike sigmoid, Softmax function, see Figure 3.1 (d), is an unbounded and non-linear mathematical function that maps a real-valued input to an output between 0 and 1, with the property that the outputs for each input vector sum to 1. The

Softmax activation function is commonly used in artificial neural networks, convolutional neural networks (CNN) and reinforcement learning models to provide classification outputs. It is commonly used in logistic regression models for multi-class classification tasks such as image recognition and natural language processing (NLP).

Sigmoid and Softmax are two commonly used activation functions in ANNs. Both functions introduce nonlinearity to the model, allowing it to capture more complex relationships between input and output. Table 3.1, shows some relevant characteristics between sigmoid and Softmax AF.

Table 3.1: DIFFERENCE BETWEEN SIGMOID AND SOFTMAX AF.

Sigmoid function	**Softmax function**
Often used in the hidden layers	Regularly used in the output layer
Used when the output of the neural network is continuous.	Used when the output of the neural network is categorical.
Bounded function that maps a real-valued input to an output in between 0 and 1.	Unbounded function that maps a real-valued input to an output in between 0 and 1 that sums to 1 for each input vector.

The selection of activation functions in neural networks is a pivotal design consideration, which depends on the specific tasks. Different activation functions introduce varying levels of non-linearity, influencing the model's capacity to learn complex relationships in data.

3.2.1 Analog Softmax AF Design

Among the AF implementations, the Softmax function - normalizing with respect to all input signals of the output layer - is frequently utilized to simulate the neuron output in multi-class problems. By assigning probabilities to each class, the Softmax is a sigmoid function normalized, combining the input signals of other neurons belonging to the same level with its corresponding input to drive output.

While an overwhelming majority of implementations have been in the digital implementations [126–129], analog circuit-oriented proposals for Softmax are especially thin on the ground, with only two references available [121, 130].

In this work, we collaborated with the proposal of a low-power analog current-mode Softmax topology, where both transfer-function slope and amplitude can be dynamically adjusted. This circuit is composed of three stages: the first implements a linear current–voltage conversion of the input signal, the second performs the exponential function of the signal coming from the first stage, and the third one acts as an analog divider.

The topology can also operate with voltage-mode inputs by using only the second and the third stages. Simulation results demonstrated that the proposed topology features a good match to the theoretical Softmax, a low voltage operation and a low power dissipation, and a strong robustness against PVT variations, exploiting the variability of the slope and of the amplitude of the transfer function [1].

An M-sized Softmax function, also known as normalized exponential function, composed by an array of M elements performing the normalization to the (0:1) interval of an array of M real-number input signals (i.e., the outputs of the multiply-and-accumulate operations). The analytical expression of the Softmax is given in Equation 3.1, which shows that the probability associated with each i-th class is proportional to the exponential of the corresponding x_i, and normalized by the sum of the exponentials performed on each input:

$$f(x) = \frac{e^{\alpha x_1}}{\sum_{k=1}^{M} e^{\alpha x_k}} \tag{3.1}$$

The proposal pretend to develop an analog circuit to imitate the Softmax Activation function utilizing the device physics of MOSFET by exploiting exponential function, sum, and division. Figure 3.2 represents a block-level

representation of the Softmax circuit. The proposed circuit is implemented in a modular fashion, being composed of three building blocks, which can be replicated and shared, to achieve a Softmax function with an arbitrary number of inputs and outputs. The first stages linearly convert the input current signals to voltage signals, the second stages implement a voltage-to-current exponential conversion, and the last stage realizes the analog division [1].

Therefore, Figure 3.3 showcases the transistor-level schematic of the current-voltage conversion and exponential blocks (a) and analog divider (b).

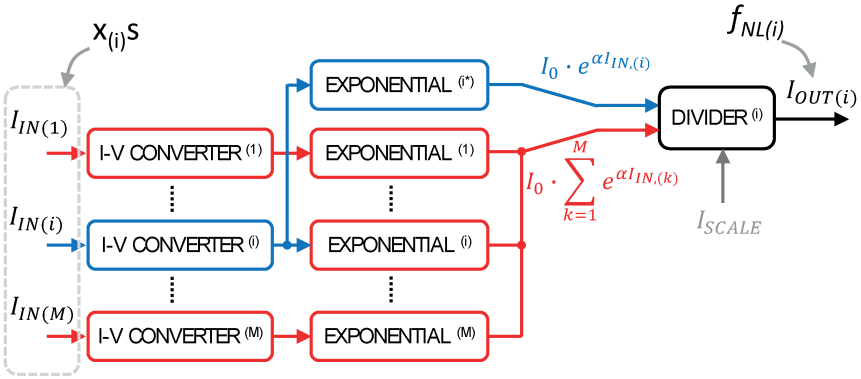


Figure 3.2: Softmax diagram, composed of M conversion blocks, M + 1 exponentials, and one analog divider. Exponential blocks and the analog divider must be replicated to produce the other outputs [1].

Analog Softmax Circuit Design and Performance

The proposed Softmax circuit was designed and simulated with the 180 nm TSMC technology node using a supply voltage (V_{DD}) of 500 mV. The nominal full-scale output current of 10 nA corresponds to the 1 output of the Softmax operation (i.e., 100% probability). Softmax transfer characteristics were measured by sweeping just one normalized input from -5 to 5. The input scale was normalized to get a nominal slope α equal to 1 for an easy comparison with the theoretical equation (see Equation 3.1).

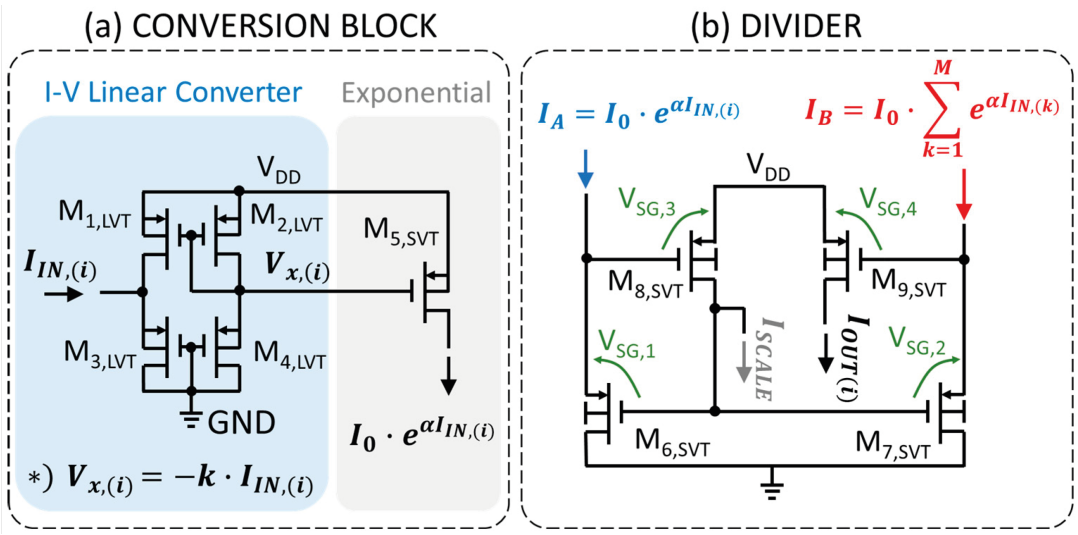


Figure 3.3: Transistor-level schematics of the (a) input conversion block (current-to-voltage linear conversion and exponential conversion) and (b) the analog divider block [1].

From Figure 3.3(b) it is possible to express the I_{OUT} as following:

$$I_{OUT} = I_{SCALE} \cdot \frac{I_A}{I_B} = I_{SCALE} \cdot \frac{I_{EXP(1)}}{\sum_{k=1}^{M} I_{EXP(k)}} \tag{3.2}$$

I_{SCALE} is set to a fixed value, since it represents the Softmax amplitude.

Figure 3.4(a) illustrates the proposed circuit implementation compared with the theoretical Softmax model considering the input range of the transfer characteristics into three regions: in regions I and III showing exponential approximations, whereas in region II, it demonstrates nearly linear behavior.

Figure 3.4(b) displays the relative error, which is the deviation between the transfer characteristics of the circuit and the theoretical function. The proposal shows a peak error of 2.2% in region II, which can be ascribed to an input offset, and an average value of $\sim$1.4%.

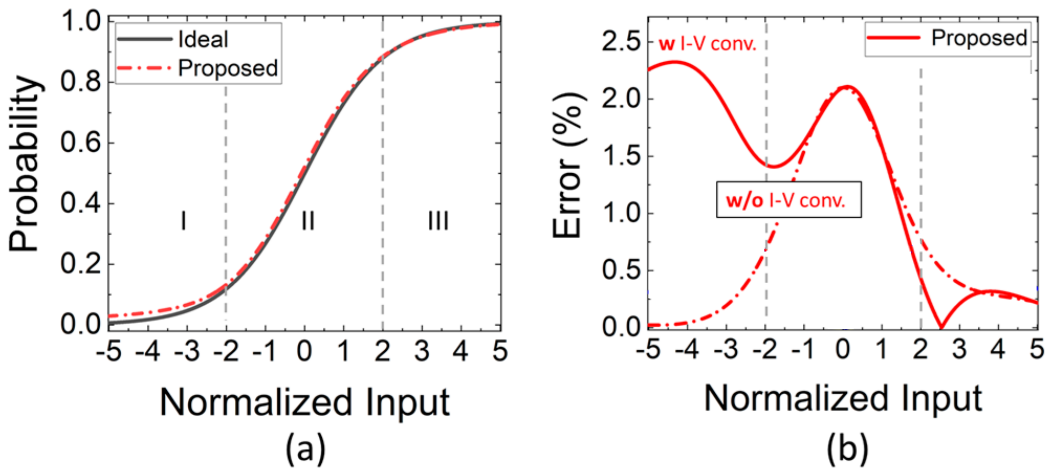


Figure 3.4: (a) Proposed Softmax design simulated transfer function and theoretical analytical model (M = 2). The simulated input signals have been arbitrarily normalized to get a Softmax slope $\alpha = 1$, while the output has been normalized to the output full scale (10 nA). (b) Relative error of the proposed Softmax[1].

Mismatch and Process Variations

Figure 3.5 displays the circuit behavior regarding mismatch and process variations, where transfer characteristics were computed for 1000 statistical Monte Carlo runs.

In Figure 3.5(a), the impact of mismatch variations is mainly related to the deviation of the characteristic amplitude. The maximum standard deviation of the output current variation is 2.97% with respect to the mean value. On the other hand, the process variation behavior is depicted in Figure 3.5(b), it leads mainly to a variation of the slope. Specifically, the ratio between standard deviation and mean value is about of 16.83%, with a negligible variation of the amplitude.

Impact of the Technology Node Scaling

The process of technology node scaling involves of reducing the size of transistors and other components on integrated circuits to enhance performance and reduce costs. In this regard, the following simulations analyze the impact of scaling on the

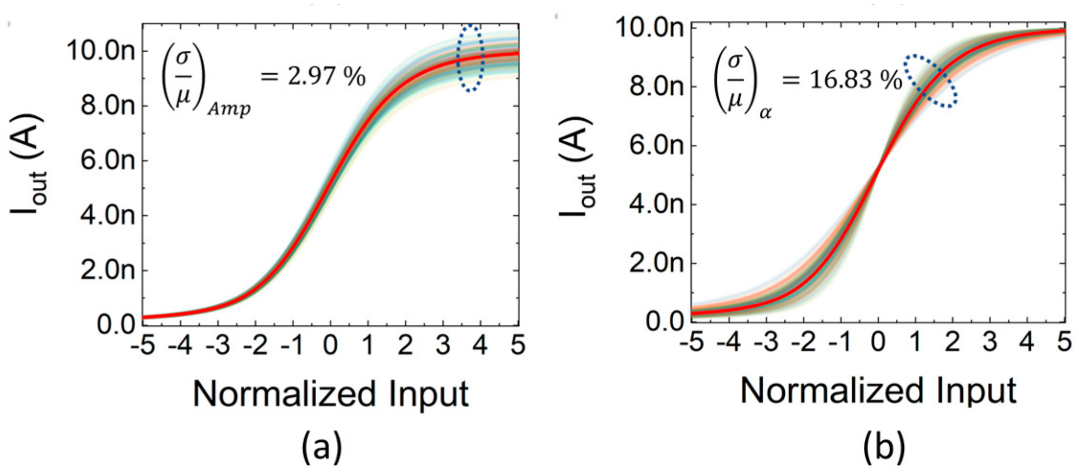


Figure 3.5: Impact of (a) mismatch and of (b) process variations on the Softmax transfer characteristics for 100 MC runs. [1].

proposed Softmax implementation. Figure 3.6a depicts the Softmax activation function characteristic at three distinct technology nodes, namely TSMC 180 nm, 65 nm, and 40 nm. On the other hand, Figure 3.6 illustrates the corresponding error concerning the theoretical equation.

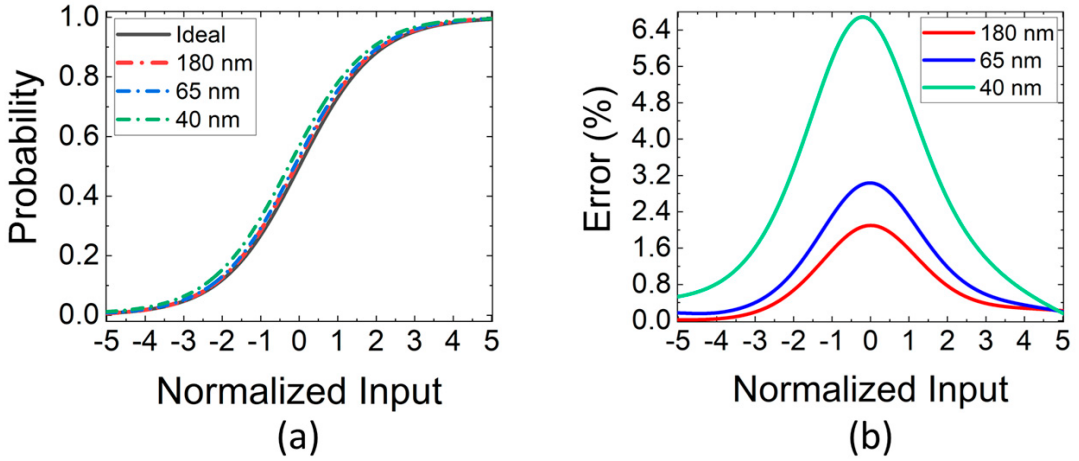


Figure 3.6: (a)Transfer characteristic and corresponding relative error (b) for three technology node (180 nm, 65 nm, 40 nm) Softmax circuits. [1].

After all, we can observe that as the design implementation decrease in technological node, the absolute error increases. This is attributed to an increased

offset resulting from the adjustment of I_{SCALE} to match the upper part.

Moreover, Figure 3.7 displays the area estimation of the proposed design considering two blocks of conversion and one of division, giving a result of about 134 μm^2.

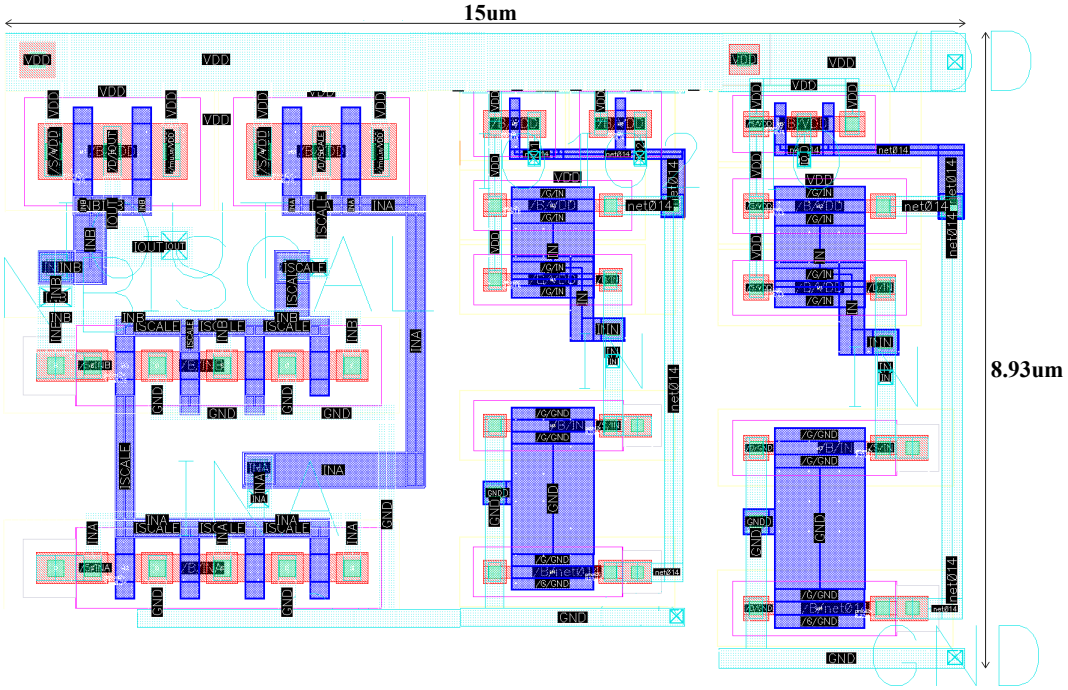


Figure 3.7: Layout schema for Softmax AF at 180 nm technology node.

Finally, a novel analog implementation of the Softmax activation function is presented, and the main features of the circuit are the good match to the theoretical function. In addition the impact of mismatch and process variation are quite acceptable with a ratio standard deviation and mean value of about 16.83%. These improvements are achieved with limited precision degradation, considering and average relative errors respect to the theoretical Softmax equation,of 0.9% only.

3.2.2 Analog Sigmoid AF Design for ANN

The development of efficient and accurate methods for realizing activation functions in hardware is essential for the widespread adoption of ANNs in various fields such

as image processing, speech recognition, natural language processing, predictive analytics, robotics, etc [131–135].

Hardware implementations of neural networks can use both digital and analog modules to realize activation functions, depending on the specific neural network being used. While digital implementations are more flexible and easier to scale, analog implementations can offer improved power efficiency and potentially higher performance. Neural networks can be generally categorized into three groups: digital [136–139], analog [3, 118, 119, 121, 140], and hybrid (mixed-signal) [20, 141–143] neural networks.

Furtherance in CMOS technology has played a crucial role in addressing the growing need for energy-efficient high-performance computing solutions. As the demand for processing power and data storage grows, energy consumption has become an essential factor in the design and operating computing systems. CMOS technology has allowed for greater integration density and power efficiency within integrated circuits. Improved analog design for hardware implementations have enabled greater advantage of transistors' unique features. This has led to the development of more efficient and effective electronic systems.

Proposed Sigmoid AF

The proposed neuron circuit is based on the Resistive-Type Sigmoidal Neuron introduced in [118]. Sigmoid function is a nonlinear function which determines the output of a neuron based on the weights of its inputs by converting input values to probability-like outputs in the range of 0 and 1. The sigmoid function is a common S-shaped curve defined as follows:

$$y = \varphi(v) = \frac{1}{1 + e^{-\alpha v}} \tag{3.3}$$

where α is the steepness of the slope in the linear region $(v \approx 0)$.

The Sigmoid circuit in this reference uses transistors operating both in triode and saturation regions, and converts the total input summation current into a voltage at the output node. The sigmoid function is designed by using a nonlinear resistive load that takes current as input and delivers a voltage at the output of the neuron.

In order to upgrade the design developed in [118], we add a pseudo-differential input I- V conversion stage to enable a voltage signal as input data to generate the sigmoid function by using 180nm technology which is more affordable for fabrication.

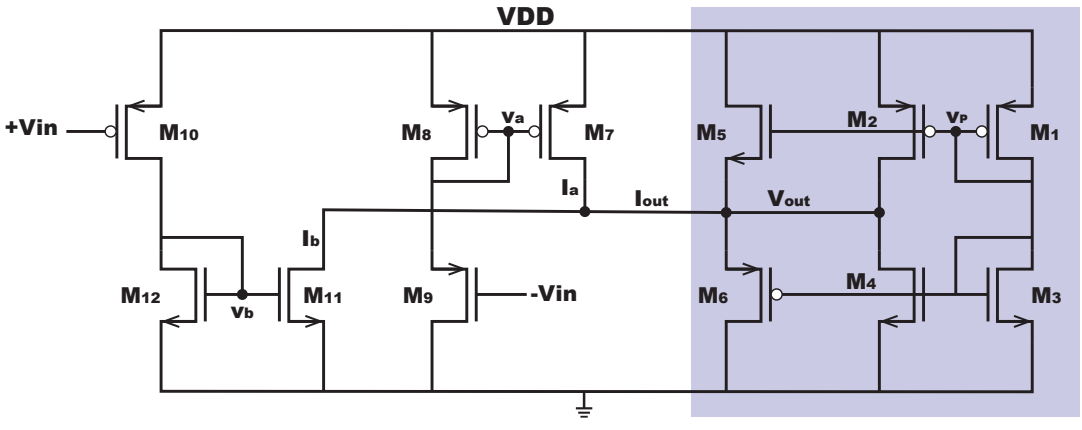


Figure 3.8: Schematic of the proposed sigmoidal neuron.

Figure 3.8 shows the circuit implementing the sigmoid activation function. The circuit consists of two parts: the shaded area, on the right, represents the sigmoid function generator of [118] having the current I_{tot} as input and producing the sigmoid function as V_{out}. The un shaded area represents the V to I converter enabling AF generator to take V_{in} as input and produce I_{tot} to drive the right-hand circuit.

Transistors M1 - M6 represent the core from referenced design, M1 and M3 are sized to generate the biasing voltage of $VP \approx V_{DD}/2$ and M2-M4-M5-M6 generate the sigmoid function by controlling the gain in linear operation and therefore the steepness of the curve i.e. the α parameter in Equation 3.3.

All the transistors of the circuit are biased using only a single supply voltage V_{DD}. By operating the MOSFETs both in triode and saturation regions, the core of the neuron circuit can provide an accurate approximation of the sigmoid function.

The operation of the right-hand is designed such that when $I_{tot} = 0$, $V_{out} = V_{DD}/2$. When I_{tot} is negative, transistors M6 and M4 are in off and deep triode region $(V_{DS4} \sim 0)$, respectively, while the other two transistors M5 and M2 are in saturation region, causing V_{out} to be close to ground, (Equation 3.4). As I_{tot} increases towards 0, i.e. becomes less negative, transistors M4 gets into saturation having its V_{DS} and therefore also V_{out} increase; M2 stays in saturation and the other two transistors M5-M6 are off, causing V_{out} to increase towards $V_{DD}/2$, (Equation 3.5). As I_{tot} increases and becomes positive, transistor M5 and M2 eventually are in off and deep triode region $(V_{DS2} \sim 0)$, respectively, connecting V_{out} to V_{DD}, while M6 and M4 are in saturation, (Equation 3.6) [118]. Current in each state is approximated as follows:

$$\begin{aligned} I_{tot} = &-\frac{1}{2}k_n S_5 \left(V_P - V_{out} - V_{th}\right)^2 - \frac{1}{2}k_p S_5 \left(V_{DD} - V_P - V_{th}\right)^2 \\ &+ k_n S_4 \left[(V_P - V_{th})V_{out} - \frac{1}{2}V_{out}\right]^2 \end{aligned} \tag{3.4}$$

$$I_{tot} = -\frac{1}{2}k_p S_2 \left(V_{DD} - V_P - V_{th}\right)^2 + \frac{1}{2}k_n S_4 \left(V_P - V_{th}\right)^2 \tag{3.5}$$

$$\begin{aligned} I_{tot} = &-\frac{1}{2}k_n S_6 \left(V_{out} - V_P - V_{th}\right)^2 - \frac{1}{2}k_n S_4 \left(V_P - V_{th}\right)^2 \\ &- k_p S_2 \left[(V_{DD} - V_P - V_{th})(V_{DD} - V_{out})V_{out} - \frac{1}{2}(V_{DD} - V_{out})\right]^2 \end{aligned} \tag{3.6}$$

For simplicity channel modulation is ignored ($\lambda = 0$), such assumption is just to have an idea of the behavior of the circuit. In all the equations the threshold voltage

for P-channel transistor and N-channel transistor are expressed as follows:

$$V_{th} = V_{th_p} = V_{th_n}$$

The variable k is expressed as follows:

$$k_{n/p} = \mu_{n/p} C_{ox}$$

where $\mu_{n/p}$ is the charge-carrier effective mobility and C_{ox} is the gate oxide capacitance per unit area. The ratio between the gate length and width is represented by $S_i = \frac{W_i}{L_i}$

The added input stage M7-M12 transforms the input voltage into current (I_{tot}) by using an unbalanced pseudo-differential structure and a current mirror to generate the input current to the shaded circuit [118] is shown in Fig. 4. The voltage V_{in} is swept from $-V_{in}$ to $+V_{in}$ with the + connected to M10 and the – to M9. When the input voltage is negative transistor M10 is on, the current in transistor M10 flows through the current mirror M11-M12 to replicate and obtain I_b, which corresponds to I_{tot} negative (flowing out of V_{out}), as depicted in Figure 3.9.

As the input voltage increases the current I_b decreases and when the input voltage becomes positive, transistor M10 is off and I_b is zero. The opposite is the case with the transistor M9, when V_{in} is negative the gate voltage at M9 (V_{in}) is positive therefore the transistor is off; as voltage V_{in} becomes positive, the V_{GS9} turns negative and the transistor turns on, consequently the current I_b decreases and most current, I_a, flows through the current mirror M7-M8 into V_{out} as I_{tot}.

Current that comes from transistors M7 (I_a) and M11 (I_b) is defined as follows:

$$I_a = -\frac{W_7}{W_8} k_p S_9 \left(V_a - V_{in\_} - V_{th}\right)^2 \tag{3.7}$$

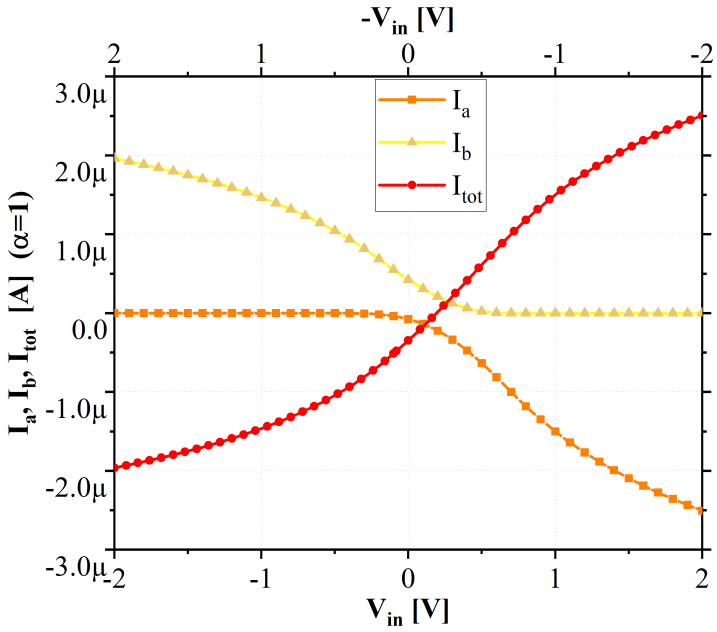


Figure 3.9: Current behavior for I_a, I_b and I_{tot}

$$I_b = \frac{W_{11}}{W_12} k_p S_10 \left(V_{DD} - V_{in_+} - V_{th}\right)^2 \tag{3.8}$$

We defined the relative errors as the difference between the circuit transfer-characteristics and the mathematical function in the transition region between 0 and 1, as follows:

$$error = \frac{|\phi_{THEORICAL} - \phi_{SIMULATED}|}{\phi_{THEORICAL}} \times 100 \tag{3.9}$$

Circuit Sizing

The advantages of this solution over earlier AF circuits include that it does not require an additional I-V or V-I conversion unit, reducing circuit complexity. The implementation can be employed in the node of the hidden layer of the neural network also in the output layer. To set biasing and steepness parameter, transistors M1-M4 are sized to generate biasing $V_P \approx V_{DD}/2$, and M9-M10 are sized to

generate enough current to obtain the desired value of the α parameter. The sizing is asymmetrical for both transistors including their respective current mirrors and varies according to the steepness parameter and the smallest error.

Notice that M10 will drive a current I_b through the mirror current M11-M12 while M9 will drive the current I_a, through the mirror current M7-M8. It is important to highlight that M10 will conduct if $V_{DD} - V_{in+} > V_{th}$ while M9 will conduct when $V_a - V_{in-} > V_{th}$ where $V_a \approx V_{D7}$. This asymmetric configuration ensures to cause drain current when V_{DS9} is greater than $-V_{th}$, this current controls the positive part of the Sigmoid function.

Table 3.2: TRANSISTOR SIZING OF THE PROPOSED SIGMOID FUNCTION .

Transistor	$\alpha = 1$		$\alpha = 2$		$\alpha = 10$	
	L[µm**]**	**W[** µm**]**	**L[** µm**]**	**W[** µm**]**	**L[** µm**]**	**W[** µm**]**
M1	0.18	6	0.18	4	0.18	0.22
M2	0.18	6.7	0.18	4.7	0.18	0.3
M3-M4	0.18	4.5	0.18	3.1	0.18	1
M5	0.18	0.22	0.18	0.22	0.18	0.22
M6	0.18	0.22	0.18	0.22	0.18	0.22
M7-M8	0.18	3	0.18	3	0.18	6
M9	3	0.22	2	0.22	0.5	0.22
M10	11	0.22	7	0.22	6	0.22
M11	0.18	4.6	0.18	3.7	0.18	1.1
M12	0.18	0.22	0.18	0.22	0.18	0.22

Based on Equation 3.7 and Equation 3.8, Table 3.2 lists the size of the transistors used in the implementation to generate a sigmoid approximation when the steepness parameter α for three different values of 1, 2 and 10. As reported in Table 3.2, it can be determined that in order to increase α, it is necessary to reduce the widths of transistors M1-M4 and the lengths of transistors M9 and M10.

Simulation Results

The proposed neuron is simulated in Virtuoso with HSPICE models using 180 nm CMOS TSMC technology and selecting the minimum allowable transistor sizes. The circuit design in Figure 3.8 is asymmetric having rail-to-rail supply of 1V.

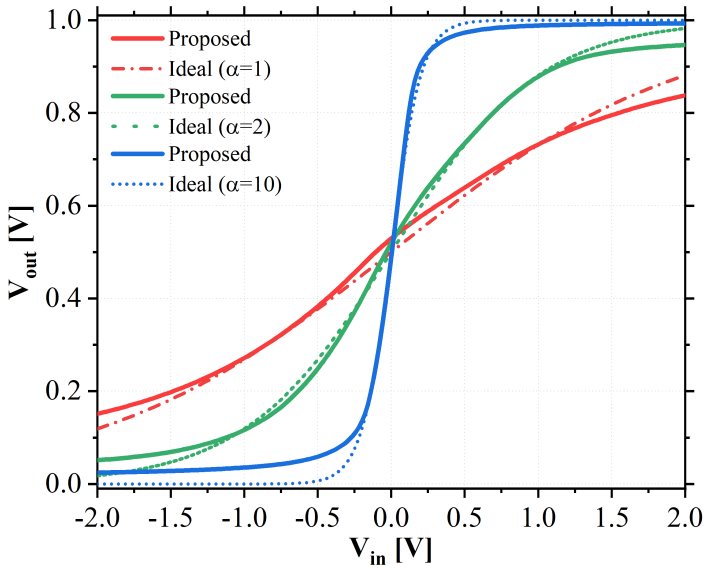


Figure 3.10: Comparison of the proposed sigmoidal neuron and the ideal Sigmoid function at different values of steepness parameter (α).

Figure 3.10 illustrates the Input/Output characteristics of the proposed neuron by varying input voltage V_{in} from $-2[V]$ to $+2[V]$. Furthermore, the graph compares the ideal Sigmoid function (referenced as Equation 3.3) for three different values of the steepness parameter.

In contrast Figure 3.11, Figure 3.12 and Figure 3.13 summarizes the behavior of I_a, I_b and I_{tot} for different values of steepness.

Finally, Figure 3.14 shows the corresponding errors between the proposed neuron and the sigmoid activation function, when the steepness parameter α is set to 1 / 2 / 10, the maximum and average error are determined at 2.87% / 3.27% / 5.29% and

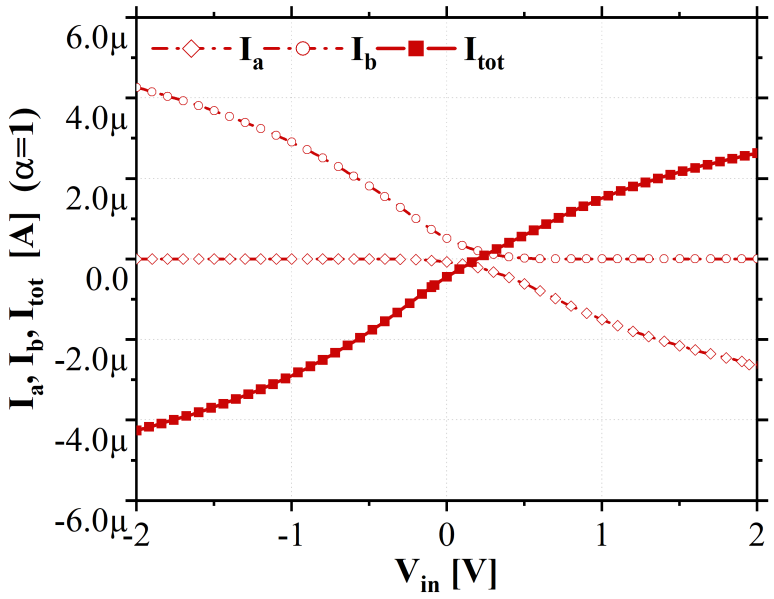


Figure 3.11: Current behavior when $\alpha = 1$ for I_a, I_b and I_{tot}

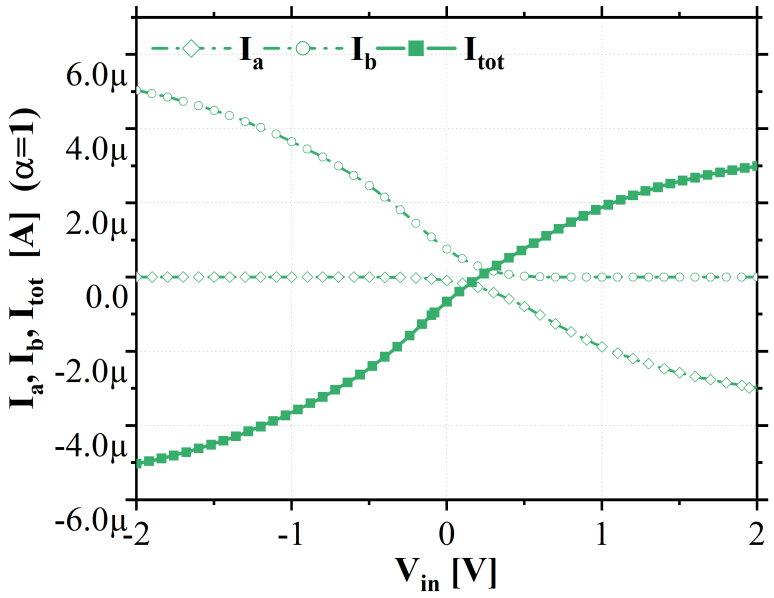


Figure 3.12: Current behavior when $\alpha = 2$ for I_a, I_b and I_{tot}

1.09% / 1.12% / 1.94%, respectively.

Figure 3.15 illustrates the average power consumption variation with different

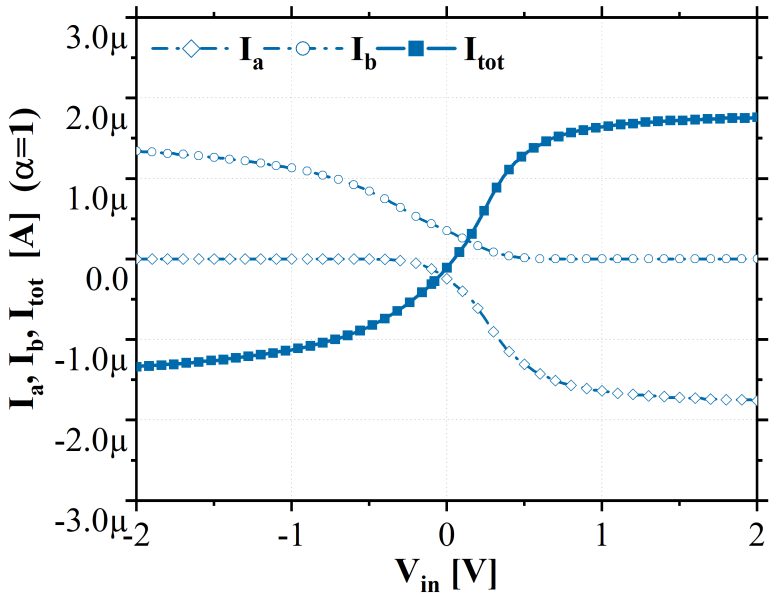


Figure 3.13: Current behavior when $\alpha = 10$ for I_a, I_b and I_{tot}

steepness levels. Specifically, with a steepness parameter of 1, power consumption reaches 18.21 µW whereas at a steepness parameter of 2, consumption drops to 1.14 µW. Finally, at a steepness parameter of 10, power consumption measuring is 6.77 µW.

Table 3.3 presents a comparison between our proposed design and the reported work. The circuit presented in [118] uses a single supply voltage and combines NMOS/PMOS transistors to approximate the Sigmoid function, with an I-V Input/Output characteristic. In contrast, our design achieves lower power consumption and is implemented with twice as many MOSFETs at a more affordable 180nm technology node, rather than 90 nm.

Similarly, in [144] showcases a circuit which employs two differential pairs to produce both tan-sigmoid and log-sigmoid neuron Activation Functions, with an I-I Input/Output characteristic. This design allows for the external programming of slope and threshold levels through the adjustment of bias voltages. In contrast, our

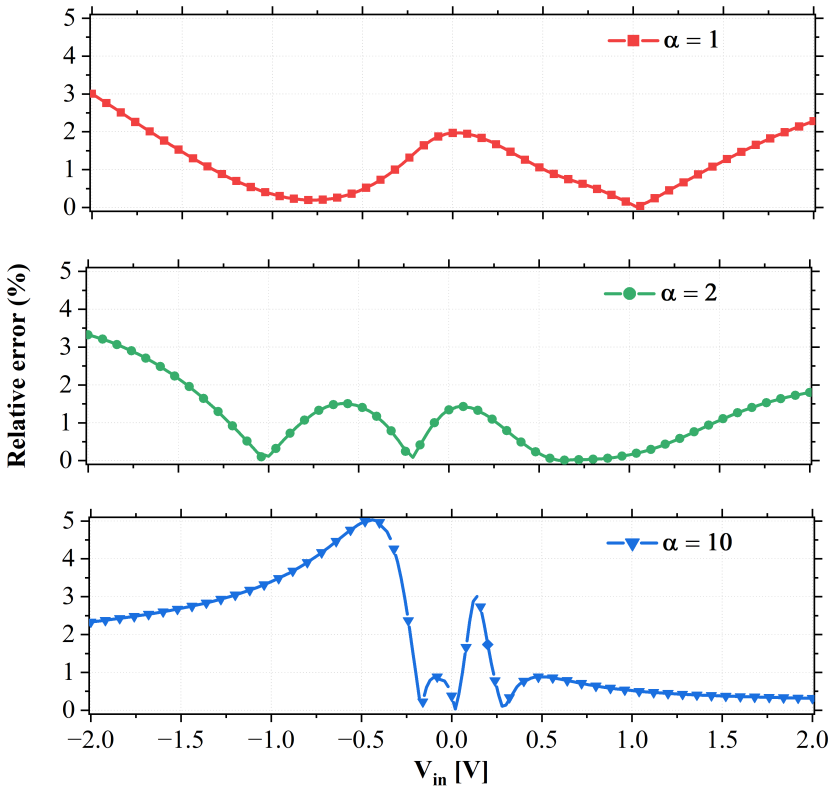


Figure 3.14: Relative error between proposed sigmoid neuron and sigmoid function for $\alpha = 1$, $\alpha = 2$ and $\alpha = 10$

proposed system adjusts the steepness parameter by modifying the size of transistors, providing a means to achieve lower power consumption.

The circuit design presented in [119], utilizing 180 nm TSMC CMOS technology, is a Sigmoid Activation Function neuron that operates across three phases: an input signal weighting circuit , a current-voltage conversion circuit, and a Sigmoid AF fitting circuit. This circuit employs differential pairs to establish the current-voltage relationship of the Sigmoid function. The total area of the layout is measured at 375 µm x 238 µm. However, our proposed design outperforms this circuit in terms of area, since the maximum value obtained is measured at 20.72 µm and 8.10 µm when considering the design for steepness parameters equal to 1.

Figure 3.16 depicts an estimation of the behavior of power consumption and

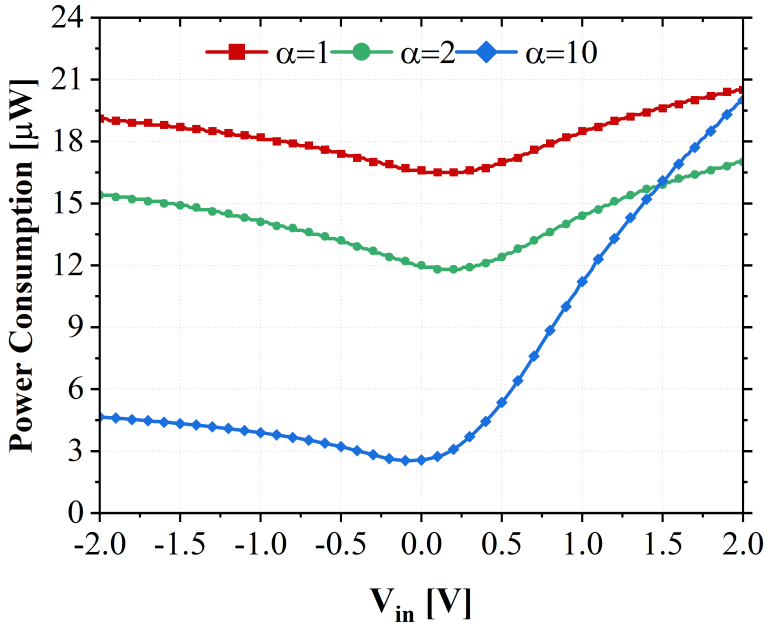


Figure 3.15: Power consumption when $\alpha = 1$, $\alpha = 2$ and $\alpha = 10$.

Table 3.3: COMPARISON OF PROPOSED SIGMOID FUNCTION IMPLEMENTATIONS.

Reference	**Tech** [nm]	**Supply** [V]	**Avg Power** [µW]	**Error** [%]	**Transistor number**
[118]	90	1.2	21.6	7.67	6
[144]	90	1.5	8.4	3	17
[119]	180	1.8	8.02	1.76	10
Prop. ($\alpha = 1$)	180	1	18.21	1.09	12
Prop.($\alpha = 2$)	180	1	14.13	1.12	12
Prop. ($\alpha = 10$)	180	1	6.7	1.93	12

relative error, contrasted across different steepness levels.

Furthermore, we developed the area estimation of the proposed designs by using Calibre-Cadence Virtuoso for all configurations at different steepness parameter. For instance, as shown in Figure 3.17 and Figure 3.18.

The proposed designs were evaluated using TSMC 180 nm CMOS technology giving an area cost when the steepness parameter is 1, 2 and 10 of about 167.83 μm^2, 156.86 μm^2 and 113.98 μm^2, respectively.

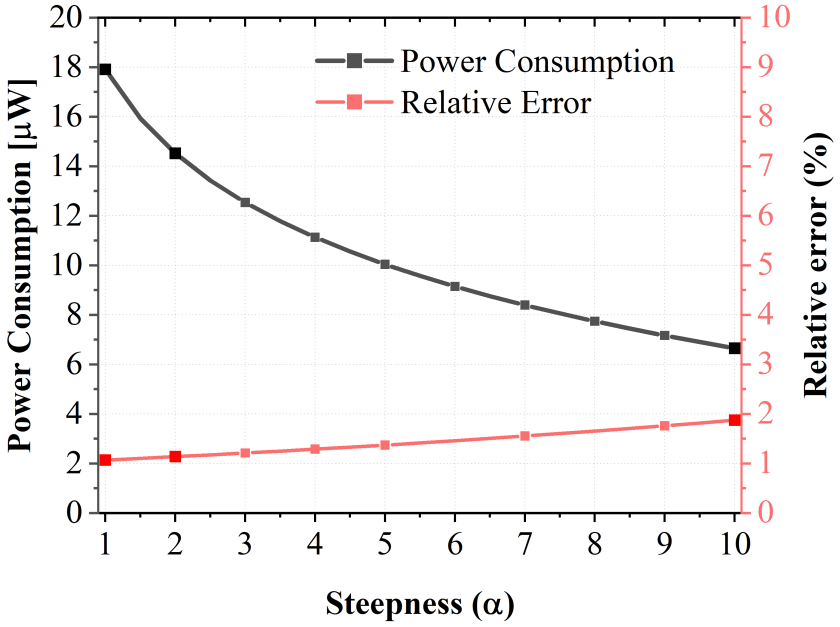


Figure 3.16: Approximation of power consumption and error behavior using the results from the implementation at different values of steepness.

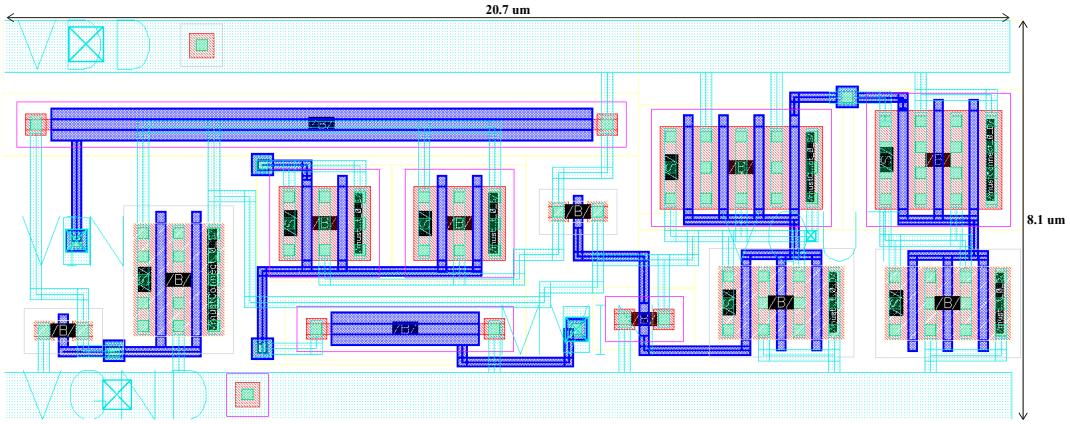


Figure 3.17: Layout schema at steepness parameters of 1 .

Table 3.4 presents an minimum increase in terms of the absolute error which is calculated considering Pre- and Post-Layout Simulation of the Sigmoid AF fitting circuit.

The proposed circuitry introduces a single-bias voltage V to V Sigmoidal neuron, which relies on the transfer characteristics of a CMOS pseudo-differential

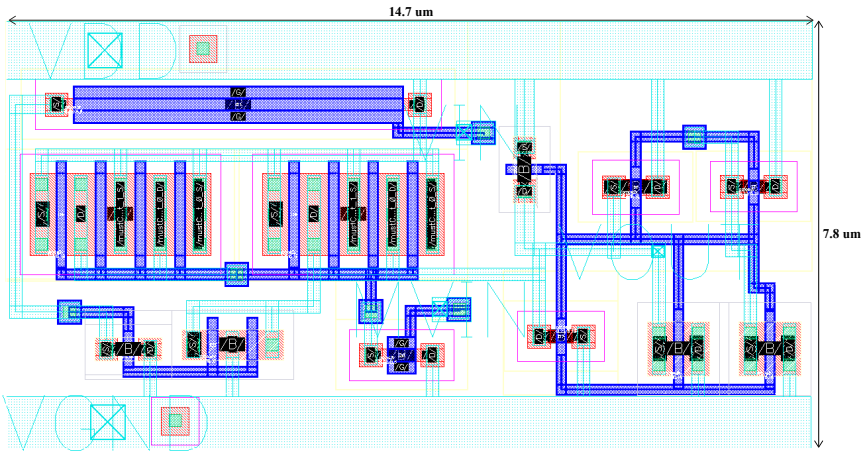


Figure 3.18: Layout schema at steepness parameters of 10 .

Table 3.4: AREA AND ABSOLUTE ERROR CALCULATION BETWEEN PRE- AND POST-LAYOUT ANALYSIS FOR THE PROPOSED SIGMOID AF.

Case	**Layout Dimensions**			**Absolute Error** (mV)	
α	**X** (µm)	**Y** (µm)	**A** (µm^2)	**Pre-Layout**	**Post-Layout**
1	20.72	8.10	167.83	16.99	20.55
2	17.39	9.02	156.86	15.25	16.34
10	14.65	7.78	113.98	21.93	29.84

pair. The circuit's analysis has been assessed across varying steepness parameters, demonstrating more precise approximations of the sigmoid function compared to earlier designs. The results from this circuit present a favorable topology for AF confirming the superior performance of the proposed design. Additionally, it highlights the cost advantage attributed to the use of the 180nm technology.

3.3 Discussion Softmax and Sigmoid

Sigmoid and Softmax are both activation functions used in ANNs to introduce non-linearity into the network's computations. Being crucial components, particularly in classification tasks.

Sigmoid is used for binary classification task, while Softmax is employed for

multi-class classification tasks. Instead, it's crucial to appreciate that each of these functions serves a unique purpose.

In summary, the choice between sigmoid and softmax depends on the specific task and the number of classes involved. Both have their strengths and weaknesses, and their suitability should be considered in the context of the specific problem at hand.

The proposed solutions, tailored to their respective contexts, emerge as compelling and promising approaches that hold significant potential for addressing diverse challenges in ANN.

3.4 Conclusion

A sigmoidal V-to-V neuron with only one bias voltage has been proposed. The circuit design utilizes the transfer characteristics of a CMOS pseudo-differential pair and the performance of this circuit has been evaluated at various steepness parameter values. The minimum error between the output of Sigmoid AF and the ideal Sigmoid function is only 1.09% considering the steepness parameter 1 and the minimum power consumption is 6.77 μW when the steepness parameter is 10. Shows a very accurate approximation compared with other references, with improvements of absolute error, power consumption and area.

Besides, a novel analog implementation of the softmax AF largely used in deep neural networks is presented in this paper. The proposed circuit is implemented in a modular fashion, being composed of three building blocks, which can be replicated and shared, to achieve a softmax function with an arbitrary number of inputs and outputs. A ten-input/ten-output implementation of the proposed softmax circuit, designed in a 180 nm CMOS technology. These improvements are achieved with limited precision degradation, considering that the maximum and average relative

errors, with respect to the theoretical softmax equation, are of 2.2% and 0.9% only, respectively.

CHAPTER 4

STT-MRAM DEVICES FOR NEURO-INSPIRED COMPUTING USING NEUROSIM

The terminology ”neuromorphic” is utilized to describe systems and devices that aim to mimic some of the functionalities of biological neural systems. Carver Mead, located at the California Institute of Technology, was responsible for coining the term in the latter part of the 1980s [145, 146].

Neuro-inspired computing is an emerging field that seeks to mimic the behavior and functionality of the human brain using artificial intelligence and computing technology. It encompasses various approaches including analog [147–149], digital [150, 151], and hybrid circuits using resistive [152, 153], phase change [154], and other non-volatile memory technologies [29, 155]. The goal of neuro-inspired computing is to develop high-performance computing systems that can perform cognitive tasks in a way that is more efficient, robust, and adaptable than traditional computing systems. This field is quickly developing and is expected to lead to significant advancements in artificial intelligence and computing technology in the coming years. This technology will be important for the future of computing, but much of the work in neuromorphic computing has focused on hardware development [156, 157].

Indeed, conventional Von-Neumann computer architecture faces the ”memory wall” problem due to the slow data transfer speeds between the microprocessor and off-chip memory/storage, which becomes more severe when large amounts of data are required for neural network training and testing [158, 159]. Neuro-inspired

architecture provides a promising solution to this problem, as it leverages the distributed computing in neurons and localized storage in synapses to perform large-scale matrix operations directly on-chip, thereby taking advantage of parallelism at a finer grain level, see Figure 4.1.

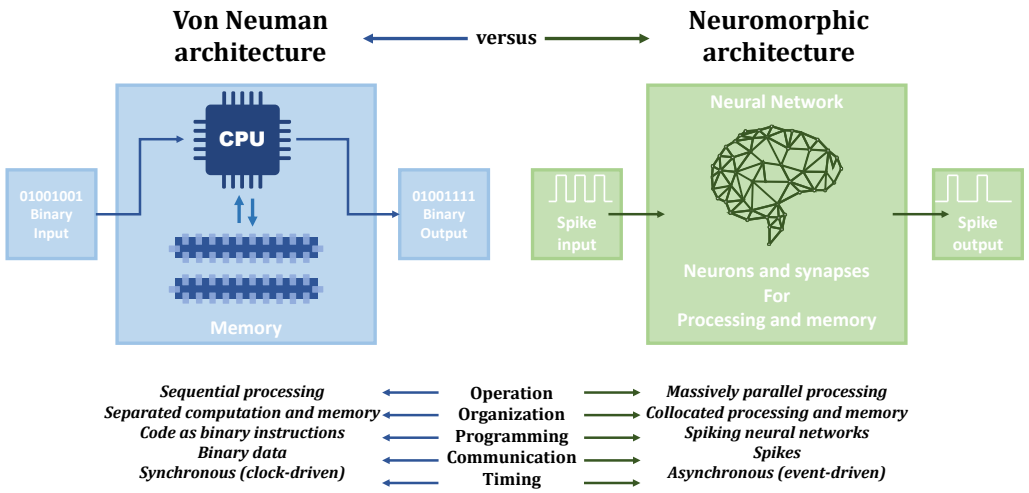


Figure 4.1: Comparison of the von Neumann architecture with the neuromorphic architecture.[160].

Furthermore, neuro-inspired computing presents various advantages in comparison to conventional computing approaches, such as enhanced energy efficiency, accelerated execution speed, increased accuracy, and improved resilience against local failures. These benefits are primarily achieved through the utilization of eNVMs that include RRAM, PCM, STT-MRAM, and FeFET, which offer enhanced flexibility for the development of DNN.

In order to emulate the synaptic connections between neurons in an artificial neural network, a computational unit known as a synaptic core is implemented. The synaptic core is responsible for performing weighted summation and weight updates [114, 115]. Depending on the type of bitcell used, the synaptic core architecture can be either analog, digital, or hybrid.

Although analog synapse eNVM-based architectures could be competitive in terms of energy and latency, they mainly suffer from low online learning accuracy [161]. To deal with this issue, digital synapse based architectures have been widely considered [162, 163].

As potential eNVM candidate for digital synapse devices, STT-MRAM cell offers low operating voltage, enough good speed operation, high-density, relatively large endurance, low fabrication cost, low-power consumption, and scalability [38, 164, 165].

The following chapter explores the outstanding features when considering SMTJ and DMTJ-based STT-MRAM cell on DNN, by using Cadence-Virtuoso environment for circuit-level simulations, along with the MLP + NeuroSimV3.0 simulator computing-in-memory based neural network accelerator [163].

4.1 Neurosim: Simulation Framework

NeuroSim simulator allows to estimate the algorithm-level performance by emulating the online learning and offline classification scenario with MNIST handwritten dataset in a 2-layer multilayer perceptron (MLP) neural network based on SRAM, eNVM and FeFET array architectures [163, 166–168]. NeuroSim enables a comprehensive framework for circuit-level performance estimation; therefore, it is used to compare neurologically inspired architectures in terms of circuit-level metrics such as area, latency, dynamic energy, and leakage power. Precisely, we propose an impact evaluation based on SMTJ and DMTJ devices.

The input parameters of the simulation tool include memory type, non-ideal device parameters, transistor technology node, network topology and array size, training dataset and traces, etc. For the full list of input parameters/variables, the reader is referred to [163]. The outputs of the simulator include: (1) the memory

architecture-level performance metrics, such as area, latency, dynamic energy, and (2) algorithm-level learning accuracy in run-time.

Figure 4.2 displays an overview of NeuroSim framework considering for the whole system from device and bitcell levels to memory architecture and algorithm levels.

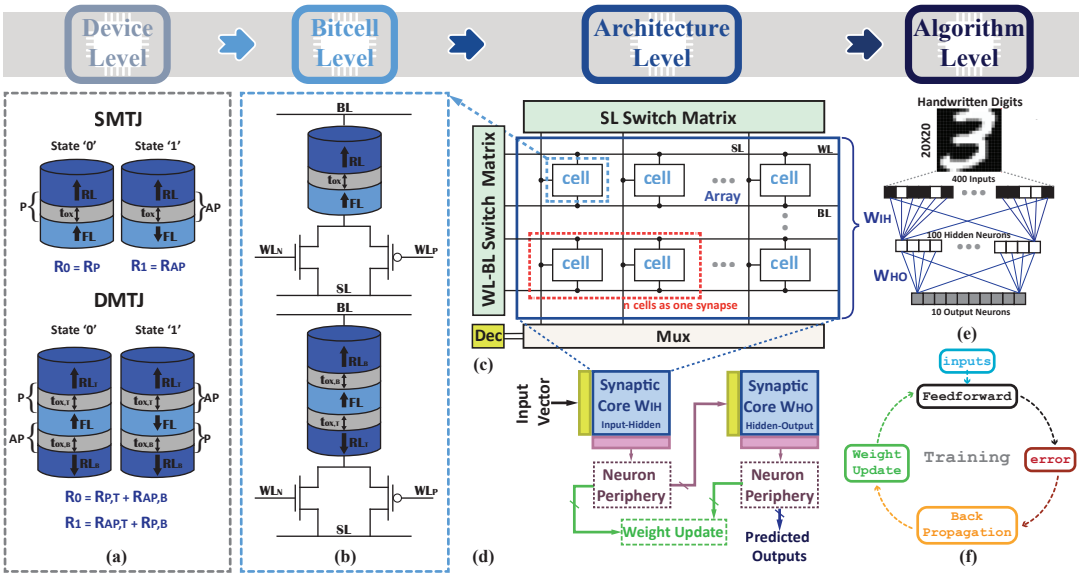


Figure 4.2: Overview of NeuroSim framework from Device to Algorithm-level, (a) STMJ and DMTJdevice, (b) SMTJ-based and DMTJ-based bitcell configurations, (c) Circuit block diagram of digital eNVM synaptic core, (d) Circuit block diagram for hardware implementation of the 2-layer MLP NN. The weights are mapped through synaptic cores (e) Training flow of Neural Network, the MNIST images are crooped and encoded into black and white data for simplification on hardware implementation.

4.1.1 Algorithm-Level

NeuronSim is a hierarchical structure from the algorithm layer to the device layer, taking into account detailed properties of the synaptic array and realistic devices. It can be viewed as an independent functional simulator capable of assessing the learning accuracy at the circuit level performance specifically for the synaptic array during learning [166].

Furthermore, it is a simple 2-layer MPL neural network for performance benchmark. As shown in Figure 4.2(e) the network consists of an input layer, hidden layer and output layer (the input layer is not included when counting the number of layers). The considered MLP is a fully connected neural network, where each neuron node in one layer connects to every neuron node in the following layer. We remind that the connection between two neuron nodes is through a synapse with its strength representing the weight. The connections between input-hidden and hidden-output layers represent the weight matrix W_{IH} and W_{HO}, respectively.

The MNIST dataset itself consists of input image data depicting handwritten digits, with each image comprising 20x20 pixels. Besides, Figure 4.2(e) displays the network topology, consisting of an input layer with 400 neurons (corresponding to the 20x20 MNIST image), a hidden layer with 100 neurons, and an output layer with 10 neurons (representing the 10 classes of digits).

During the training process, see Figure 4.2(f), it consists of two key operations: feed forward and back propagation. In feed forward, input data are passed from the input layer to the output layer via a series of weighted sum operations and neuron activation functions. The output is then compared to the correct answer to calculate the prediction error. In back propagation, this error is used to adjust the weights of each layer in order to minimize the prediction error. Stochastic gradient descent is used to update the weights during back propagation, with the back propagation performed after the feed forward of each image. During testing and classification, only the feed forward operation is used to make predictions and the weights are not changed.

Lastly, considering the training time, we employ 15 epochs (i.e., number of training iterations), 8000 and 1000 MNIST images for training and testing, respectively, giving a total of 12000 MNIST images being trained. We used the online learning in hardware configuration, which handle testing and training for

both weight sum and weight update all in hardware.

4.1.2 Memory Architecture-Level

Among the available design options for the synaptic cores, we considered the digital eNVM based on pseudo-crossbar array as shown in Figure 4.2(c), as the digital synaptic devices can only store binary 0 or 1, the digital synaptic devices must be grouped together to represent the weight precision [163].

By contrast Figure 4.2(d) depicts the circuit block diagram for hardware implementation of the 2-layer MLP considered. Each synaptic core is a computation unit specifically designed for weighted sum and weight update [163, 166].

The embedded non-volatile memory makes up the majority of the digital synaptic core and is responsible for storing the synaptic weights of a neural network. It plays a crucial role in the processing of neural network information through write and read operations, allowing for efficient and accurate execution of machine learning tasks. The size of the synaptic core area is an important factor to consider in achieving optimal performance and energy efficiency in digital synapse devices, as reducing its size can lead to a smaller overall circuit architecture and potentially lower power consumption [166].

Therefore, the bit-cell selected for the digital synapse core is an STT-MRAM cell based on SMTJ and DMTJ.

4.1.3 MTJ Devices and operation-bitcells

As shown in Figure 4.2(a), we consider STT-SMTJ/DMTJ devices, whose main physical and performance parameters are listed in Table 4.1. Chapter 4 provides comprehensive details regarding SMTJ and DMTJ devices. The main device parameters considered in the referenced work [38] for both perpendicular-MTJ and FinFET devices is at 28 nm technological node and T = 300 K.

The STT-MTJs are described through Verilog-A based compact models [169, 170], which have been validated against full micromagnetic and experimental results. This parameters are extracted since are needed to incorporated the SMTJ and DMTJ parameters at the device level in NeuroSim tool (see Table 4.1). In particular, the STT-MTJ models utilize experimental data reported in [171]. These models further account for the impact of process variability on the STT-MTJs. Specifically, the variability, modeled by incorporating Gaussian-distributed variations, was set to 1% for both the free-layer and oxide thickness, 3% for TMR ratio, and 5% for the cross-section area [38].

Table 4.1: SMTJ AND DMTJ DEVICE PARAMETERS [38].

Parameter	**Units**	**Value**
Diameter (d)[a]	nm	28
Saturation magnetization (Ms)[a]	A/m	1000×10^3
Magnetic damping (α)[a]	-	0.025
Spin-polarization factor (η)[a]	-	0.67
FL thickness (t_{FL})[a]	nm	1.2
SMTJ oxide thickness	nm	0.85
DMTJ top oxide thickness	nm	0.85
DMTJ bottom oxide thickness	nm	0.4
TMR at 0 V (TMR(0))[c]	%	150

[a] Same value for SMTJ and DMTJ devices.
[c] Same value for SMTJ barrier and DMTJ top/bottom barriers

This analysis considers the SMTJ-based and DMTJ-based bit-cell developed by E.Garzón *et al* [37, 38]. Figure 4.2(b) shows the considered SMTJ-based and DMTJ-based bitcell configurations designed exploiting a 28 nm FinFET technology featuring a nominal supply voltage of 0.8 V. These are referred to the two complementary FinFET and one MTJ (2T1MTJ) cells in reverse and standard connection (2T1MTJ-RC and 2T1MTJ-SC) for the SMTJ- and DMTJ-based bitcells, According to the study carried out in [38], those considered are the most write energy-efficient bitcell configurations.

4.2 Efficiency of DMTJ-based Digital eNVM

NeuroSim framework shown in Figure 4.2 was properly calibrated with the 0.8 V FinFET technology parameters, along with the bitcell electrical characteristics of the considered 2T1MTJ-based bitcells, which are the cells of the pseudo-crossbar eNVM digital synaptic core. Bitcell-level results consider both SMTJ/DMTJ and FinFET device-to-device variability through extensive Monte Carlo simulations. Table 4.2 shows the bitcell-level parameters of the energy-optimal cell size and configurations (refer to Figure 4.2(b)).

It is worth to mention that these results are carried out at parity of TMR, and oxide thickness, i.e., $t_{ox,SMTJ} = t_{ox,t,DMTJ} = 0.85$ nm. Performance results for write and read operations are obtained, assuring a write-error-rate (WER) of 10^{-7} and read disturbance rate (RDR) of 10^{-9}, respectively.

Table 4.2: NOMINAL VALUES FOR STT-SINGLE AND DOUBLE BARRIER CELL (t_{ox} = 0.85 nm)

Parameters		Units	SMTJ	DMTJ
	Cell Area	F^2	231	131
	Resistance ON	Ω	9513	11370
	Resistance OFF	Ω	16390	22170
	Read Pulse Width	ns	1.00	1.00
Read Mode	Read Current	fJ	26.12	7.20
	Read Power	ns	20.89	5.76
	Reset Current	µA	54.77	49.27
Reset Mode	Reset Pulse	ns	3.39	1.16
	Reset Energy	pJ	0.1929	0.0552
	Set Current	µA	82.85	49.62
Set Mode	Set Pulse	ns	3.39	1.16
	Set Energy	pJ	0.177	0.0409

From Table 4.2, it is clear that thanks to the reduced switching and read currents, the DMTJ-based bitcell is the most energy-efficient alternative under write/read operations. Overall, at bitcell-level, the DMTJ-based alternative shows energy

savings of about 72% and 97% for read and write operations, while assuring faster (65.7%) switching in contrast to the SMTJ-based bitcell.

Moreover, from Table 4.2, Resistance ON is related to the Low-Resistance-State (LRS) and Resistance OFF corresponds to the High-Resistance-State (HRS) when sensing (i.e,. during reading operation). Reset Mode and Set Mode corresponds to the operation during writing when writing a "zero"(AP→P) and "one"(P→AP) into STT-MRAM cell, respectively. Likewise, Reset Pulse and Set Pulse stand for the pulse duration over the AP→P and P→AP switching operation, respectively.

The calculation of parameters that must be inserted when setting up the eNVM bitcell digital specifications for the synaptic core are; Read Voltage, Read Energy, Read and Write Pulse Width, Write Energy, Write Voltage long-term depression and long-term potentiation (LTD and LTP). As mention in the following equations:

$$ReadVoltage = ReadCurrent \cdot \frac{R_{ON} + R_{OFF}}{2} \tag{4.1}$$

$$ReadEnergy = ReadPower \cdot ReadPulseWidth \tag{4.2}$$

$$WriteEnergy = \frac{ResetEnergy + SetEnergy}{2} \tag{4.3}$$

LTP and LTD involve modifying the strength of the connections between neurons by increasing or decreasing the synaptic weight over time. Below the equation is presented as:

$$WriteVoltageLTP = ResetCurrent \cdot R_{OFF} \tag{4.4}$$

$$WriteVoltageLTD = SetCurrent \cdot R_{ON} \tag{4.5}$$

In summary, Table 4.3 displays the parameters obtained from the equations shown above, in order to evaluate the algorithm-level performance of 2-layer MPL neural network.

Table 4.3: BITCELL-LEVEL PARAMETERS FOR SMTJ AND DMTJ (t_{ox}=0.85 nm)

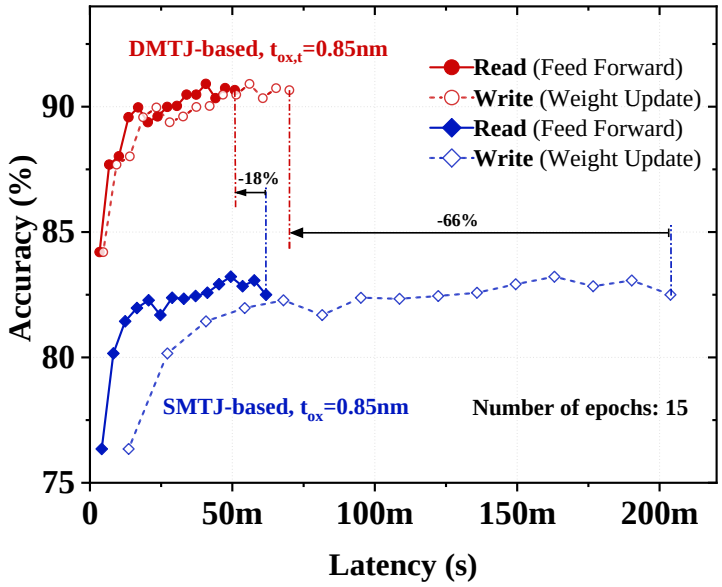
	Parameter	Unit	STMJ	DTMJ
bitcell	Cell Area	F^2	231	131
	Resistance ON	Ω	9513	11370
	Resistance OFF	Ω	16390	22170
	Conductance ON/OFF	–	1.79	1.97
	Read Voltage	V	0.338	0.121
	Read Energy	fJ	20.9	5.76
	Read Pulse Width	ns	1.00	1.00
	Write Energy	fJ	185	4.80
	Write Voltage LTD	V	0.788	1.09
	Write Voltage LTP	V	0.898	0.564
	Write Pulse Width	ns	3.39	1.16

4.2.1 Performance Analysis

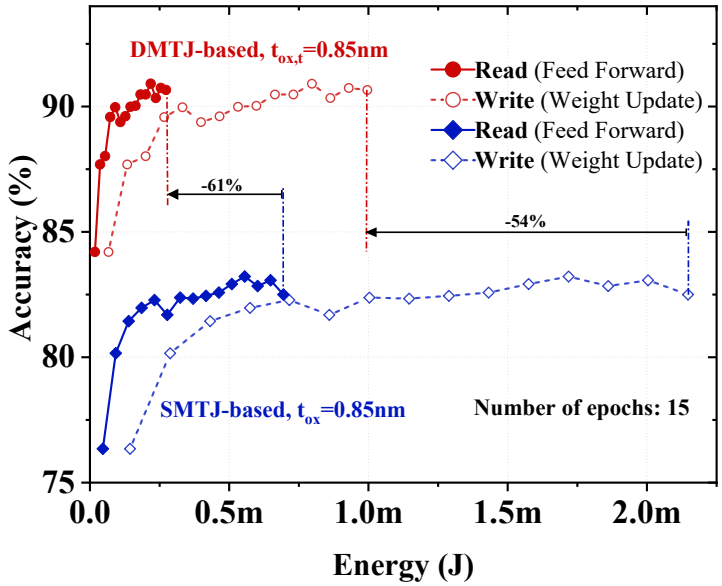
The SMTJ- and DMTJ-based 2-layer MLP neural network performance is evaluated in terms of learning accuracy versus latency and energy consumption, calculated at the run-time.

The read (weighted sum-feed forward operation) and write (weight update operation) latency and energy are shown in Figure 4.3. We can observe that the weighted sum and weight update operations associated to the DMTJ-based eNVM cell achieve the highest accuracy much faster as compared to the SMTJ-based counterpart, while at the same time ensuring less energy consumption. This is due to the reduced energy/write-pulse width of the DMTJ-based bitcell (refer to Table Table 4.2).

From Figure 4.3(a), it is worth noting that the delta latency (i.e, time between



(a)



(b)

Figure 4.3: Trace of Latency and Energy in feed forward and weight update during online learning for both STMJ- and DMTJ-based when considering a top barrier of $t_{ox,SMTJ} = t_{ox,t,DMTJ} = 0.85$ nm

iterations) in feed forward operation, for both STMJ- and DMTJ-based alternatives, is roughly the same, mainly do to the similar requirement for the read pulse width.

As for the weight update operation, the delta latency between each epoch is 14ms

and 4.7ms, respectively. This can be explained due to the larger pulse width required for writing operation. As compared with the SMTJ-based alternative, the DMTJ-based cell shows an improvement in terms of latency, of about 18% and 66% in feed forward and weight update operations, respectively, during online learning. Similar results have been obtained for the energy consumption, as shown in Figure 4.3(b). The DMTJ-based cell shows lower energy consumption as compared to the SMTJ-based alternative, owing to its reduced bitcell read/write energy. The results showed an improvement of about 61% and 54% during feed forward and weight update, respectively.

The benchmark results show that, while the DMTJ-based solution achieves a good accuracy of $(> 90\%)$, the SMTJ-based neural network reaches a learning accuracy of about 83%.

The cause of degradation in terms of learning accuracy is attributed to the devices' poor conductance ON/OFF ratio [161].

In addition, we estimate the area occupation as extracted from NeuroSim. Fig. Figure 4.4 shows the total area footprint. The area occupation for the SMTJ-based and DMTJ-based alternatives is $0.0788\,\mathrm{mm}^2$ and $0.0531\,\mathrm{mm}^2$, respectively. DMTJ-based bitcell can achieve the smallest area footprint due to the smaller bitcell area (see Table 4.3), which corresponds to the energy-optimal cell size.

4.2.2 Impact of Synaptic Device Properties on Accuracy

During the weight update, the conductance of the device should be sufficiently large, i.e., the lowest conductance state (OFF-state) should be low enough to represent the zero weight in the algorithm [161]. To quantify the impact of the device properties on the learning accuracy, we carried out an analysis for both STT-MTJ alternatives by varying $t_{ox}/t_{ox,t}$.

If we decrease the oxide thickness for both devices, the ON and OFF resistance

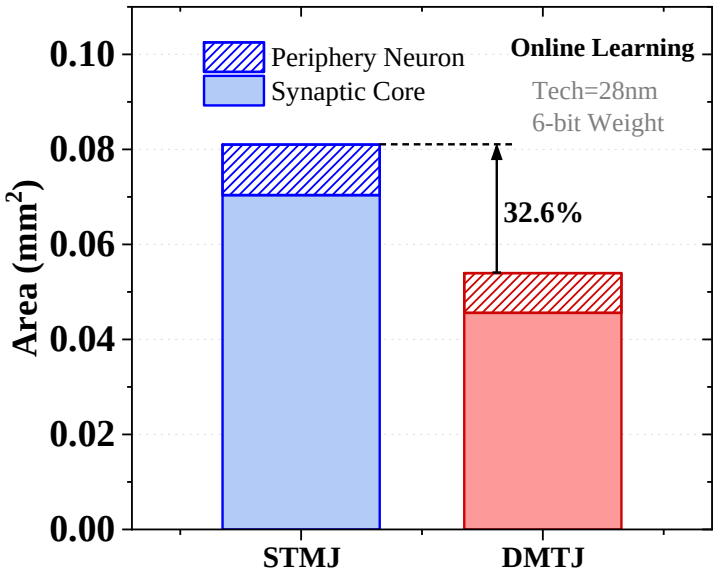


Figure 4.4: Area of MLP NN architecture for both SMTJ-based and DMTJ-based synaptic cores.

of the bitcell will be affected. Accordingly, the performance results for write and read operations obtained at oxide thickness $t_{ox,SMTJ}$=$t_{ox,t,DMTJ}$ = 0.80 nm are shown in Table 4.4

Table 4.4: NOMINAL VALUES FOR STT-SINGLE AND DOUBLE BARRIER CELL (t_{ox} = 0.80 nm)

Parameters		Units	SMTJ	DMTJ
	Cell Area	F^2	231	131
	Resistance ON	Ω	6975	8838
	Resistance OFF	Ω	13340	16650
	Read Pulse Width	ns	1.00	1.00
Read Mode	Read Current	µA	26.12	7.20
	Read Power	µW	20.89	5.76
	Reset Current	µA	69.44	60.96
Reset Mode	Reset Pulse	ns	2.119	0.971
	Reset Energy	pJ	0.1462	0.0533
	Set Current	µA	110.0	56.88
Set Mode	Set Pulse	ns	2.119	0.971
	Set Energy	pJ	0.1503	0.0394

Likewise, Table 4.5 depict SMTJ-base and DMTJ-base bitcell parameters

required for the NeuroSim tool.

Table 4.5: BITCELL-LEVEL PARAMETERS FOR SMTJ AND DMTJ (t_{ox}=0.80 nm)

	Parameter	Unit	STMJ	DTMJ
bitcell	Cell Area	F^2	231	131
	Resistance ON	Ω	6975	8838
	Resistance OFF	Ω	13340	16650
	Conductance ON/OFF	–	1.913	1.88
	Read Voltage	V	0.265	0.0917
	Read Energy	fJ	20.9	5.76
	Read Pulse Width	ns	1.00	1.00
	Write Energy	fJ	148	46.4
	Write Voltage LTD	V	0.767	0.503
	Write Voltage LTP	V	0.926	1.01
	Write Pulse Width	ns	2.119	0.971

When considering a top barrier of $t_{ox,SMTJ}$ = $t_{ox,t,DMTJ}$ = 0.80 nm, the conductance ON/OFF ratio for SMTJ- and DMTJ-based cell are 1.91 and 1.88, respectively. Therefore the conductance ON/OFF ratio for SMTJ-based cell increases by 6.4%, while DMTJ-based cell decreases by 4.8%, as shown in Figure 4.5. The reduced ON/OFF conductance ratio in the DMTJ-based cell can be explained by the presence of the second oxide barrier.

The read (weighted sum-feed forward operation) and write (weight update operation) latency and energy, when oxide thickness of the SMTJ-base and DMTJ-based bitcell is 80nm, are shown in Figure 4.6.

We can notice that during the feed forward and weight update operation, DMTJ-based eNVM is much faster to reach the highest accuracy as compared to the SMTJ-based counterpart, conversely ensuring less energy consumption.

Furthermore, we can observe that during the read and write operations associated to the SMTJ-based and DMTJ-based eNVM cell the achieved accuracy are almost close, giving results of 90.52% and 89.86 %, respectively. Therefore, the accuracy for SMTJ-based cell increases by 5.9%, while DMTJ-based cell decreases by 2.4%,

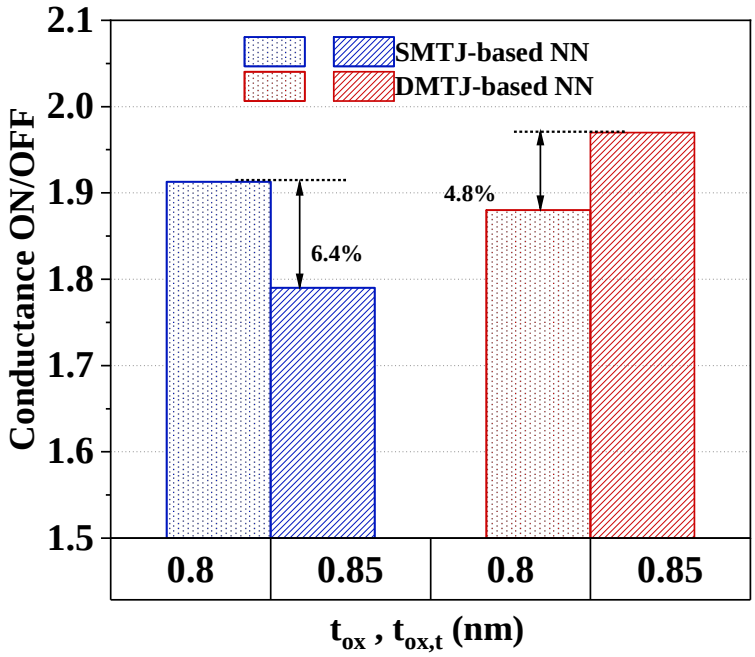


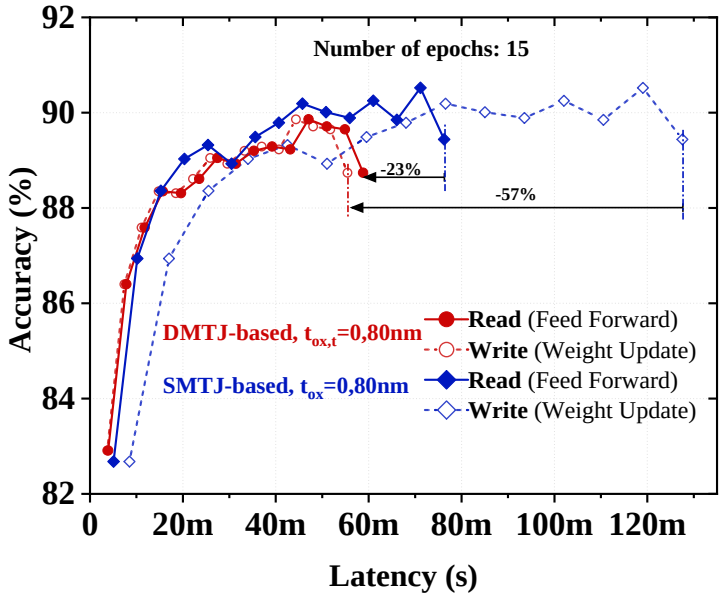
Figure 4.5: Learning accuracy versus oxide thickness (t_{ox} or $t_{ox,t}$) for SMTJ- and DMTJ-based neural networks.

see Figure 4.7.

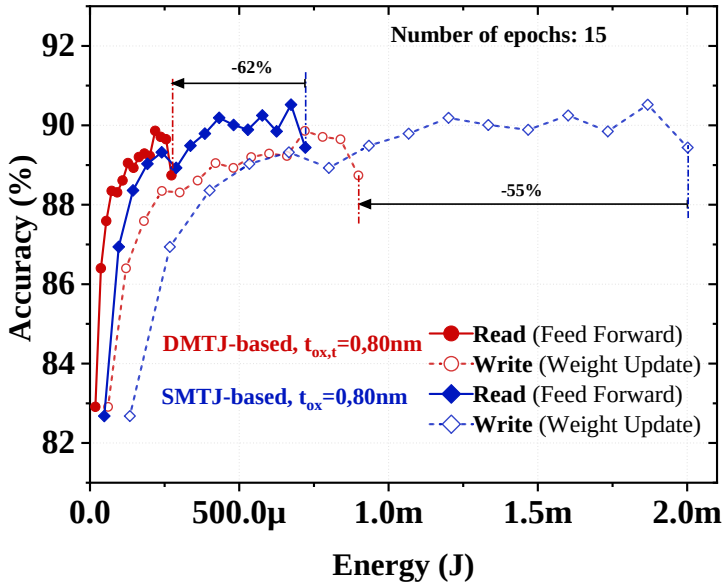
Note that the use of very thin oxide barriers could lead to breakdown of the MTJ structure. To deal with this reliability issue, the write voltages have to be reduced [172].

Table 4.6 shows the assessment of energy, latency, accuracy, and area results obtained at different values of oxide thickness, for SMTJ- and DMTJ-based cells.

From table Table 4.6, the SMTJ-based cell at t_{ox}=0.85 nm has less latency and energy consumption compared with SMTJ-based cell at t_{ox}=0.80 nm in feed forward operation. In contrast, during the weight update, the latency and energy consumption increases when t_{ox}=0.85 nm. Moreover, during the feed forward and weight update operation the DMTJ-based cell at t_{ox}=0.80 nm results less energy hungry than its t_{ox}=0.85 nm counterpart. Furthermore, the DMTJ-based cell at t_{ox}=0.85 nm is faster compared with t_{ox}=0.80 nm along the weight sum. During the



(a)



(b)

Figure 4.6: Trace of Latency and Energy in feed forward and weight update during online learning for both STMJ- and DMTJ-based when considering a top barrier of $t_{ox,SMTJ} = t_{ox,t,DMTJ} = 0.85$ nm

weight update, the DMTJ-based cell at t_{ox}=0.80 nm has improved latency over the t_{ox}=0.85 nm counterpart.

Finally, we have also performed the comparative study of the DMTJ- and

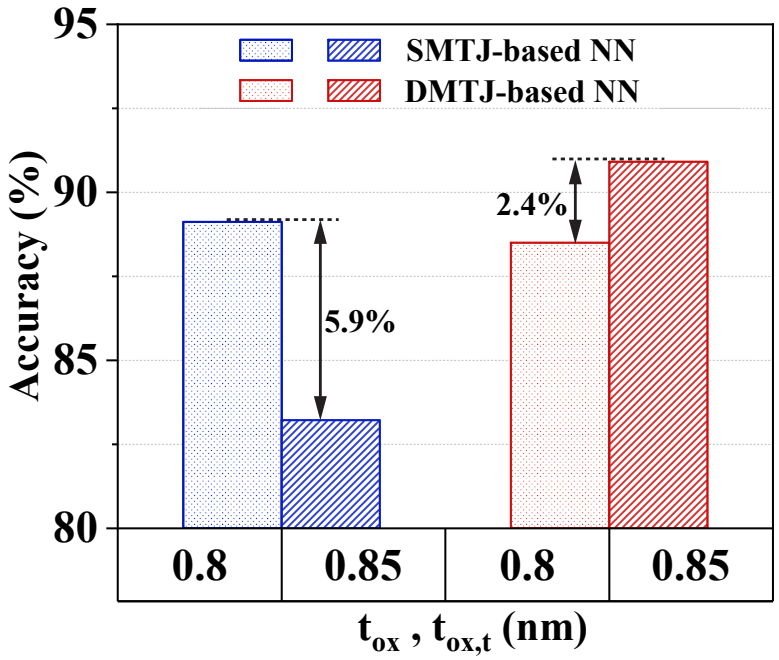


Figure 4.7: Learning accuracy versus oxide thickness (t_{ox} or $t_{ox,t}$) for SMTJ- and DMTJ-based neural networks.

Table 4.6: BENCHMARK RESULTS OF SMTJ- AND DMTJ-BASED CELL AT $T_{OX,SMTJ} = T_{OX,T,DMTJ} = 0.85$ nm AND $T_{OX,SMTJ} = T_{OX,T,DMTJ} = 0.80$ nm.

Parameter	Oxide thickness	Energy (mJ)		Latency (ms)	
		Read	Write	Read	Write
SMTJ	**0.80** nm	0.712	2	76.2	128
	0.85 nm	0.695	2.15	61.8	204
DMTJ	**0.80** nm	0.273	0.899	58.8	55.4
	0.85 nm	0.2731	0.996	50.9	70.0
SMTJ (0.8 nm vs 0.85 nm) (%)		2.45	-6.98	23.3	-37.25
DMTJ (0.8 nm vs 0.85 nm) (%)		-0.04	-9.74	15.52	-20.86
DMTJ vs SMTJ (@ 0.80 nm) (%)		-61.66	-55.05	-22.83	-56.72

SMTJ-based solutions considering t_{ox}=0.80 nm. The DMTJ-based cell shows an improvement in terms of latency, of about 23% and 57% in feed forward and weight update operations, respectively, compared with the SMTJ-based cell. As for the energy consumption, the analysis shows similar results compared with $t_{ox,SMTJ}$ =

$t_{ox,t,DMTJ}$ = 0.85 nm, showing accuracy improvements of about 62% and 55% during feed forward and weight update, respectively.

4.3 Conclusion

We have explored the STT-MTJ synaptic pseudo-crossbar array architecture and device/transistor models in NeuroSim. Our results show that, at parity of TMR and oxide thickness, as compared to the conventional SMTJ-based alternative, the DMTJ-based solution proves to be faster during feed forward and weight update operations of about 18% and 66%, respectively, more energy efficient under read (-60.7%) and write operation (-53.7%), and less area hungry (-35%) at an energy-optimal bitcell configuration/size. This occurs while also achieving an accuracy closed to 91% when running the neural network with the MNIST dataset.

CHAPTER 5

STT-MTJ BASED SMART MATERIAL IMPLICATION ARCHITECTURE FOR IN-MEMORY COMPUTING

Material implication (IMPLY) logic shows great potential as a prospective solution for defining LIM architectures designed to execute fast and energy-efficient computations directly within memory units. This approach effectively mitigates the von-Neumann bottleneck of conventional computing platforms, specifically the need to read/write data to/from off-chip memories [31, 173–180]. However, the conventional IMPLY logic scheme faces significant challenges, including the degradation of the logic states and the limited design flexibility associated with operating voltages [181]. To overcome these drawbacks, an alternative smart IMPLY (SIMPLY) LIM scheme was recently proposed [181–184]. The SIMPLY solution integrates an output comparator into the classical IMPLY scheme. This comparator is exploited to execute a preliminary 2-bit read operation, which is then used to execute the SET operation selectively, based on the specific need (i.e., only when both inputs are at low logic level [181]). Such an approach effectively alleviates the issue of logic state degradation, while also reducing energy consumption with minimal impact on circuit area and complexity when compared to the conventional IMPLY scheme [181].

Although the SIMPLY logic was originally proposed and validated for RRAM devices [181–184], recent investigations have extended its applicability to

STT-MRAM devices[185, 186]. The latter represents an appealing option for LIM applications owing to faster read/write operations, very low standby power consumption, and high endurance [5, 29, 38, 71, 72, 187]. According to findings in [185], the STT-MRAM-based SIMPLY scheme exhibits the expected advantage of improved energy-efficiency and reliability as compared to its IMPLY counterpart. However, as highlighted in [186], the reliability of STT-MRAM-based SIMPLY logic is significantly impacted by the preliminary read operation. This influence is primarily due to the relatively narrow read memory window inherent in STT-MRAM devices, constrained by their TMR ratio [188–190]. Consequently, this limitation results in suboptimal read margins, which lead to a higher level of design complexity in the sensing circuitry [186].

In response to the above issue, this chapter introduces SIMPLY+, i.e., an advanced STT-MRAM-based SIMPLY logic scheme that allows enhanced operation reliability compared to its conventional counterpart. SIMPLY+ scheme, which is an STT-MTJ-based SIMPLY scheme with large read sensing margins, reliable for IMPLY operations, is exhaustively evaluated by means of extensive Monte Carlo (MC) simulations and benchmarked against the conventional SIMPLY logic.

5.1 STT-MTJ Modeling

The behavior and characteristics of the STT-MTJ are described using an analytical macrospin-based Verilog-A compact model [191]. The main physical parameters of the 30- nm STT-MTJ device considered in this work are summarized in Table 5.1, referring to room temperature (300 K) [192–194].

As results of the modeling, Figure 5.1 shows the trend of the resistance and switching characteristic across temperatures.

Table 5.1: STT-MTJ parameters (300 K)

Parameter	Description	Value	Units
d	Diameter	30	nm
t_{FL}	FL thichness (variability)	1.15	nm
t_{OX}	Oxide thickness (variability)	0.85	nm
RA	Resistance-area product	10	$\Omega\cdot$ µm^2
η	Spin-polarization factor	0.66	–
V_H	Bias voltage for $TMR = 0.5 \cdot TMR(0)$	0.5	V
M_S	Saturation magnetization	1.58	T
α	Gilbert damping factor	0.03	–
Ki	Interfacial perperdicular anisotropy constant	1.3	mJ/ m^2
Δ	Thermal stability factor	∼44	–

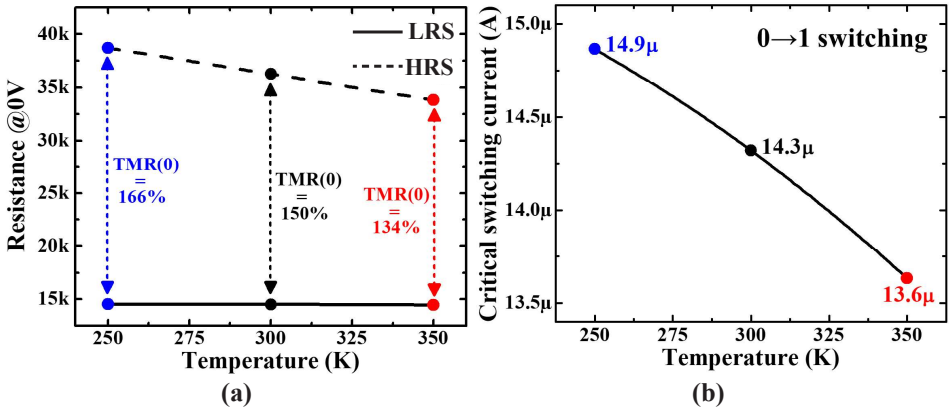


Figure 5.1: STT-MTJ structure and its resistive states. (a) Resistance in low (LRS) and high (HRS) states at zero bias voltage with tunnel magnetoresistance (TMR) ratio values at 250K, 300K, and 350K. (b) Critical switching current (Ic) for $0 \rightarrow 1$ switching (i.e., from antiparallel to parallel state)[186].

5.2 Logic-in-Memory Architectures

BNN implementation based on the SIMPLY architecture have gained significant attention in recent years due to their energy efficiency and potential for hardware implementation.

Figure 5.2 shows the top-level diagram of the designed STT-MRAM SIMPLY-based architecture. A control logic, equipped with analog tri-state buffers (TSBs)

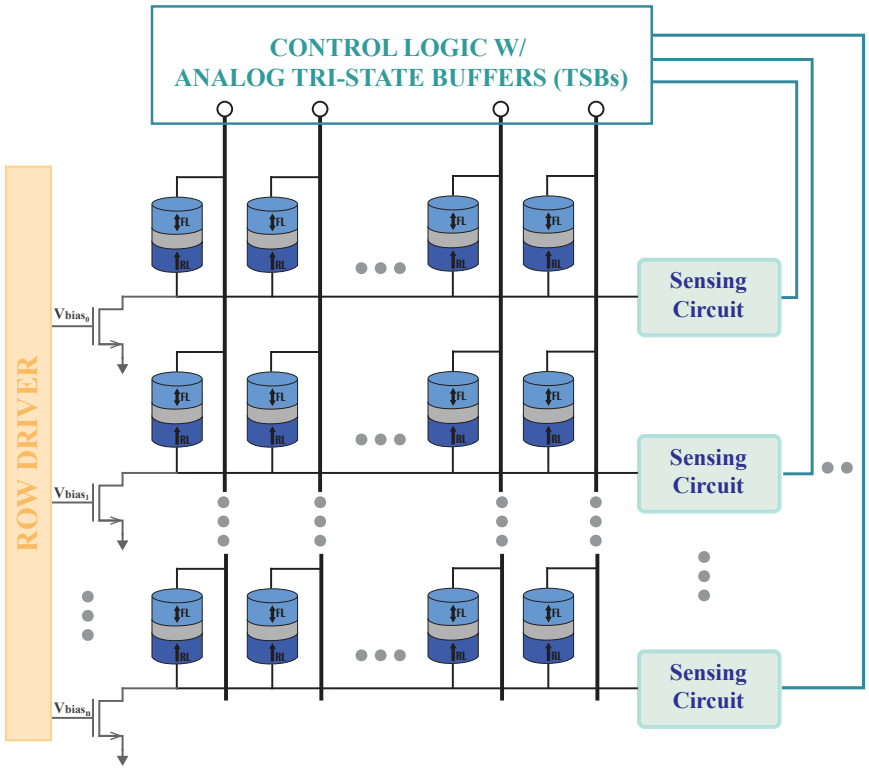


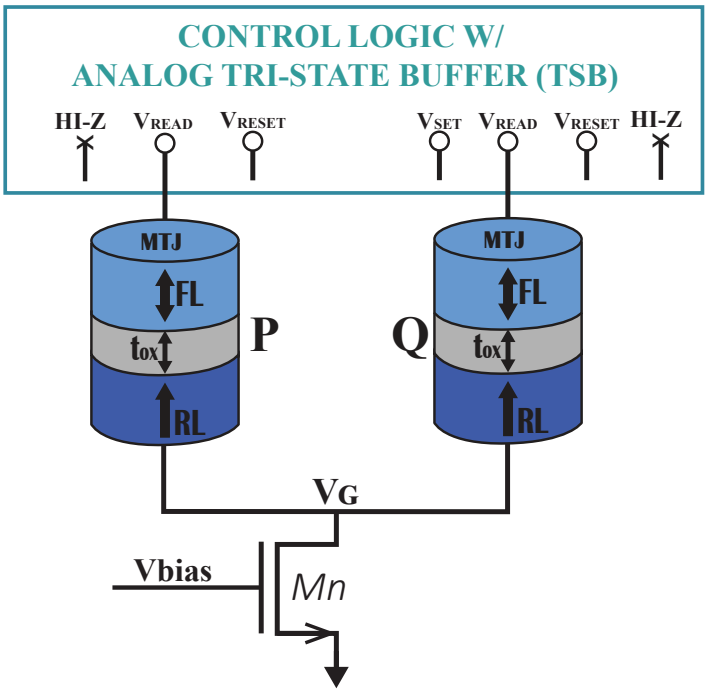
Figure 5.2: Top-level STT-MRAM SIMPLY architecture.

delivers the appropriate voltages to the STT-MRAM devices while the sensing circuit involves the topology needed to perform the IMPLY operation. The architecture also employs transistors to enable specific array columns and to connect adjacent columns.

5.2.1 Conventional IMPLY Scheme

The IMPLY logic is based on two logic operations: IMPLY and sFALSE. The IMPLY operation involves two inputs (P and Q) with a load resistor RG and one output, while the sFALSE operation is a one-input one-output operation that always results in a logic-'0'.

These operations can be implemented with MTJ devices, and replace the tail resistor [186] with a NMOS transistor (refer to the Mn in Figure 5.3), along with a control logic and analog tri-state buffers, to apply the appropriate voltages to the



(Q'= P IMPLY Q)

Input		Output
P	Q	Q'
0	0	1
0	1	1
1	0	0
1	1	1

Q' ≠ Q only for P=Q=0

Figure 5.3: (a) Conventional STT-MTJ based IMPLY logic gate circuit with a tail transistor (instead of resistor) and truth table of the IMPLY logic operation.

MTJs. IMPLY truth table presented at the right of Figure 5.3. To perform the P IMPLY Q operation, the control logic sends voltage pulses to P and Q. Then the output Q' is determined by the state read on Q.

The input P maintains its state regardless of the input combination, while Q should only switch from '0' to '1' in the case where P and Q are both '0'. As for the sFALSE operation, a negative voltage pulse is applied to a single device. Due to logic state degradation [181], the resistance that is supposed to keep the logic '0' during IMPLY computation gets slightly reduced. To deal with this, IMPLY operation is repeated. However, this leads into bit corruption after a few cycles, thus requiring inefficient memory refresh cycles [195].

5.2.2 SIMPLY Scheme

Figure 5.4 sketches the conventional SIMPLY scheme [183], which was introduced to overcome the shortcoming of the traditional IMPLY design [181–184].

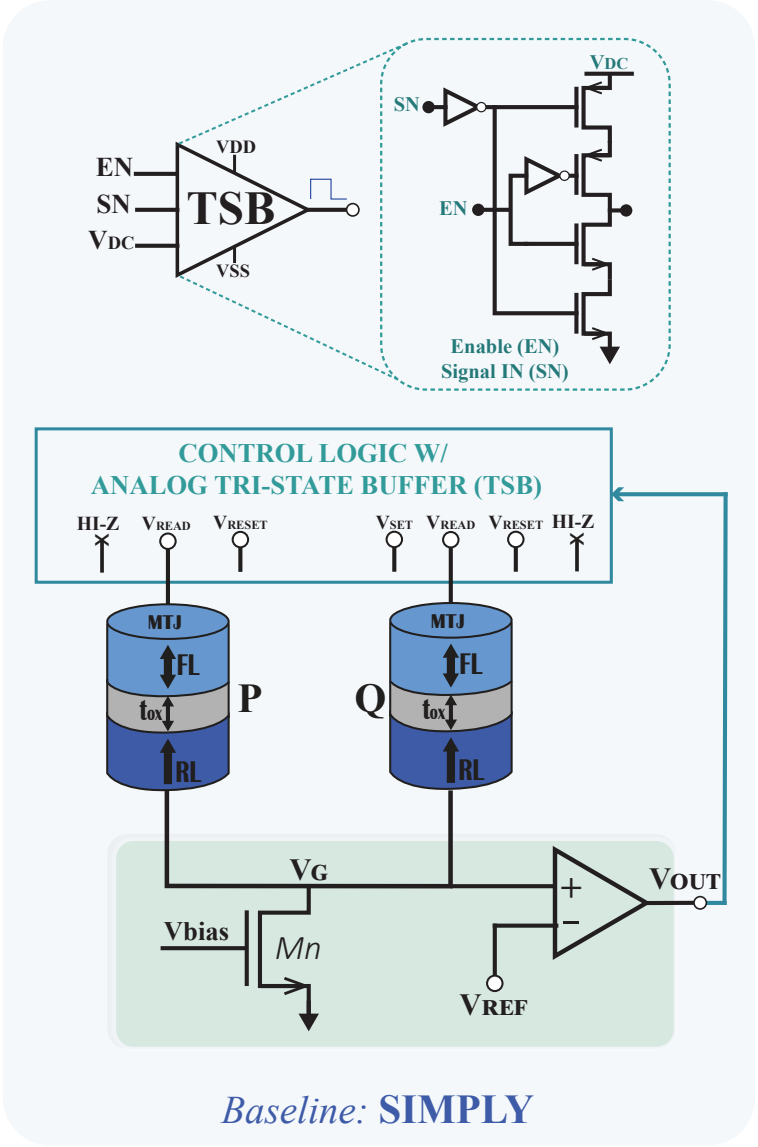


Figure 5.4: Conventional STT-MTJ based SIMPLY logic gate circuit with a tail transistor (instead of resistor).At the top: Tri-state buffer topology (TSB).

Indeed, both IMPLY and FALSE operations are performed more efficiently within this architecture as SIMPLY and sFALSE operations, respectively [182]. The IMPLY or SIMPLY operation involves two inputs and one output, according to the

truth table reported in Figure 5.3 (right). The inputs are represented by the initially stored states of the two MTJs P and Q in Figure 5.4, while the output is given by the final state of the MTJ Q after the execution. On the other hand, the FALSE or sFALSE operation is a one-input one-output operation that always results in a logic '0' stored in the considered MTJ. Accordingly, as shown in Figure 5.4, the core of the conventional SIMPLY logic scheme consists of two MTJs (P and Q) and an NMOS tail transistor (Mn) driven by the Vbias voltage, here used to implement the load resistor R_G [186]. Moreover, the SIMPLY scheme presents an output comparator and a control logic block including analog tri-state buffers (TSBs), shown at the top of Figure 5.4, to apply the appropriate voltages to the MTJs and to manage the execution of the operations.

Within this scheme, ensuring the proper execution of the P IMPLY Q operation involves maintaining the state of MTJ P, regardless of the input combination. At the same time, the MTJ Q should only switch from '0' to '1' when P = Q = '0'. A preliminary read operation is performed with the aim of distinguishing the input combination P = Q = '0' from all other possibilities. To accomplish this, a proper voltage pulse with an amplitude V_{READ} and width t_{READ} is applied the top electrode of both MTJ devices through the control logic block. Then the voltage V_G across the transistor Mn is compared to an appropriate reference voltage V_{REF} using the output comparator. In this way, the P = Q = '0' input combination is effectively detected, thus allowing the subsequent SET operation on MTJ Q to take place only in this specific case. This is achieved by applying an appropriate voltage pulse with amplitude V_{SET} and width t_{SET} on Q, while keeping the P driver in a high impedance (HI-Z) state, as shown in Figure 5.5(a). On the contrary, for the other input combinations the control logic forces both MTJ drivers into a HI-Z state, thus enabling significant energy savings [181, 183, 186], see Figure 5.5(b).

Similarly to the SIMPLY operation, the sFALSE operation also requires a

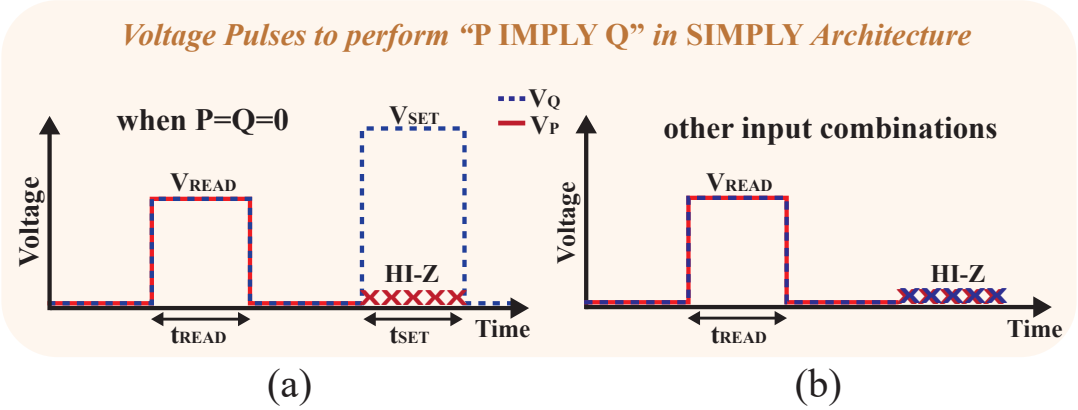


Figure 5.5: Timing diagram of applied voltage pulses when the sensing circuitry detects the input condition P=Q='0' and in all other cases [186].

preliminary read operation, but on a single device. Thereby, the subsequent RESET operation requiring a negative voltage pulse is performed only when the detected MTJ state is '1' [182].

5.2.3 Proposed SIMPLY+

Figure 5.6 shows the SIMPLY+ scheme proposed in this work and purposely designed to enhance the reliability of the preliminary read operation. This latter is the most critical operation for the STT-MRAM-based SIMPLY framework, owing to the relatively narrow read memory window offered by MTJ devices [186]. Our approach involves the use of a CS amplifier stage with a diode-connected load ($M1$-$M2$). The CS input is represented by the voltage V_G across the transistor Mn, while the output drives the gate terminal of the same transistor. This allows for significantly enlarging the read margins in terms of V_G voltages developed for the P = Q = '0' input combination and the others,thus enabling more reliable operation, as demonstrated in the subsequent subsection.

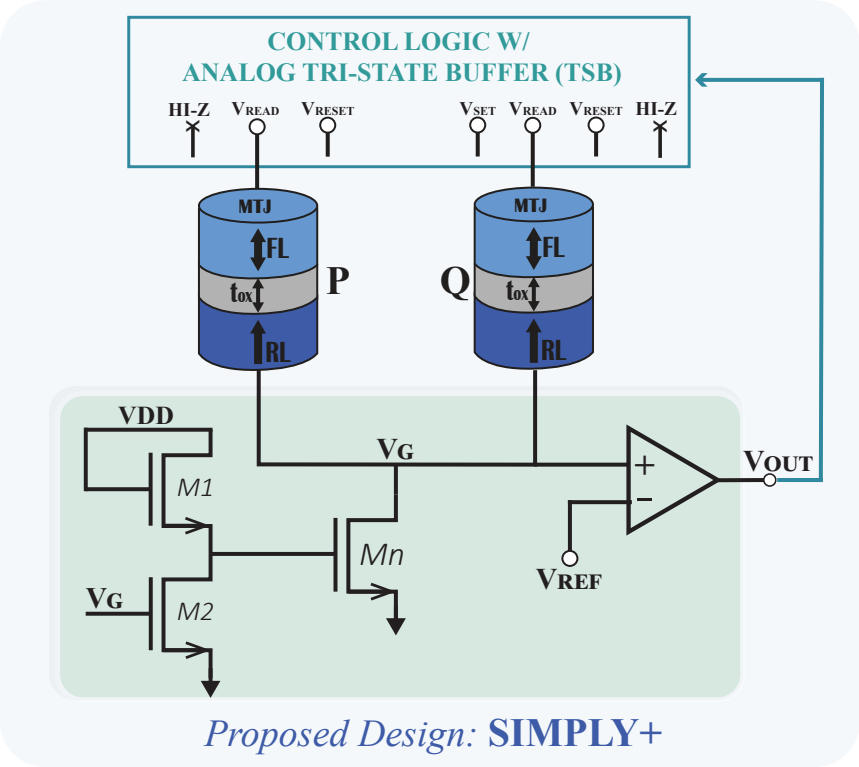


Figure 5.6: STT-MRAM-based SIMPLY+ scheme, including the tail transistor, a common source stage with diode-connected load, and the output comparator.

5.3 Simulation Results and Discussion

In the following, we will present and discuss the simulation results referred to the preliminary two-device read operation in the SIMPLY+ scheme, while also benchmarking it against the conventional SIMPLY counterpart. All the reported data is based on electrical simulations performed at room temperature (300 K) by using the Cadence Virtuoso environment. Transistors' modeling refers to a commercial 65 nm 1.2 V CMOS process, while our Verilog-A compact model [191] is used for the 30- nm STT-MTJ devices.

Figure 5.7 shows the timing diagram for key signals involved in the preliminary read operation when using both the conventional SIMPLY and the enhanced SIMPLY+ approaches. The data refers to a nominal simulation with the transistor

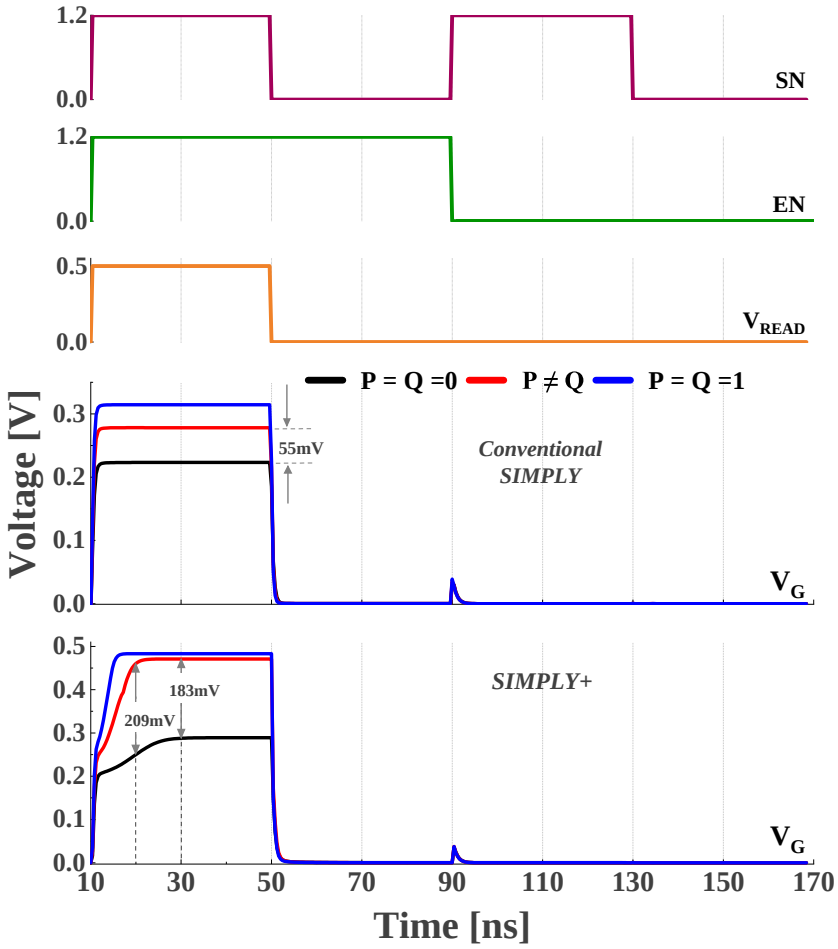


Figure 5.7: Timing diagram of the signals involved in the SIMPLY and SIMPLY+ schemes during the preliminary read operation as obtained from nominal simulations at 300 K considering Wn/Ln = 1 µm/3 µm size for the transistor Mn, V_{READ} = 0.5 V and Vbias = 1 V.

Mn size of Wn/Ln = 1 µm/3 µm and V_{READ} = 0.5 V in both schemes. In the SIMPLY scheme, Vbias is set to 1 V. From Figure 5.7, when the EN and SN signals are switched ON in the TSBs, the V_{READ} is set to 0.5 V. This results in a corresponding VG voltage across transistor Mn, dependent on the input combinations, , specifically the states of the MTJs P and Q. For the conventional SIMPLY architecture, the V_G voltages obtained for the scenarios where P = Q = '0' and P $\neq$ Q yield a nominal read margin (RM) of just 55 mV. On the other hand, due to the additional CS stage, the SIMPLY+ scheme exhibits a nominal RM of 209 mV

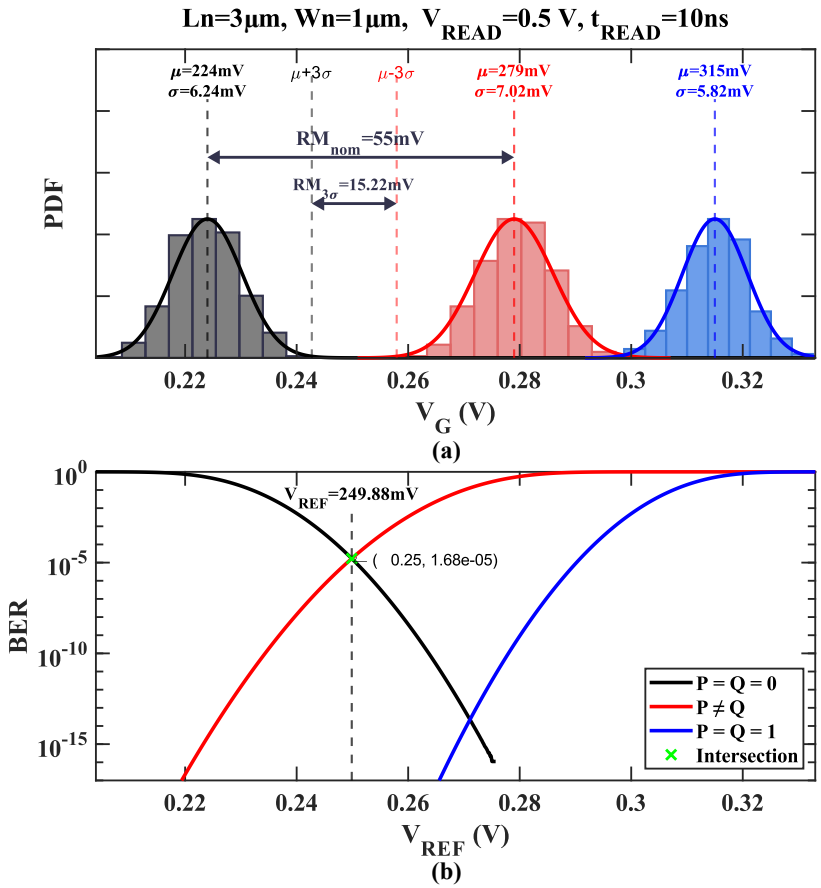


Figure 5.8: Simulation results of the conventional SIMPLY scheme for the preliminary read operation under process variations at 300 K considering Wn/Ln = 1 µm/3 µm size for the tail transistor Mn, V_{READ} = 0.5 V, Vbias = 1 V and t_{READ} =10 ns. (a) V_G statistical distributions for the different input combinations and (b) estimation of the bit error rate (BER) and reference voltage (V_{REF}).

at 10 ns and 183 mV at 20 ns. This leads to an improvement of 3.8 × and 3.3 ×, respectively, when compared to its conventional counterpart.

The reliability of the preliminary read operation in both schemes was also investigated while considering the effect of process variations on transistor and MTJ devices. This analysis was performed by means of extensive MC simulations. Transistor process variations were included using statistical models provided by the commercial PDK. For the MTJs, we assumed Gaussian distributed variations in the adopted Verilog-A compact model by setting the variability (defined as the ratio of the standard variation (σ) to the mean value (μ)) of 1% and 5% for the oxide

thickness (t_{OX}) and the cross-section area, respectively [185, 186, 191].

Figure 5.8(a)-(b), Figure 5.9(a)-(b) and Figure 5.10(a)-(b) show the MC simulation results obtained for the SIMPLY and SIMPLY+ architectures. More specifically, these figures report the statistical distributions of the voltage V_G for the different input combinations, while highlighting the corresponding estimated values for the RM, bit error rate (BER), and V_{REF}. The RM is evaluated both at the nominal corner (as given by the difference between the mean V_G values associated with the P = Q = '0' and P $\neq$ Q cases) and at the 3σ corner ($RM_{3\sigma}$), with σ being the standard deviation of V_G distributions. The BER refers to the failure probability in distinguishing the input combination P = Q ='0' from the other combinations during the preliminary read operation [186] and it is estimated by properly setting the V_{REF} used as input of the output comparator. In particular, the appropriate V_{REF} is determined by the voltage value that results in the same BER for the cases P = Q = '0' and P $\neq$ Q [104]. It is worth pointing out that in our analysis, the BER was evaluated by assuming an ideally stable V_{REF} and an ideal comparator with zero offset.

Figure 5.8(a)-(b) show the MC simulation results achieved within the conventional SIMPLY scheme for t_{READ} =10 ns. From Figure 5.8(a), the obtained nominal RM is equal to 55 mV, whereas the corresponding $RM_{3\sigma}$ is about 15 mV. From Figure 5.8(b), the V_{REF} to be used in the SIMPLY scheme is about 250 mV, which leads to a BER of 1.68×10^{-5} for the P = Q = '0' and P$\neq$Q input combinations, i.e., the worst-case BER.

Figure 5.9(a)-(b) report the statistical results of the SIMPLY+ scheme for t_{READ} =10 ns. The nominal RM and $RM_{3\sigma}$ values are respectively 202.5 mV and 103.4 mV, i.e., 3.7 $\times$ and 6.8 $\times$ larger than the conventional SIMPLY scheme. The appropriate V_{REF} is 362.9 mV, which corresponds to a worst-case BER of 4.37×10^{-10}, i.e., more than four orders of magnitude better as compared to its conventional counterpart.

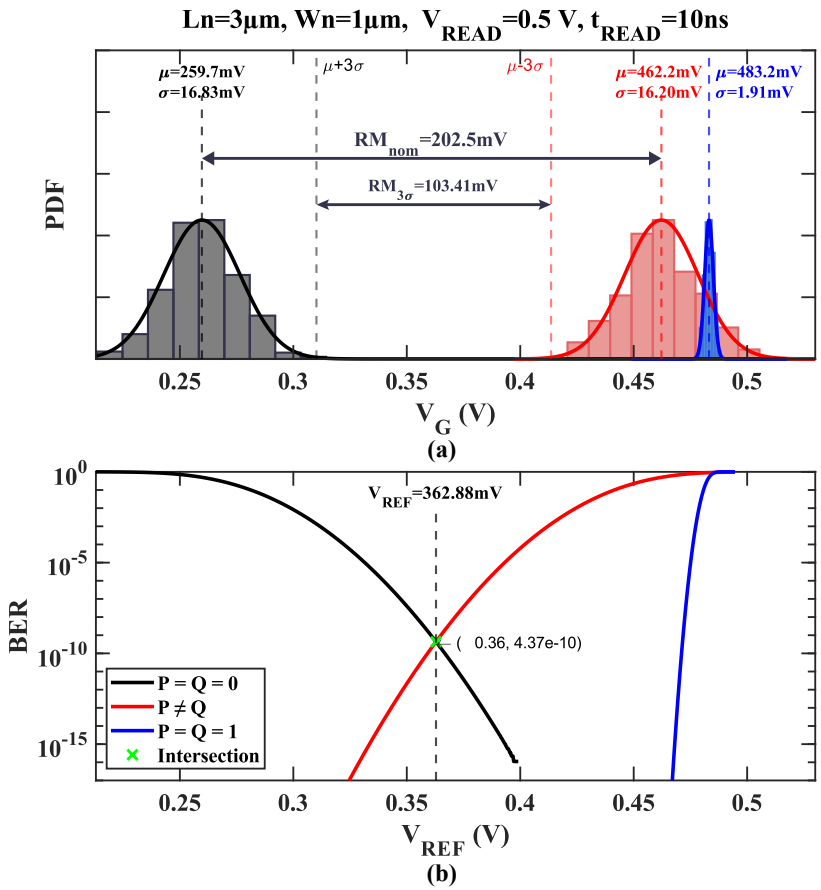


Figure 5.9: Simulation results of the SIMPLY+ scheme for the preliminary read operation under process variations at 300 K considering Wn/Ln = 1 µm/3 µm size for the tail transistor Mn, V_{READ} = 0.5 V and t_{READ} =10 ns. (a) V_G statistical distributions for the different input combinations and (b) estimation of the bit error rate (BER) and reference voltage (V_{REF}).

The reliability of the preliminary read operation in the SIMPLY+ scheme can be further improved by enlarging the read voltage pulse duration. This can be observed in Figure 5.10(a)-(b), which show the statistical results of the SIMPLY+ scheme for t_{READ} =20 ns. Indeed, despite a reduction of the nominal RM down to 181.1 mV compared to the 202.5 mV obtained at t_{READ} =10 ns, increasing the t_{READ} up to t_{READ} =20 ns leads to a $RM_{3\sigma}$ of about 121.6 mV, i.e., 8 × and 1.2 × larger than the conventional SIMPLY scheme and the SIMPLY+ scheme at t_{READ} =10 ns, respectively, owing to the reduced standard deviation values of V_G distributions.

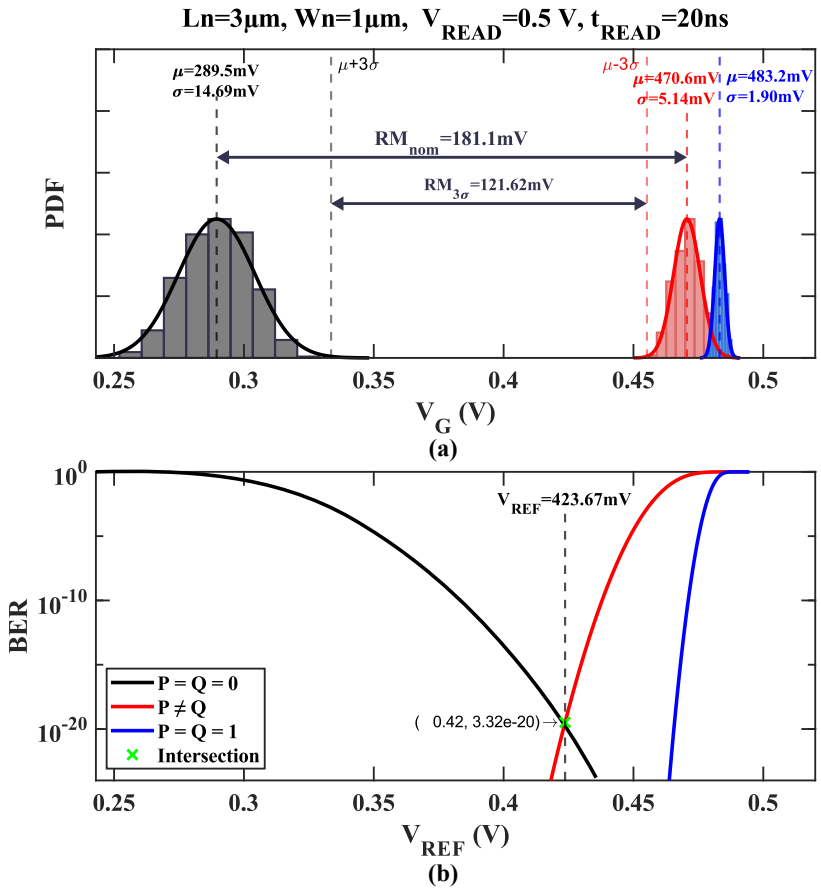


Figure 5.10: Simulation results of the SIMPLY+ scheme for the preliminary read operation under process variations at 300 K considering Wn/Ln = 1 µm/3 µm size for the tail transistor Mn, V_{READ} = 0.5 V and t_{READ} =20 ns. (a) V_G statistical distributions for the different input combinations and (b) estimation of the bit error rate (BER) and reference voltage (V_{REF}).

This results into a worst-case BER of 3.32×10^{-20} at V_{REF} = 423.7 mV, which corresponds to an improvement by more than fourteen and ten orders of magnitude as compared to the the conventional SIMPLY scheme and the SIMPLY+ scheme for t_{READ} =10 ns, respectively. Obviously, such an improvement comes at the cost of higher energy consumption. This can be observed in Table 5.2, which summarizes the comparative results obtained within the SIMPLY and SIMPLY+ schemes for the preliminary read operation under process variations.

Our analysis was also extended with the aim of evaluating the effect of the tail

Table 5.2: Comparative results SIMPLY vs SIMPLY+ for the preliminary read operation under process variations

	SIMPLY (t_{READ} =10 ns)		**SIMPLY+ (t_{READ} =10 ns)**		**SIMPLY+ (t_{READ} =20 ns)**	
	P=Q='0'	**P≠Q**	**P=Q='0'**	**P≠Q**	**P=Q='0'**	**P≠Q**
μ of V_G (mV)	224.0	279.0	259.7	462.2	289.5	470.6
σ of V_G (mV)	6.24	7.02	16.83	16.20	14.69	5.14
RM_{nom} (mV)	55.0		202.5		181.1	
$RM_{3\sigma}$ (mV)	15.2		103.4		121.6	
Worst-case BER	1.7×10^{-5}		4.4×10^{-10}		3.3×10^{-20}	
Energy (pJ)	1.73		2.60		5.19	

transistor (Mn) sizing on SIMPLY+ performance during the preliminary read operation. In this regard, Figure 5.11(a)-(f) show the color maps of the nominal RM, the RM at the 3σ corner, the worst-case read disturbance rate (RDR), i.e., referred to the case P = Q ='0' [186], the V_{REF}, the worst-case BER, and the worst-case overall read error rate (RER), both referred again to the case P = Q ='0' [186]. All this data was obtained from statistical simulations at 300 K, V_{READ} = 0.5 V and t_{READ} =10 ns while varying the size (Ln and Wn) of Mn, i.e., its strength. In particular, the RDR is an important metric to assess the reliability of the read operation performed within the SIMPLY/SIMPLY+ framework, as it refers to the probability of unintentionally switching the stored data during this operation [38, 186]. Accordingly, for a given Mn size, the overall RER is given by the combination of the RDR and BER. From Figure 5.11(a)-(b), we can observe that the Mn sizing strongly affects the RM. More specifically, we can identify a relatively small design space for size around Wn/Ln = 1 µm/3 µm leading to nominal RM and $RM_{3\sigma}$ values in the neighborhood of 200 mV and 100 mV, respectively. More precisely, we obtain a nominal RM of

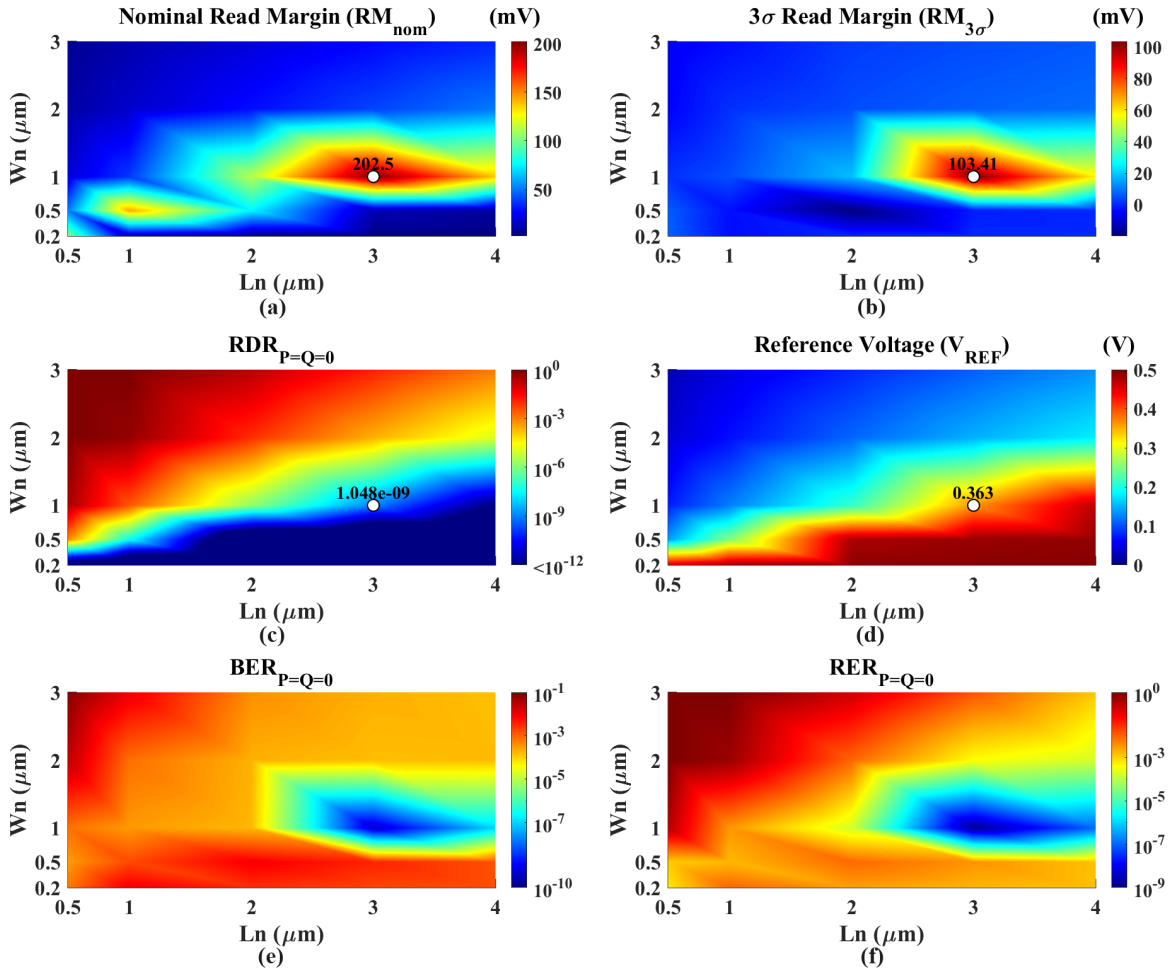


Figure 5.11: Simulation results of the SIMPLY+ scheme for the preliminary read operation under process variations at 300 K, V_{READ} = 0.5 V and t_{READ} =10 ns while varying the size (Ln and Wn) of the tail transistor Mn: (a) nominal read margin (RM), (b) RM at the 3σ corner, (c) worst-case read disturbance rate (RDR) referred to the case P = Q =‘0’, (d) reference voltage (V_{REF}), (e) worst-case bit error rate (BER) referred to the case P = Q =‘0’, and (f) worst-case overall read error rate (RER) again referred to the case P = Q =‘0’.

202.5 mV and a $RM_{3\sigma}$ of 103.4 mV for Wn/Ln = 1 µm/3 µm, according to Figure 5.9(a). From Figure 5.11(c), the worst-case RDR increases when increasing the strength of the transistor Mn, i.e., for larger (smaller) Wn (Ln), as given by the corresponding increase in the current flowing through the MTJ devices. In particular, for Wn/Ln = 1 µm/3 µm we obtain a worst-case RDR equal to 1.05×10^{-9}.

An opposite trend as compared to that of the RDR can be seen for the V_{REF} in Figure 5.11(d), where its value tends to increase when decreasing the transistor strength, i.e., when increasing its resistance. From Figure 5.11(e), the worst-case BER expectedly shows a similar trend to the RM, hence with an optimal design space for Mn size around Wn/Ln = 1 µm/3 µm leading to BER values in the order of 10^{-10} as in Figure 5.9(b). The discussed trends of the RDR and BER thus result in the map of the RER shown in Figure 5.11(f), where its optimal value of 1.05×10^{-9} is achieved at Wn/Ln = 1 µm/3 µm size, i.e., that used in the above analysis.

5.4 Improved SIMPLY+ design

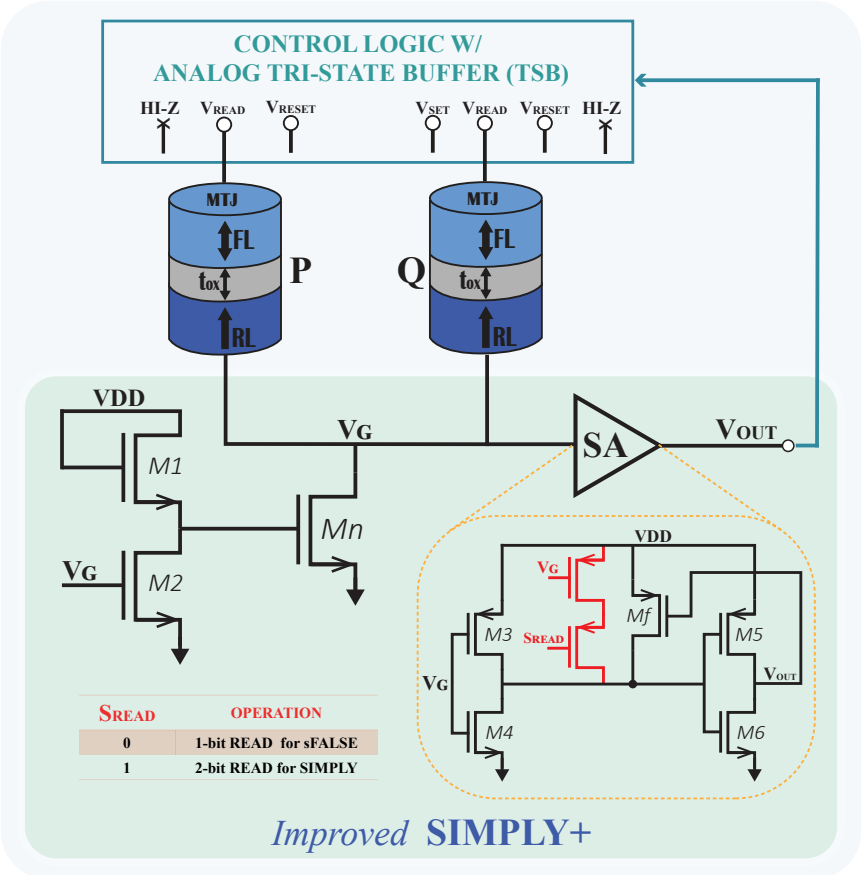


Figure 5.12: (a) Improved SIMPLY+ scheme, including a common source (CS) stage with diode-connected load and a two-stage inverter as output block.

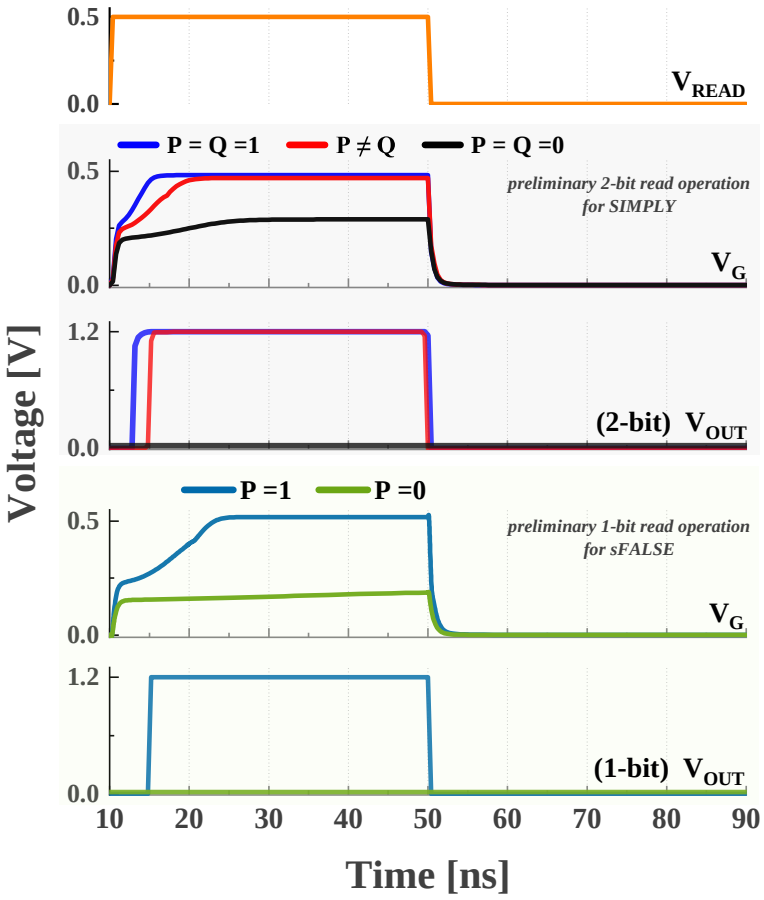


Figure 5.13: Time diagram of V_{READ}, V_G and V_{OUT} signals involved in the preliminary 1-bit and 2-bit read operations to be performed respectively for sFALSE and SIMPLY operations within the improved SIMPLY+ scheme of Figure 5.12, as obtained from nominal simulations at 300 K and V_{READ} = 0.5 V.

In addition to the introduction of the CS stage as discussed in the previous section, we propose a further modification of the conventional SIMPLY scheme, which consists of replacing the conventional sense amplifier-based output comparator [182, 185, 186] with a lower complexity two-stage buffer, thus resulting in an enhanced SIMPLY+ scheme, as depicted in Figure 5.12. The two-stage inverter-based output block acts as a sense amplifier to discriminate the V_G values associated with the different input combinations, using the logic threshold as the reference voltage. To this aim, skewed inverters are employed to strengthen the

output data (V_{OUT}) and to improve the capability to discriminate the input combinations.

Furthermore, transistors highlighted in red in Figure 5.12 are used to enable both 1-bit and 2-bit read operations within the same circuit block, respectively needed to execute sFALSE and SIMPLY operations. This is achieved by properly setting the S_{READ} signal, which allows adjusting the logic threshold of the first inverter in the output block on the basis of the operation to be performed. More specifically, S_{READ} = '0' ('1') enables the 1-bit (2-bit) read operation for the sFALSE (SIMPLY) execution, as reported in Figure 5.12. The proposed approach is demonstrated in Figure 5.13, which shows the timing diagram of the V_{READ}, V_G and V_{OUT} signals when performing the 2-bit and 1-bit read operations within the scheme of Figure 5.12. Here, data refers to nominal simulations at 300 K and V_{READ} = 0.5 V. From this figure, we can observe that the proper tuning of the logic threshold through the S_{READ} signal allows distinguishing the input combination P = Q ='0' (i.e., V_{OUT} ='0') from the others (i.e., V_{OUT} ='1') in the 2-bit read operation, as well as the case P ='0' (i.e., V_{OUT} ='0') from the case P ='1' (i.e., V_{OUT} ='1') in the 1-bit read operation. The improved SIMPLY+ presents an average energy consumption of about 2.07 pJ. This is $\sim 25\%$ less as compared to the SIMPLY+ scheme of Figure 5.12.

5.5 Conclusion

We proposed SIMPLY+, a reliability enhanced STT-MTJ-based LIM logic scheme.

Our design exploits a commercial 65 nm CMOS PDK, as well as a macrospin-based Verilog-A compact model used to describe the behavior of a 30 nm diameter perpendicular STT-MTJ device. We evaluate SIMPLY+ circuit performance by means of exhaustive Monte Carlo simulations. As a main result of our study, the SIMPLY+ nominal read margin is about $4\times$ compared with the

convectional SIMPLY scheme. The SIMPLY+ scheme proves a better performance, accordingly the BER and RDR by more than four orders of magnitud. In addition the improved SIMPLY+ achieves higher energy efficient, meaning an energy saving of $\sim 25\%$, as compared with the SIMPLY+ scheme.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The earlier chapters of this thesis focused on overcoming the key challenges that hardware solutions for ANNs have faced, especially when exploring the integration of emerging memory devices like STT-MRAMs with CMOS technology. This chapter summarizes the main contributions and presents areas for research improvement.

6.1 Key thesis contribution and related future work

As starting point, a sigmoid and softmax AF were introduced to explore the capabilities of an ANN at the output node. In Chapter 3, the proposed solutions demonstrated optimal performance compared to existing references. A sigmoidal V-to-V neuron with only one bias voltage is proposed. The sigmoidal circuit proposed has been evaluated for different values of steepness parameter. The simulation results show that the proposed design approximates the sigmoid function more accurately than previous designs, being superior in terms of power consumption, error, and area.

Contrastingly, a low-power, low-voltage analog implementation of the softmax activation function used in deep neural networks is proposed. The softmax circuit are good match to the theoretical function, leading to good stability performance against process and temperature variations. The design occupies a small area and low power consumption compared to the digital counterparts. The design shows limited precision degradation and lower average relative errors, with respect to the

theoretical softmax equation.

In regard to the integration of emerging memory devices (STT-MRAMs) with CMOS technology, Chapter 4 evaluates the impact of STT-MRAM cells based on DMTJ against the conventional SMTJ counterpart on the performance of a two-layer MLP neural network. The STT-MTJ synaptic pseudo-crossbar array architecture and device/transistor models in NeuroSim is explored. Considering the NeuroSim emulator to evaluate the learning accuracy with 2-layer MPL neural networks at the run-time of online learning in eNVM devices such as MTJ-based STT-MRAM. Our results show that, at parity of TMR and oxide thickness, as compared to the conventional SMTJ-based alternative, the DMTJ-based solution proves to be faster during feed forward and weight update operations, more energy efficient, and less area hungry at an energy-optimal bitcell configuration/size. This occurs while also achieving a high accuracy when running the neural network with the MNIST dataset. Our study suggests that DMTJ-based eNVM synaptic cores are good candidates to replace conventional SRAM-based solutions.

Likewise, Chapter 5 proposes SIMPLY+, a new architecture designed for in-memory computing built from the smart material implication logic and perpendicular MTJ based STT-MRAM technologies. The proposed architecture is benchmarked against its conventional counterpart. Obtained results show a significant improvement in terms of reliability in terms of nominal read margin and exhibit a better performance in terms of BER. Our results prove that the SIMPLY+ scheme is a very promising solution for designing reliable in-memory computing architectures. Such results prove that SIMPLY+ scheme is an outstanding solution for the development of reconfigurable in-memory computing architectures.

Finally, the capabilities of today's hardware platforms are limited by the need to transfer large volumes of data between memory and compute units, also known as the memory wall [196]. The existing hardware accelerator confronts several challenges

in achieving a design that meets the desired performance and cost criteria. These challenges encompass power/energy consumption, throughput, area, speed, learning performance, and resource consumption.

To address these challenges, future work will be centered on performing analog in-memory vector-matrix operations through the integration of emerging technologies. The overarching goal is to develop robust architecture models, focusing on frameworks like the STT-MTJ-based SIMPLY+ logic scheme which is performed for BNN inference. SIMPLY+ can be used as an effective solution for the in-memory computation of logic operations (e.g., XNOR [195], full adders [197]). Also thanks to its reconfigurability, SIMPLY+ enables the possibility to easily implement different neural networks topologies.

Furthermore, exploring new approaches will be pivotal in this endeavor. The main objective is to exploit novel architectures to fulfill optimal computational requirements. By leveraging emerging technologies and innovative design paradigms, future work aims to overcome the limitations of existing hardware accelerators and usher in a new era of high-performance, cost-effective computing solutions.

6.2 Impact of Logic-in Memory

LIM is a solution to overcome the limitations of Von Neumann's architecture. By integrating simple logic circuits within or near memory elements, local computations can be performed without the need to carry data from the main memory.

From a design perspective, the integration of logic modules with STT-MRAM memory is highly feasible due to the seamless compatibility between STT-MRAMs and CMOS circuits. This integration allows for the implementation of logic using non-volatile or CMOS logic, as well as the modification of the readout circuit to

enable logic operations in the analog domain. It notable that this approach not only proves advantageous for standard CMOS technology but also leverages the potential of emerging technologies.

Moreover, several proposed sensing schemes have demonstrated improvements in reliability, energy efficiency, and area efficiency within LIM architectures. These findings highlight that the impact of this thesis extends beyond the realm of STT-MRAM memories and has the potential to shape the development of promising architectures appropriate for high-priority tasks.

REFERENCES

[1] M. Vatalaro, **T. Moposita**, S. Strangio, L. Trojman, A. Vladimirescu, M. Lanuzza, and F. Crupi, "A low-voltage, low-power reconfigurable current-mode softmax circuit for analog neural networks," *Electronics*, vol. 10, no. 9, p. 1004, 2021.

[2] **T. Moposita**, E. Garzón, F. Crupi, L. Trojman, A. Vladimirescu, and M. Lanuzza, "Efficiency of Double-Barrier Magnetic Tunnel Junction-Based Digital eNVM Array for Neuro-Inspired Computing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 3, pp. 1254–1258, 2023.

[3] **T. Moposita**, L. Trojman, F. Crupi, M. Lanuzza, and A. Vladimirescu, "Voltage-to-voltage sigmoid neuron activation function design for artificial neural networks," in *2022 IEEE 13th Latin America Symposium on Circuits and System (LASCAS)*, 2022, pp. 1–4.

[4] **T. Moposita**, E. Garzón, F. Crupi, L. Trojman, A. Vladimirescu, and M. Lanuzza, "Efficiency of Double-Barrier Magnetic Tunnel Junction-Based Digital eNVM Array for Neuro-Inspired Computing," in *2023 IEEE 14th Latin America Symposium on Circuits and System (LASCAS)*, IEEE, 2023.

[5] E. Garzón, B. Zambrano, **T. Moposita**, R. Taco, L.M. Prócel, and L. Trojman, "Reconfigurable CMOS/STT-MTJ non-volatile circuit for logic-in-memory applications," in *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, 2020, pp. 1–4.

[6] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[7] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1253,

[8] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–6.

[9] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[10] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, pp. 529–544, 2020.

[11] 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.

[12] J. Zhang, Z. Wang, and N. Verma, "A machine-learning classifier implemented in a standard 6T SRAM array," *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pp. 1–2, 2016.

[13] S. K. Gonugondla, M. Kang, and N. R. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pp. 490–492, 2018.

[14] H. Yang *et al.*, "Scaling of 32nm low power SRAM with high-K metal gate," in *2008 IEEE International Electron Devices Meeting*, IEEE, 2008, pp. 1–4.

[15] Z. Guo *et al.*, "10-nm SRAM design using gate-modulated self-collapse write-assist enabling 175-mV V MIN reduction with negligible active power overhead," *IEEE Solid-State Circuits Letters*, vol. 4, pp. 6–9, 2020.

[16] J. Chang *et al.*, "12.1 a 7nm 256mb sram in high-k metal-gate finfet technology with write-assist circuitry for low-v min applications," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, IEEE, 2017, pp. 206–207.

[17] M. N. I. Khan and S. Ghosh, "Comprehensive study of security and privacy of emerging non-volatile memories," *Journal of Low Power Electronics and Applications*, vol. 11, no. 4, 2021.

[18] M. N. I. Khan, S. Bhasin, B. Liu, A. Yuan, A. Chattopadhyay, and S. Ghosh, "Comprehensive study of side-channel attack on emerging non-volatile memories," *Journal of Low Power Electronics and Applications*, vol. 11, no. 4, 2021.

[19] W. Law and S. Wong, *Emerging Non-Volatile Memory Technologies*. Springer, 2021.

[20] S. Oh, "Energy Efficient Hardware Implementation of Neural Networks Using Emerging Non-Volatile Memory Devices," Ph.D. dissertation, UC San Diego, 2023.

[21] B. Li, B. Yan, and H. Li, "An overview of in-memory processing with emerging non-volatile memory for data-intensive applications," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, 2019, pp. 381–386.

[22] M. Si, H.-Y. Cheng, T. Ando, G. Hu, and P. D. Ye, "Overview and outlook of emerging non-volatile memories," *MRS Bulletin*, vol. 46, no. 10, pp. 946–958, 2021.

[23] 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. 100 004, 2022.

[24] G. Krishnan *et al.*, "Exploring model stability of deep neural networks for reliable rram-based in-memory acceleration," *IEEE Transactions on Computers*, vol. 71, no. 11, pp. 2740–2752, 2022.

[25] E. P.-B. Quesada *et al.*, "Experimental assessment of multilevel rram-based vector-matrix multiplication operations for in-memory computing," *IEEE Transactions on Electron Devices*, vol. 70, no. 4, pp. 2009–2014, 2023.

[26] X. Wang, W. Li, Z. Luo, K. Wang, and S. P. Shah, "A critical review on phase change materials (PCM) for sustainable and energy efficient building: Design, characteristic, performance and application," *Energy and Buildings*, vol. 260, p. 111 923, 2022.

[27] A. Ehrmann, T. Blachowicz, G. Ehrmann, and T. Grethe, "Recent developments in phase-change memory," *Applied Research*, e202200024, 2022.

[28] N. Li *et al.*, "Optimization of Projected Phase Change Memory for Analog In-Memory Computing Inference," *Advanced Electronic Materials*, p. 2 201 190, 2022.

[29] T. Moposita, E. Garzón, F. Crupi, L. Trojman, A. Vladimirescu, and M. Lanuzza, "Efficiency of Double-Barrier Magnetic Tunnel Junction-Based Digital eNVM Array for Neuro-Inspired Computing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 3, pp. 1254–1258, 2023.

[30] E. Garzón, R. Taco, L. M. Prócel, L. Trojman, and M. Lanuzza, "Voltage and Technology Scaling of DMTJ-based STT-MRAMs for Energy-Efficient Embedded Memories," in *2022 IEEE 13th Latin America Symposium on Circuits and System (LASCAS)*, IEEE, 2022, pp. 01–04.

[31] A. Musello, E. Garzón, M. Lanuzza, L. M. Prócel, and R. Taco, "XNOR-bitcount operation exploiting computing-in-memory with STT-MRAMs," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 3, pp. 1259–1263, 2023.

[32] T. Soliman *et al.*, "FELIX: A ferroelectric FET based low power mixed-signal in-memory architecture for DNN acceleration," *ACM Transactions on Embedded Computing Systems*, vol. 21, no. 6, pp. 1–25, 2022.

[33] S. Thomann, H. Nguyen, P. Genssler, and H. Amrouch, "All-in-memory brain-inspired computing using fefet synapses," *Frontiers in Electronics*, vol. 3, 2022.

[34] S. Chatterjee, S. Kumar, A. Gaidhane, C. K. Dabhi, Y. S. Chauhan, and H. Amrouch, "Ferroelectric FDSOI FET modeling for memory and logic applications," *Solid-State Electronics*, vol. 200, p. 108 554, 2023.

[35] K. Roy, I. Chakraborty, M. Ali, A. Ankit, and A. Agrawal, "In-memory computing in emerging memory technologies for machine learning: An overview," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, IEEE, 2020, pp. 1–6.

[36] N. Sayed, R. Bishnoi, F. Oboril, and M. B. Tahoori, "A cross-layer adaptive approach for performance and power optimization in STT-MRAM," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2018, pp. 791–796.

[37] E. Garzon, R. De Rose, F. Crupi, L. Trojman, and M. Lanuzza, "Assessment of STT-MRAM performance at nanoscaled technology nodes using a device-to-memory simulation framework," *Microelectronic Engineering*, vol. 215, p. 111 009, 2019.

[38] E. Garzon *et al.*, "Assessment of STT-MRAMs based on double-barrier MTJs for cache applications by means of a device-to-system level simulation framework," *Integration*, vol. 71, pp. 56–69, 2020.

[39] J.-M. Hung *et al.*, "An 8-Mb DC-current-free binary-to-8b precision ReRAM nonvolatile computing-in-memory macro using time-space-readout with 1286.4-21.6 TOPS/W for edge-AI devices," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, IEEE, vol. 65, 2022, pp. 1–3.

[40] W. Ye *et al.*, "A 28-nm RRAM Computing-in-Memory Macro Using Weighted Hybrid 2T1R Cell Array and Reference Subtracting Sense Amplifier for AI Edge Inference," *IEEE Journal of Solid-State Circuits*, 2023.

[41] N. Noor, S. Muneer, R. S. Khan, A. Gorbenko, and H. Silva, "Enhancing Programming Variability in Multi-Bit Phase Change Memory Cells for Security," *IEEE Transactions on Nanotechnology*, vol. 19, pp. 820–828, 2020.

[42] W. Chen, D. Li, Y. Zhong, and Y. Tang, "A Novel Non-Volatile Memory Update Mechanism for 6G Edge Computing," *IEEE Transactions on Network Science and Engineering*, 2022.

[43] P. Barla, V. K. Joshi, and S. Bhat, "Spintronic devices: a promising alternative to CMOS devices," *Journal of Computational Electronics*, vol. 20, no. 2, pp. 805–837, 2021.

[44] A. Chen, "A review of emerging non-volatile memory (NVM) technologies and applications," *Solid-State Electronics*, vol. 125, pp. 25–38, 2016, Extended papers selected from ESSDERC 2015.

[45] S. Kim *et al.*, "NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning," in *2015 IEEE international electron devices meeting (IEDM)*, IEEE, 2015, pp. 17–1.

[46] H.-S. P. Wong *et al.*, "Phase change memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010.

[47] M. N. Baibich *et al.*, "Giant magnetoresistance of (001) Fe/(001) Cr magnetic superlattices," *Physical review letters*, vol. 61, no. 21, p. 2472, 1988.

[48] I. Ennen, D. Kappe, T. Rempel, C. Glenske, and A. Hütten, "Giant magnetoresistance: basic concepts, microstructure, magnetic interactions and applications," *Sensors*, vol. 16, no. 6, p. 904, 2016.

[49] N. Tezuka and T. Miyazaki, "Giant magnetic tunneling effect in Fe/Al2O3/Fe junction," *Journal of Magnetism and Magnetic Materials*, vol. 139, pp. L231–L234, 1995.

[50] J. Gregg, I. Petej, E. Jouguelet, and C. Dennis, "Spin electronics—a review," *Journal of Physics D: Applied Physics*, vol. 35, no. 18, R121, 2002.

[51] Y. Emre, C. Yang, K. Sutaria, Y. Cao, and C. Chakrabarti, "Enhancing the reliability of STT-RAM through circuit and system level techniques," in *2012 IEEE Workshop on Signal Processing Systems*, IEEE, 2012, pp. 125–130.

[52] D. Wang, C. Nordman, J. Daughton, Z. Qian, and J. Fink, "70% TMR at room temperature for SDT sandwich junctions with CoFeB as free and reference Layers," *IEEE Transactions on Magnetics*, vol. 40, no. 4, pp. 2269–2271, 2004.

[53] J. Mathon and A. Umerski, "Theory of tunneling magnetoresistance of an epitaxial Fe/MgO/Fe (001) junction," *Physical Review B*, vol. 63, no. 22, p. 220 403, 2001.

[54] W. Butler, X.-G. Zhang, T. Schulthess, and J. MacLaren, "Spin-dependent tunneling conductance of Fe— MgO— Fe sandwiches," *Physical Review B*, vol. 63, no. 5, p. 054 416, 2001.

[55] S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzuki, and K. Ando, "Giant room-temperature magnetoresistance in single-crystal Fe/MgO/Fe magnetic tunnel junctions," *Nature materials*, vol. 3, no. 12, pp. 868–871, 2004.

[56] S. S. Parkin *et al.*, "Giant tunnelling magnetoresistance at room temperature with MgO (100) tunnel barriers," *Nature materials*, vol. 3, no. 12, pp. 862–867, 2004.

[57] D. D. Djayaprawira *et al.*, "230% room-temperature magnetoresistance in CoFeB/ MgO/ CoFeB magnetic tunnel junctions," *Applied physics letters*, vol. 86, no. 9, 2005.

[58] W. Jin, G. Zhang, H. Wu, L. Yang, W. Zhang, and H. Chang, "Room-Temperature and Tunable Tunneling Magnetoresistance in Fe3GaTe2-Based 2D van der Waals Heterojunctions," *ACS Applied Materials & Interfaces*, vol. 15, no. 30, pp. 36 519–36 526, 2023.

[59] K. Ando *et al.*, "Spin-transfer torque magnetoresistive random-access memory technologies for normally off computing," *Journal of Applied Physics*, vol. 115, no. 17, 2014.

[60] Y. Chen, H. H. Li, I. Bayram, and E. Eken, "Recent technology advances of emerging memories," *IEEE Design & Test*, vol. 34, no. 3, pp. 8–22, 2017.

[61] T. Endoh, H. Koike, S. Ikeda, T. Hanyu, and H. Ohno, "An overview of nonvolatile emerging memories—Spintronics for working memories," *IEEE*

journal on emerging and selected topics in circuits and systems, vol. 6, no. 2, pp. 109–119, 2016.

[62] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of candidate device technologies for storage-class memory," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 449–464, 2008.

[63] D. Apalkov *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.

[64] D. M. Bromberg, D. H. Morris, L. Pileggi, and J.-G. Zhu, "Novel STT-MTJ device enabling all-metallic logic circuits," *IEEE transactions on Magnetics*, vol. 48, no. 11, pp. 3215–3218, 2012.

[65] Z. Xu, C. Yang, M. Mao, K. B. Sutaria, C. Chakrabarti, and Y. Cao, "Compact modeling of STT-MTJ devices," *Solid-State Electronics*, vol. 102, pp. 76–81, 2014.

[66] K. Cao *et al.*, "Novel metallization processes for sub-100 nm magnetic tunnel junction devices," *Microelectronic Engineering*, vol. 209, pp. 6–9, 2019.

[67] L. Zhang *et al.*, "A high-reliability and low-power computing-in-memory implementation within STT-MRAM," *Microelectronics Journal*, vol. 81, pp. 69–75, 2018.

[68] K. C. Chun, H. Zhao, J. D. Harms, T.-H. Kim, J.-P. Wang, and C. H. Kim, "A scaling roadmap and performance evaluation of in-plane and perpendicular MTJ based STT-MRAMs for high-density cache memory," *IEEE journal of solid-state circuits*, vol. 48, no. 2, pp. 598–610, 2012.

[69] J. Choe, "Memory technology 2021: Trends & challenges," in *2021 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, IEEE, 2021, pp. 111–115.

[70] I. Yoon, M. A. Anwar, R. V. Joshi, T. Rakshit, and A. Raychowdhury, "Hierarchical memory system with STT-MRAM and SRAM to support transfer and real-time reinforcement learning in autonomous drones," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 3, pp. 485–497, 2019.

[71] 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.

[72] H. Cai *et al.*, "Proposal of analog in-memory computing with magnified tunnel magnetoresistance ratio and universal STT-MRAM cell," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 4, pp. 1519–1531, 2022.

[73] G. Zhang and Y. Jiang, "Novel multi-bit parallel pipeline-circuit design for STT-MRAM," *AIP Advances*, vol. 13, no. 2, 2023.

[74] R. De Rose *et al.*, "A Compact Model with Spin-Polarization Asymmetry for Nanoscaled Perpendicular MTJs," *IEEE Transactions on Electron Devices*, vol. 64, no. 10, pp. 4346–4353, 2017.

[75] R. De Rose, M. d'Aquino, G. Finocchio, F. Crupi, M. Carpentieri, and M. Lanuzza, "Compact modeling of perpendicular STT-MTJs with double reference layers," *IEEE Transactions on Nanotechnology*, vol. 18, pp. 1063–1070, 2019.

[76] C. Surawanitkun *et al.*, "Modeling of switching energy of magnetic tunnel junction devices with tilted magnetization," *Journal of Magnetism and Magnetic Materials*, vol. 381, pp. 220–225, 2015.

[77] J. C. Slonczewski, "Current-driven excitation of magnetic multilayers," *Journal of Magnetism and Magnetic Materials*, vol. 159, no. 1-2, pp. L1–L7, 1996.

[78] K. Lee *et al.*, "1Gbit high density embedded STT-MRAM in 28nm FDSOI technology," in *2019 IEEE International Electron Devices Meeting (IEDM)*, IEEE, 2019, pp. 2–2.

[79] N. Maciel, E. Marques, L. Naviner, Y. Zhou, and H. Cai, "Magnetic tunnel junction applications," *Sensors*, vol. 20, no. 1, p. 121, 2019.

[80] G. Hu *et al.*, "STT-MRAM with double magnetic tunnel junctions," in *2015 IEEE International Electron Devices Meeting (IEDM)*, IEEE, 2015, pp. 26–3.

[81] M. Carpentieri *et al.*, "Micromagnetic analysis of statistical switching in perpendicular magnetic tunnel junctions with double reference layers," *IEEE Magnetics Letters*, vol. 9, pp. 1–5, 2018.

[82] S.-E. Lee, Y. Takemura, and J.-G. Park, "Effect of double MgO tunneling barrier on thermal stability and TMR ratio for perpendicular MTJ spin-valve

with tungsten layers," *Applied Physics Letters*, vol. 109, no. 18, p. 182 405, 2016.

[83] T. Nozaki, A. Hirohata, N. Tezuka, S. Sugimoto, and K. Inomata, "Bias voltage effect on tunnel magnetoresistance in fully epitaxial MgO double-barrier magnetic tunnel junctions," *Applied Physics Letters*, vol. 86, no. 8, p. 082 501, 2005.

[84] A. Giordano, G. Finocchio, L. Torres, M. Carpentieri, and B. Azzerboni, "Semi-implicit integration scheme for Landau–Lifshitz–Gilbert-Slonczewski equation," *Journal of Applied Physics*, vol. 111, no. 7, p. 07D112, 2012.

[85] M. Natsui *et al.*, "A 47.14- µW 200-MHz MOS/MTJ-Hybrid Nonvolatile Microcontroller Unit Embedding STT-MRAM and FPGA for IoT Applications," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 11, pp. 2991–3004, 2019.

[86] Y. Li, W. Kang, K. Zhou, K. Qiu, and W. Zhao, "Experimental Demonstration of STT-MRAM-based Nonvolatile Instantly On/Off System for IoT Applications: Case Studies," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 2, pp. 1–24, 2023.

[87] D. Edelstein *et al.*, "A 14 nm embedded stt-mram cmos technology," in *2020 IEEE International Electron Devices Meeting (IEDM)*, IEEE, 2020, pp. 11–5.

[88] Y.-D. Chih *et al.*, "13.3 a 22nm 32mb Embedded STT-MRAM with 10ns read speed, 1m cycle write endurance, 10 years retention at 150 c and high immunity to magnetic field interference," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, IEEE, 2020, pp. 222–224.

[89] W. Zhao *et al.*, "A Low-Latency and High-Endurance MLC STT-MRAM-Based Cache System," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 1, pp. 122–135, 2022.

[90] J. J. Kan *et al.*, "A study on practically unlimited endurance of STT-MRAM," *IEEE Transactions on Electron Devices*, vol. 64, no. 9, pp. 3639–3646, 2017.

[91] W. Kim *et al.*, "Extended scalability of perpendicular STT-MRAM towards sub-20nm MTJ node," in *2011 International Electron Devices Meeting*, IEEE, 2011, pp. 24–1.

[92] J. Kim, A. Chen, B. Behin-Aein, S. Kumar, J.-P. Wang, and C. H. Kim, "A technology-agnostic MTJ SPICE model with user-defined dimensions for STT-MRAM scalability studies," in *2015 IEEE custom integrated circuits conference (CICC)*, IEEE, 2015, pp. 1–4.

[93] X. Fong, S. H. Choday, and K. Roy, "Bit-cell level optimization for non-volatile memories using magnetic tunnel junctions and spin-transfer torque switching," *IEEE Transactions on Nanotechnology*, vol. 11, no. 1, pp. 172–181, 2011.

[94] Q. K. Trinh, S. Ruocco, and M. Alioto, "Voltage scaled STT-MRAMs towards minimum-energy write access," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 3, pp. 305–318, 2016.

[95] D. Gajaria, K. A. Gomez, and T. Adegbija, "A Study of STT-RAM-based In-Memory Computing Across the Memory Hierarchy," in *2022 IEEE 40th International Conference on Computer Design (ICCD)*, IEEE, 2022, pp. 685–692.

[96] X. Jiang, J. Bao, L. Zhang, and L. Bai, "A novel dual-reference sensing scheme for computing in memory within STT-MRAM," *Microelectronics Journal*, vol. 121, p. 105 355, 2022.

[97] K. Cho and S. K. Gupta, "XNOR-VSH: A Valley-Spin Hall Effect-based Compact and Energy-Efficient Synaptic Crossbar Array for Binary Neural Networks," *arXiv preprint arXiv:2306.05219*, 2023.

[98] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, "Computing in Memory With Spin-Transfer Torque Magnetic RAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 470–483, 2018.

[99] R. Salonik *et al.*, "PIMBALL: Binary Neural Networks in Spintronic Memory," *ACM Trans. Arch. Code Optim.*, vol. 37, no. 4, 2018.

[100] N. Xu *et al.*, "STT-MRAM Design Technology Co-optimization for Hardware Neural Networks," in *2018 IEEE International Electron Devices Meeting, IEDM 2018*, ser. Technical Digest - International Electron Devices Meeting, IEDM, Institute of Electrical and Electronics Engineers Inc., Jan. 2019, pp. 15.3.1–15.3.4.

[101] G. Zhao *et al.*, "An in-memory computing multiply-and-accumulate circuit based on ternary STT-MRAMs for convolutional neural networks," *IEICE Electronics Express*, vol. 19, no. 20, pp. 20 220 399–20 220 399, 2022.

[102] M. Morsali, R. Zhou, S. Tabrizchi, A. Roohi, and S. Angizi, "XOR-CiM: An Efficient Computing-in-SOT-MRAM Design for Binary Neural Network Acceleration," in *2023 24th International Symposium on Quality Electronic Design (ISQED)*, IEEE, 2023, pp. 1–5.

[103] S. Angizi, Z. He, A. Awad, and D. Fan, "MRIMA: An MRAM-Based In-Memory Accelerator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 5, pp. 1123–1136, 2020.

[104] Q. K. Trinh, S. Ruocco, and M. Alioto, "Novel boosted-voltage sensing scheme for variation-resilient STT-MRAM read," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 10, pp. 1652–1660, 2016.

[105] K. Kim, H. Shin, J. Sim, M. Kang, and L.-S. Kim, "An energy-efficient processing-in-memory architecture for long short term memory in Spin Orbit Torque MRAM," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, IEEE, 2019, pp. 1–8.

[106] A. Sridharan, F. Zhang, and D. Fan, "MnM: A Fast and Efficient Min/Max Searching in MRAM," in *Proceedings of the Great Lakes Symposium on VLSI 2022*, 2022, pp. 39–44.

[107] P. Kim, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*. Apress, 2017.

[108] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," vol. 60, no. 6, 2017.

[109] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[110] E. Garzón, A. Teman, M. Lanuzza, and L. Yavits, "AIDA: Associative In-Memory Deep Learning Accelerator," *IEEE Micro*, vol. 42, no. 6, pp. 67–75, 2022.

[111] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[112] J. Liu, Y. Fang, Z. Yu, and T. Wu, "Design and Construction of a Knowledge Database for Learning Japanese Grammar Using Natural Language Processing and Machine Learning Techniques," pp. 371–375, 2022.

[113] D. Wang, H. He, and D. Liu, "Intelligent Optimal Control With Critic Learning for a Nonlinear Overhead Crane System," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 2932–2940, 2018.

[114] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," *2017 IEEE International Electron Devices Meeting (IEDM)*, pp. 6.1.1–6.1.4, 2017.

[115] P.-Y. Chen and S. Yu, "Technological Benchmark of Analog Synaptic Devices for Neuroinspired Architectures," *IEEE Design & Test*, vol. 36, no. 3, pp. 31–38, 2019.

[116] Y. LeCun, C. Cortes, and C. J. Burges, *THE MNIST DATABASE of handwritten digits*, http://yann.lecun.com/exdb/mnist/.

[117] D. Fan, Y. Shim, A. Raghunathan, and K. Roy, "STT-SNN: A Spin-Transfer-Torque Based Soft-Limiting Non-Linear Neuron for Low-Power Artificial Neural Networks," *IEEE Transactions on Nanotechnology*, vol. 14, no. 6, pp. 1013–1023, 2015.

[118] G. Khodabandehloo, M. Mirhassani, and M. Ahmadi, "Analog implementation of a novel resistive-type sigmoidal neuron," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 4, pp. 750–754, 2011.

[119] S. Xing and C. Wu, "Implementation of a neuron using sigmoid activation function with CMOS," in *2020 IEEE 5th International Conference on Integrated Circuits and Microsystems (ICICM)*, IEEE, 2020, pp. 201–204.

[120] A. Kagalkar and S. Raghuram, "CORDIC based implementation of the softmax activation function," in *2020 24th International Symposium on VLSI Design and Test (VDAT)*, IEEE, 2020, pp. 1–4.

[121] R. Zunino and P. Gastaldo, "Analog implementation of the softmax function," in *2002 IEEE international symposium on circuits and systems. Proceedings (Cat. No. 02CH37353)*, IEEE, vol. 2, 2002, pp. II–II.

[122] A. H. Namin, K. Leboeuf, R. Muscedere, H. Wu, and M. Ahmadi, "Efficient hardware implementation of the hyperbolic tangent sigmoid function," in *2009 IEEE International Symposium on Circuits and Systems*, IEEE, 2009, pp. 2117–2120.

[123] F. M. Shakiba and M. Zhou, "Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 11, pp. 10 856–10 867, 2020.

[124] J. Shamsi, A. Amirsoleimani, S. Mirzakuchaki, A. Ahmade, S. Alirezaee, and M. Ahmadi, "Hyperbolic tangent passive resistive-type neuron," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2015, pp. 581–584.

[125] P. Priyanka, G. Nisarga, and S. Raghuram, "CMOS implementations of rectified linear activation function," in *VLSI Design and Test: 22nd International Symposium, VDAT 2018, Madurai, India, June 28-30, 2018, Revised Selected Papers 22*, Springer, 2019, pp. 121–129.

[126] A. A. Mohammed and V. Umaashankar, "Effectiveness of hierarchical softmax in large scale classification tasks," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2018, pp. 1090–1094.

[127] B. Alabassy, M. Safar, and M. W. El-Kharashi, "A high-accuracy implementation for softmax layer in deep neural networks," in *2020 15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, IEEE, 2020, pp. 1–6.

[128] I. Kouretas and V. Paliouras, "Hardware implementation of a softmax-like function for deep learning," *Technologies*, vol. 8, no. 3, p. 46, 2020.

[129] X. Dong, X. Zhu, and D. Ma, "Hardware implementation of softmax function based on piecewise LUT," in *2019 IEEE International Workshop on Future Computing (IWOFC*, IEEE, 2019, pp. 1–3.

[130] I. M. Elfadel and J. L. Wyatt Jr, "The" softmax" nonlinearity: Derivation using statistical mechanics and useful properties as a multiterminal analog circuit element," *Advances in neural information processing systems*, vol. 6, 1993.

[131] B. Yegnanarayana, "Artificial neural networks for pattern recognition," *Sadhana*, vol. 19, pp. 189–238, 1994.

[132] C. Yang, S. O. Prasher, J. Landry, and A. DiTommaso, "Application of artificial neural networks in image recognition and classification of crop and weeds," *Canadian agricultural engineering*, vol. 42, no. 3, pp. 147–152, 2000.

[133] M. Schrimpf *et al.*, "Artificial neural networks accurately predict language processing in the brain," *BioRxiv*, pp. 2020–06, 2020.

[134] Y. Kuvvetli, M. Deveci, T. Paksoy, and H. Garg, "A predictive analytics model for COVID-19 pandemic using artificial neural networks," *Decision Analytics Journal*, vol. 1, p. 100 007, 2021.

[135] J. L. Patel and R. K. Goyal, "Applications of artificial neural networks in medical science," *Current clinical pharmacology*, vol. 2, no. 3, pp. 217–226, 2007.

[136] V. F. Koosh and R. Goodman, "VLSI neural network with digital weights and analog multipliers," in *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196)*, IEEE, vol. 3, 2001, pp. 233–236.

[137] Z. Li, H. Li, X. Jiang, B. Chen, Y. Zhang, and G. Du, "Efficient FPGA implementation of softmax function for DNN applications," in *2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, IEEE, 2018, pp. 212–216.

[138] I. Tsmots, O. Skorokhoda, and V. Rabyk, "Hardware implementation of sigmoid activation functions using FPGA," in *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, IEEE, 2019, pp. 34–38.

[139] C. Pappas *et al.*, "Programmable Tanh-, ELU-, Sigmoid-, and Sin-based Nonlinear Activation Functions for Neuromorphic Photonics," *IEEE Journal of Selected Topics in Quantum Electronics*, 2023.

[140] A. Ghomi and M. Dolatshahi, "Design of a new cmos low-power analogue neuron," *IETE Journal of Research*, vol. 64, no. 1, pp. 67–75, 2018.

[141] D. Miyashita, S. Kousai, T. Suzuki, and J. Deguchi, "A neuromorphic chip optimized for deep learning and CMOS technology with time-domain analog and digital mixed-signal processing," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 10, pp. 2679–2689, 2017.

[142] M. H. Amin, M. Elbtity, M. Mohammadi, and R. Zand, "MRAM-based analog sigmoid function for in-memory computing," in *Proceedings of the Great Lakes Symposium on VLSI 2022*, 2022, pp. 319–323.

[143] B. Zamanlooy and M. Mirhassani, "Efficient VLSI implementation of neural networks with hyperbolic tangent activation function," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 1, pp. 39–48, 2013.

[144] V. S. Babu and M. Biju, "NOVEL CIRCUIT REALIZATIONS OF NEURON ACTIVATION FUNCTION AND ITS DERIVATIVE WITH CONTINUOUSLY PROGRAMMABLE CHARACTERISTICS AND LOW POWER CONSUMPTION," 10, vol. 5, 2014, pp. 185–200.

[145] R. Douglas, M. Mahowald, and C. Mead, "Neuromorphic analogue VLSI," *Annual review of neuroscience*, vol. 18, no. 1, pp. 255–281, 1995.

[146] C. Mead, "Author Correction: How we created neuromorphic engineering," *Nature Electronics*, vol. 3, no. 9, pp. 579–579, 2020.

[147] M. E.-D. Abo El-Soud, H. H. Soliman, R. A. AbdelRassoul, and L. M. El-ghanam, "Low-Voltage CMOS Circuits for Analog VLSI Programmable Neural Networks.," *MEJ. Mansoura Engineering Journal*, vol. 28, no. 4, pp. 21–30, 2021.

[148] M. Yamaguchi, G. Iwamoto, H. Tamukoh, and T. Morie, "An energy-efficient time-domain analog VLSI neural network processor based on a pulse-width modulation approach," *arXiv preprint arXiv:1902.07707*, 2019.

[149] J. G. Elias and D. P. Northmore, "Analog VLSI neuromorph with spatially extensive dendritic tree," in *World Congress On Neural Networks-San Diego*, Routledge, 2021, pp. II–543.

[150] Z. Kuang, J. Wang, S. Yang, G. Yi, B. Deng, and X. Wei, "Digital implementation of the spiking neural network and its digit recognition," in *2019 Chinese Control And Decision Conference (CCDC)*, IEEE, 2019, pp. 3621–3625.

[151] W. Zhang *et al.*, "Neuro-inspired computing chips," *Nature electronics*, vol. 3, no. 7, pp. 371–382, 2020.

[152] B. Tang *et al.*, "Wafer-scale solution-processed 2D material analog resistive memory array for memory-based computing," *Nature Communications*, vol. 13, no. 1, p. 3037, 2022.

[153] Q. Xue *et al.*, "Nonvolatile resistive memory and synaptic learning using hybrid flexible memristor based on combustion synthesized Mn-ZnO," *Journal of Materials Science & Technology*, vol. 119, pp. 123–130, 2022.

[154] Q. Wang *et al.*, "Reliable Ge2Sb2Te5 based phase-change electronic synapses using carbon doping and programmed pulses," *Journal of Materiomics*, vol. 8, no. 2, pp. 382–391, 2022.

[155] Y. Shi, "Neuro-inspired Computing Using Emerging Non-Volatile Memories," 2023.

[156] F. Liu, S. Deswal, A. Christou, Y. Sandamirskaya, M. Kaboli, and R. Dahiya, "Neuro-inspired electronic skin for robots," *Science Robotics*, vol. 7, no. 67, eabl7344, 2022.

[157] Y. Sandamirskaya, M. Kaboli, J. Conradt, and T. Celikel, "Neuromorphic computing hardware and neural architectures for robotics," *Science Robotics*, vol. 7, no. 67, eabl8419, 2022.

[158] P. Rohini, S. Tripathi, C. Preeti, A. Renuka, J. L. A. Gonzales, and D. Gangodkar, "A study on the adoption of Wireless Communication in Big Data Analytics Using Neural Networks and Deep Learning," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, IEEE, 2022, pp. 1071–1076.

[159] D. Ghimire, D. Kil, and S.-h. Kim, "A survey on efficient convolutional neural networks and hardware acceleration," *Electronics*, vol. 11, no. 6, p. 945, 2022.

[160] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, "Opportunities for neuromorphic computing algorithms and applications," *Nature Computational Science*, vol. 2, no. 1, pp. 10–19, 2022.

[161] P.-Y. Chen and S. Yu, "Technological benchmark of analog synaptic devices for neuroinspired architectures," *IEEE Design & Test*, vol. 36, no. 3, pp. 31–38, 2018.

[162] Y. Luo, X. Peng, and S. Yu, "MLP+ NeuroSimV3. 0: Improving on-chip learning performance with device to algorithm optimizations," in *Proc. of the Int. Conference on Neuromorphic Systems*, 2019, pp. 1–7.

[163] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *IEEE International Electron Devices Meeting (IEDM)*, 2017, pp. 6–1.

[164] N. Xu *et al.*, "STT-MRAM design technology co-optimization for hardware neural networks," in *2018 IEEE International Electron Devices Meeting (IEDM)*, IEEE, 2018, pp. 15–3.

[165] K. Zhang *et al.*, "High on/off ratio spintronic multi-level memory unit for deep neural network," *Advanced Science*, vol. 9, no. 13, p. 2 103 357, 2022.

[166] P.-Y. Chen *et al.*, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3067–3080, 2018.

[167] A. Lu, X. Peng, W. Li, H. Jiang, and S. Yu, "NeuroSim validation with 40nm RRAM compute-in-memory macro," in *IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1–4.

[168] X. Peng, S. Huang, H. Jiang, A. Lu, and S. Yu, "DNN+ NeuroSim V2. 0: An end-to-end benchmarking framework for compute-in-memory accelerators for on-chip training," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 40, no. 11, pp. 2306–2319, 2020.

[169] R. De Rose, M. Lanuzza, *et al.*, "A compact model with spin-polarization asymmetry for nanoscaled perpendicular MTJs," *IEEE Transactions on Electron Devices*, vol. 64, no. 10, pp. 4346–4353, 2017.

[170] R. De Rose, M. d'Aquino, *et al.*, "Compact modeling of perpendicular STT-MTJs with double reference layers," *IEEE Transactions on Nanotechnology*, vol. 18, pp. 1063–1070, 2019.

[171] Y. Zhang *et al.*, "Compact model of subvolume MTJ and its design application at nanoscale technology nodes," *IEEE Transactions on Electron Devices*, vol. 62, no. 6, pp. 2048–2055, 2015.

[172] E. Garzón, R. De Rose, F. Crupi, L. Trojman, A. Teman, and M. Lanuzza, "Relaxing non-volatility for energy-efficient DMTJ based cryogenic STT-MRAM," *Solid-State Electronics*, vol. 184, p. 108 090, 2021.

[173] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'Memristive'switches enable 'stateful'logic operations via material implication," *Nature*, vol. 464, no. 7290, pp. 873–876, 2010.

[174] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based material implication (IMPLY) logic:

Design principles and methodologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2054–2066, 2013.

[175] E. Garzón, A. Teman, M. Lanuzza, and L. Yavits, "AIDA: Associative in-memory deep learning accelerator," *IEEE Micro*, vol. 42, no. 6, pp. 67–75, 2022.

[176] Q. Chen, X. Wang, H. Wan, and R. Yang, "A logic circuit design for perfecting memristor-based material implication," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 2, pp. 279–284, 2016.

[177] H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberherr, "Implication logic gates using spin-transfer-torque-operated magnetic tunnel junctions for intrinsic logic-in-memory," *Solid-State Electronics*, vol. 84, pp. 191–197, 2013.

[178] M. Lanuzza, M. Margala, and P. Corsonello, "Cost-effective low-power processor-in-memory-based reconfigurable datapath for multimedia applications," in *Proceedings of the 2005 international symposium on Low power electronics and design*, 2005, pp. 161–166.

[179] H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberher, "Reliability analysis and comparison of implication and reprogrammable logic gates in magnetic tunnel junction logic circuits," *IEEE Transactions on Magnetics*, vol. 49, no. 12, pp. 5620–5628, 2013.

[180] F. M. Puglisi, L. Pacchioni, N. Zagni, and P. Pavan, "Energy-efficient logic-in-memory I-bit full adder enabled by a physics-based RRAM compact model," in *2018 48th European Solid-State Device Research Conference (ESSDERC)*, IEEE, 2018, pp. 50–53.

[181] F. M. Puglisi, T. Zanotti, and P. Pavan, "SIMPLY: Design of a RRAM-based smart logic-in-memory architecture using RRAM compact model," in *ESSDERC 2019-49th European Solid-State Device Research Conference (ESSDERC)*, IEEE, 2019, pp. 130–133.

[182] T. Zanotti, F. M. Puglisi, and P. Pavan, "Reconfigurable smart in-memory computing platform supporting logic and binarized neural networks for low-power edge devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 478–487, 2020.

[183] T. Zanotti, F. M. Puglisi, and P. Pavan, "Smart logic-in-memory architecture for low-power non-von neumann computing," *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 757–764, 2020.

[184] T. Zanotti, F. M. Puglisi, and P. Pavan, "Circuit reliability analysis of in-memory inference in binarized neural networks," in *International Integrated Reliability Workshop (IIRW)*, IEEE, 2020, pp. 1–5.

[185] R. De Rose, T. Zanotti, F. M. Puglisi, F. Crupi, P. Pavan, and M. Lanuzza, "STT-MTJ based smart implication for energy-efficient logic-in-memory computing," *Solid-State Electronics*, vol. 184, p. 108 065, 2021.

[186] R. De Rose, T. Zanotti, F. M. Puglisi, F. Crupi, P. Pavan, and M. Lanuzza, "Smart Material Implication Using Spin-Transfer Torque Magnetic Tunnel Junctions for Logic-in-Memory Computing," *Solid-State Electronics*, vol. 194, p. 108 390, 2022.

[187] E. Garzón, M. Lanuzza, A. Teman, and L. Yavits, "AM 4: MRAM crossbar based CAM/TCAM/ACAM/AP for in-memory computing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 408–421, 2023.

[188] S. Wang and H. Cai, "Computing-in-Memory with Enhanced STT-MRAM Readout Margin," *IEEE Transactions on Magnetics*, 2023.

[189] J.-W. Ryu and K.-W. Kwon, "Self-adjusting sensing circuit without speed penalty for reliable STT-MRAM," *Electronics Letters*, vol. 53, no. 4, pp. 224–226, 2017.

[190] T. Na, S. H. Kang, and S.-O. Jung, "STT-MRAM sensing: a review," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 12–18, 2020.

[191] R. De Rose *et al.*, "A Compact Model with Spin-Polarization Asymmetry for Nanoscaled Perpendicular MTJs," *IEEE Transactions on Electron Devices*, vol. 64, no. 10, pp. 4346–4353, 2017.

[192] J. J. Nowak *et al.*, "Dependence of voltage and size on write error rates in spin-transfer torque magnetic random-access memory," *IEEE Magnetics Letters*, vol. 7, pp. 1–4, 2016.

[193] J. C. Sankey, Y.-T. Cui, J. Z. Sun, J. C. Slonczewski, R. A. Buhrman, and D. C. Ralph, "Measurement of the spin-transfer-torque vector in magnetic tunnel junctions," *Nature Physics*, vol. 4, no. 1, pp. 67–71, 2008.

[194] S. Ikeda *et al.*, "A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction," *Nature Materials*, vol. 9, no. 9, pp. 721–724, 2010.

[195] T. Zanotti, F. M. Puglisi, and P. Pavan, "Reliability and performance analysis of logic-in-memory based binarized neural networks," *IEEE Transactions on Device and Materials Reliability*, vol. 21, no. 2, pp. 183–191, 2021.

[196] M. Ali, S. Roy, U. Saxena, T. Sharma, A. Raghunathan, and K. Roy, "Compute-in-Memory Technologies and Architectures for Deep Learning Workloads," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 11, pp. 1615–1630, 2022.

[197] T. Zanotti *et al.*, "Reliability of logic-in-memory circuits in resistive memory arrays," *IEEE Transactions on Electron Devices*, vol. 67, no. 11, pp. 4611–4615, 2020.