2024

Adaptive Beyond Von-Neumann Computing Devices and Reconfigurable Architectures for Edge Computing Applications

Mousam Hossain
University of Central Florida

ADAPTIVE BEYOND VON-NEUMANN COMPUTING DEVICES AND RECONFIGURABLE ARCHITECTURES FOR EDGE COMPUTING APPLICATIONS

by

MOUSAM HOSSAIN
Master of Science, North Dakota State University, 2019

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2024

Major Professor: Ronald F. DeMara

ABSTRACT

The Von-Neumann bottleneck, a major challenge in computer architecture, results from significant data transfer delays between the processor and main memory. Crossbar arrays utilizing spin-based devices like Magnetoresistive Random Access Memory (MRAM) aim to overcome this bottleneck by offering advantages in area and performance, particularly for tasks requiring linear transformations. These arrays enable single-cycle and in-memory vector-matrix multiplication, reducing overheads, which is crucial for energy and area-constrained Internet of Things (IoT) sensors and embedded devices.

This dissertation focuses on designing, implementing, and evaluating reconfigurable computation platforms that leverage MRAM-based crossbar arrays and analog computation to support deep learning and error resilience implementations. One key contribution is the investigation of Spin Torque Transfer MRAM (STT-MRAM) technology scaling trends, considering power dissipation, area, and process variation (PV) across different technology nodes. A predictive model for power estimation in hybrid CMOS/MTJ technology has been developed and validated, along with new metrics considering the Internet of Things (IoT) energy profile of various applications.

The dissertation introduces the Spintronically Configurable Analog Processing in-memory Environment (SCAPE), integrating analog arithmetic, runtime reconfigurability, and non-volatile devices within a selectable 2-D topology of hybrid spin/CMOS devices. Simulation results show improvements in error rates, power consumption, and power-error-product metric for real-world applications like machine learning and compressive sensing, while assessing process variation impact. Additionally, it explores transportable approaches to more robust SCAPE implementations, including applying redundancy techniques for artificial neural network (ANN)-based digit recognition applications. Generic redundancy techniques are developed and applied to hybrid spin/CMOS-based ANNs, showcasing improved/comparable accuracy with smaller-sized networks. Furthermore, the dissertation examines hardware security considerations for emerging memristive device-based applications, discussing mitigation approaches against malicious manufacturing interventions. It also discusses reconfigurable computing for AI/ML applications based on state-of-the-art FPGAs, along with future directions in adaptive computing architectures for AI/ML at the edge of the network.

To my dear husband, beloved parents, doting sister, and loving in-laws, thank you for your unwavering support and being my source of inspiration. Your belief in me gives me the utmost strength.

"*Whether you think you can, or you think you can't – you're right*." – Henry Ford.

ACKNOWLEDGMENTS

I wish to convey my sincere gratitude to Dr. Ronald F. DeMara for graciously accepting me into his UCF Computer Architecture Lab (CAL) research group during my doctoral journey. His invaluable guidance and counsel have consistently guided me in the right direction with my research, professional growth and active participation in scholarly endeavors. I would also like to extend my gratitude to the members of my committee who dedicated their valuable time to provide their support: Dr. Kalpathy Sundaram, Dr. Mingjie Lin, Dr. Yaser Fallah, and Dr. Niels da Vitoria Lobo. Furthermore, I would also like to thank my team members with whom I enjoyed collaborating at different times and learned so much. Finally, I am thankful to the National Science Foundation (NSF) through ECCS-1810256 and the Army Research Office (ARO) under Grant Number W911NF-20-1-0174 for partly supporting my work in various capacities.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AAS	Analog Activation Stage
ADAS	Autonomous Driver Assistance Systems
ADC	Analog-to-Digital Conversion
AlO_x	Aluminum Oxide
AHE	Anomalous Hall Effect
ANN	Artificial Neural Network
ApGAN	Approximate Generative Adversarial Network
AVSS	Advanced Video Surveillance Systems
BEOL	Back End of Line
CAPP	Content Addressable Parallel Processor
CMOS	Complementary Metal Oxide Semiconductor
CNTFET	Carbon Nanotube Field Effect Transistor
CPU	Central Processing Unit
CS	Compressive Sensing
DMTJ	Double-barrier Magnetic Tunnel Junction
DBN	Deep Belief Networks
DRAM	Dynamic Random Access Memory
FIMS	Field-Induced Magnetic Switching
FPGA	Field Programmable Gate Array
GAAF	Generalizable Analog Activation Function

GAN	Generative Adversarial Network
GPU	General Purpose Unit
IC	Integrated Circuit
ILLIAC	Illinois Automatic Computer
IoT	Internet of Things
IRAM	Intelligent RAM
ISA	Instruction Set Architecture
IRDS	International Roadmap for Devices and Systems
LFSR	Linear Feedback Shift Register
MAD	Mean Active Duration
mAP	Mean Average Precision
MC	Monte Carlo
MIMD	Multiple Instruction Multiple Data
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
MgO	Magnesium Oxide
MRAM	Magnetoresistive Random Access Memory
MSD	Mean Standby Duration
MTJ	Magnetic Tunnel Junction
NN	Neural Network
NVM	Non-Volatile Memory
PCM	Phase Change Memory

PCU	Prediction Comparator Unit
PCHB	Pre-Charge Half Buffer
PDSR	Power Dissipation Scaling Ratio
PE	Processing Element
PEP	Power Error Product
PIM	Processing-In-Memory
PIN-Sim	Probabilistic Inference Network Simulator
PTMR	Progressive Temporal Modular Redundancy
PIR	Probabilistic Interpolation Recoder
PTM	Predictive Technology Model
PV	Process Variation
RBM	Restricted Boltzmann Machines
RC	Resistor-Capacitance
RRAM	Resistive Random-Access Memory
SCL	Sleep Convention Logic
SHE-MTJ	Spin Hall Effect Magnetic Tunnel Junction
SIMD	Single Instruction Multiple Data
SCAPE	Spintronically Configurable Analog Processing-in-Memory Environment
SNAP-1	Semantic Network Array Processor parallel AI prototype
SNR	Signal to Noise Ratio
SoC	System-on-Chip
SOT-MRAM	Spin Orbit Torque - Magnetoresistive Random Access Memory

SRAM	Static Random Access Memory
SSD	Solid State Drive
STMR	Spatial Triple Modular Redundancy
STT-SRAM	Spin Torque Transfer Magnetic Random Access Memory
6T-SRAM	Six Transistor – Static Random Access Memory
TAS	Thermally Assisted Switching
TMR	Tunneling Magnetoresistance Ratio
TPU	Tensor Processing Unit
2T-1R MRAM	Two Transistor-1 Resistor Magnetoresistive Random Access Memory
VMM	Vector Matrix Multiplication
VMMS	Vector Matrix Multiplication Stage
VOT	Visual Object Tracking
VTB	Visual Tracker Benchmark
WCHB	Weak Charge Half Buffer

CHAPTER 1: INTRODUCTION[1]

1.1 Introduction, Motivation, and Research Objectives

Machine learning (ML) and Artificial Neural Networks (ANN) have found ubiquitous applications in numerous fields of research spanning image recognition, video processing, speech recognition, Internet of Things (IoT), among many others. Software-based acceleration of ANN training and inference phases on platforms such as Central Processing Units (CPU), Field Programmable Gate Arrays (FPGA), and Graphics Processing Units (GPU) has witnessed significant advancements in research. However, when it comes to resource-constrained edge-of-the-network applications, each of these approaches has its own set of limitations. CPUs encounter the power wall issue in deep sub-micron technology nodes, which is caused by the separation of memory and computational units and is an inherent characteristic of Von-Neumann machines. GPUs and FPGAs, despite offering advantages like multiple processing cores and massively parallelized operations, respectively, are not without their limitations either. The former is characterized by drawbacks including high power consumption and bandwidth allocation, while the latter is hindered by routing congestion and resource utilization constraints. Notwithstanding the fact that Vector Matrix Multiplication (VMM) operations have been extensively rehosted from a general-purpose computing paradigm to FPGAs, Tensor Processing Units (TPUs), and GPUs, they still encounter key challenges such as memory-wall barriers and excessive energy consumption.

[1] ©IEEE. Part of this chapter is reprinted, with permission, from [1], [2], [3], and [4]

1.1.1 Need for Tunable and Intrinsically based Computing Architectures

Due to the high bandwidth demands of data transfer inherent in VMM-intensive applications, the Von-Neumann architectural model of data transfer between discrete memory and processing units is being reconsidered as it suffers from large latency and energy costs. Efforts have been made to alleviate this bottleneck by enabling in-memory computing functionality using emerging hybrid-memristive devices. The most mature among these efforts use crossbar arrays of non-volatile spin-based components, such as Spin Orbit Torque-Magnetoresistive Random Access Memory (SOT-MRAM) based on Magnetic Tunnel Junction (MTJ) devices. The 2021 International Roadmap for Devices and Systems (IRDS) Beyond CMOS Roadmap [5] and 2020 Magnetism Roadmap [6] lists nanomagnetic devices as one of the most promising post-CMOS options for embedded MRAM, with MTJs seeing increased commercialization [7]. Additionally, the advantages of non-Von-Neumann architectures are sought for cutting-edge applications, hardware-aware intelligent edge devices, and neuromorphic computing. With this motivation, one of the primary *objectives* of my doctoral research has been to *design evolvable computing architectures for machine learning (ML) and edge-IoT applications*, including *generalizability of activation functions,* hardware reconfigurability, high performance and low-power designs, and emerging technology-based next-generation computing.

1.1.2 Need for Energy Cognizant Usage of Memory Device Technology

Non-volatile memories (NVMs), such as MRAM, are a viable alternative to conventional CMOS-based memory cells, owing to their near-zero leakage dissipation, lower area footprint, faster read access, and backend compatibility with existing CMOS fabrication processes.

However, the cost comparison between conventional memories and emerging NVMs is not straightforward, as the performance depends significantly on the target applications' activity profiles. For instance, for intermittently powered edge and IoT applications, the asymmetry between read/write energy consumption and active/standby duty cycles must be considered during tradeoff analysis. Previous models in the literature did not account for such crucial parameters when estimating the power efficiency of emerging technology NVMs. To address this gap, the research presented herein also *focuses* on *developing an analytical predictive model and proposed novel metrics* considering the workload-driven parameters of the components' mean standby and active memory duty cycles.

1.1.3 Need for Reliable ANN Hardware for Selected Applications

From an application standpoint, intelligent edge devices that operate on an energy budget are in great need of hardware that is adaptable, energy-efficient, and resilient. Recently, research on neural network (NN) computation using emerging memristive devices has gained momentum due to their unique capability of performing extremely low-energy analog computations. However, the effect of process variation on these device architectures can affect the system robustness. There exist several methods that focus on improving robustness of such emerging technology based robust hardware. However, many of such approaches are often bulky and complex circuits, which restricts their implementation scope for many applications due to the increased area and energy overheads or, are over-reliant on the NN model structure making them only applicable to certain neural network model architectures, which necessitates domain specific knowledge and expertise of the target application. To overcome these limitations, a part of my doctoral research presented herein is centered on *designing a robust*

NN hardware implementation for energy-constrained applications, utilizing an alternative approach to attain energy/area efficiency without compromising the performance accuracy.

1.1.4 Need for Trustworthy Post-CMOS ML Accelerators

Emerging technologies based on spin have the capacity to bring about significant transformations across multiple domains; however, apprehensions regarding privacy and security could impede their widespread implementation. The inclusion of spin-based devices in a variety of computing systems necessitates the assurance of the hardware's integrity for the system's overall security. Analysis of hardware security guarantees that spin-based devices can be relied upon to carry out their designated functions in a secure manner. Gaining insight into the hardware security environment of spin-based devices empowers developers to customize security protocols in a manner that efficiently mitigates particular risks. Through comprehensive examination of hardware security elements and remediation of any detected susceptibilities, developers have the ability to foster confidence among users and stakeholders, thereby expediting the integration of these technologies. In order to construct *dependable applications utilizing these emergent spin-based architectures and devices*, this dissertation addresses a limited number of physical device-level threats that may result in performance issues at the application level.

1.2 Dissertation Overview

1.2.1 Power Consumption Scaling in Hybrid Spin/CMOS Devices Used in Memory

The CMOS design paradigm still dominates today’s semiconductor industry. However, with the ever-decreasing feature size to keep up with Moore’s law of doubling the count of transistors on chip every two years, new challenges have surfaced since the past two decade or so. In deep sub-micron region, CMOS technology suffers from high static power dissipation, which is the power dissipated in idle state, i.e., when the circuit is not performing any useful work. Static power now accounts for more than 50% of the power consumption profile of any device, which is a major challenge for IoT devices like handheld System-on-Chips (SoCs) with limited power source [15]. This has resulted in renewed interest in the post-CMOS domain, where alternate technologies, such as Carbon Nanotube Field Effect Transistors (CNTFETs) and FinFETs, are being explored, both in the synchronous and asynchronous digital design domain [8], [9], [10], [11], [12], [13], [14]. Most of the static power consumption occurs in the memory units such as Static Random Access Memory (SRAM) caches, where the power cannot be turned off as the volatile nature of CMOS would result in data loss. Therefore, new emerging technologies are being explored as possible alternatives to CMOS that are non-volatile in nature but offer comparable performance and fast read/write speeds like conventional SRAM.

Magnetic Tunnel Junction (MTJ) devices have garnered attention as a prospective substitute for CMOS-based memory due to their advantageous characteristics, including near-zero standby power, efficient area utilization, fast read speed, and backend compatibility with pre-existing CMOS devices, among others. MTJs are presently manufactured commercially by leading semiconductor companies such as Intel, Everspin, IBM, etc., in the form of embedded magnetic RAM (MRAM), solid state drives (SSD), and dynamic RAM (DRAM). As of now, transistor characteristics scaled in accordance with Dennard scaling equations in a relatively predictable manner with supply voltage and transistor size, except when device sizes entered the deep sub-micron region. In deep sub-micron regions, classical scaling equations may differ by approximately one to two orders of magnitude [53] as compared to calculations obtained utilizing Predictive Technology Model (PTM) models [16]. There have been several models for estimating the scaling trends for 6T-SRAM memory array-based structures in the past. However, not many such holistic power estimation models exist for memory structures utilizing emerging logic devices that take into account the critical design aspects, such as, lifetime operating profiles, asymmetric read/write cycles, etc., among others. Hence, there is a need for more such models to be developed for emerging device-based memory structures, which can demonstrate the scaling trends for power dissipation and other device characteristics with respect to various practical and current technology node sizes. This could be of great assistance to architects in their early stages of design decision-making, as it provides an approximation of the compromises between power consumption and device performance based on scaling trends for the targeted technology library and target application.

To extend beyond recent efforts for modeling of 6-Transistor SRAM (6T-SRAM) cells, this dissertation considers technology scaling trends of Spin Torque Transfer Magnetic Random Access Memory (STT-MRAM) with respect to power dissipation and area, including the impact of Process Variation (PV). These effects have been incorporated herein into models utilized via SPICE simulation along with Predictive Technology Model (PTM) libraries [16] to ascertain MRAM vs. SRAM technology inflection tradeoff points. Quantitative results obtained also refine lifetime energy estimates over the operational lifetime, which are parameterized in terms of three new metrics Mean Standby Duration (MSD) and Mean Active Duration (MAD), and Power Dissipation Scaling Ratio (PDSR).

1.2.2 Processing in-Memory (PiM) with Hybrid Spin/CMOS Analog and Digital Blocks

Machine learning techniques are increasingly being applied to data-intensive applications, such as image processing, video processing, computer vision, audio and speech processing, etc., owing to the recent technological developments [17] [18] [19]. Concurrently, these are sought after to operate under energy constraints imposed by edge-of-network based embedded components. In particular, artificial neural network (ANN) architectures and edge-of-network applications make significant use of vector-matrix multiplication (VMM) operations, which impose significant memory transfer demands [20]. ANN processing designed for emergent real-world applications at the edge of the computing network fundamentally includes VMM operations and various compressive sensing tasks. To overcome the memory bottleneck, devices and architectures that go beyond Von Neumann architectural principles are increasingly adopted to offer processing capability closer to where the data resides. This has given rise to the study of memory-centric strategies to attain improved throughput and energy efficiency, both with and without modification to the underlying storage/switching devices utilized in the design of the Processing-in-Memory (PiM) component itself. Most recently, with the fabrication, demonstration, and preliminary commercialization of post-CMOS devices, e.g., Spin Transfer Torque Magnetic Tunnel Junctions (STT-MTJs), Spin Orbit Torque Magnetic Tunnel Junctions (SOT-MTJs), and Spin Hall Effect Magnetic Tunnel Junctions (SHE-MTJs), such emerging devices are being thoroughly investigated towards advancing PiM paradigm to enable emerging opportunities for future edge-of-network computing platforms. Taking inspiration from various technical attributes of the milestone works in PiM approaches spanning the last five decades, this dissertation considers new roles and approaches to PiM for machine learning applications and compressive sensing. Specifically, this work furthers the efforts in edge-of-network PiM via hardware implementation of a *generalized activation function* in a *Spintronically Configurable Analog Processing-in-Memory Environment (SCAPE)* architecture for selected applications.

1.2.3 Emerging Technology based Area and/or Energy Efficient Robust ANN Hardware

The past few years have seen several implementations of ANNs and Deep Belief Networks (DBNs) utilizing memristive crossbars and MRAM-based stochastic neurons and synapses, which have lower energy and area footprint compared to prior CMOS-based hardware implementations. However, such implementations suffer from low accuracy on image classification tasks due to stochasticity of the MRAM-based neuron as well as PV effects of the intrinsic MTJ devices. Accuracy, area footprint, and power efficiency are crucial in safety-critical and/or resource-constrained applications at the edge-of-network. To enable the adoption of current emerging technologies in a wide range of NN-based applications and suitably leverage the benefits they present, particularly with respect to the implementation of VMM operations in a crossbar, optimization methodologies such as synaptic weight binarization, model compression, and weight quantization have been widely explored in literature to shrink the model size. In this dissertation, a low area- and energy- footprint stable ANN implementation approach has been proposed utilizing generic redundancy schemes. The methodology is portable and can be extended and implemented on additional deep learning networks and models, including alternative emerging memristive technologies, in order to reduce resource overhead while maintaining comparable recognition accuracy as a viable substitution to larger neural network models and/or model compression techniques.

1.2.4 Analyzing the PV Sensitivity Effects on Spin-based ANN from a Hardware Security Perspective

Recent advancements in hardware designs and emerging devices have shown promising potential for accelerating ML and NN-driven computations [21]. Such computation requires rapid and reliable operations at the hardware level to ensure minimal loss in the algorithm accuracy. Hardware accelerators commonly employ a range of emerging technologies, including resistive random-access memory (RRAM), SOT-MRAM, STT-MRAM, SHE-MTJ, phase change memory (PCM), and others. By doing so, they are capable of accelerating computations by orders of magnitude while reducing their energy consumption [22]. Among these emerging devices, SOT-MRAM has been shown to be a highly promising technology in its category, which can be readily integrated with the traditional baseline CMOS design with minimal incurred fabrication cost. SOT-MRAM devices benefit from non-volatility, high endurance, compact cell size, low read/write energy, extremely low leakage, and faster read/write capability.

Furthermore, with technological advancements in the semiconductor processing industry, the cost of maintaining and creating tools for integrated circuit (IC) manufacturing has increased rapidly. Thus, many IC design companies, with a few exceptions, have adopted a fabless business model that utilizes a distributed global supply chain. This approach necessitates several distinct stages for the design, manufacturing, and validation of ICs. Moreover, the globalization of the IC supply chain has resulted in the emergence of several hardware vulnerabilities and threats [23]. Current IC supply chain model allows adversaries to introduce malicious design modifications at various stages of the process [24]. Notably, these include Intellectual Property (IP) piracy, IC overuse, reverse engineering [25], hardware trojan [26], counterfeiting [27], and side channel attacks [28].

Trustworthiness of the hardware platform attains significance due to exposures of authorized and unauthorized accesses during various manufacturing processes. If the security of an IC is compromised, it would result in vulnerabilities to algorithms running on the platform, as well as to other hardware components within the platform. Therefore, it is imperative to understand the supply chain exposures, especially in applications such as ML where the output behavior is intricate and well-recognized as challenging to observe. Currently, however, there is a gap in the research regarding the reliability of the computing operations and the security threats affecting the hardware components, including the ML hardware accelerators. Thus, it is critical to analyze such designs to optimize the computation speed and minimize the overhead in terms of energy and area, while ensuring the security and reliability of the hardware.

Fig. 1 illustrates the highlights of this doctoral research.

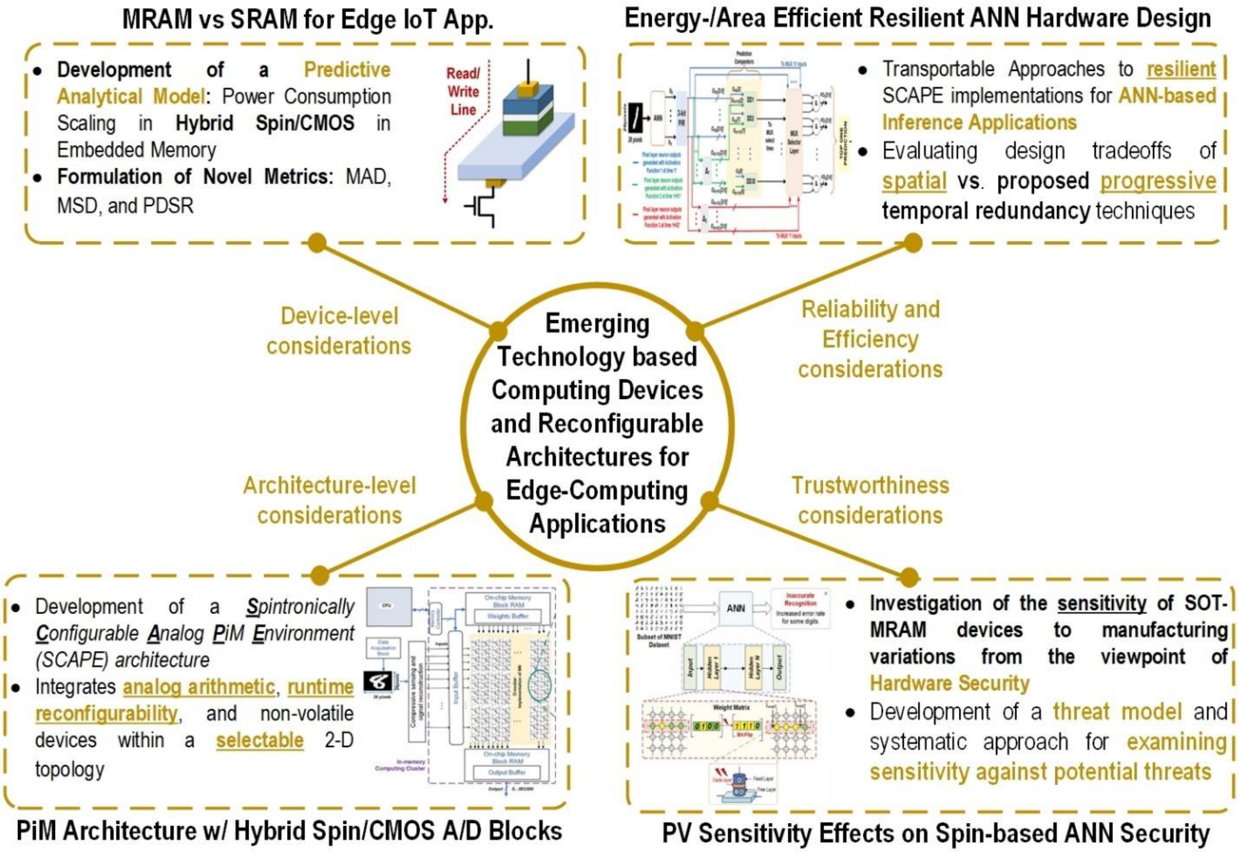

MRAM vs SRAM for Edge IoT App.
Development of a Predictive Analytical Model: Power Consumption Scaling in Hybrid Spin/CMOS in Embedded Memory
Formulation of Novel Metrics: MAD, MSD, and PDSR
Read/ Write Line
Energy-/Area Efficient Resilient ANN Hardware Design
Transportable Approaches to resilient SCAPE implementations for ANN-based Inference Applications
Evaluating design tradeoffs of spatial vs. proposed progressive temporal redundancy techniques
Device-level considerations
Reliability and Efficiency considerations
Emerging Technology based Computing Devices and Reconfigurable Architectures for Edge-Computing Applications
Architecture-level considerations
Trustworthiness considerations
Development of a Spintronically Configurable Analog PiM Environment (SCAPE) architecture
Integrates analog arithmetic, runtime reconfigurability, and non-volatile devices within a selectable 2-D topology
PiM Architecture w/ Hybrid Spin/CMOS A/D Blocks
Investigation of the sensitivity of SOT-MRAM devices to manufacturing variations from the viewpoint of Hardware Security
Development of a threat model and systematic approach for examining sensitivity against potential threats
PV Sensitivity Effects on Spin-based ANN Security

Figure 1: Research Overview

1.3. Research Contributions

The doctoral research presented in this dissertation has the following major contributions:

- ***Study of power consumption scaling of MRAM vs. SRAM in IoT devices:*** A predictive model has been developed to analyze the scaling effects of SRAM and STT-MRAM in terms of static and write power dissipation for sub-micron technology nodes, considering the presence of PV. The determination of the total power dissipation impacts of candidate memory cell designs is achieved through the utilization of workload-driven parameters, namely the Mean Standby

Duration (MSD) and Mean Active Duration (MAD), of components. A novel Power Dissipation Scaling Ratio (PDSR) metric has been developed, which extends technology scaling to embedded MRAM devices. Quantification of power static and dynamic power dissipation bit-cell 6T-SRAM and 2T-1R MRAM cell has been inferred based on MSD, MAD, and PDSR.

- ***PiM architecture utilizing hybrid spin/CMOS based reconfigurable neurons for sensing and reasoning applications:*** A novel crossbar topology for in-memory processing, which provides in-field configurability of hybrid Spin/CMOS-based analog/digital blocks, has been developed. Various neuron designs, including the use of SHE-MTJs for memristive-based computation and activation function calculation, are evaluated. A generalizable spin-based activation function has been proposed to achieve run-time configurability, while increasing recognition rate. Analog computation of the generalized activation function demonstrates acceptable accuracy, reduced area, and decreased energy consumption, as evaluated on Modified National Institute of Standards and Technology database (MNIST) dataset. As a dynamic and transportable performance metric, the power-error-product (PEP) concept has been applied to the evaluation of various activation functions. By employing the Monte Carlo (MC) method, the effects of PV on SHE-MTJ devices have been quantified, and the results for the deviation in activation function of neurons with respect to PV are presented. A maximum standard deviation of 5% has been accounted for MTJ parameters including length, width, and thickness.

- ***Energy and/or area efficient ANN-based inference for digit recognition application via spin-based progressive redundancy:*** Generic redundancy schemes that are extendable and applicable to other emerging spin-device based ANN models and classification tasks have been explored. A Spatial Triple Modular Redundancy (STMR) flow for spin based ANNs with re-configurable activations based on ensemble learning has been designed and evaluated. A novel Progressive Temporal Modular Redundancy (PTMR) approach with varied activations, which can be applied after the output layer and therefore, can be implemented without causing intervention in the internal layers and hardware structure of the neural network, has been proposed. A comparative analysis was undertaken to examine the impact of these two distinct redundancy techniques on prediction accuracy and area utilization when applied to a smaller-sized ANN as opposed to a baseline, more complex ANN without redundancy. The objective is to identify design tradeoffs that are feasible for ANN inference on edge computing applications.

- ***An analytical approach to study PV sensitivity of physical fabrication parameters to identify possible hardware security threats for spin-based architectures:*** The sensitivity of SOT-MRAM devices to manufacturing parameter variations from the viewpoint of hardware security has been comprehensively investigated. The effects of internal changes in different layers of the device with respect to its behavior as well as the impact on the performance of the ML accelerators designed using these devices have been quantitatively analyzed. Simulations involving detailed comparison with an ideal SOT-MRAM device has been utilized to identify how a physically modified SOT-MRAM device performs under specific conditions. This study specifically illustrates how a malicious global change to the oxide layer thickness, denoted as T_{ox}, across the wafer during the fabrication phase of the supply chain can introduce a gainful vulnerability to the ML recognition system. Table 1 summarizes the research outcomes of this doctoral research work.

Table 1: Research Questions and Limitations Addressed in this Dissertation

Research Questions	Current Limitations	Research Outcomes	Evaluation Metrics
How to better estimate the tradeoffs between emerging and conventional memory usage for edge IoT applications?	Existing predictive models do not consider the intermittent power profiles of IoT devices.	Development of a workload driven analytical model of SRAM vs. MRAM for edge-of-network applications	Mean Standby Duration (MSD), Mean Active Duration (MAD), Power Dissipation Scaling Ratio (PDSR)
Can runtime adaptable PiM be instrumental for reasoning and/or sensing applications?	User runtime reconfigurability and tunabilty of neuron activations in PiM is underexplored	Development of a spin based PiM architecture for reasoning applications with generalizable activations	Power consumption, inference accuracy, and PV analysis
Can larger NN models in PiM be replaced by more efficient smaller networks with a comparable or improved accuracy?	Existing approaches are complex, costly in terms of energy and area utilization, and/or lacks portability, making them not suitable for resource-constrained applications	Development of generic redundancy schemes leveraging dynamic redundancy and *Ensemble Learning* to achieve comparable accuracy and energy efficiency	Power consumption, inference accuracy, and area utilization

Research Questions	Current Limitations	Research Outcomes	Evaluation Metrics
Can subtle, imperceptible variation in the physical parameter of spin-devices present potential threat to the application security?	Limited amount of works exploring the device-level manufacturing threats in spin-based architectures and their impacts at the application level	A comprehensive sensitivity analysis of spin based ANNs and underlying device manufacturing parameters from a hardware security perspective	T_{ox} variation effect on read/write disturbance and inference accuracy

CHAPTER 2: BACKGROUND AND RELATED WORKS[2]

2.1 Emerging Spin-based Devices

2.1.1 Magnetic Tunnel Junction (MTJ)

MTJs are composed of two ferromagnetic (FM) layers: a fixed layer and free layer, separated by a thin oxide barrier, as depicted in Fig. 2. Based on the magnetic orientation of the FM layers, MTJs can be classified into two broader categories: in-plane MTJ, which has the FM layer's magnetic orientation in alignment with the MTJ plane; and perpendicular MTJ, characterized by FM layers with magnetic orientations perpendicular to the MTJ plane. MTJs have two resistive states determined by the relative orientation of the free-layer magnetization with respect to the fixed layer. One is called the low resistance Parallel (P) state, where the free layer and the fixed layers' magnetic orientations are in the same direction (i.e., $\theta = 0°$, where ϑ is the angle between the magnetization orientations of fixed layer and free layer). The other state of resistance is referred to as the high resistance anti-parallel (AP) state. In this state, the magnetic orientations of the free and fixed layers are in opposite directions (i.e., $\theta = 180°$). The resistance of P and AP states are represented in literature as R_P and R_{AP}, respectively. The performance of an MTJ is indicated by a parameter known as a Tunneling Magnetoresistance (TMR) ratio, which arises primarily due to the insulating layer in between (typically MgO or AlO_x), separating the two FM layers. TMR ratio can be derived from the following equation,

$$TMR = \frac{R_{AP}-R_P}{R_P} = \frac{G_P-G_{AP}}{G_{AP}} \tag{2.1}$$

[2] ©IEEE. Part of this chapter is reprinted, with permission, from [1], [2], [3], and [4]

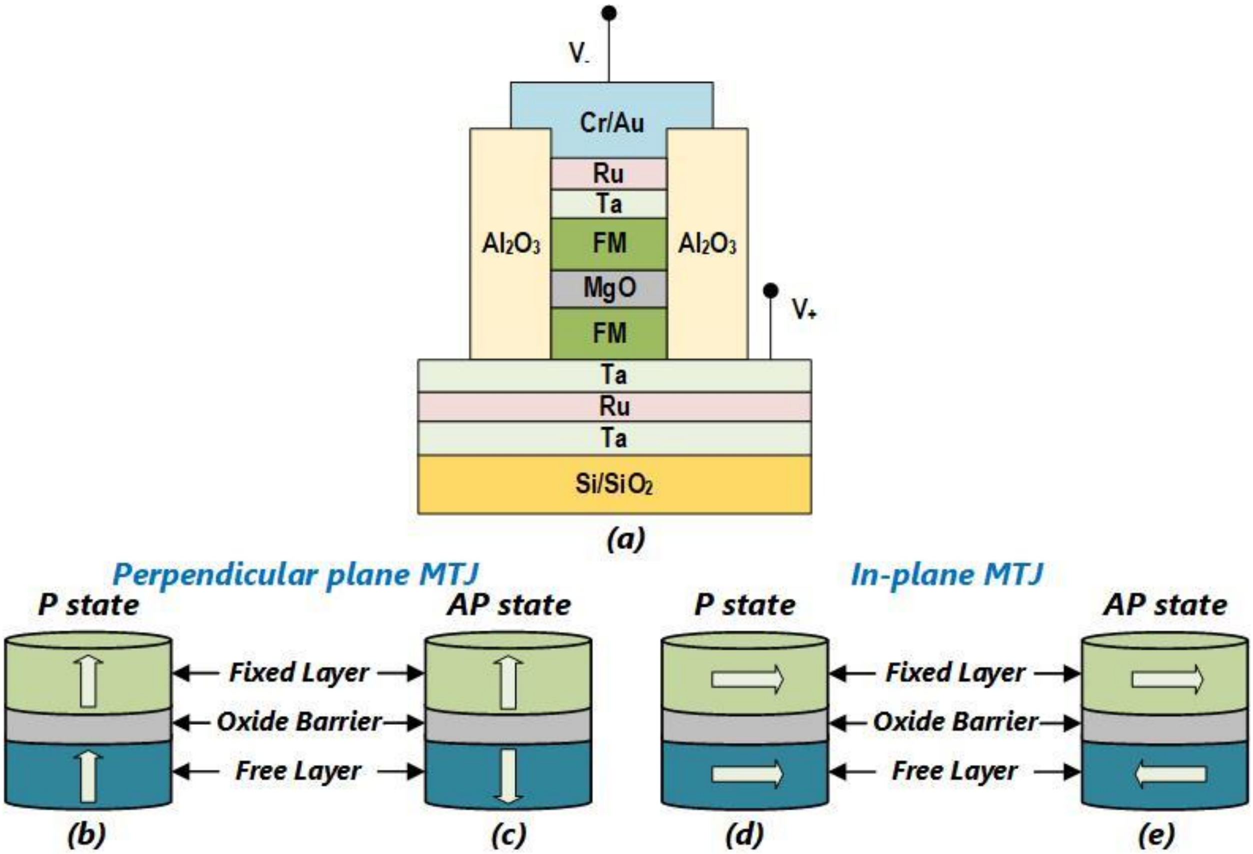


Figure 2: (a) MTJ (Vertical Stack Structure) [29], (b) P-state Perpendicular MTJ, (c) AP-state Perpendicular MTJ, (d) P-state In-plane MTJ, and (e) AP-state In-plane MTJ.

TMR is a quantum mechanical effect, which depends on spin polarization. If P_1 and P_2 are the spin polarizations of the fixed and free layer, respectively, then the TMR ratio can be obtained utilizing the Julliere’s model as follows [30], [31],

$$TMR = \frac{2P_1P_2}{1-P_1P_2} \tag{2.2}$$

In CMOS/spin hybrid circuits employing MTJ devices, a substantial value of the TMR ratio guarantees an adequate voltage gap to enable a distinct differentiation between the high and low logic states (as represented by logic '1' and '0'). The typical value of TMR ratio lies between

50-200% for practical applications [32]. The MTJ resistance at any angle of polarization, θ, can be derived using the following equations:

$$R_{MTJ}(\theta) = \frac{2R_{MTJ}(1+TMR)}{2+TMR+TMR\times cos\theta} = \begin{cases} R_P = R_{MTJ}, \quad \theta = 0° \\ R_{AP} = R_{MTJ}(1+TMR), \quad \theta = 180° \end{cases} \tag{2.3}$$

$$R_{MTJ} = \frac{t_{ox}}{F \times A\sqrt{\varphi}} \exp(1.025 t_{ox}\sqrt{\varphi}) \tag{2.4}$$

where TMR is the tunneling magnetoresistance, t_{ox} is the thickness of the oxide layer, F is a parameter that depends on the resistance-area product of the MTJ, A is the MTJ surface area, and φ is the energy barrier height of the oxide layer. MTJs have been fabricated at varying resistance levels ranging from the $K\Omega$ [33] to $M\Omega$ [34] range.

During the early phases, the switching of MTJ between the *P* and *AP* states was achieved by applying external trigger agents such as magnetic-field and temperature, which are known in the literature as field-induced magnetic switching (FIMS) [35] and thermally assisted switching (TAS) [36], respectively. In FIMS, *P-to-AP* (or, *AP-to-P*) switching is triggered by applying a current through bit and digit lines, which generates a magnetic field, resulting in the desired switching based on the current direction. TAS is an improvement over FIMS, as TAS requires a *heating* current to be passed through a single current line, generating a switching magnetic field of sufficient strength. The major drawbacks of FIMS and TAS are the requirement of a large switching current (>10mA) and a significantly longer cooling period post switching, respectively. Moreover, both the techniques warrant significant switching energy utilization, which deems them not suitable for low-power and energy-constrained applications. [37] addressed this shortcoming by proposing an alternative switching mechanism known as spin transfer torque (STT), which achieved widespread commercialization.

2.1.2 Spin Transfer Torque-Magnetic Tunnel Junction (STT-MTJ)

Different types of spin polarized currents play a crucial role in the switching of MTJs. Table 2 below lists the different types of current based on charge and spin. In STT-MTJ, a bidirectional spin-polarized current is applied to carry out the desired switching operation. An important parameter that determines the corresponding switching behavior in STT-MTJs is the critical current (I_{cc}), which can be derived from the following equations for in-plane (I_{CC_IP}) and perpendicular STT-MTJs (I_{CC_P}) [38], [39],

$$I_{CC\_IP} = 2\alpha e M_S V \frac{H_C + \frac{H_{eff}}{2}}{g(\theta) P \hbar} \tag{2.5}$$

$$I_{CC\_P} = \frac{\alpha e M_S V \gamma\, H_k}{\mu_B\, g(\theta)} \tag{2.6}$$

where α is the damping constant, γ is the gyromagnetic ratio, μ_B is the Bohr magneton, V is the free layer volume, H_C is the in-plane coercive field, H_{eff} is the effective out-of-plane demagnetization field, H_k is the anisotropy field, M_S is the saturation magnetization, $\hbar$ is the reduced Planck's constant, and e is the electric charge. In order to switch, the spin-current applied to the MTJ should exceed the critical current, I_{CC}. The switching duration can be calculated using the Sun Model [40], as follows:

$$\frac{1}{T_{sw}} = \left(I_{MTJ} - I_{CC}\right)\left[\frac{2P\mu_B}{em(P^2+1)(E_C+\ln(\pi^2\Delta))}\right] \tag{2.7}$$

where T_{sw} is the mean switching duration, Δ is the thermal stability factor, E_C is the Euler's constant (= 0.577), and m is the free layer magnetic moment.

Table 2: Different types of currents in MTJ.

Type	Features	Resulting Current
Unpolarized Current	Equal number of up and down spin electrons flowing in the same direction	Only charge current; no spin current
Polarized Current	Majority spin-up electrons and minority spin-down electrons, flowing in the same direction	Non-zero spin and charge currents where, charge current > Spin current
Fully Spin-Polarized Current	Flow of only spin-up or only spin-down electrons in one direction	Charge current = Spin current
Pure Spin Current	Equal number of up and down spin electrons flowing in the opposite direction	No charge current; only spin current

Fig. 3 depicts a conventional STT-MTJ structure. While STT switching promises significant advantages as compared to its predecessors, TAS and FIMS, mostly in terms of scalability due to denser layout, it still has certain disadvantages. STT switching utilizes the same line for current flow for reading form and writing to the device. This may cause unreliable read operations, causing unintended writes to the device. In addition, the device exhibits asymmetric write energy utilization between AP-to-P and P-to-AP states, which presents design issues in regularizing device operations in larger constituent circuits.

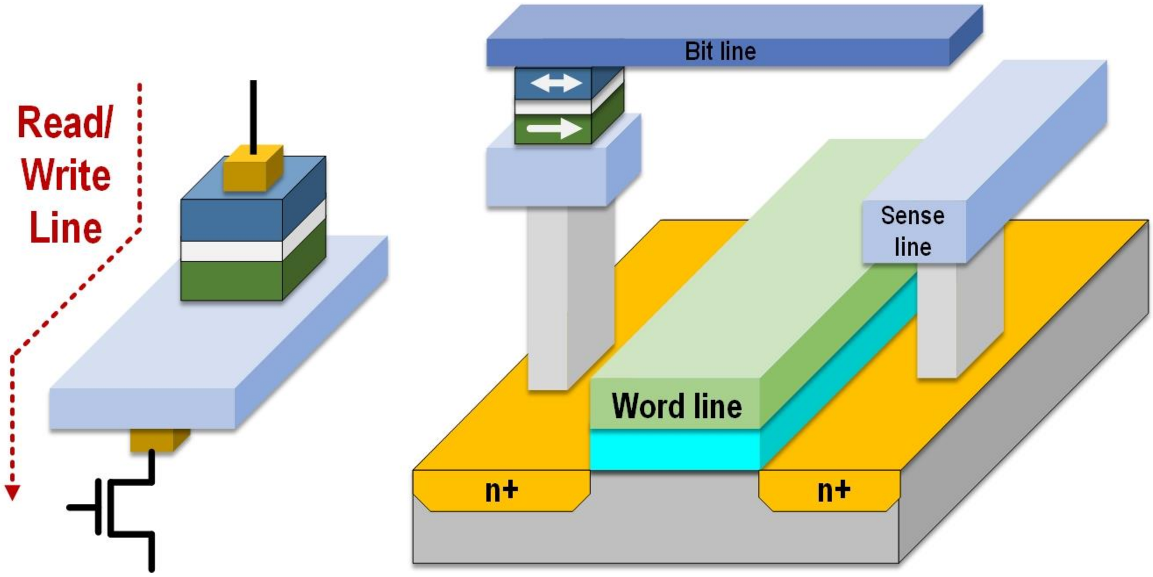


Figure 3: Spin Transfer Torque Magnetic Tunnel Junction (STT-MTJ) Structure

2.1.3 Spin Hall based MTJ Switching

In STT-MTJs, the write duration is intentionally kept longer than the switching duration to achieve reliable writes. This is because STT switching mandates a certain delay to be considered, which is referred to as incubation delay, arising mostly due to highly stochastic thermal fluctuations. This limits the speed of operation as well as results in unnecessary energy utilization. Spin Hall Effect (SHE) MTJs can overcome the limitations of STT-MTJs. Unlike the conventional 2-terminal MTJs, SHE-MTJs are three-terminal devices with the presence of distinct read and write paths and a heavy-metal layer adjacent to the free layer of the MTJ. The switching principle resembles the Anomalous Hall Effect (AHE). In AHE, applying a charge current produces a transverse spin current in the device, when the ferromagnetic material is placed in the presence of an external magnetic field. However, in SHE, a pure spin current is generated in the heavy-metal layer with a high atomic number, without the necessity of an external magnetic field.

Fig. 4 depicts the physical structure of a SHE-MTJ along with the peripheral write circuitry. The *P-to-AP* or *AP-to-P* switching is achieved by passing a charge current through the heavy metal, as shown in Fig. 4. The unpolarized charge current through the heavy metal layer along the X-axis results in a change in magnetization along the Y-axis and generation of spin-polarized current along the Z-axis direction orthogonal to that of the unpolarized current. The spin current so produced transfers its angular momentum to the free layer resulting in switching behavior as shown in Fig. 4a and Fig. 4b. Generally, the magnitude of the generated spin-current is greater than that of the charge current, making the spin-hall injection efficiency (SHIE), which is the ratio of spin current to charge current, greater than 1. The SHIE can be derived from the below equation,

$$SHIE = \frac{\pi}{4}\left(\frac{w_{MTJ}\times\, l_{MTJ}}{w_{HM}.t_{HM}}\right)\ \theta_{SHE}\left[1-\mathrm{sech}\left(\frac{t_{HM}}{\lambda_{sf}}\right)\right] \tag{2.8}$$

where l_{MTJ} and w_{MTJ} are the length and width of the MTJ, respectively, t_{HM} and w_{HM} are the thickness and width of the heavy-metal layer, respectively, θ_{SHE} is the spin-hall angle, and λ_{sf} is the heavy-metal's spin flip length.

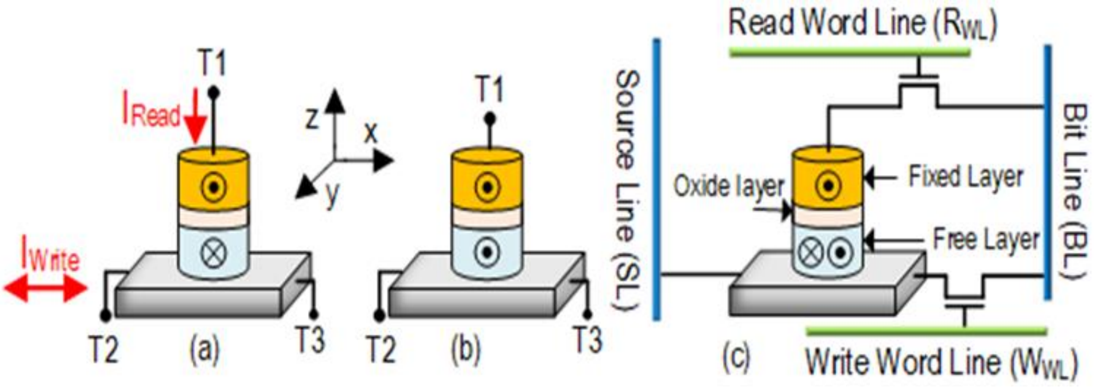


Figure 4: SHE-MTJ (a) AP Configuration, (b) P Configuration, and (c) Bit-cell with Read/Write Circuitry [1]

2.1.4 Probabilistic Bit (P-Bit) Devices

An NMOS with transistor and MTJ comprise the probabilistic bit, or p-bit, an emerging hybrid device that converts an analog input signal to a digital output. Fig. 5(a) depicts the p-bit structure [41], [42]. The MTJ used in a p-bit device is a low-barrier MTJ with energy barrier, E_B, where $E_B \ll 40kT$. Under this condition, thermal fluctuations at room temperature are sufficient to change the state of the device. The probability of the digital output being a high logic (i.e., 1) is determined by the supplied input voltage. The p-bit's configuration as a voltage divider between a low-barrier MTJ and NMOS transistor enables this functionality. An increase in the gate voltage utilized in the transistor leads to a decrease in the drain-source voltage, denoted as r_{ds}. This decrease in voltage enhances the likelihood of supplying an adequate amount of current to the inverter's input, thereby producing a logic 1 output.

The p-bit output can be determined using the following equation,

$$V_{out} = V_{DD}\, sgn\, \{\tanh\left(\frac{V_b}{V_0}\right) + random\ (-1, 1)\} \tag{2.9}$$

where V_{DD}, V_b and V_0 are the supply voltage, bias voltage, and model parameter, respectively, $random\ (-1, 1)$ is a random number in [-1, 1], and sgn represents the *sign* function. The below equation computes the probability of obtaining a high p-bit output,

$$P(1) = \frac{1}{2}\left(1 + \tanh\left(\frac{V_b}{V_0}\right)\right) \tag{2.10}$$

The p-bit output is averaged to implement the hyperbolic tangent function through Eq. 2.10.

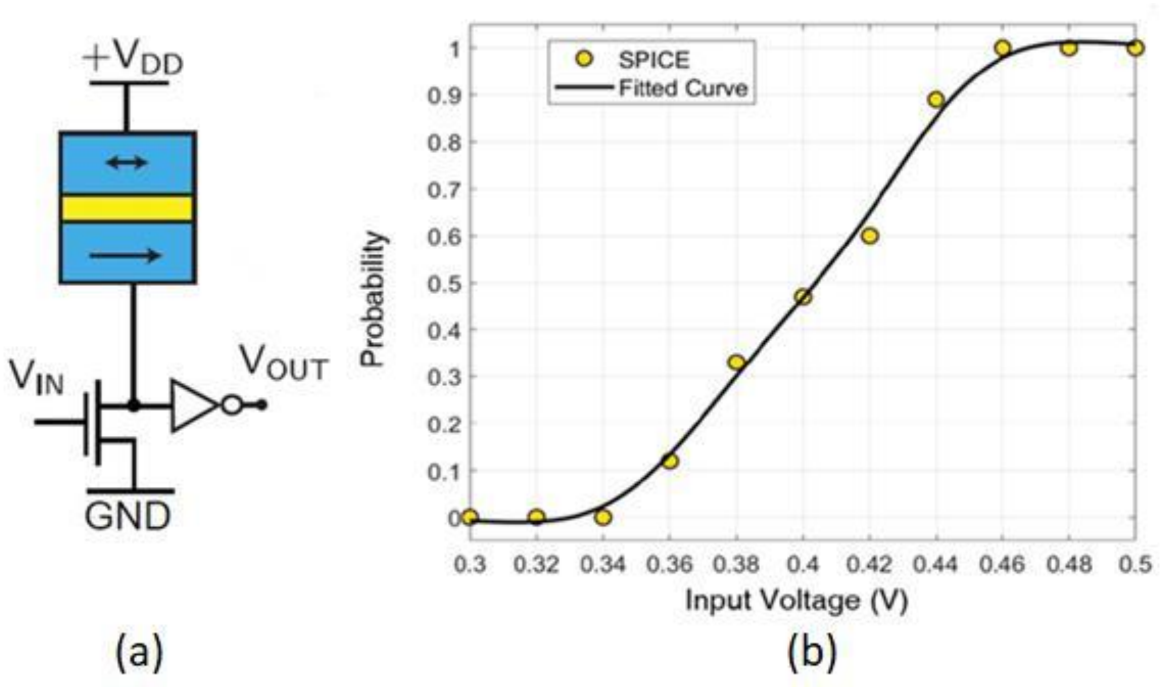


Figure 5: (a) A p-bit Structure, and (b) Probability of a High-Logic at the Output [43]

2.2 Deep Belief Networks (DBNs)

2.2.1 Restricted Boltzmann Machines (RBMs)

Restricted Boltzmann Machines, referred to as RBMs, are a category of recurrent stochastic neural networks [43] in which the following equation specifies the energy of the network in state *k*:

$$E(k) = -\sum_i s_i^k\, b_i - \sum_{i<j} s_i^k\ s_j^k w_{ij} \tag{2.11}$$

where w_{ij} denotes the weight between nodes i and j, while s_i^k signifies the state of node i when the network is in state k. The probability that each node in an RBM is in state 1 is given by the equation below,

$$P\ (s_i = 1) =\ \sigma\left(b_i + \sum_j w_{ij} s_j\right) \tag{2.12}$$

where σ is the sigmoid function. Over time, a Boltzmann distribution is attained, which establishes the following probability for locating the system in state *j*:

$$P(k) = \frac{e^{-E(k)}}{\sum_u e^{-E(u)}} \tag{2.13}$$

where the summation in the denominator is taken over all possible states of the system. An RBM is a two-layer neural network consisting of a visible layer and hidden layer; by stacking RBMs, it is possible to realize a DBN of arbitrary length [43].

2.2.2 Probabilistic Inference Network Simulator (PIN-Sim)

At a software and hardware level, DBN simulations on the MNIST dataset are easily realizable via the Probabilistic Inference Network Simulator (PIN-Sim) [43]. PIN-Sim comprises five modules. The first module, *trainDBN*, reads the training images in MATLAB and produces weight and bias matrices that describe the DBN. *mapWeight* is the second module in MATLAB, which transforms the weight and bias data into device conductance values. Following this, SPICE representations of multiple crossbar weighted arrays are generated by the *mapRBM* Python module, based on the *mapWeight* outputs and the specified network topology. The final Python module, *testDBN*, determines the classification error rate and power consumption of the DBN through the execution of a SPICE circuit simulation. The *testDBN* module receives as inputs the results obtained from *mapWeight* and *mapRBM*, in addition to the neuron module, which is a SPICE representation of the circuit utilized for activation function computation. The logic flow of PIN-Sim is depicted in Fig. 6.

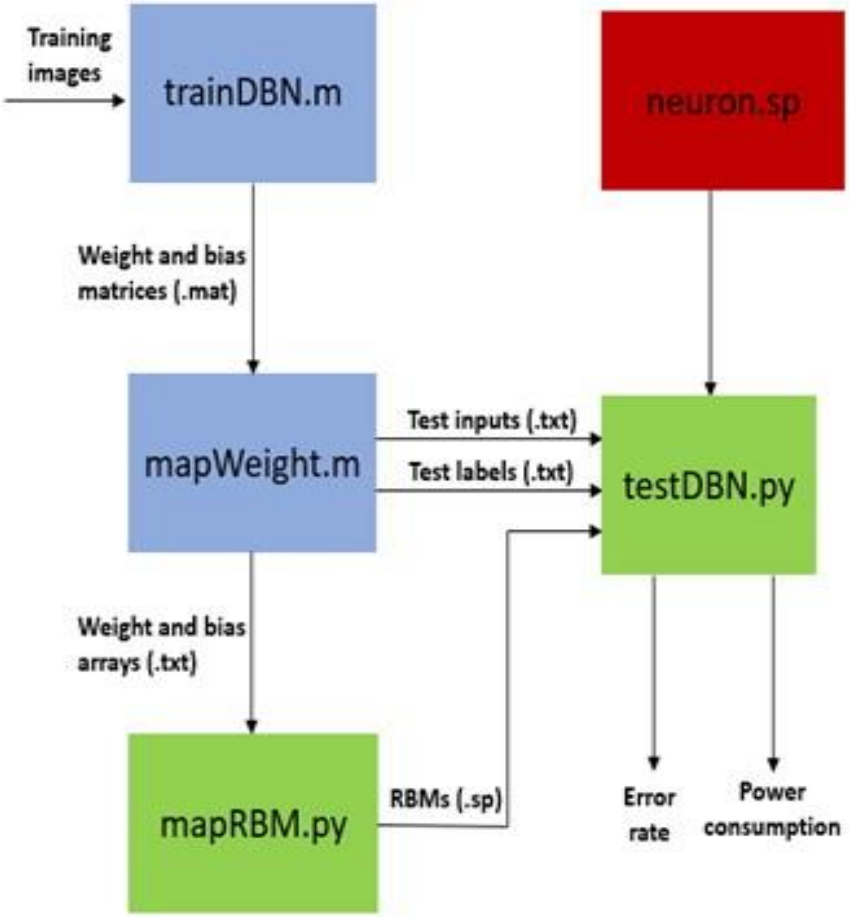


Figure 6: The Five Main Modules of the PIN-Sim Framework [43]

CHAPTER 3: MRAM vs. SRAM: A COMPREHENSIVE ANALYSIS OF SCALING TRENDS AND LIMITS FOR IOT DEVICES[3]

3.1 Background

At the deep submicron level, SRAM bit cells encounter significant leakage power dissipation and limited storage density [44]. In contrast, STT-MRAM provides embedded bit cells with vertical integration and near-zero standby power dissipation [45]. As the technology scales, static power dissipation constitutes an increasing proportion of total bit cell power loss, which can be especially prominent during standby (idle) periods. Fig. 7 depicts these considerations in terms of a taxonomy spanning their most significant operational and design viewpoints. Switching current is primarily impacted by two components which are the intensity of write current and average duration of IoT device activity. Meanwhile, leakage current of the bit cell technologies is primarily influenced by the availability of non-volatile retention and duration of the standby interval of the IoT device. Finally, sensing reliability aspects are constrained by the circuit's immunity to process variation, especially the sense amplifier in the case of resistive bit cells such as MRAM [46]. These factors combine to influence the overall scalability of the device technology at deep sub-micron nodes.

[3] ©IEEE. Part of this chapter is reprinted, with permission, from [2]

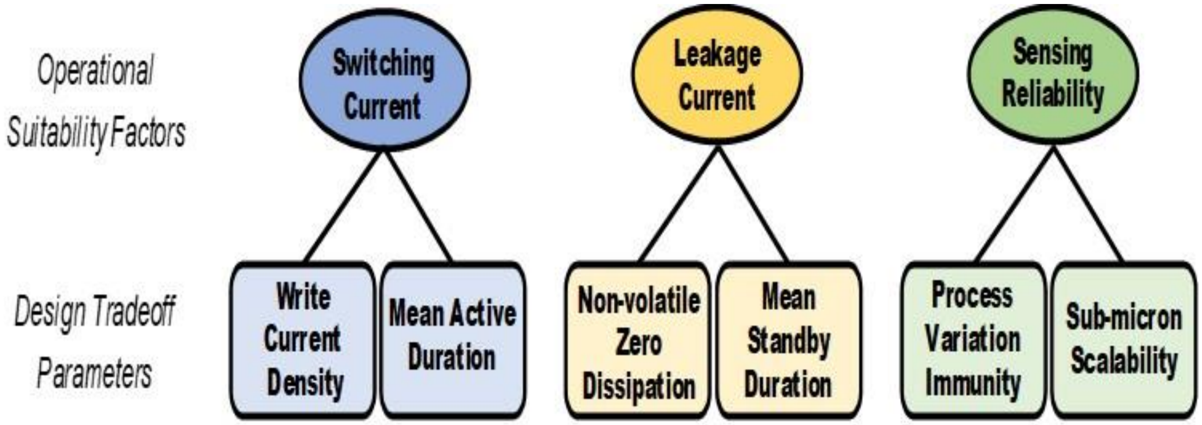


Figure 7: SRAM vs MRAM Co-design Considerations for IoT Devices [2]

An initial comparison is made qualitatively between the power profiles of SRAM and MRAM technologies for embedded intermittently powered devices. The static power dissipation can be modeled via direct superposition of underlying mechanisms such as gate tunneling and conduction through reverse-biased p-n junctions, as certain non-idealities stemming from the subthreshold leakage current result from their CMOS transistors [47]. Additionally, dynamic power dissipation is a result of the active operation of the memory cell, mainly during write operations.

Moreover, the reliability of write operations are directly related to the device characteristics. For SRAM, write reliability might not be of much importance since the overhead cost of an additional write operation in terms of delay and power dissipation is negligible. However, using MTJ devices, overhead cost of an additional write operation would be significant, and it can become a burden on the dynamic power dissipation, and some manufacturers conduct two write cycles [48]. MTJ characteristics outline the MRAM's reliability in terms of write failures and read disturb failures, which mainly arise due to thermal instability of the MTJ nanomagnet, insufficient switching duration, readability degradation due to technology scaling on the access transistor of the MTJ device and write polarization asymmetry while switching between ‘0’ and ‘1’ states of the MTJ [49].

3.2 Related Works

The primary contributor to leakage power dissipation in CMOS transistors is the subthreshold leakage that is caused by the current that flows from drain to source when the transistor is in standby mode. The leakage current, I_{lkg}, can be modeled by the following equation [44],

$$I_{lkg} = I_{S0} \times \left(e^{-\frac{v_{off}}{nV_t}}\right) \times \left(e^{-\frac{V_{th}}{nV_t}}\right) \tag{3.1}$$

where, n is the threshold swing factor, V_{gs} is the gate to source voltage, V_{th} is the threshold voltage, I_{S0} is the current dependent on the transistor’s geometry, $V_t = kT/q$ is the thermal voltage, k represents the Boltzmann constant, T accounts for the external temperature, and q is the electron charge.

The static and dynamic power dissipations of the SRAM cell can be estimated using the following equations,

$$P_{dynamic} = C_L \times {V_{CC}}^2 \times f_{clk} \tag{3.2}$$

$$P_{static} = I_{lkg} \times V_{CC} \tag{3.3}$$

where, C_L is the capacitive load and f_{clk} is the clock frequency. Total power dissipation is determined through dynamic and static power, with near negligible power contributed from the short circuit current that will be ignored for our calculations. The leakage current can be modeled via direct superposition of underlying mechanisms such as gate tunneling and conduction through reverse-biased p-n junctions comprising the 6 MOSFET cell [47].

Given [50], equations for STT-MRAM static and dynamic power dissipation are provided below, where $I(t)$ is the write current that passes through the MTJ device and T is the pulse duration for the write operation. Since the MTJ devices have near-zero leakage as discussed earlier, most of the leakage power dissipation comes from the write transistors. The following simplified equations are used to provide a model to estimate the power dissipation of STT-MRAM bit-cells.

$$P_{dynamic} = f_{data} \times V_{CC} \times \int_0^T I(t)dt \tag{3.4}$$

$$P_{static} = I_{lkg} \times V_{CC} \tag{3.5}$$

Several models exist for estimating the static and dynamic power scaling trends for SRAM devices and corresponding memory array structures developed over the past two decades [44], [51], [52]. However, some of those suffer from inaccuracies because of node capacitances and can be difficult to estimate using simple analytical models. Furthermore, there has been significant interest in the development of models that estimate power dissipation and the effects of scaling in memories using emergent devices. Stillmaker and Baas developed a fast and scalable model for determining the area, power and delay performance of a CMOS system based on a cascaded chain of inverters [53]. Smullen et al. [54] developed a simulation and modeling system (STeTSiMS) for STT- MRAM based devices and demonstrated their model using three different designs of the MTJ based memory cell. Meanwhile, Togashi et al. designed a 16bit/32bit binary counter using MTJ devices and compared power dissipation with corresponding CMOS based design for 45nm and 16nm technology nodes [55].

Chun et al. compared the scaling trends of the read and write performances of STT-MRAM to those of SRAM in [45], where Monte Carlo (MC) simulations were conducted to account for PV effects. In addition, Jaiswal et al. examined scaling trends for a range of performance parameters, including write current, TMR, area, read failures, and write failures, for bit-cells comprising three different MTJ-based design structures, from 45nm to 11nm [49]. The impact of reducing the size of the bit-cell from 28nm to 20nm on various aspects of double-barrier MTJ (DMTJ) STT-MRAM memory was investigated by Garzón et al [56]. This included resistance, write access time, and energy consumption due to scaling. Furthermore, they expanded their findings from a device-level to an architecture-level framework, wherein they examined scaling patterns on STT-MRAM and contrasted the outcomes with those of conventional SRAM implementations utilized for the final level cache. Table 3 summarizes the models developed in recent times, as discussed above.

Table 3: Analysis of Scaling Trends of CMOS and STT-MRAM: A Summary of Recent Works

Author	Focused Model	Equations for Predictive Modeling	Parameters Considered
Stillmaker & Baas **2017**	Predictive polynomial model for delay, energy, and power scaling for CMOS devices.	$DelayFactor = a_{d3}V^3 + a_{d2}V^2 + a_{d1}V + a_{d0}$ $EnergyFactor = a_{e2}V^2 + a_{e1}V + a_{e0}$ $PowerFactor = a_{p2}V^2 + a_{p1}V + a_{p0}$	$D_x = \frac{DelayFactor_x}{DelayFactor_y} \cdot D_y$ $E_x = \frac{EnergyFactor_x}{EnergyFactor_y} \cdot E_y$ $P_x = \frac{PowerFactor_x}{PowerFactor_y} \cdot P_y$ Power/ delay/ energy of any technology node can be anticipated if power/ delay/energy of one technology node is known.
Chun et al. **2013**	Scaling analysis of write delay, sensing delay, and Read Disturb Rate (RDR) of in-plane and perpendicular STT-MRAM memory based on semi-empirical model.	$I = \exp\left\{-\left(\frac{E}{k_B T}\right)\left(1 - \frac{I_{MTJ}}{I_{C0}}\right)\right\};$ $t_p = t_0 \exp\left\{\left(\frac{E}{k_B T}\right)\left(1 - \frac{I_{MTJ}}{I_{C0}}\right)\right\} + t_{t\to p}\left(I_{t\to p}/I_{MTJ}\right)^2;$ $V_{TMR} = V_{ctrl} * \exp\left({V_{MTJ}}^2/2c^2\right)$	Thickness of free layer, thermal stability factor, $(J_{C0} * RA/V_{DD})$ ratio factor
Jaiswal et al. **2016**	Predictive scaling trends for write power, TMR %, and area of different anisotropy based STT-MRAM bit cells based on LLG equations and NEGF.	$J_C = J_{CO}\left\{1 - \left(\frac{KT}{E_b}\right) ln\left(\frac{\tau_P}{\tau_O}\right)\right\};$ $J_{CO} = \frac{2e\alpha M_s t_{FL}}{\hbar\eta}\left(H_{K\|\|} + 2\pi M_s\right)$	Satuaration Magnetization, Damping constant, Energy Barrier, Length, Aspect Ratio, Technology Node, free-layer thickness.
De Rose et al. **2017**	Verilog-A device model, and 0.8V Fin-FET library used for scaling analysis of nominal write current to critical current ratio, worst-case write delay and write energy in hybrid FinFET/MTJ memory array structures.	$I_{C(P\to AP),(AP\to P)} = \frac{\beta_{c(P\to AP),(AP\to P)} e\gamma_0\mu_0 M_S^2 V_{FL}}{2\eta g\mu_B};$ $TMR(V) = \frac{TMR(0)}{1+\left(\frac{V_{MTJ}}{V_H}\right)^2};$ $\varDelta = \frac{\mu_0 M_S^2 V_{FL} k_{eff}}{2k_B T};$ etc.	Voltage-dependent perpendicular magnetic anisotropy, temperature, thermal heating/cooling, MTJ process variations, and the spin-torque asymmetry.

All the related works that correspond to STT-MRAM mentioned above provide an insight for the scaling effects on write delay, write energy, and/or sensing delay. However, they do not provide a general rule or metric to assist researchers and circuit designers to be able to estimate the power dissipation of hybrid CMOS/MTJ designs at scaled technology nodes. Moreover, previous works do not account for the power efficiency of STT-MRAM memory compared to conventional SRAM memory within IoT applications, where most of the time system is in the standby mode and leakage power dissipation is dominant. Thus, there is a need for a model that considers the time that the memory spends in standby mode compared to that spent in reading from or writing to the memory.

This dissertation presents an analysis and comparison of static and dynamic power dissipation trends for a 6T-SRAM bit-cell and a STT-MRAM bit-cell. The comparison is based on technology scaling and four sub-micron technology nodes ranging from 45nm to 16nm. This study introduces two performance parameters, Mean Standby Duration (MSD) and Mean Active Duration (MAD), which are crucial factors in IoT applications utilizing emerging non-volatile devices. Additionally, a novel metric for comparing power dissipation across various technology nodes has been developed, surpassing previous research in the field.

The subsequent sections provide further details regarding the proposed approach for modeling power dissipation, as well as our simulation environment and setup.

3.3 Experimental Setup and Approach

The motivation for this research is to quantify trends in power efficiency achievable using STT-MRAM memory bit-cell versus SRAM down to 16 nm technology nodes. Considering array structure implementations of SRAM and STT-MRAM since the peripheral circuitry vary depending on different implementations, the proposed method focuses on a single memory bit-cell comparison. Fig. 8(a) depicts a 6-T SRAM memory cell and Fig. 8(b) illustrates an STT-MRAM memory cell consisting of an MTJ device and a transmission gate as the write circuit. The 2T-1R structure is chosen as it is shown to be the optimal configuration to achieve low write energy [56].

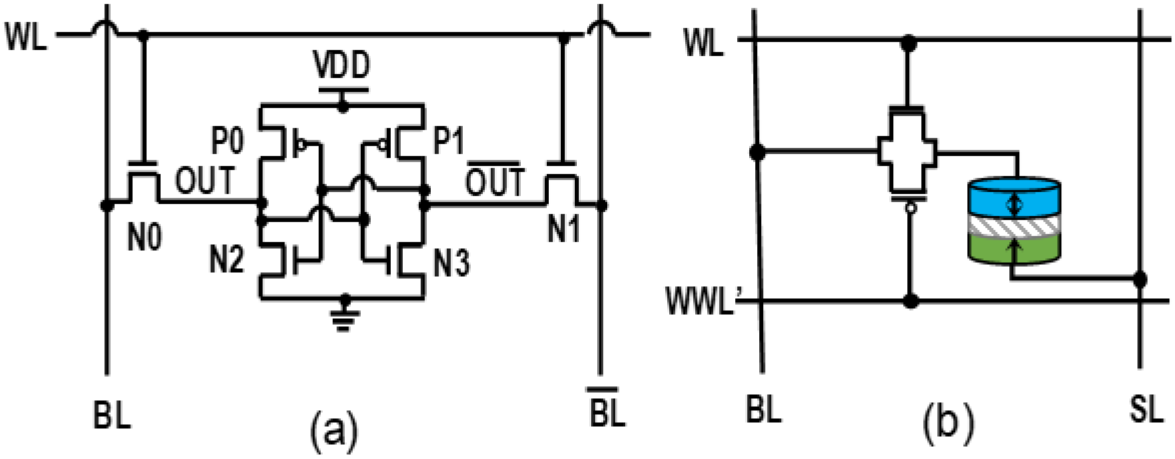


Figure 8: (a) 6-T SRAM Bit Cell, (b) STT-MRAM Bit Cell [2]

The PTM, as given in [16], has been utilized for the NMOS and PMOS transistors in both SRAM and STT-MRAM bit-cells while the model in [57] was used for the MTJ device. We used SPICE for circuit simulations using the parameters listed in Table 4 or otherwise typical of two-terminal STT-MTJ in the literature [55], [58]. The SRAM transistor parameters are defined to achieve reliable and stable SRAM operation [59], which has shown to be feasible for fabrication as discussed in the literature [7]. All simulations are carried out at temperature, T=298K and for switching operation with iso-write time of 5ns for the STT-MRAM bit-cell as well as 6T-SRAM bit-cell. These simulations were performed utilizing four contemporary and trending sub-micron technology nodes, namely 45nm, 32nm, 22nm, and 16nm with nominal voltages of 1.0V, 0.9V, 0.8V, and 0.7V, respectively. Parasitic node capacitance values are important for correct modeling and simulation of memory bit-cells in SPICE. Therefore, the accuracy of our simulation results was further improved by accounting for the parasitic node capacitances by designing the layout for the SRAM and STT-MRAM bit-cells and extracting those parasitic capacitances and area from the layout, for the parameters listed in Table 4.

The layout for SRAM and STT-MRAM bit-cells are shown in Fig. 9(a) and 9(b), respectively. According to the layout, the bit-cell area of SRAM and STT-MRAM are estimated to be $861F^2$ and $336F^2$, respectively, where *F* is the feature size or technology node. While this research examines device technology scaling consideration to evaluate the leakage and dynamic power dissipation, readers interested in detailed post-layout considerations, fabrication aspects, and associated challenges can refer to [60], [61], [62]. Furthermore, as technology nodes scale, the reliability challenges for both SRAM and STT-MRAM bit-cells increase, mainly due to PV effects that necessitate increased dynamic power dissipation. Thus, we perform 1,000 Monte Carlo (MC) simulation runs considering the variations discussed to achieve more accurate simulation results in terms of reliability with regards to scaling. For the MC simulations, we have considered 1% variation on the *width* and *length*, along with 10% worst case variation on the *threshold voltage* of NMOS and PMOS transistors. Consideration of 1% variation on the *width*, *length*, and *thickness* of the MTJ free layer as well as 1% variation on the *oxide thickness* [63] were considered.

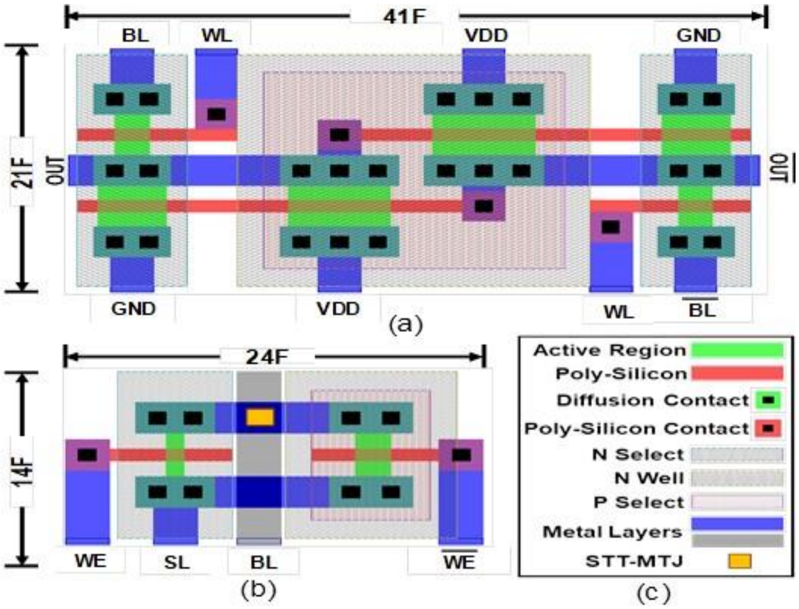


Figure 9: (a) 6-T SRAM Layout, (b) STT-MRAM Layout, and (c) Layout Legend

Table 4: Technology Parameters for SRAM and MRAM Bit Cells

Parameter			**Mean Value and Description**			
			45nm	**32nm**	**22nm**	**16nm**
SRAM	**PMOS**	W/L	270/45nm	192/32nm	132/22nm	96/16nm
	NMOS	W/L	180/45nm	128/32nm	88/22nm	64/16nm
		W/L Access Transistor	90/45nm	64/32nm	44/22nm	32/16nm
STT-MRAM	**PMOS**	W/L	90/45nm	64/32nm	44/22nm	32/16nm
	NMOS	W/L	45/45nm	32/32nm	22/22nm	16/16nm
		tox	Oxide Thickness			1.5nm
		Ms0	Saturation Magnetizaion			456
		P_0	Polarization Factor			0.69
		α	Damping Factor			0.007
		T	Temperature			298
		RA0	Resistance Area Product			5 Ω - μm^2
		t_c	Critical Thickness			1.5nm
		TMR	Tunnel Magnetoresistance			120%

The MTJ write operation is asymmetric nature, which requires greater write pulse duration to switch from *AP* state to *P* state compared to switching from *P* state to *AP* state. These variations have been considered to account for possible write failures that may occur during the write operation. The parameters of the STT-MRAM bit-cell and the write pulse duration were carefully selected to maintain <0.001 write error rate in the 1,000 MC simulation runs, while achieving a fair comparison with the SRAM bit-cell. Finally, once the simulations are performed, we utilize a polynomial curve fitting using MATLAB on the results to provide a more accurate model for power dissipation of memory bit-cells that can be generalized for scaled technology nodes ranging from 45nm to 16nm and beyond.

3.4 Simulation Results

Simulation results for the SRAM and STT-MRAM bit-cell configurations considering iso-write pulse duration of 5ns for all four technology nodes ensures reliable switching of both bit cells. The power dissipations for both bit-cell configurations are listed in Table 5. Additionally, we have plotted the write and static power measurements in logarithmic scale for both SRAM and STT-MRAM power dissipation in different technology nodes for comparison in Figs. 10(a) and 10(b), respectively. The dynamic power dissipation for both SRAM and STTMRAM bit-cells are reduced with technology scaling. However, due to the high energy barrier of the MTJ device to maintain increased thermal stability and high endurance, the dynamic power dissipation of STT-MRAM is greater than SRAM. On the other hand, the static power consumption of the SRAM bit-cell increases with technology scaling, which is an expected behavior according to the literature and is mainly caused by short-channel effects and increased leakage current in scaled technology nodes. In contrast, the static power consumption of the STT-MRAM reduces as the technology scales. We conclude from the simulation results that although the static power dissipation of STT-MRAM is significantly smaller than SRAM, the impact of the static power dissipation can be considerably significant in scaled technology nodes. This is because a majority of the lifetime of the memory circuit is spent in standby mode, thus leakage power dissipation becomes a significant contributor to the total power dissipation.

Table 5: Static and Dynamic Power Dissipation for Iso-Write Duration

Technology Node	STT-MRAM			SRAM		
	Dynamic Power (nW)	Static Power (pW)	Total Power (nW)	Dynamic Power (nW)	Static Power (pW)	Total Power (nW)
45 nm	46286.3	2.2975	46286	171.628	21377.3	193.01
32 nm	30007.8	1.6289	30007	106.604	29541.5	136.14
22 nm	19068.2	1.2880	19068	63.9465	53366.8	117.31
16 nm	12758	0.9869	12758	36.4842	120950	157.43

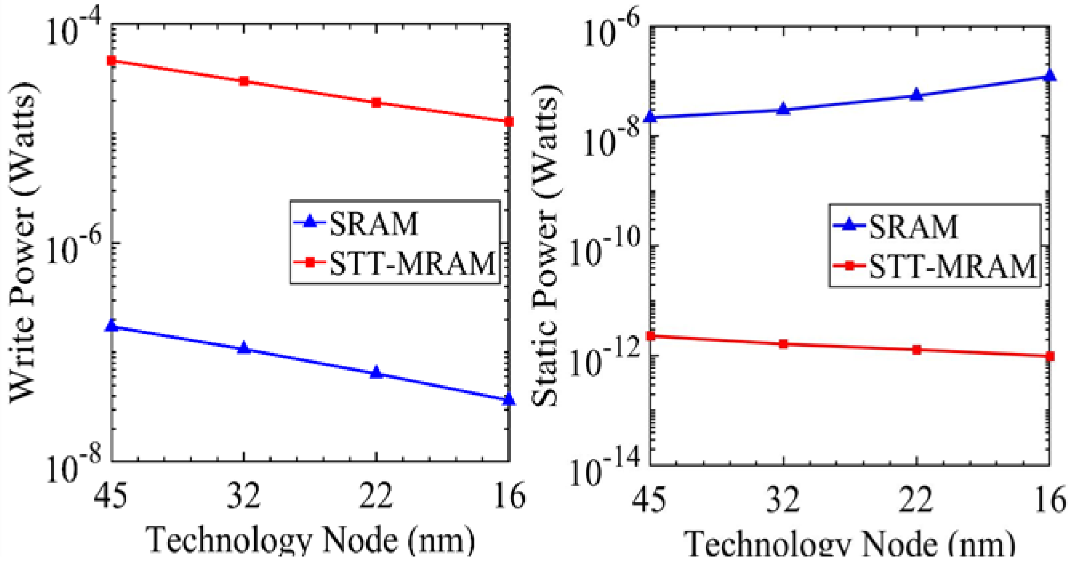


Figure 10: SRAM vs. STT-MRAM: (a) Dynamic Power Dissipation, and (b) Static Power Dissipation

3.5 Analytical Model and Performance Metrics

Two parameters, namely Mean Active Duration (MAD) and Mean Standby Duration (MSD) are introduced, to more accurately model power dissipation scenarios for intermittently active IoT devices. MAD provides a metric for the ratio of the memory time spent performing the write operation, while MSD provides a metric for the ratio of the memory time spent in the standby mode. Considering these two key parameters we can compare total power dissipation of SRAM and STT-MRAM bit-cells in each technology node using the following metric called Power Dissipation Scaling Ratio (PDSR) described as follows,

$$PDSR = \frac{P_{total_{SRAM}}}{P_{total_{MRAM}}}$$

$$= \frac{MAD_{SRAM} \times P_{dynamic\_SRAM} + MSD_{SRAM} \times P_{static\_SRAM}}{MAD_{MRAM} \times P_{dynamic\_MRAM} + MSD_{MRAM} \times P_{static\_MRAM}} \quad (3.7)$$

Values for MAD and MSD are determined according to architectural benchmarking for entire memory array and averaged for each bit-cell. By scaling the value of MSD from 0 to 1 in steps of 0.001, and simultaneously varying MAD as (1-MSD), simulations were performed to see the effects of MSD and MAD on the PDSR scaling ratio for memory array size of 256x256 bits. Fig. 11 is a logarithmic plot of PDSR vs. MSD, which shows that for MSD >0.995, i.e., only if memory is in standby mode for more than 0.995 fractions of the mean workload profile time of the intermittently powered device, the performance of MRAM cell in terms of power dissipation becomes better than SRAM cell. For improved readability of the logarithmic plot, only the values of MSD ranging from 0.5 to 1.0 have been shown in the figure, as the effect of MSD on PDSR was not found to be pronounced for MSD <0.5. Note that, embedded MRAM offers a nonvolatile memory technology option over Flash memory for intermittently active applications, such as data logging application in sensors for IoT devices [64] and is influenced by the frequency of transition between active and sleep mode. This is because Flash memory has longer write-time, dissipates more power, and has significantly lower write endurance of approximately a thousand write-cycles. MRAM is also a viable alternative to SRAM for ultra-low power IoT devices, which operate at lower frequencies [64]. Table 6 lists some real-world IoT applications [65], [66] as examples of intermittently active devices that also substantiate MSD metric thresholds mentioned above. Other embedded applications operating under intermittent computational conditions span FPGA-based [67], long-duration deployment sustainability [68], and high reliability signal processing systems [69].

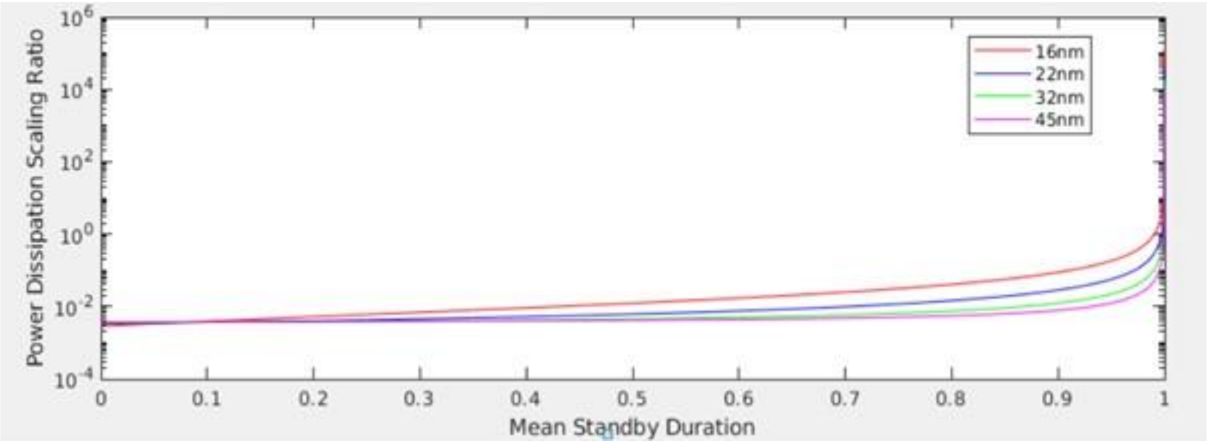


Figure 11: Power Dissipation Scaling Ratio (PDSR) vs. Mean Standby Duration (MSD) for 16 nm, 22 nm, 32 nm, and 45 nm Technology Nodes

Table 6: Duty Cycle Calculation of Some Real World IoT Applications

IoT Applications	Write Frequency	Embedded Memory	Estimated MSD
Smart Grid	2X hour	Cloud Storage	0.998
Surveillance Camera	1X hour	2GB	0.976
NFC Controllers	1X hour	1280KB	0.999
Key fob	1X minute	4KB	0.999
Thermostat	1X 10 years	512MB	0.999

For instance, considering the GRID energy dataset, the IoT based smart meter system operates on a 30-minute repetition loop. Considering an operational duration not exceeding 3 seconds per repetition cycle under conditions of maximum memory utilization, a standby duty cycle of 0.9983 is calculated. These results can provide valuable new insights on the fact that the tradeoffs between utilizing traditional vs non-volatile memory components in the design depends largely on the workload profiling of IoT device applications. Furthermore, once we gathered the SPICE simulation results, a polynomial curve fitting is performed using MATLAB to obtain predictive models for power dissipation and its scaling trends with a coefficient of determination, $R^2>0.95$. Table 7 lists the polynomial equations for average dynamic and static power dissipations of single SRAM and STT-MRAM bit-cells, where *x* is the target technology node, *p1*, *p2*, and *p3* are coefficients for the equations, and *nF* is the Normalizing Factor, which is the ratio of target technology node over normalized mean technology node. For example, the static power dissipation for 45nm node STT-MRAM bit-cell is estimated to be 2.44 pW using the proposed model considering *nF=1.565*, which has an error rate *<6%* compared to the simulated values provided in Table 5.

Table 7: Proposed Power Dissipation Estimation Model

Unit (Watts)	Equation	p1	p2	p3
$\mathbf{P_{st\_SRAM}}$	$nF \times (p1 \times x^{p2})$	5.675×10^{-24}	-2.093	0
$\mathbf{P_{dy\_SRAM}}$	$nF * (p1 \times x^2 + p2 \times x + p3)$	2.771×10^{-9}	5.814×10^{-8}	9.277×10^{-8}
$\mathbf{P_{st\_STT\text{-}MRAM}}$	$nF * (p1 \times x^2 + p2 \times x + p3)$	4.257×10^{-14}	5.483×10^{-13}	1.52×10^{-12}
$\mathbf{P_{dy\_STT\text{-}MRAM}}$	$nF * (p1 \times x^2 + p2 \times x + p3)$	9.398×10^{-7}	1.436×10^{-5}	2.635×10^{-5}

*Normalized Factor (nF) = (Desired technology node) ⁄ (Normalized Mean Node)

3.6 Summary and Discussions

As CMOS technologies continue to scale, static power dissipation continues to become a significant design issue constraining intermittently active application. Results indicate that SRAM bit-cell dynamic power decreases with scaling due to decrease in transistor write current and nominal threshold voltage, whereas static power increases exponentially due to sub-threshold leakage. On the other hand, for an STT-MRAM bit-cell, both static and dynamic power decreases with scaling due to near-zero leakage of the MTJ devices and reduced transistor count compared to 6T-SRAM. These provide utilization trends and limits, quantified using the novel metrics of MAD and MSD, for a more accurate estimation of the power dissipation for SRAM and STT-MRAM comparison in scaled technology nodes. Namely, it was deduced that when MSD is very high (~0.995), the power dissipation of embedded MRAM attained energy advantages over SRAM. Moreover, further assimilation of transistor-scaling impact with non-volatile devices furthers the understanding of energy profiles for low duty cycle applications.

CHAPTER 4: SCALABLE REASONING AND SENSING USING PROCESSING-IN-MEMORY WITH HYBRID SPIN/CMOS-BASED ANALOG/DIGITAL BLOCKS[4]

4.1 Processing-in-Memory (PiM): Overview and Architectural Milestones

Architectural advancements in pursuit of PiM computational paradigms have targeted various gainful attributes for special-purpose computing over the last five decades. Although a comprehensive summary would be too extensive, Fig. 12 delineates the progression of the noteworthy research milestones that have laid the foundation for the research herein. Specifically, application-specific PiM approaches have continued to evolve from distributed memory modules in conventional array processors up through hybrid spin/CMOS-based memory/processing cells capable of intrinsic execution of selected computations. Starting with segmented memory distributed physically across an ensemble of Processing Elements (PEs), Slotnick et al. fielded the *Illinois Automatic Computer (ILLIAC)* by researching the concept of distributed memory closely coupled with localized parallel processing operations via the association of segmented memory among identical PEs [70]. Next, by drilling down to the bit-cell level while focusing on the referencing capability of data when resident inside the memory component, Foster advocated the benefits and capabilities of a *Content Addressable Parallel Processor (CAPP)* [71]. The CAPP provided an umbrella term for hardware implementation of Boolean logic gates elements replicated within each SRAM bit cell, which tagged contents as

[4] ©IEEE. Part of this chapter is reprinted, with permission, from [1].

responders for further processing without involving off-chip processor/memory transactions. Leveraging the concept of content addressability for PiM, DeMara developed the *Semantic Network Array Processor parallel AI prototype* (SNAP-1) which used in-place computation initiated with Single Instruction Multiple Data (SIMD) broadcast mode [72]. The responder PEs storing the semantic network then launched a Multiple Instruction Multiple Data (MIMD) model of spreading-activation to conduct reasoning tasks without bus transactions using a multi-ported memory approach. Later, when microprocessors became ubiquitous in the computing landscape, including the MIPS chip he designed and helped to commercialized, Patterson advocated the case for I*ntelligent RAM (IRAM)* to unify logic elements within a DRAM memory module, thereby bridging the memory-wall between the processor and memory [73]. Next, while furthering the IRAM-style PIM paradigm, Elliot et al. researched tightly coupled integrations of more complex logic networks to capture data parallelism via SIMD architectural implementations of PiM. Elliot evaluated transistor count and area costs versus throughput benefits of embedding PiM of various granularities up through rudimentary ALUs consisting of a few hundred transistors [74].

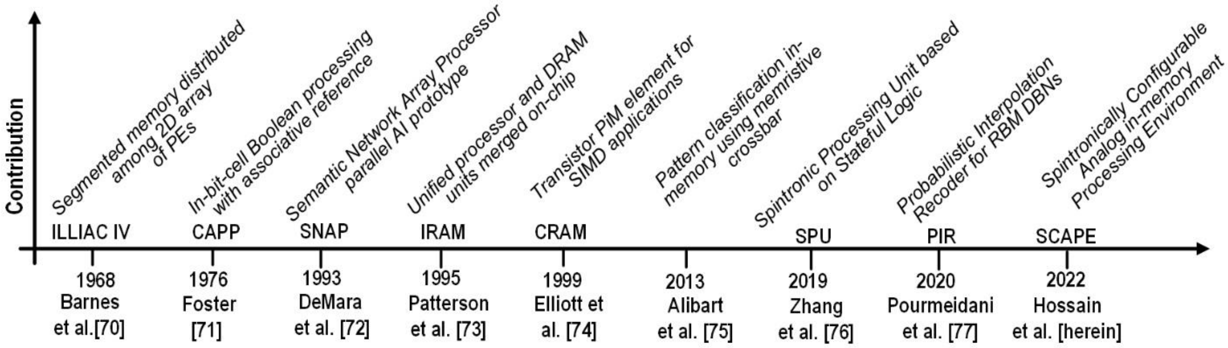

Contribution
Segmented memory distributed among 2D array of PEs
In-bit-cell Boolean processing with associative reference
Semantic Network Array Processor parallel AI prototype
Unified processor and DRAM units merged on-chip
Transistor PiM element for SIMD applications
Pattern classification in-memory using memristive crossbar
Spintronic Processing Unit based on Stateful Logic
Probabilistic Interpolation Recoder for RBM DBNs
Spintronically Configurable Analog in-memory Processing Environment
ILLIAC IV
CAPP
SNAP
IRAM
CRAM
SPU
PIR
SCAPE
1968 Barnes et al.[70]
1976 Foster [71]
1993 DeMara et al. [72]
1995 Patterson et al. [73]
1999 Elliott et al. [74]
2013 Alibart et al. [75]
2019 Zhang et al. [76]
2020 Pourmeidani et al. [77]
2022 Hossain et al. [herein]

Figure 12: Timeline of Foundational Works towards Hybrid Spin/CMOS-based Application-Specific Processing-in-Memory [1]

During the last decade, the aforementioned works promoted considerable research interest to extend the PiM paradigm beyond the use of transistors alone. These utilize emerging logic devices, such as memristors and spintronic devices as alternatives to CMOS-based memory designs. For instance, Strukov et al. in [75] showed emerging memristive devices could be used in a 2D-crossbar layout to conduct pattern recognition tasks leveraging the intrinsic switching behaviors of titanium-dioxide-based memristive devices within a *Computational RAM (CRAM)* component. Zhang et al. in [76] present a PiM platform called *Spintronic Processing Unit (SPU)*, configurable at the individual cell level for performing different logic functions using memory-like read and write operations. Different logic functions are computed by altering the final state of the memory cell based on different input operands. The final state of an STT-MRAM bit-cell is given by $B_{i+1} = AC + A'B_i$; where, A and C are the inputs to the WL and BL, respectively, and B_i and B_{i+1} are the initial data and the final result stored in the MTJ device, respectively. Different Boolean functions are achieved by altering the input variables A, C, and B_i. This work also shows how the *Instruction Set Architecture (ISA)* can be modified with additional instructional support such as `MOV` and `LOG`, for moving data to the target bit-cell and carrying out the logic operation based on value of input operands, respectively.

Although intrinsic switching functionalities of memristors in this context were shown to offer a viable new approach to PiM, the limited endurance of their write cycles and substantial drift of ON/OFF resistances presented new challenges. Thus, Pourmeidani et al [77] advanced a crossbar of non-volatile tunable stochastic elements based on MTJs by developing *Probabilistic Interpolation Recoder (PIR)* for Deep Belief Networks (DBNs). The MTJ devices were used to realize near-zero energy barrier switching supporting an unlimited endurance approach to PiM, whereas PIR provided a stochastic based energy and area efficient alternative to conventional interpolation technique of using resistor-capacitance (RC) tanks and analog-to-digital (ADC) convertors. The use of MTJ-based Non-Volatile Memories (NVMs) like commercialized Magnetic Random-Access Memories (MRAMs) allows feasibility for performing arithmetic and logic operations inside memory word lines. This memory word line approach to PiM led to energy-efficient hardware implementation of a *Restricted Boltzmann Machine (RBM) based Deep Belief Network (DBN)* using a conventional sigmoidal activation function. Furthermore, it was found that MTJs can be employed to realize area-efficient and wire-count efficient realization of neurons and synapses, elevating them as an emerging device technology useful for accelerating neural networks [78], [79]. Their properties of near-zero standby power, compatibility with CMOS *Back End of Line (BEOL)* fabrication process offering high integration density enables the implementation of efficient hybrid MRAM/CMOS circuits to combine the benefits of both technologies.

Taking inspiration from various technical attributes of these milestones in PiM approaches spanning the last five decades, herein we consider new roles and approaches to PiM for CS and ML applications. Specifically, we further the efforts in edge-of-network PiM with hardware implementation of a *Generalized Analog Activation Function (GAAF)* in a *Spintronically Configurable Analog Processing-in-Memory Environment (SCAPE)* architecture for selected applications.

4.2 Sensing and Reasoning Operations Amenable to Processing-in-Memory

Advancing beyond the foundational works on PiM, the last several years have witnessed interest in pursuing beyond Von Neumann approaches for efficient processing of data in edge-of-network applications such as compressive sensing and automated reasoning. Research has spanned multiple layers of the system stack, ranging from execution model and architectural topology down to algorithmic formulation, as well as the data representation and fundamental signal encoding methods. At the signal encoding stage, emerging spintronic devices enable new tradeoffs beyond the use of digital computation exclusively. In addition to providing computation ability to storage bit-cells in the memory, spintronic devices, due to their vertical-integration capability on MOS transistors, also offer potential area benefits at the cost of incurring additional fabrication complexity. A single bit-cell size comparison of different memory technologies found in [80] shows that STT-MRAM technology has lower cell size than SRAM but may be comparable to cell size of DRAM technologies. On the other hand, benefits of analog-based computations include reduced wire counts and device counts when compared to digital implementation of non-linear operations such as multiplication and exponentiation, spanning computer vision, signal processing, and machine learning applications.

For example, a conventional digital implementation of multiplication and exponentiation operations can result in substantial increases in both area and delay in the digital realm, requiring >12 clock cycles to execute and hundreds of Boolean logic gates [77]. Analog computation can be especially compatible in edge-of-network application domain owing to the tolerance for approximate computation. Analog circuits trade computational accuracy to minimize power dissipation and area overheads. This tradeoff is particularly appealing for power-/area constrained error-tolerant applications, such as IoT devices. Analog computation offers enhanced advantages when applied to vector-valued data, as the resulting data can be directly sent to a memristive crossbar array for additional processing, eliminating the requirement for digital-to-analog conversion. A multitude of applications rely heavily on multiplication and exponentiation operations, such as machine learning, signal processing, and computer vision. Such applications rely extensively upon VMM, wherein its fundamental operation of multiplication requires execution that is efficient and co-located near the data being operated upon. For instance, square root may function as an activation function for neural networks [78]. In signal processing applications, square and square root are frequently used to normalize vectors. An instance of a representative use case incorporating VMM is compressive sensing (CS), which entails compression and transmission of a spectrally sparse signal, which is then reconstructed at the receiving end. Another example is NL via neural networks. Herein, we propose a device to architecture level compound PiM implementation based on hybrid spin/CMOS, analog as well as digital computational blocks, re-distributed within the memory fabric, inter-communicating via simple control logic modifications to the peripheral circuitry. The major contributions of this chapter include:

1) development of a novel crossbar topology for PiM, which provides in-field configurability of hybrid spin/CMOS-based analog/digital blocks. Integrating memory devices into a PiM array should address various important metrics of both storage and computation. In this dissertation, the use of spintronic devices for PiM has been explored, as opposed to other

alternatives such as titanium dioxide based memristors, due to their virtually unlimited write endurance documented as 10^{16} write cycles. Various synapse and neuron designs are evaluated including use of SHE-MTJs for memristive-based computation and activation function calculation.

2) development of a generalizable activation function to mitigate the gradient decay problem while increasing recognition rate. Analog computation of the generalized activation function demonstrates acceptable accuracy, reduced area, and decreased energy consumption, as evaluated on MNIST dataset.
3) the concept of Power Error Product (PEP) is introduced as a transportable performance metric and is evaluated for various activation functions.
4) quantification of process variation (PV) effects when using SHE-MTJ devices. Approach and results for PV versus neuron activation function deviation are provided using Monte-Carlo method. Standard deviations of 5% for MTJ Parameters such as length, width, thickness are considered.

4.3 Proposed Spintronically Configurable Adaptive In-Memory Processing Environment (SCAPE) Architecture

Recently SHE-MTJs have been explored as means to realize in-memory computing architectures. This dissertation elaborates the developed *Spintronically Configurable Analog Processing in-memory Environment (SCAPE)* architecture, which incorporates top-down architectural approaches along with bottom-up intrinsic device switching behaviors of SHE-MTJs. Key technical objectives of *SCAPE* are to provide explicit hardware support collocated with large amounts of data that the edge of network devices must encounter, to process and send only higher-level information up the network to the cloud. Such applications have high data requirements, whereas they are typically streaming data as well as large templates of matrices, which can stress the memory bottleneck. Therefore, PiM is desirable. Both applications also manipulate data elements via dot product and rely on a large number of VMM operations at various precisions. In the context of machine learning domain, both the synapse and neuron have mathematical operations to perform. The synapse conducts a multiplication operation, while the neuron must perform activation based on thresholding using some type of activation function such as a sigmoid limiter. As mentioned in the previous section, a memristive crossbar conducts the synapse operation as an analog multiplication using current based representation of the values to be multiplied. For these operations, beyond-CMOS devices can add capability to calculate them as intrinsic behaviors of the switching device itself without having complex and area-consuming floating-point hardware units distributed throughout the memory.

An innovation in this research has been to provide a PiM element that can perform generalized analog multiplication and a *Generalizable Analog Activation Function (GAAF)*. Fig. 13 shows the high-level topology of the proposed SCAPE architecture. The memory component is laid out as a 2D crossbar array implementation to realize memristance at crossbar nodes. The SCAPE topology can embed an ANN within the memory as visible layers at the input/output interface of the memory component, and internal cascaded hidden layers, connected as per the machine learning network specification. Each of these layers can be abstracted into three distinct phases/stages:

(1) a Vector Matrix Multiplication Stage (VMMS) depicting the synaptic connections between the multiple nodes in each layer and computing the weighted dot-product of the input signals via the crossbar implementation,

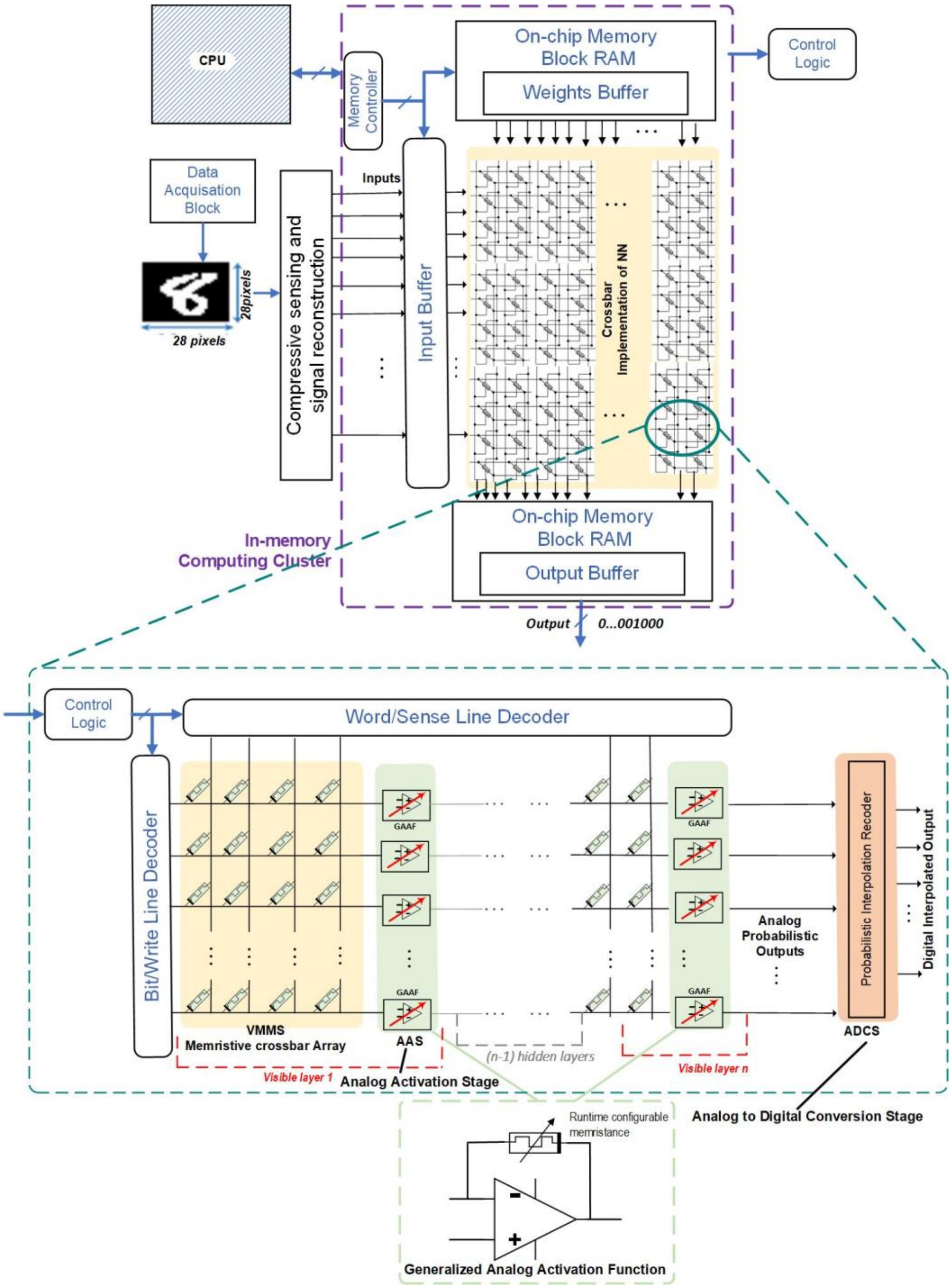


Figure 13: Proposed SCAPE Architecture [1]

(2) an Analog Activation Stage (AAS), consisting of the proposed GAAF blocks, which are composed of hybrid spin-analog components realizing various activations of the neuron in response to inputs, and

(3) an Analog-to-Digital Conversion (ADC) Stage, consisting of a spin-based Probabilistic Interpolation Recoder (PIR) [77], which converts the analog outputs of the AAS stage to digital at a low energy and area footprint.

For illustration, the process flow for an edge-of-the-network system has been showed, where an image from a benchmark dataset such as MNIST may be acquired from an input image acquisition block, and then via the on-board sensing and signal reconstruction stored into the input buffer of the memory unit. In the case of compressive sensing dot-product also needs to be performed which can be conducted intrinsically by the SHE-MTJ, as elaborated in Section 4.5.7. The training weights of the dataset are stored on the on-chip block RAM for efficient and quick access. The input buffer data and weights are then fed into the crossbar implementation of the ANN to produce dot products via analog computation. The weighted sums of inputs then propagate through the hidden layers of the neural network, and the corresponding activation layers comprised of the proposed GAAF blocks. A GAAF block consists of an analog hybrid-spin based three stage op-amp, with runtime configurable resistance providing the user with an in-field selectable range of more expressive activation functions, which can be configured at runtime to achieve high accuracy as per the data set to be inferred, as elaborated in Section 4.5.4. Finally, the outputs of the last visible layer are fed to the PIR [77] to achieve the digital outputs to be interfaced with other embedded digital system for further processing. Within this paper we describe the design and tradeoffs using various approaches to embed these processing steps within the memory element. We also evaluate its performance for real world applications of handwritten digit recognition for the MNIST dataset.

4.4 Hybrid Spin-CMOS Synapse Design

One way to realize machine learning at the edge of the network is to apply a Short-Term Memory-Long Term memory (STM-LTM) approach. A crossbar-based synapse interconnect can be efficient, as delimited in [81], [82]. Alternative mechanisms can be exploited through a variety of hybrid configurations of device technologies; for instance, capacitive synapses can be utilized in lieu of resistive coupling due to their extremely low static power dissipation [83]. A capacitive neural network that performs VMM operation using a charge-based capacitor crossbar has been proposed in [84]. By utilizing capacitive coupling and voltage division, these designs accomplish the weighted summation of inputs in order to produce an output in a read-like operation executed by memory devices.

4.4.1 Memory Unit Design - Capacitor as STM

In recent times, numerous research works have investigated the potential of memories based on capacitors for neural network applications [83], [84]. Achieving precise training of neural networks necessitates the implementation of successive minor weight modifications, which render NVMs non-ideal in this regard owing to their constrained speed and endurance. In contrast, DRAM presents an appropriate mechanism for online (in-situ) training on account of its relatively high-speed and symmetrical read/write capabilities with significantly high level of endurance. This attribute is particularly crucial for networks, such as IoT edge devices, that require continuous training over an extended duration [85]. Achieving a parallel computation with a low bit width, without requiring any ADC/DAC peripherals as in Re-RAM-based accelerators [86], is possible with digital capacitor-based accelerators [87], in which each memory BL is capable of performing bitwise digital Boolean logic operations and each capacitor stores a binary synaptic weight.

Based on the biologically inspired STM-LTM characteristics, Shiekhfaal et al. [81] implement a capacitive crossbar augmented with an NVM in a novel fashion. The capacitor of each memory bit-cell represents a binary synaptic weight ('1' or '0') in the form of a 'charged' or 'discharged' capacitor state. WL controls the access transistor of the STM T1 in Fig. 14(c)], which permits selective write/read operations on the cells contained within a row. Two critical duties that must be executed are the storage of the network weights in the STM via a write operation and the reinforcement of the memory through an STM-to-LTM transfer.

In both operations, the capacitor is in the pre-charged state (P.S.), whereby the voltage driver sets the BL voltage to (V_{DD}/2). The memory decoder needs to first activate the corresponding WL and set the BL to high (V_{DD}) or low (GND) to save weight on a capacitor. This will provide enough bias voltage to change the capacitor data in a DRAM fashion. STM-to-LTM transfer, or computation will subsequently be performed utilizing the synaptic weight representing STM.

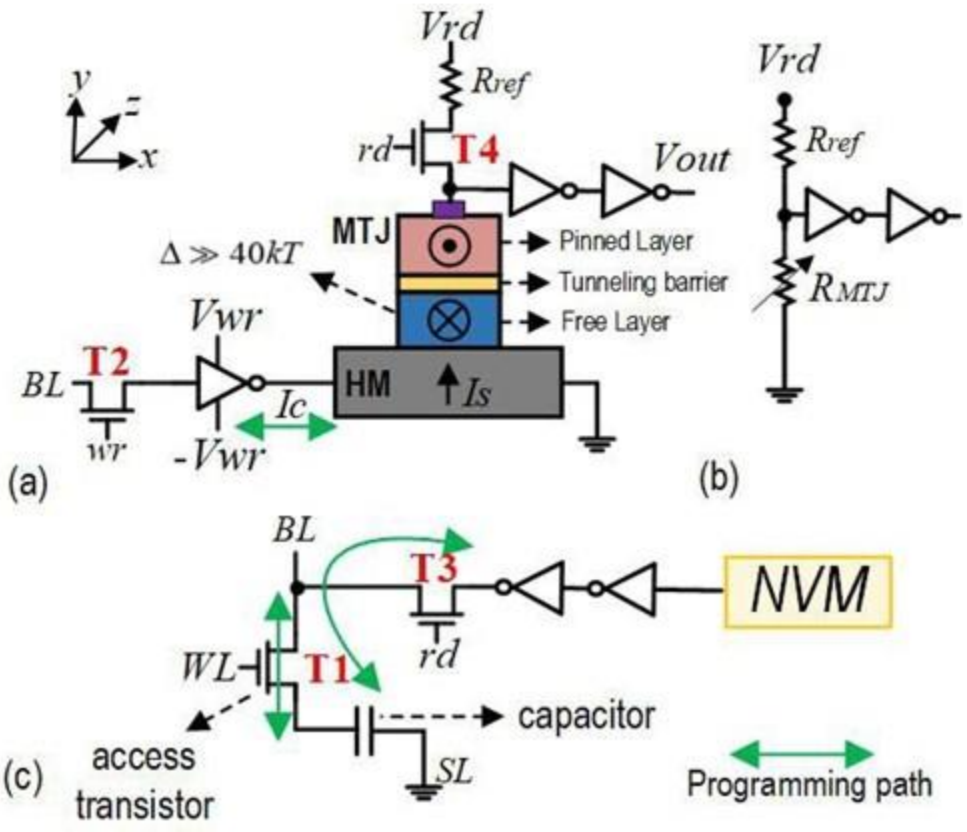


Figure 14: (a) Device Structure. (b) Read Circuitry, and (c) the Programming Path [1], [81], [82]

4.4.2 Memory Unit Design - SHE-MTJ as LTM

As depicted in Fig. 14 (a), the NVM component of the STM-LTM memory architecture is a SHE-MTJ that employs a stable nanomagnet and two CMOS inverters to amplify the output. A charge current (I_c) is introduced into the heavy-metal layer in the *+x(-x)*-direction, as elaborated in Chapter 2.1.3, in order to manipulate the free-layer magnetization and store the data in the SHE-MTJ. Fig. 14 (b) depicts the read circuit of a SHE-MTJ. To read data from the SHE-MTJ, a read voltage is applied to sense the resistance of the device. This can be achieved by implementing a resistive voltage divider. This study examines the utilization of three access transistors, as depicted in [81], for controlling the SHE-MTJ in relation to the volatile element

Transistors *T3* and *T4* activate the read path while transistor *T2* controls NVM and VM data transfer.

4.5 Hybrid Spin-Analog Neuron Design

4.5.1 Previous Neuron Designs

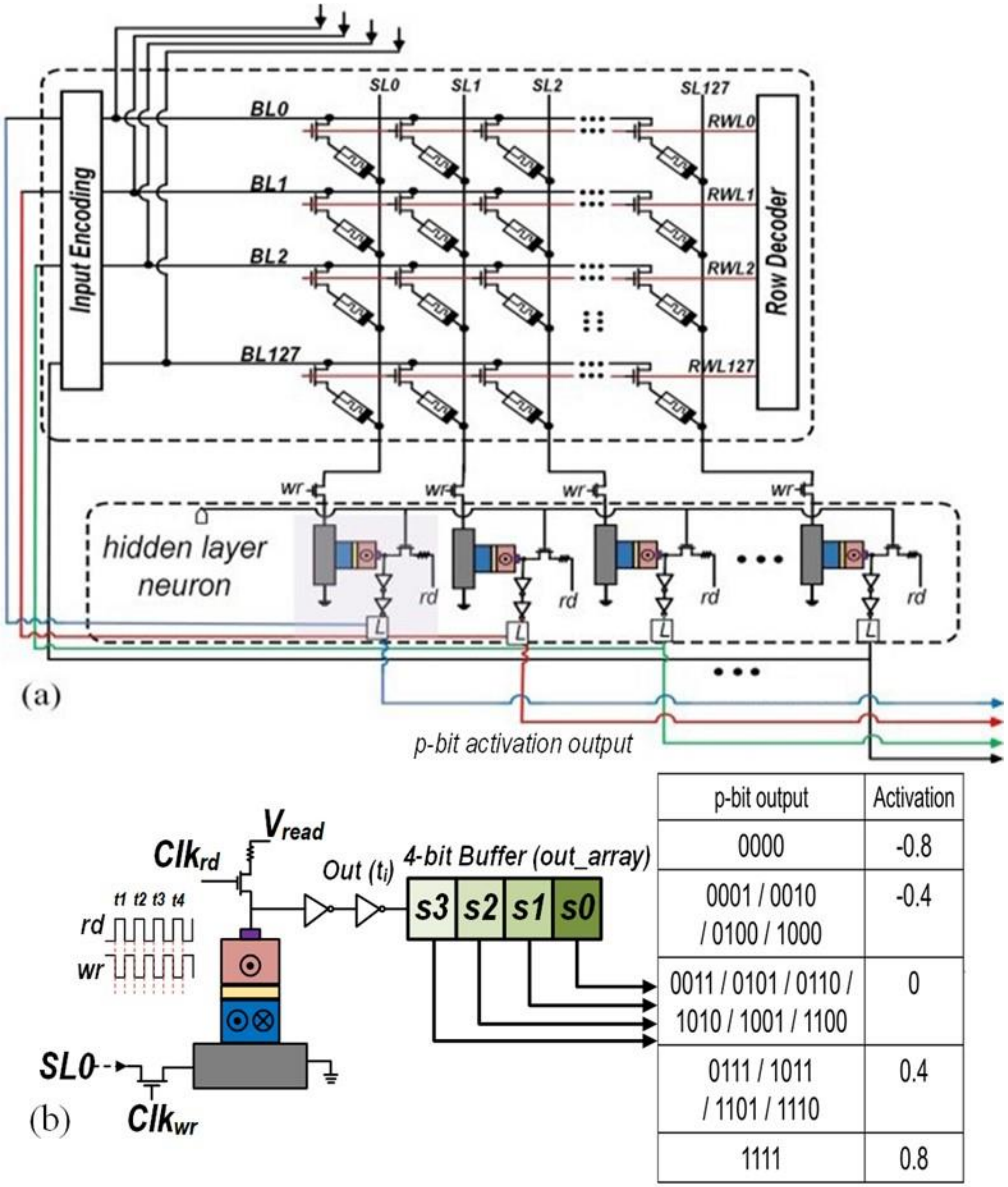
Prior research on Re-RAM crossbar-based PiM demonstrated that CMOS-based neuron implementations necessitated large built-in truth tables with added clock cycles, which resulted in increased area and energy utilization [88], and [89]. Recently efficient hardware implementations of brain inspired neurons utilizing emerging NVM devices is being widely explored, to implement VMM operations via the intrinsic weighted summation capability of crossbar designs based on PiM architecture. The SHE-MTJ device shown in Fig. 14 (a) is considered to be low-barrier under the condition energy barrier $\mathrm{E_B} \ll 40kT$, in which case thermal fluctuations at room temperature are sufficient to change the state of the device.

4.5.2 Binary and Non-Binary Neurons

The Long Short-Term Memory (LSTM) networks requires sigmoid and tanh-based neurons for multiple gating purposes. Fig. 14 (a) shows circuit implementation of a sigmoidal behavior achieved by connecting an inverter to V_{DD} and GND, provided the SHE-MTJ used in the circuit has $E_B << 40kT$. The device's time-averaged output has the capability to exhibit *sigmoid* and *tanh* function behaviors by means of marginally distinct circuit designs [1], [79], and [82]. The voltage values from the output are stored and mapped to a low-overhead LUT. The hardware implementation of p-bit based stochastic neuron has been improved as delineated in

[79] by adding two components, as shown in Fig. 15 (b), along with a NN implementation shown in Fig. 15 (a).

To latch the output, a 4-bit buffer is inserted first corresponding to the four times of applying the crossbar output. Second, the neuron output is formed using LUT. Two complementary signals, *wr* and *rd*, are taken into account, as illustrated in Fig. 15. The *wr* signal goes high for each sample and the p-bit device is programmed in accordance with the crossbar output current. To generate the output bit and read the device resistance, the *wr* and *rd* signals go low and high, respectively.



p-bit output	Activation
0000	-0.8
0001 / 0010 / 0100 / 1000	-0.4
0011 / 0101 / 0110 / 1010 / 1001 / 1100	0
0111 / 1011 / 1101 / 1110	0.4
1111	0.8

Figure 15: The LSTM Network with Non-Binary Neurons [1], [79], and [82]

The converter LUT, which is pre-loaded with the sampled floating-point activation values that correspond to the output combinations in the buffer, is then provided with the 4-bit buffered data. For instance, the LUT selects 0 as the output when the buffer content is 0011. This value can be triggered by any of 0101/0110/1010/1100 output bitstreams. This type of non-binary neuron design is practical for numerous ANN applications that require deterministic and non-linear tanh and sigmoid activation functions.

4.5.3 Configurable Analog Multiplier for Generalizable Activation Function

The reconfigurable analog multiplier in [78], [90] is based on the op-amp design presented in Fig. 18 (a). The op-amp consists of two cascaded stages: an input stage consisting of a differential amplifier, followed by a gain stage. A simple op-amp design consisting of only 10 CMOS transistors as shown in Fig. 18 (b) is chosen to optimize power consumption as well as area and simulated using models from the PTM 14nm LSTP library, at V_{DD} = 0.8V. The trans-linear principle is applied to attain exponentiation of the input signal [91]. As shown in Fig. 18 (b), a three-stage design has been proposed whose output is a power function of the input. The design accepts a single input for performing exponentiation operations; the design can also be reconfigured to accept two inputs for performing analog multiplication. The first stage, outlined in red in Fig. 18 (b), is a logarithmic amplifier:

$$V_1 = -A_{OL}V_0 \tag{4.1}$$

$$-\frac{V_0 - V_{in}}{R_1} = I_{S1}\left[\exp\left(\frac{V_0 - V_{in}}{V_T}\right) - 1\right] \tag{4.2}$$

where A_{OL} is open loop gain and $I_{S1}$is the saturation current of diode D_1.

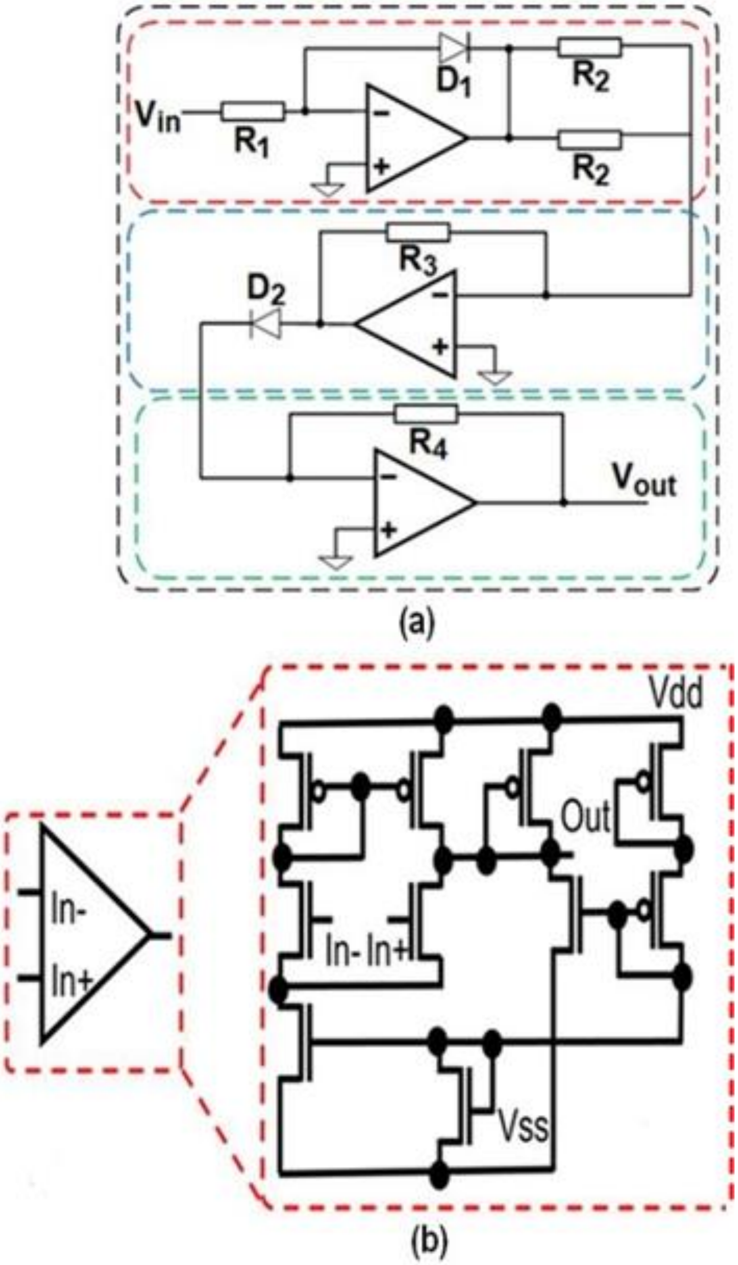


Figure 16: (a) Three Stage Analog Multiplier, and (b) Internal Structure of an Op-amp Implementation [1], [78], and [90].

Solving the systems of equations consisting of Eq. 4.1 and Eq. 4.2 simultaneously yields,

$$V_1\left(1+\frac{1}{A_{OL}}\right) = -V_T\left(\frac{V_{in}+\frac{V_1}{A_{OL}}}{R_1 I_{S1}}+1\right) \tag{4.3}$$

Considering an infinite open loop gain and a large input voltage, Eq. 4.3 can be approximated as,

$$V_1 = -V_T \ln\left(\frac{V_{in}}{R_1 I_{S1}}\right) \tag{4.4}$$

The second stage is an analog adder, whereby a similar analysis yield $V_2 = \frac{2V_1 R_3}{R_2}$. The third and final stage is an anti-log amplifier whose output is roughly defined as follows:

$$V_{out} = -R_4 I_{S2} e^{\frac{V_2}{V_T}} \tag{4.5}$$

where I_{S1} represents diode's (D_2) saturation current. Overall, the output of this circuit is given by the following equation:

$$V_{out} = -e^{\frac{V_2}{V_T}} \frac{R_4 I_{S2}}{(R_1 I_{S1})^a} (V_{in})^a \tag{4.6}$$

where $a = 2\frac{R_3}{R_2}$, realizing any positive power function of the input as depicted in Fig. 18(b). Additionally, to obtain an analog multiplier, a dual-input stage comprised of two logarithmic amplifiers may be inserted. Ultimately, inverse power functions can be achieved by inserting an inverting amplifier between the second and third stages [78], [90].

4.5.4 Proposed Selectively Reconfigurable Activation Function Neuron Functionality and Design

As mentioned, the *sigmoid* and *tanh* activation functions are the most employed activation functions for inferencing tasks on neural networks. In this dissertation, the research goes beyond previous work by realizing hardware for more expressive activation functions, which can be runtime configured within the memory to achieve different variations in activation functions as per the target dataset/application. Fig. 19 demonstrates the hardware implementation of the proposed GAAF, based on the op-amp design presented in Fig. 18. Diverse exponential functions may be generated using the analog multiplier by manipulating $a = 2\,R_3/R_2$, as described in reference [78], [90]. Therefore, it is possible to generate more expressive activation functions by modifying the resistances of R2, R3, or both. Parallel (P) and anti-parallel (AP) magnetization states (R_A, R_{AP}), which are determined by intrinsic device parameters, enable SHE-MTJs to provide configurable variable resistances at runtime. In this study, the hardware implementation (as illustrated in Fig. 19) replaces only R3 in the feedback path of the operational amplifier. This modification provides a control mechanism that demonstrates the generation of different activation functions. For the MNIST dataset, error rates, performance metrics, and the effects of process variation of GAAF are evaluated on the following network sizes: 784x200x10 and 784x500x10. The input V_{in} to the GAAF is a *sigmoid* function output of the device shown in Fig. 15 (b) and elaborated in Section 4.6.1.

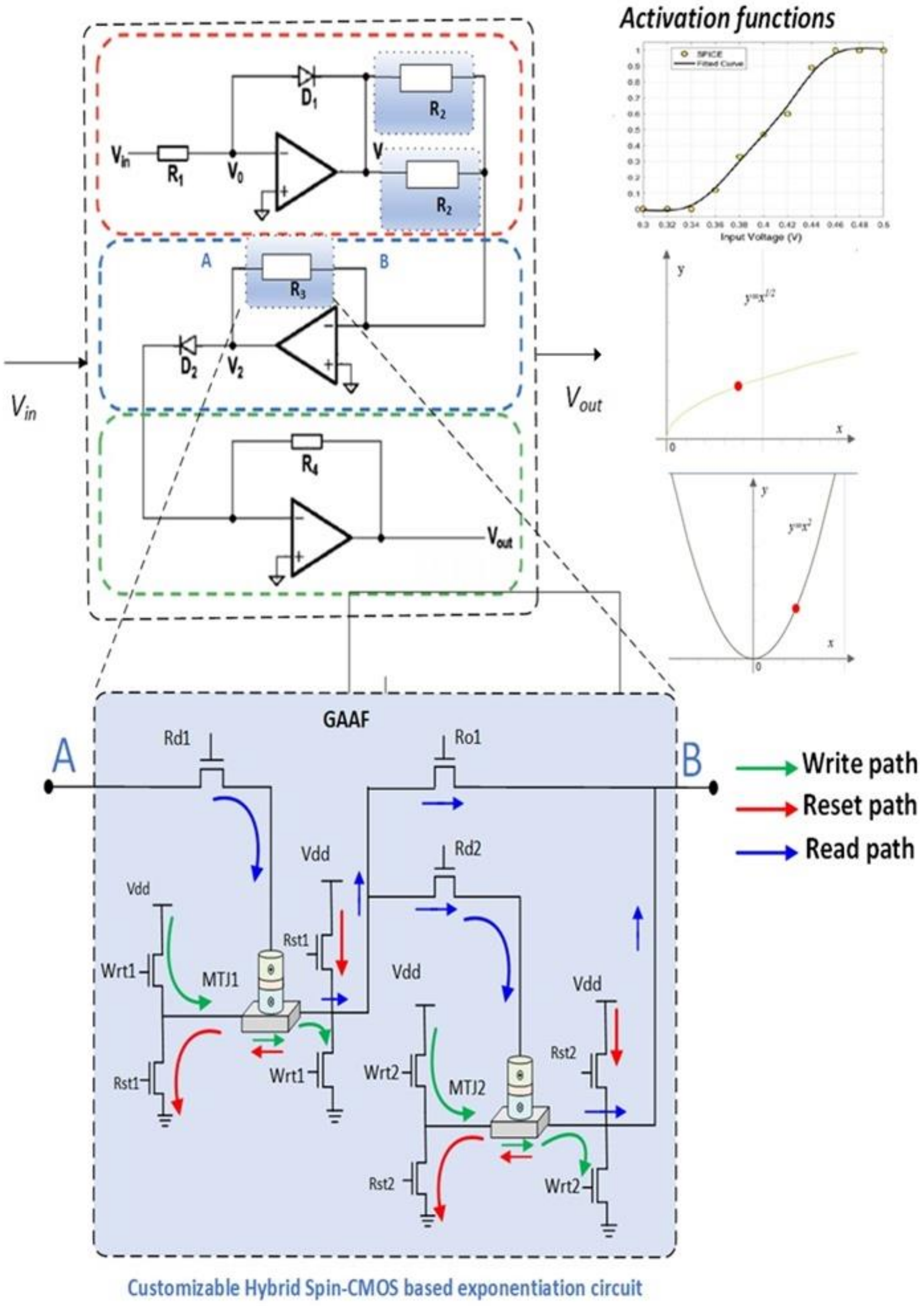


Figure 17: Hybrid Spin/CMOS Device-based GAAF neuron structure [1]

4.5.5 Application Mapping and Execution Model

In this dissertation, a series connection of two SHE-MTJs in the feedback path of the op-amp has been implemented to analyze the feasibility of the proposed design approach and the different activations achievable. Identical SHE-MTJs having design parameters listed in Table 8 are employed. The *P* and *AP* resistance values obtained via SPICE simulations show the R_P and R_{AP} resistances of 2.8 KΩ and 5.6 KΩ, respectively.

Table 8: SHE-MTJ Simulation Parameters [1], [90]

Symbol	***Parameter***	***Value***
R_P/R_{AP}	P/AP MTJ Resistance	2.8 KΩ/5.6 KΩ
TMR	Tunnel Magnetic Ratio	100%
α	Damping Coefficient	0.007
t_f	Free layer thickness	1.3nm
T	Temperature	300K
P	Spin Polarization	0.52
V_{t_p}/V_{t_n}	P/NMOS Threshold	0.46 V/0.50 V
W_p/W_n	P/NMOS Width	44nm/22nm
θ_{she}	Spin Hall Angle	0.4
ρ_{hm}	Resistivity of HM layer	0.2 mΩ.cm
MTJ Area	MTJ Length × MTJ Width × $\pi/4$	60nm×30nm×$\pi/4$
HM Volume	L × W × T	100 nm×60 nm×3 nm

KΩ = kilo-ohm, K = Kelvin, mV = milli-volt, nm=nanometer.

Table 9 lists the control signals re-quired for configuration of the two SHE-MTJs during write phase, i.e., MTJ1 and MTJ2 in Fig. 19 in P-OFF, AP-OFF, P-P, P-AP, AP-AP states, respectively, where *P* is the parallel, *AP* anti-parallel, and OFF is the turned off state of MTJs, V_{DD}=0.8 V. Since, in this phase the MTJs are being written their resistances, hence all the read signals are set to low (GND).

Table 9: GAAF Configuration Phase Control Logic [1]

Switching transitions		***Control Signals***						
MTJ1	**MTJ2**	***Rd1***	***Rd2***	***Ro1***	***Wrt1***	***Rst1***	***Wrt2***	***Rst2***
P → AP	OFF	0	0	0	V_{DD}	0	0	0
AP → P	OFF	0	0	0	0	V_{DD}	0	0
P → AP	P → AP	0	0	0	V_{DD}	0	V_{DD}	0
AP → P	P → AP	0	0	0	0	V_{DD}	V_{DD}	0
P → AP	AP → P	0	0	0	V_{DD}	0	0	V_{DD}
AP → P	AP → P	0	0	0	0	V_{DD}	0	V_{DD}

Table 10 lists the corresponding resultant resistance values and activation functions generated from the GAAF unit upon reading the MTJs with a read voltage of 0.8V, and all the write and reset signals are set to low in this phase. Initially, MTJ1 is configured in parallel magnetization state and MTJ2 cutoff from the circuit by *Ro1* signal set to V_{DD} via the pass transistor. In this case, the equivalent MTJ resistance evaluates to 2.8 KΩ and the output of GAAF evaluates the sigmoidal square root activation function. To switch the device to *AP* state, *Wrt1* is set to V_{DD}=0.8 V, and read signal *Rd1*, reset signal *Rs1* are kept low, such that write current passes along the heavy metal layer and the free layer magnetization switches to *AP* state. In this stage, with MTJ1 in *AP* state and MTJ2 OFF, the resultant equivalent MTJ series resistance evaluates to 5.8 KΩ, and inverted sigmoidal activation is evaluated by the GAAF. In a similar fashion, sigmoidal power of 3/2 activation function can also be produced by the GAAF unit, by suitably setting the control signals to their corresponding values in Table 9. A control unit takes care of the timing and setting of different control signals to appropriate voltages. Fig. 20 shows the corresponding timing diagram of the various control signals and corresponding switching behavior of the two SHE-MTJs, evaluated on SPICE [1], [90].

Table 10: GAAF Evaluation/Read Phase Operation and Control Logic [1]

Resistance		***Total Series***	***Control Signals***							***Activation***
MTJ1	**MTJ2**	***Resistance***	***Rd1***	***Rd2***	***Ro1***	***Wrt1***	***Rst1***	***Wrt2***	***Rst2***	***function***
P	OFF	R_{P1}=2.8KΩ	V_{DD}	0	V_{DD}	0	0	0	0	*Sig. Sq. root* $\surd V_{in}$
AP	OFF	R_{AP1}=5.6KΩ	V_{DD}	0	V_{DD}	0	0	0	0	*Inv.Sig. -* V_{in}
P	P	R_{P1}+ R_{P2}=5.6KΩ	V_{DD}	V_{DD}	0	0	0	0	0	*Inv.Sig. -* V_{in}
AP	P	R_{AP1}+ R_{P2}=8.4KΩ	V_{DD}	V_{DD}	0	0	0	0	0	*Sig. Pow(3/2)* $V_{in}^{(3/2)}$
P	AP	R_{P1}+ R_{AP2}=8.4KΩ	V_{DD}	V_{DD}	0	0	0	0	0	*Sig. Pow(3/2)* $V_{in}^{(3/2)}$
AP	AP	R_{AP1}+ R_{AP2}=11.2kΩ	V_{DD}	V_{DD}	0	0	0	0	0	*Sig. Sq.* V_{in}^2

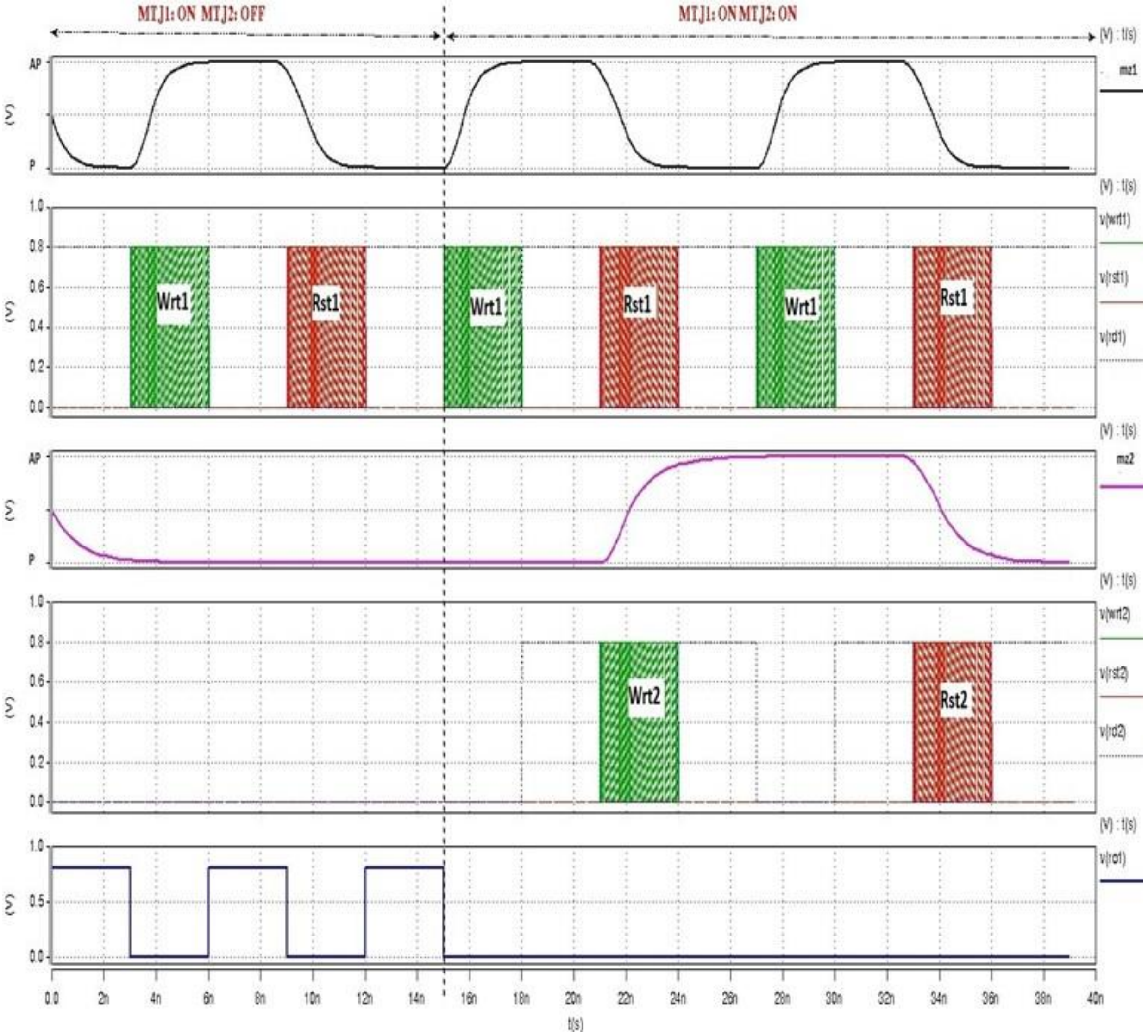


Figure 18: Control Signal Mapping for GAAF Configuration and Evaluation Stages [1]

4.5.6 High-Level Exploration of Additional Software Support for SCAPE Utilization

For software applications to utilize the SCAPE architecture, the execution mechanism needs additional software support. This is done congruent with the concept of Gather/ Scatter techniques as illustrated in [92]. Although the premise of [92] and our work is distinct, the concept is expanded to support our architecture in the scenario of activation and access/write to multiple target cells located in a crossbar memory layout. This requires additional circuitry including modification to the control logic and memory decoder structure. Communication between the CPU and SCAPE is established via a 64-bit data bus and an address bus serving each crossbar layer. The proposed approach to utilize SCAPE capabilities is via additions to the ISA including a `SET` operation for writing data to the array, `SCATTER` for activating word lines, and `GATHER` for reading output data. Besides the dynamic activation of multiple word lines for synaptic weight calculation as exhibited in [81], SCAPE provides infield configurability of Hybrid Spin/ CMOS-based Analog/Digital Blocks to enable hardware for more expressive neural network activation functions. Thus, the GAAF units can be runtime configured within the memory array to achieve various activation functions as per the target dataset/application. A generalized activation function is developed in the manuscript, which is shown to achieve better recognition rate for MNIST dataset. The activation of target GAAF neurons is achievable by introducing two new instructions into the ISA, i.e., `ACTIVATE` and `EVALUATE`. The following is an overview of the ISA modifications required for functionality of SCAPE:

1: `SET (REGID, addr)` which is used to write the data from CPU register specified by `REGID` to a specific SCAPE memory cell specified by `addr`. In this context, `addr` can be broken down to {`layerID`, `rowID`, `columnID`} to identify a specific crossbar memory cell. `SET` is used to load matrix data, and input data, into SCAPE; a `columnID` of 0 is used to denote input vector data.

2: `SCATTER (REGID, layerID, WL1, WL2)` which is used to set all the word lines between `WL1` and `WL2` in a specified layer of SCAPE, using the configuration data initially stored in `REGID`. This is achieved at the hardware level through a latch/reset mechanism like that described in [93].

3: `GATHER (REGID, layerID, BL1, BL2)` which is used to load output data from a range of bit lines in a specific layer of SCAPE into the CPU register labeled `REGID`.

4: `ACTIVATE (layerID, configID)` that configures the GAAF units by setting internal MTJ values to their required *P* or *AP* or OFF (disconnected from circuit) orientations based on the desired neuron activation functions. The parameter `layerID` identifies the GAAF enhanced neuron layer in the SCAPE to be activated. The `configID` in SCAPE is a 3-bit identifier corresponding to each of the six unique combinations of MTJ1 and MTJ2 resistance states in the GAAF neuron, as listed in Table 10, which achieves a specific activation function, by generating the corresponding control signals through the control logic circuitry. For instance, in order to generate an inverted sigmoidal activation function, MTJ1 and MTJ2 are configured to be in *AP* and *OFF* states in the circuit, respectively. As such, a `configID` of ‘000’ generates the required control signal values as listed in row one of Table 8, to set the MTJs to their required states.

5: `EVALUATE (layerID, funcID)` that generates the desired neuron activation function at the GAAF output. The `funcID` denotes the type of activation function that we want the GAAF neuron to output. The `funcID` is encoded as a 2-bit identifier generating the control signals corresponding to evaluating one of the four unique functions: inverted sigmoid, $(\text{sigmoid})^{2}$, $(\text{sigmoid})^{3/2}$, and $(\text{sigmoid})^{1/2}$ at the GAAF output, as listed in Table 10.

4.5.7 Intrinsic VMM on SCAPE: CS Applications

By employing Compressive Sensing (CS) to sample at the information rate as opposed to the Nyquist rate, transmission and storage overheads for spectrally sparse and wideband data can be reduced [78], [90]. Thus, CS offers a resolution to the unparalleled difficulties linked to 5G communication, such as the intricacy and power consumption associated with expanded bandwidths, by limiting the quantity of samples taken per frame. However, hardware implementation of CS sampling and reconstruction poses unique challenges and is not straightforward. A random number generator is utilized to carry out the sampling operation. Conventionally, this is accomplished through the implementation of a Linear Feedback Shift Register (LFSR), which may entail substantial power and area overheads. In their study, Qian et al. [94] proposed memristive crossbar arrays to support VMM operations during CS sampling. It was observed that the Signal to Noise Ratios (SNRs) obtained from signal reconstruction using $\ell 1$-minimization were comparable to those obtained from employing a Gaussian matrix. In CS, a signal of length n is sampled using m measurements, where $m \ll n$. Sampling is achieved through the linear transformation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^{n}$ is the signal vector, $\boldsymbol{y} \in \mathbb{R}^{m}$ is the measurement vector, and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the measurement matrix. At the receiving end, the signal gets reconstructed by solving the basis pursuit problem:

$$\hat{\boldsymbol{x}} = \operatorname{argmin}\|\boldsymbol{x}\|_1 \text{ s.t. } \boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{x}} \tag{4.7}$$

where $\|\boldsymbol{x}\|_1$ represents the ℓ_1 norm of $\boldsymbol{x}$. It can be shown that the signal vector can be reconstructed if the signal is sufficiently sparse, and $\boldsymbol{A}$ satisfies the Restricted Isometry Property, i.e., if for any k-sparse vector $\boldsymbol{x}$,

$$\|\boldsymbol{x}\|_2^2(1-\delta) \leq \|\boldsymbol{\Phi x}\|_2^2 \leq \|\boldsymbol{x}\|_2^2(1+\delta), \quad 0 < \delta < 1 \tag{4.8}$$

As alternatives to basis pursuit, numerous algorithms have been devised to facilitate CS reconstruction. For instance, *Approximate Message Passing (AMP)*, as shown in Algorithm 1, serves as a soft thresholding algorithm optimized for fast convergence [95].

Table 11: Approximate Message Passing Algorithm [1], [90]

```
Algorithm 1 Approximate Message Passing
Inputs: Measurement matrix, $\boldsymbol{\varphi}$, Measurement vector $\mathbf{y}$, # of measurements $\boldsymbol{m}$
Outputs: Approximate signal vector, $\hat{\boldsymbol{x}}$
Procedure: 1) Initialize residual $\boldsymbol{r_0} = \boldsymbol{y}$, Signal approximation $\hat{\boldsymbol{x}} = \mathbf{0}$, counter $\boldsymbol{i} = \mathbf{1}$
           while $\boldsymbol{i} < \boldsymbol{k}$ do
           2) $\theta = ||r_{i-1}||/\sqrt{m}$
           3) $a = \hat{\boldsymbol{x}}_{i-1} + \varphi^T r_{i-1}$
           4) $\hat{\boldsymbol{x}}_i = sign(a)\max(|a| - \theta, 0)$
           5) $b_i = \frac{||\hat{x}_i||_0}{m}$
           6) $\mathbf{r}_i - y - \varphi\hat{\boldsymbol{x}}_i + b_i r_{i-1}$
           end while
```

In Line 1, the AMP algorithm initializes the residual vector, $\boldsymbol{r_0}$, to the measurement vector y, as well as initializing the estimate of the signal vector $\hat{\boldsymbol{x}}$ to zero. Line 2 computes the threshold, *q*, as the root-mean-square error of the residual. Next, Lines 3 – 4 follow the Iterative

Soft Thresholding technique [96] to generate an estimate of the reconstructed signal vector. The notation in Line 4 refers to elementwise operations on the components of vector *a*, with the function *sign(x)* defined as -1 when x < 0 and as 1 when x > 0. Finally, Lines 5 – 6 update the residual, $\|\hat{\boldsymbol{x}}_{\boldsymbol{i}}\|_0$, based on the current estimate of the signal as well as the residual of the previous iteration, $\boldsymbol{r}_{\boldsymbol{i-1}}$.

The AMP algorithm is implemented using the SCAPE hardware architecture presented in Fig. 13. AMP requires vector-matrix multiplication operations, which are executed using the VMMS. Furthermore, a three-stage analog circuit based on the design shown in Fig. 18 is used for basic arithmetic operations, including multiplication (by use of a dual first-input stage), addition (using the second computational stage) and exponentiation operations such as square, square root and inverse square root. Besides the operations listed above, AMP requires thresholding operations which are also achievable with the AAS using the simple analog design in [90].

The function *y* = sign(*x*) is computed by an analog comparator circuit when $V_{ref} = 0$. Two additional functions are computed using a three-stage design based on a chain of inverters: *y=sign1(x, ref)*, which is defined as 1 when *x* is less than *ref* and 0 when *x* is greater than *ref*, and *y=sign2(x, ref)*, which is defined as 1 when *x* is greater than *ref* and 0 when *x* is less than *ref*. The computation of the three remaining functions required for AMP is feasible with this hardware. First, *y*=|*x*| is rewritten as *y = xsign(x)*. Next, *y = max(x,0)* is equivalent to *y = xsign2(x,0)*. Finally, *y*=$\|\boldsymbol{x}\|_0$ is roughly equivalent to $y = \sum\big(sign1(x, 0.05) + sign2(x, -0.05)\big)$, assuming any input with an absolute value greater than 0.05 is considered as “non-zero.”

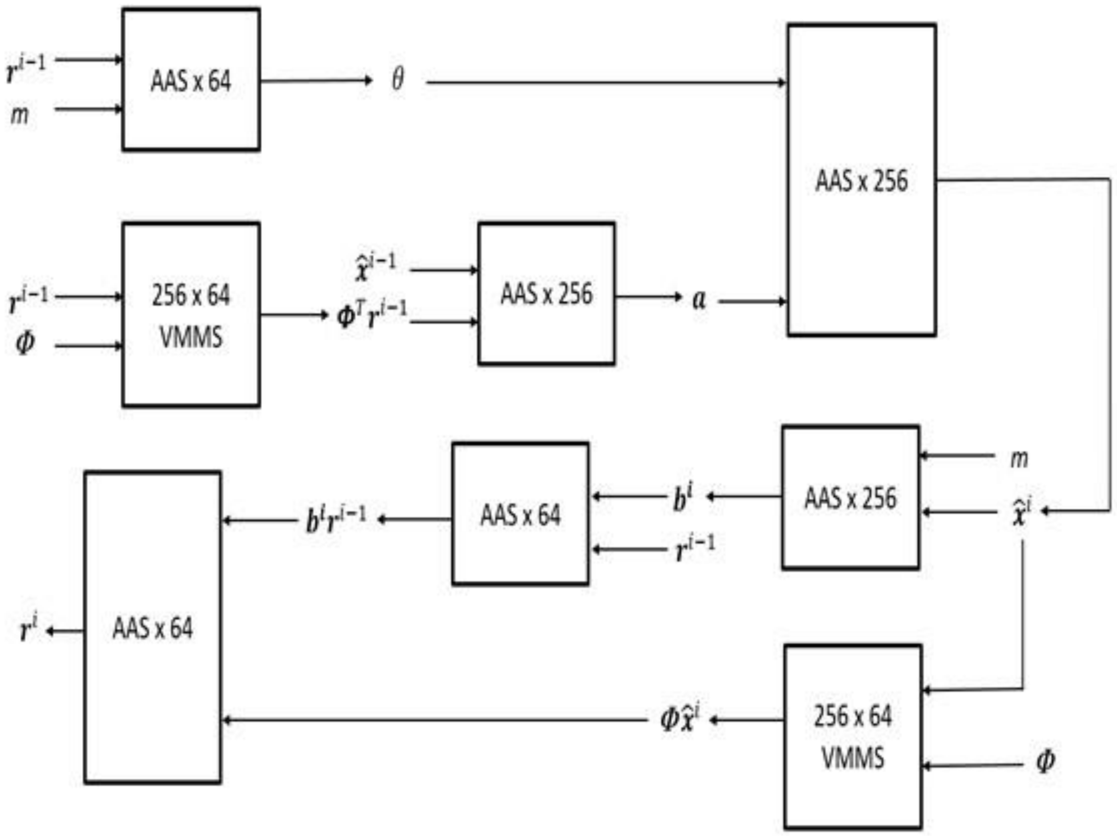


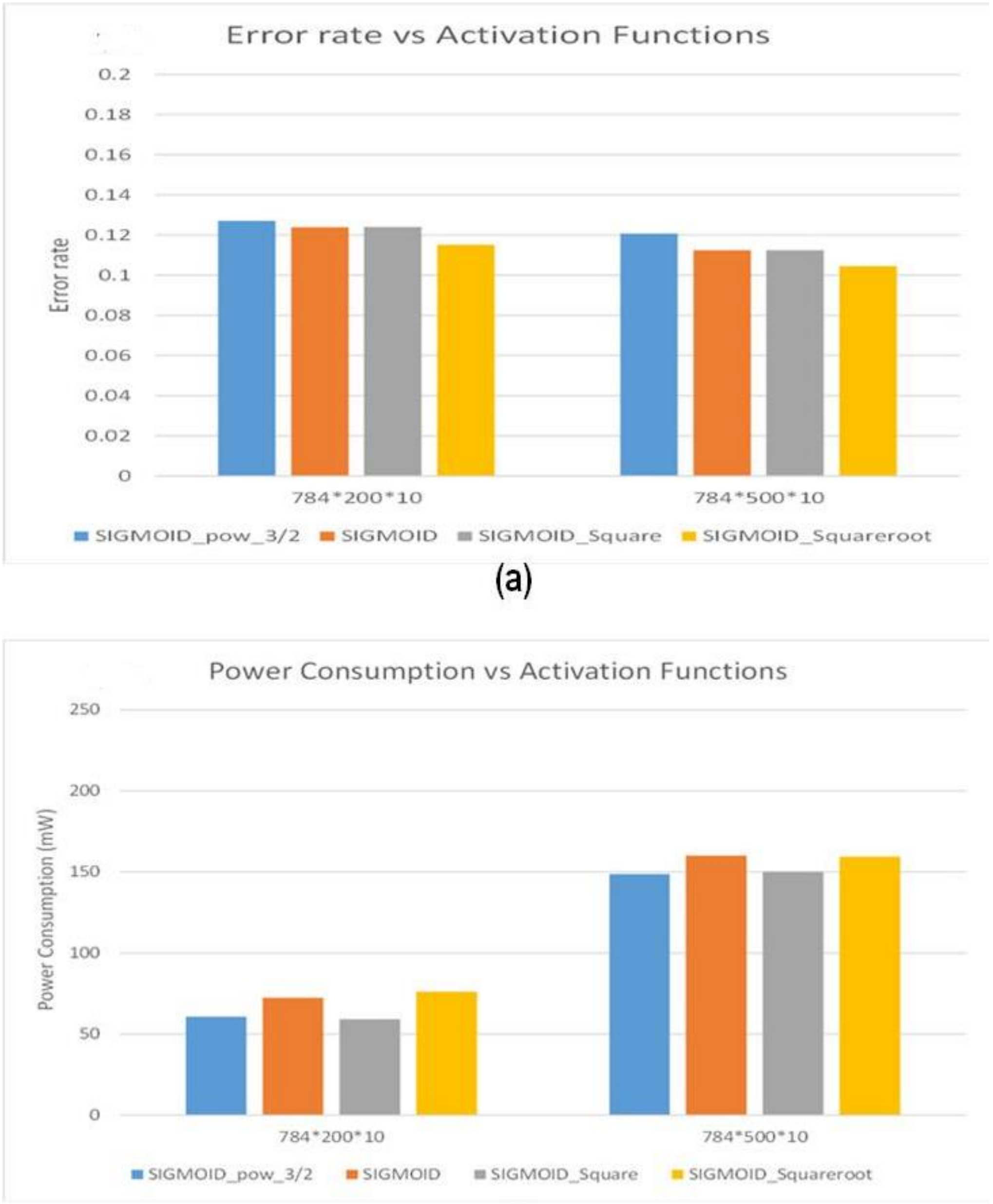
Figure 19: Hardware Implementation of AMP Algorithm [1], [90]

Fig. 19 demonstrates a hardware implementation of one loop of the AMP algorithm. To perform VMM operations in Line 4 and Line 6, reconstruction using a signal size of $n = 256$ and $m = 64$ necessitates a 256 × 64 VMMS array. Additionally, 256 AAS functional units are required for scalar operations.

4.6 Results and Analysis

4.6.1 Benchmark Validation on MNIST Dataset for ML

For evaluating our SCAPE topology, MNIST data set containing 70,000 images has been utilized, out of which 3,000 images are employed for training the ANN. The trained weights and biases obtained for the network are accordingly assigned to the crossbar array, and testing is done for the hardware network using the 100 test images and PIN-Sim framework [43]. Fig. 20 shows the error rate obtained at the final layer of the SCAPE topology in Fig. 13, and the overall power consumption of the ANN for the four activation functions namely, $(\text{sigmoid})^2$, sigmoid, $(\text{sigmoid})^{3/2}$ and $(\text{sigmoid})^{1/2}$. Fig. 20 (a) shows that accuracy achieved by the sigmoidal square root activation function is best with lowest error rate, sigmoidal power (3/2) performs worst, whereas baseline sigmoidal and square achieve similar error rates for all the topologies for MNIST dataset evaluated using PIN-Sim [43]. Fig. 20 (b) shows that the overall power consumption for the sigmoid square root activation is comparable to the power consumption of plain sigmoidal activation function. Switching from one activation function to another is achieved by GAAF configuration as mentioned previously. Appropriate control signals are given to the block so that the MTJ's switch between *P* and *AP* states to get the desired activation function. Table 12 represents the comparison of GAAF performance for different activation functions with other digital/analog activation function generators. It can be observed that the number of components used in GAAF block is less with comparable power consumption and delay, as with other circuits in literature.



(a)

(b)

Figure 20: (a) Error Rates, and (b) Overall Power (mW) Consumption of four different GAAF Activation functions used for inference of MNIST dataset on 2 ANNs (784x200x10; 784x500x10) [1]

Table 12: Performance Comparison of GAAF [1], [90]

	[91]	**[97]**	**[98]**	**[99]**	**Herein**	**Herein**
Mode	Analog	Digital	Digital	Analog	Analog	Analog
Operation	Square	Multiplier	Square root	Square	Square root	Square
Tech node	180nm	28nm	45nm	500nm	14nm	14nm
V_{DD}	1.3V	1V	1V	1.5V	0.8V	0.8V
#Components	100	~1000	>1000	12	55+2 SHE-MTJs	55+2 SHE-MTJs
Power	149mW	126mW	21.02mW	600mW	**121mW**	**126mW**
Delay	N/A	0.8ns	3.61ns	N/A	**6.4ns**	**3.5ns**

Table 13 lists the error rate, average DBN power consumption, and power-error-product of proposed SCAPE topology for various sized ANNs, and activation functions evaluated on MNIST dataset. The Power Error Product (PEP) metric is also calculated as a product of power consumption and error rate to better establish the error efficiency of the SCAPE topology compared to plain sigmoidal activation function. PEP for sigmoidal square root activation function for 784x200x10 topology was observed to be the lowest i.e., most efficient. For datasets larger than MNIST, SCAPE limits accuracy loss and accumulated current associated with larger arrays by matrix partitioning using a similar method described in [100].

Table 13: Power Error Product of Sigmoid Activation vs. SCAPE Topology for Various Network Sizes [1]

Attributes	**Activation Function**			
	Sigmoid		**GAAF Enhanced** **Sigmoid + Square Root**	
ANN	784×200×10	784×500×10	784×200×10	784×500×10
Error rate	0.1239	0.1124	0.1152	0.1046
Power(mW)	72.4	160.1	76.1	159.5
PEP	8.97	18	8.77	16.68

4.6.2 Comparative Analysis of CS AMP Algorithm

SPICE simulations are conducted to estimate the overall computational energy expenditure for executing a single cycle of AMP. These simulations calculate the energy cost per operation of the scalar functions executed by the AAS and the per-cell energy cost of the VMMS, to determine the total energy cost of AMP. The VMMS consumes *3.15 nJ* in total energy, while the AAS consumes *2.02 nJ*, for a cumulative computational energy consumption of *5.17 nJ*. This indicates an energy overhead of *258.5 nJ* for operating AMP over the course of 50 iterations. An evaluation was conducted as in [1], [90] to assess the impact of approximations in the AAS units on signal reconstruction error. The analysis was conducted on a signal with dimensions n = 1000 and sparsity k = 100, where *n* denotes the total number of elements present in each signal frame and k signifies the total number of non-zero elements per frame. The average deterioration in accuracy due to computational error was determined to be a negligible *1.1dB*.

The energy consumed during the execution of a single AMP cycle is detailed in Table 14. An energy comparison between two recent ASIC implementations for AMP is presented in Table 15. Hardware executing the Enhanced AMP algorithm (EAMP) [101] over 50 iterations on a 400MHz system consumes *315 mW* of power and completes in *8900* clock cycles with the same CS parameters (n, m) = (256, 64). Therefore, the approximate energy consumption per sample is *27 nJ*. With 100 iterations, EAMP is comparable to the conventional AMP algorithm in terms of mean square error. This indicates that the full-analog approach to AMP offers significant energy saving while having a minimal impact on reconstruction accuracy.

Table 14: Breakdown of AMP Circuit Energy Consumption [1], [90]

Operation	**Hardware Units**	**Energy Cost**
$\|\|r^{i-1}\|\|.$	AAS	47.6 pJ
$q = \|\|r^{i-1}\|\|/\sqrt{m}$	AAS	1.1 pJ
$a = \hat{x}^{i-1} + \Phi^T r^{i-1}$	VMMS + AAS	1.654 nJ
$\hat{x}^i$ = sign(a**)** max(abs(a) – θ, 0)	AAS	1.24 nJ
$b^i = \|\|\hat{x}^i\|\|_0/m$	AAS	0.58 nJ
$r^i = y - \Phi\hat{x}^i + b^i r^{i-1}$	VMMS + AAS	1.65 nJ
Total		**5.17 nJ**

Table 15: Comparison of AMP Energy Consumption [1], [90]

	Herein	**Herein**	**[96]**	**[101]**
Tech. node	14nm	14nm	65nm	65nm
V_{DD}	0.8V	0.8V	1.2V	N/A
Array size	256x64	1024x512	1024x512	256x64
Array precision	8 bits	8 bits	26 bits	1 bit
#Iterations	50	20	20	50
Energy/sample	**1.0 nJ**	**2.1 nJ**	61 nJ	27 nJ

4.6.3 Process Variation (PV) Analysis

Two justified concerns facing analog computation are sensitivity to noise, and the ability to deliver sufficient accuracy in the computation. Approaches to mitigating variation and adapting operational tolerances span design margin, redundancy, and reconfiguration [102], and [103]. Device parameters such as Anisotropy field (H_k), Diameter (d) and Thickness (t) for the MTJ's may vary due to the process variation (PV) in MTJ fabrication, resulting in changes in R_P and R_{AP} resistance values. Inconsistencies in R_P and R_{AP} result in variations in activation function, thereby affecting the inference accuracy of the NN hardware. Fig. 21 depicts the deviation in square and square root activation functions due to PV in the GAAF MTJs, using 100-trial Monte Carlo (MC) simulation runs in SPICE with standard deviation (SD) of 5% for MTJ length, width, thickness, V_{in} represents the input to the GAAF and V_{out} represents the output obtained by using Eq. (4.6), where I_{s1}, I_{s2} are the diode saturation currents. R_1, R_2, R_4 are the resistance values in the multiplier circuit. R_3 (2.8 KΩ/5.6 KΩ/8.4 KΩ/11.2 KΩ) is decided by the state of MTJ's, thereby determining the neuron activation function in the network. A deviation of 5% in R_3 resistance value of 2.8 KΩ of the GAAF with a sigmoidal square root activation function was found to result in a maximum 5% increase in ANN inference error rates using PIN-Sim framework [43] on the MNIST dataset.

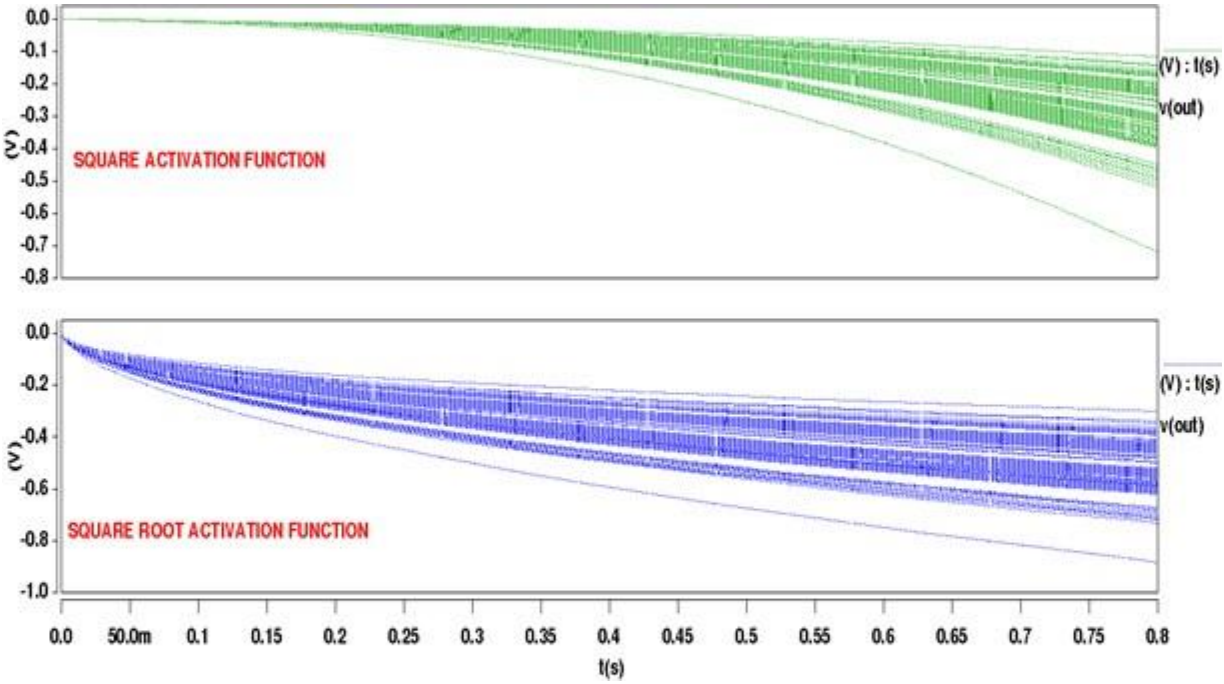


Figure 21: Effects of PV on Two GAAF Activation Functions Applying 5% SD on MTJ Length, Width, and Thickness [1]

4.7 Summary and Discussions

In this chapter, a 2D array-based approach to PiM by developing the SCAPE topology targeting efficient adaptive analog activation has been presented. Namely, an innovative GAAF based on spin-configurable activation function computes more expressive activation functions intrinsically in analog. Realization of AMP signal processing algorithm shows *~95%* reduction in energy consumption at comparable accuracy. Simulation results of power consumption and error rate for MNIST dataset using sigmoidal square root activation of GAAF shows up to *7%* accuracy improvement versus baseline conventional sigmoidal activation. This research has the potential to be expanded upon in future by adding enhanced functionality to GAAF and evaluating effects on more varied datasets for additional real-world applications.

CHAPTER 5: LOW ENERGY AND AREA FOOTPRINT ANN-BASED DIGIT RECOGNITION USING SPIN-BASED PROGRESSIVE MODULAR REDUNDANCY[5]

5.1 Context and Background

Redundancy and fault tolerant schemes have long been researched as effective approaches towards fault detection and error correction in various applications, in both synchronous and asynchronous digital design domain, thereby improving output accuracy. Among the approaches found in literature in the synchronous domain, [104] demonstrates that fault-tolerance of systems implemented with Triple Modular Redundancy (TMR) of designs based on multiple diverse logic designing techniques performs superior to TMR implemented with only minimum variations using the same logic designing technique, with negligible increase in runtime overhead. [105] proposes a heterogenous concurrent error detection hardware scheme for Discrete Cosine Transforms implemented on FPGA, but it does not analyze effects on any ANN-based use case. [106] designs a fault tolerance model, called triple modular redundancy with standby (TMRSB), applicable for FPGAs, where each TMR module has access to several independent standby configurations such that whenever a fault is detected, the physical resources within the faulty module are remapped utilizing the standby configurations to regain full functionality. [107] presents a modular adaptive redundancy technique to mitigate fault tolerance, by implementing a dual-tiered approach for fault handling, with an FPGA-based hardware step followed by evolutionary algorithm-based software

[5] ©IEEE. Part of this chapter is reprinted, with permission, from [3].

step. The system works to achieve dynamic partial self-reconfiguration of the system for fault recovery at significant power and repair time savings compared to a purely TMR based and full bitstream configuration-based approach, respectively. Some other approaches involve bitstream manipulation and dynamic redundancy leveraging partial dynamic reconfiguration on FPGA platforms [108], [109], [110], [111]. Various genetic evolution-based approaches for fault handling in synchronous domain have also been investigated [112], [113], [114]. In the asynchronous domain, [115] proposes soft-error tolerance and correction scheme for Quasi Delay Insensitive (QDI) NULL Convention Logic (NCL) circuits based on proper sizing of feedback transistor used in the threshold gate design. Moreover, [116], [117], [118], [119], [120], [121] discuss duplication-based error resiliency techniques for widely utilized asynchronous paradigms, including NCL, Sleep Convention Logic (SCL), Pre-Charge Half Buffers (PCHB), and Weak Charge Half Buffers (WCHB).

The effects of redundancy on the accuracy of ANN-based functional approximation (FA) tasks or pattern classification (PC) tasks were first explored back in the early 1990s. It draws inspiration from the fact that ANNs as cognitive models should try to emulate other biological cognitive systems, much like the human brain, wherein neural redundancy and multiple replications of same processes have been scientifically established [122]. Moreover, the authors in [122] claim redundant ANN networks are a more viable option for accurate and stable networks compared to conventional ANN, such that the hardware overhead of redundancy outweighs the benefits achieved. Previously, [123] postulated that Triple Modular Redundancy (TMR) along with the majority gate can offer a viable approach to ensure single fault tolerance in ANNs. However, these works mostly adopted software-based algorithms demonstrating their analysis and evaluation. Meanwhile, the intrinsic computational advantages of NVMs in ANN-based detection on edge devices, offer new approaches to realize less power- and resource-hungry, more reliable, and compact ANNs. [124] utilizes hybrid spin-CMOS based majority gate primitives for approximate computing, whereas [125] utilizes Generative Adversarial Network (GAN) as an adversarial learning approach to reduce hardware learning overheads of conventional deep learning methods for massive, labeled datasets, as well as implements a memristive PiM accelerator for the Approximate GAN (ApGAN). Techniques such as binarization and pruning have also been widely explored. For instance [126] proposes a flexible and regular edge pruning technique for ReRAM based DNN accelerator modeled using CACTI. They evaluate their design on an in-house software model incorporated with hardware design constraints, on an NVDIA pre-trained LeNET network for MNIST dataset, and some other neural networks and datasets. Considering hardware fault-handling, [127] presents a methodology to rectify bit-errors in memristive crossbars via the identification of vital weights, and a retraining and re-mapping algorithm for those weights in the matrix, improving recognition accuracy. The authors derive their conclusions from experimental device testing data. The redundancy approach they propose improves recognition accuracy for the MNIST dataset by as much as 98% using a two-layer ANN implementation, with 20% random bit errors retrained and 5% of them remapped. However, this approach is limited in its scope of identifying significant weights and tracking the bit-errors for only those weights. [128] explores the concept of '*autonomy*' in management of fault tolerance, implying the circuit dynamically determines if another evaluation of outputs is substantiated. *Ensemble Learning* applied to SOT-based binarized CNN is explored in [129], where various types of classifiers trained for the same task result in better accuracy and energy efficiency than even the most complex of networks.

In this dissertation, the research combines the approach of ensemble learning with Spatial Triple Modular Redundancy (STMR) and the proposed Progressive Temporal Modular Redundancy (PTMR), to quantify how spin-based smaller sized neural networks fair in comparison to more complex larger spin-based NNs. Training and inference on varied sizes of spin based RBM DBN have been conducted with reconfigurable MRAM based stochastic neurons having activations such as plain sigmoidal, sigmoidal sq. root, etc., for a digit recognition task evaluated on a subset of MNIST dataset.

5.2 Architectural Overview

As a starting point, Probabilistic Inference Network-Simulator (PIN-Sim) along with Probabilistic Inference Network (PIR) frameworks have been utilized to evaluate the efficiency of our redundancy-based ANN implementations for digit recognition task on MNIST dataset suitable for resource constrained edge-devices. PIN-Sim framework, as elaborated in 2.2.2, is a hierarchical system of several functional modules written in MATLAB, python and SPICE. It is utilized for generating a DBNs hardware circuit-level implementation for any desired model size. It utilizes memristive crossbars for the weighted connections between different ANN nodes, and MRAM-based p-bit analog neurons for activation functions at the output of each NN layer. For the MNIST benchmark dataset with $28-pixel\ X\ 28-pixel$ input image samples, the DBN has 784 nodes in the input layer and 10 neurons in the output layer for each of the ten output classes, with any depth of internal hidden layers of the DBN depending on the NN size chosen for the experiments. The PIN-Sim framework outputs an analog voltage at each of the 10 neurons' outputs in the final layer. The PIR thereafter converts this probabilistic analog voltage at each neuron output into a 3-bit digital interpolated output. The 3-bit SS-PIR design [77] has been utilized in the proposed implementations. The PIN-Sim SPICE file of the neuron, called *'neuron.sp'*, is modified into the hybrid-spin/analog based reconfigurable activation neuron design called *Generalizable Analog Activation Function* (GAAF) presented in [1], by adding the necessary components after the p-bit output in the SPICE file. Combining GAAF neuron design with PIN-Sim, spin-based DBN implementations having different types of neuronal activations have been obtained. The work in [1] presents how different activations are achievable via changing certain control signals in the PiM architecture. As the activation functions and crossbars are implemented via spintronic devices, the inherent stochasticity of the p-bits and variations in the spin-based devices itself may affect the recognition accuracy of the overall NN. Therefore, to counteract such variations, the impact of our proposed STMR and PTMR approaches on such spin-based ANN has been analyzed for a digit recognition application.

5.3 Spatial Triple Modular Redundancy (STMR) in MTJ-based ANNs

Spin-based neurons naturally exhibit stochastic switching, resulting in slightly varied predictions for classification tasks even when run with the same inputs and identical weights and biases in the crossbar implementations. Prior works elaborate a lower bound on redundancy for feedforward neural nets to achieve substantial levels of fault tolerance in ANNs via Triple Modular Redundancy using a majority voter [123]. For use in applications where reliability is crucial, it is highly desirable for ANNs to be variation tolerant. Hence, a spin-based in-memory STMR application has been designed, as shown in Fig. 22, for handwritten digit recognition task evaluated on MNIST dataset, by implementing three instances of the same sized ANN in hardware utilizing the PIN-Sim and PIR frameworks. The size of the ANNs in STMR are chosen much smaller than that of baseline ANN without any redundancy to reduce area and energy footprint. Applying the concept of *Ensemble Learning*, and *Wisdom of Crowds* [123], the three structurally identical ANNs: *ANN1*, *ANN2*, and *ANN3* (as shown in Fig. 22), have been trained and tested on three separate activation functions (AFs): *AF1*, *AF2*, *AF3*, respectively, to achieve the most accurate and stable classification results. Such change in activations, e.g., sigmoid, sigmoidal square root, sigmoidal power (3/2), inverted sigmoid etc., is achievable by changing the voltages on the control signals, without any changes in the physical hardware [1]. As ANN efficiency and accuracy largely depends on the choice of suitable activation function for the given workload and ANN model, such designs grant the user with in-field reconfigurability during the inference phase with a choice to adopt the activation function presenting the best prediction accuracy rates. The same test images of $28 - pixel\ X\ 28 - pixel$ squared are fed to each of the ANNs. The analog outputs of the ten neurons in ANN1's final layer, are given by $h_{0(1)}$ to $h_{9(1)}$, $h_{0(1)}$ being the output of '*Neuron0*', which detects the class of digit '0' in the MNIST dataset. These are then passed through 3-bit PIR module, to convert each into its corresponding 3-bit interpolated digital output, given by $O_{0(1)}[2:0]$ to $O_{9(1)}[2:0]$, respectively. The MSB bits neurons from *ANN2*, and *ANN3*, are passed through a spin-based 3-bit majority gate (MG-3) to get the majority prediction for that bit of the neuron output. Hence, to evaluate the MNIST dataset our design requires 30 MG-3s, the outputs of which are concatenated to produce the final 3-bit predictions $F_{O(1)}[2:0]$ to $F_{O(9)}[2:0]$, of which the top one prediction is chosen as final output. This design has at least thrice the resource overhead as compared to baseline without redundancy, but no increase in processing latency as all the networks compute in parallel. The results are detailed in a later section.

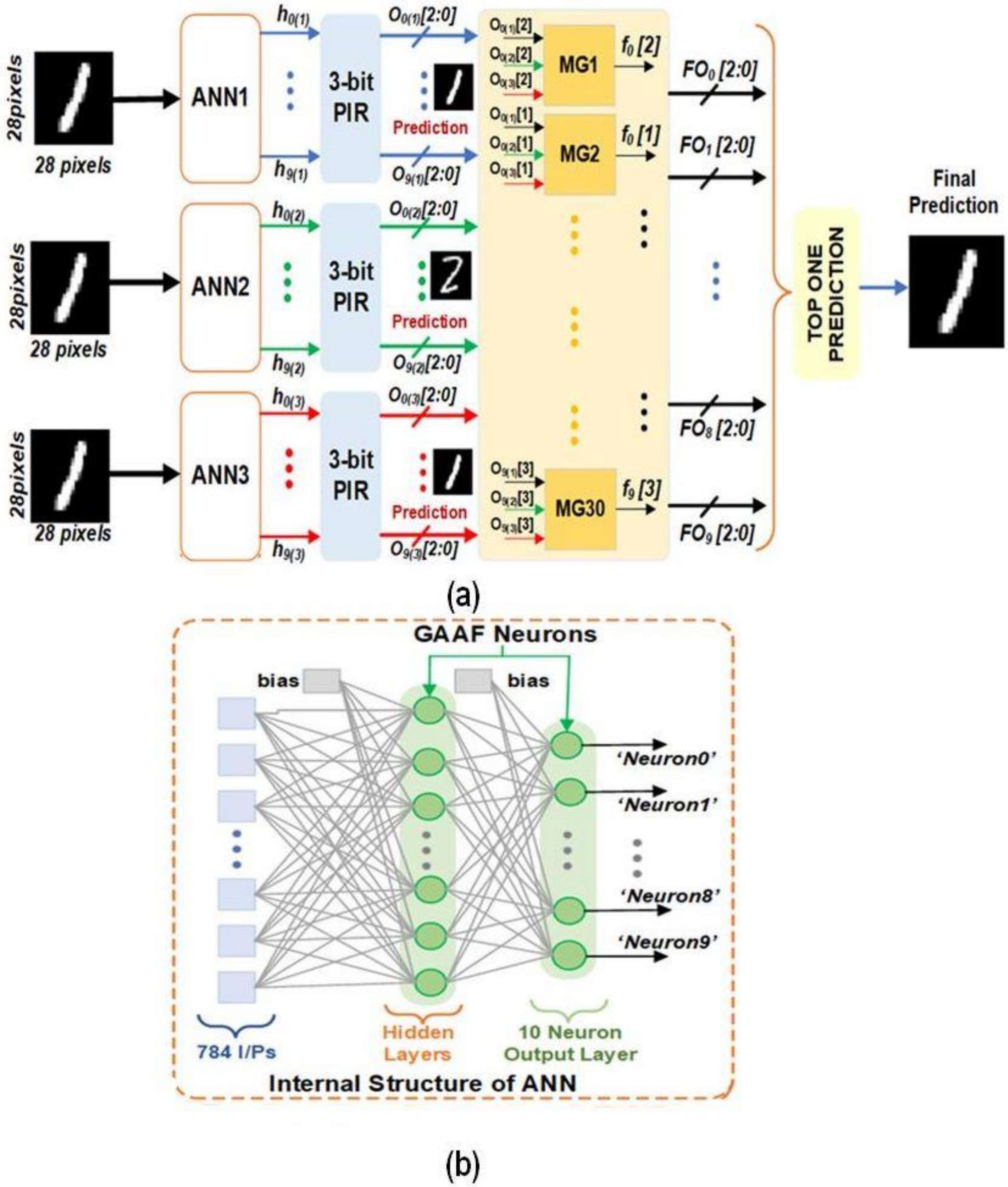


Figure 22: (a) Spatial Triple Modular Redundancy Architecture on Spin-based ANN Digit Recognition System, and (b) ANN Internal Structure [3]

5.4 Proposed Progressive Temporal Modular Redundancy (PTMR) in MTJ-based ANN

In this section, the proposed Progressive Modular Redundancy approach based on temporal redundancy (PTMR) has been detailed, which is a viable alternative to reduce the resource overhead cost of STMR while retaining similar accuracy. The overview of the proposed architecture is depicted in Fig. 23 (a). The primary differentiating factor between PTMR and STMR is the hardware implementation of a single ANN as opposed to three. The network is trained based on a particular Activation Function (AF), *AF1*, and sample the 3-bit digital PIR output for any test image of the MNIST dataset at two different time intervals, '*t*' and '*t+δ1*'. $O_{0(t)}[2:0]$ to $O_{9(t)}[2:0]$, and $O_{0(t+\delta 1)}[2:0]$ to $O_{9(t+\delta 1)}[2:0]$ are the 3-bit prediction outputs '*Neuron0*' to '*Neuron9*' in the final ANN layer, sampled at times '*t*' and '*t+δ1*', respectively. This interval, '*δ1*', between two sampling events is a design constraint, which depends on the processing time of the ANN per test image and the time required to reconfigure the ANN with a different *AF* (e.g., *AF2*). The activation reconfiguration is achieved via the assertion of the appropriate control signals within the GAAF-enhanced neurons [1]. The weights and biases of the new *AF* are stored on an on-chip buffer and written onto the crossbar in a time that is much less than '*δ1*' as elaborated in Section 5.5, ensuring accurate inference on the updated weights and biases. For very small sized ANNs, the on-board training time is only a few minutes, making this approach useful for edge devices.

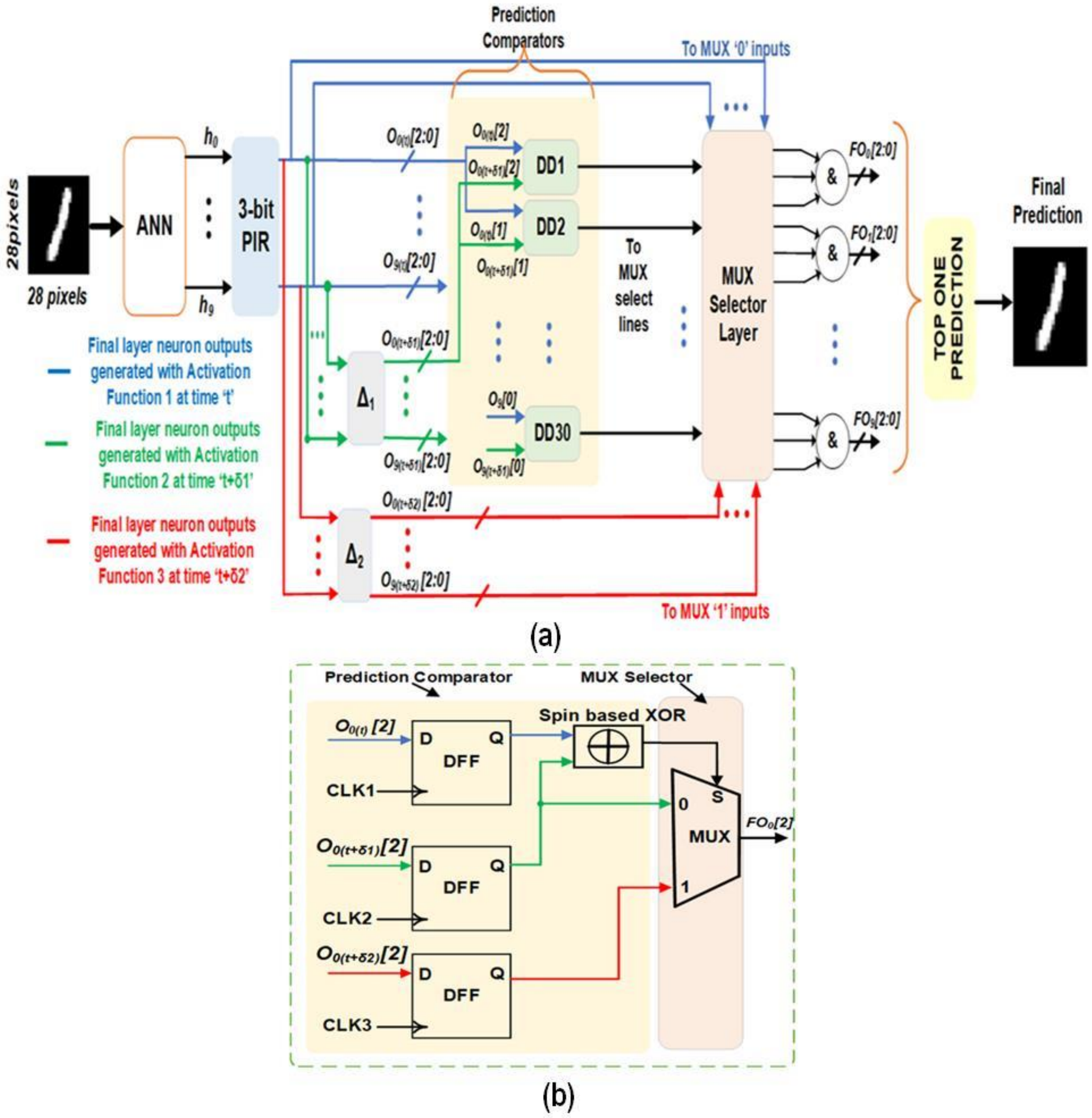


Figure 23: (a) Progressive Temporal Modular Redundancy Architecture for SCAPE, and (b) PTMR Internal Structure [3]

These outputs are then passed through a *Prediction Comparator Unit (PCU)*, as depicted in Fig. 23(b). Each bit of the prediction in the first sample is compared with the corresponding bit in the second sample using spin based XOR gates. If a discrepancy is detected, an enable signal '*En*' is generated which enables the control unit to activate the second sampling unit which proceeds to take a third sample of the test images at time '*t+2δ1'*, which decides the final prediction output. However, this is only for the test images where a discrepancy exists between the output of the first two samples, hence saving on much of the computation energy and hardware compared to STMR, at similar or sometimes improved accuracy but at the cost of increased latency. Energy consumption is reduced even further if the third sample is evaluated on a smaller ANN size compared to the first two sampled ANNs in the PTMR approach, which is a plausible option for reconfigurable hybrid spin-analog fabrics such as hybrid CMOS/spin FPGAs via a heterogenous fabric.

5.5 Experimental Setup and Evaluation

We evaluate the application-level performance of a DBN employing the STMR and PTMR redundancy techniques for 784x100x10 DBNs on SPICE against a baseline 784x500x500x10 DBN trained on sigmoid activation, without redundancy. The hybrid spin-based MG-3s and XORs utilized in STMR, and PTMR, respectively, and the GAAF neuron design, are implemented with SHE-MTJs models and device parameters like those used in [1], listed in Table 16. The flipflops, MUX-es, and other control peripherals are designed with CMOS PTM 14nm HP library, at V_{DD} = 0.8V. MG-3 and XOR gate implemented based on the designs in [130], consume 0.0273 mW and 0.0375 mW, respectively. The power consumption of the overall peripherals comprising the MG-3 gate in STMR, and XOR, FF, and MUX in PTMR are evaluated in SPICE to be 0.819 mW and 1.13 mW, respectively.

Table 16: SHE-MTJ Simulation Parameters [3]

Symbol	Parameter	Value
R_P/R_{AP}	P/AP MTJ Resistance	2.8 KΩ/5.6 KΩ
TMR	Tunnel Magnetic Ratio	100%
α	Damping Coefficient	0.007
t_f	Free layer thickness	1.3nm
T	Temperature	300K
P	Spin Polarization	0.52
V_{t_p}/V_{t_n}	P/NMOS Threshold	0.46 V/0.50 V
W_p/W_n	P/NMOS Width	44nm/22nm
θ_{she}	Spin Hall Angle	0.4
ρ_{hm}	Resistivity of HM layer	0.2 mΩ.cm
MTJ Area	MTJ Length × MTJ Width × $\pi/4$	60nm×30nm×$\pi/4$
HM Volume	L × W × T	100 nm×60 nm×3 nm

5.6 Results and Analysis

The 3 ANNs in STMR and PTMR are trained and tested on GAAF enhanced neuron activations of the sigmoid, sigmoidal square, and sigmoidal square root activations, with 3000 and 100 images, respectively, from the MNIST dataset. For the PTMR approach, the sampling time was 45 ns, i.e., 3 times the delay of the STMR architecture, plus additional delay to rewrite weights before trials. The overheads associated with training are minimized since trained weights and biases are pre-loaded into input buffers. We list our results in Table 17 based on HSPICE simulations, recognition accuracy, average power consumption, and a normalized area-overhead analysis of the nodes in weighted crossbar array in STMR and PTMR. We also calculate the Power-Error-Product (PEP) metric as,

$$PEP = (Average\ error \times Average\ power\ consumed\ in\ the\ crossbar\ and\ peripherals)$$

Table 17: Comparison based on Area, Power, and PEP [3]

	Avg. Error	**Avg. Power**	**Delay**	**Norm. X-bar Area**	**PEP**
A	30%	316.7 mW	17 ns	647,000x	95.01
B	45%	43 mW	13 ns	79,400x	19.35
B (PTMR)	**27%**	**44.02 mW**	**45 ns**	**79,400x**	**11.88**
B (STMR)	27%	167.2 mW	13 ns	238,200x	45.14

PEP gives a quantitative measure of the inefficiency of the spin based DBNs for faulty digit recognition. The results show that even though, in terms of base case of a single run using sigmoidal activation function, the accuracy of the 784x100x10 network, *B,* is significantly below that of the 784x500x500x10 network*, A,* use of modular redundancy on different activations allows comparable accuracy using the smaller network. Both *B (PTMR)* and *B (STMR)* show improvements in terms of power, area, and PEP. The STMR approach trades off area and power for performance. The PTMR allows for reduced power consumption as well as area, reporting *86.1%* and *87%* reduction in power and area overhead, at the cost of *~2.6x* increased throughput latency and *87.5%* reduction in PEP; the time-averaged power consumption for PTMR is less than the baseline 784x100x10 network, since the third stage is only necessary *~35%* of the time, meaning in all other cases the architecture stalls during the third stage.

5.7 Summary and Discussion

This chapter in the dissertation presents a novel area- and energy-efficient progressive redundancy-based approach suitable for ANN hardware inference on edge devices. Various performance tradeoffs and metrics show promising findings based on inference results of the MNIST dataset when implemented on a larger spin-based 784x500x500x10 network vs. a smaller 784x100x10 network implemented using the proposed Progressively Temporal Modular Redundancy and Spatial Triple Modular Redundancy. Both the PTMR and the STMR modified neurons show a 3% improvement in accuracy compared to the baseline case A with sigmoidal activation, whereas PTMR reports 86.1% and 87% reduction in power and area overhead, at the cost of ~2.6x increased throughput latency and 87.5% reduction in PEP.

CHAPTER 6: HARDWARE SECURITY PERSPECTIVE ON SENSITIVITY ANALYSIS OF SOT-MTJ BASED INFERENCING TO MANUFACTURING VARIATION[6]

Hardware-based acceleration approaches for Machine Learning (ML) workloads have been embracing the significant potential of post-CMOS switching devices to attain reduced footprint and/or energy-efficient execution relative to transistor-based GPU and/or TPU-based accelerator architectures. Meanwhile, the promulgation of fabless IC chip manufacturing paradigms has heightened the hardware security concerns inherent in such approaches. Namely, unauthorized access to various supply chain stages may expose significant vulnerabilities resulting in malfunctions including subtle adversarial outcomes via the malicious generation of differentially corrupted outputs. Whereas the Spin-Orbit Torque Magnetic Tunnel Junction (SOT-MTJ) is a leading spintronic device for use in ML accelerators, as well as holding security tokens, their manufacturing-only security exposures are identified and evaluated herein. Results indicate a novel vulnerability profile whereby an adversary without access to the circuit netlist could differentially influence the machine learning application’s behavior. Specifically, ML recognition outputs can be significantly swayed via a global modification of oxide thickness (Tox) resulting in bit-flips of the weights in the crossbar array, thus corrupting the recognition of selected digits in MNIST dataset differentially creating an opportunity for an adversary.

[6] ©IEEE. Part of this chapter is reprinted, with permission, from [4].

This chapter examines the sensitivity of SOT-MRAM devices to manufacturing parameter variations for secure computing. It explores how internal changes in different layers of the device can affect its behavior, as well as the impact on the performance of the ML accelerators designed using these devices and analyzes the effect on an application level. Simulation involving detailed comparison with an ideal SOT-MRAM device is used to identify how a modified SOT-MRAM device performs under specific conditions. It is shown that a malicious global change to T_{ox} across the wafer can introduce a gainful vulnerability to the ML digit recognition system.

6.1 Proposed Approach for Sensitivity Analysis

In this section, the proposed threat model that exploits the sensitivity of device characteristics to process variation has been presented. Subsequently, an approach to study the impact of such attacks at the application level has been detailed.

6.1.1 Proposed Threat Model

A white-box threat model is devised based on the following assumptions:

(1) the attacker is a hardware supply chain insider, capable of introducing variations in one or more critical MTJ parameters during fabrication,

(2) the introduced variations fall within an acceptable range while maintaining a stealthy nature, making them challenging to detect, and

(3) the attacker has prior knowledge of the memory architecture of the neural network, i.e. the knowledge of the critical nodes in the weight matrix, that when affected by bit-flips can significantly affect the accuracy.

These assumptions are valid due to side-channel information leakage in recent times, which can transpire if the attacker has a subset of the test data and uses it for inference. In Section 6.3, we provide experimental evidence that modification of device physical characteristics could leverage process variation (PV). In particular, it has been demonstrated how changing the thickness of the oxide layer, Tox, among other physical parameters, can result in modification of the resistive behavior of MTJs; thus, affecting the read current flowing through the device. Considering an ML accelerator design that utilizes a crossbar architecture with MTJs, such changes in the read current can accumulate across neighboring branches, resulting in incorrect firing of neurons within a neural network application. The potential for an attacker having this knowledge to determine the minimum threshold for variation for a stealthy attack, which falls within the acceptable range to pass functional testing, has been comprehensively investigated. Although the manipulation may go undetected during testing, this can still significantly disrupt the usual operation of a target application.

6.1.2 Approach to Examine Sensitivity against Potential Threats

Our high-level approach to examine the sensitivity of the application against such threats is depicted in Fig. 24. The goal is to demonstrate how PV changes in physical parameters at the device level can impact the performance at the application level, particularly for in-memory computing applications implemented with these devices. First, it has been analyzed how accumulated currents from multiple branches in the weight matrix, such as the read currents, *I_{read1}* and *I_{read2}*, shown in Fig. 24, may be large enough to cause either bitflips of multiple weight nodes in the crossbar array or incorrect firing of neurons in a given ANN. Such bitflips can cause incorrect firing of neurons in ANNs eventually affecting the performance, e.g., reducing the accuracy of a handwritten digit recognition application based on the MNIST dataset.

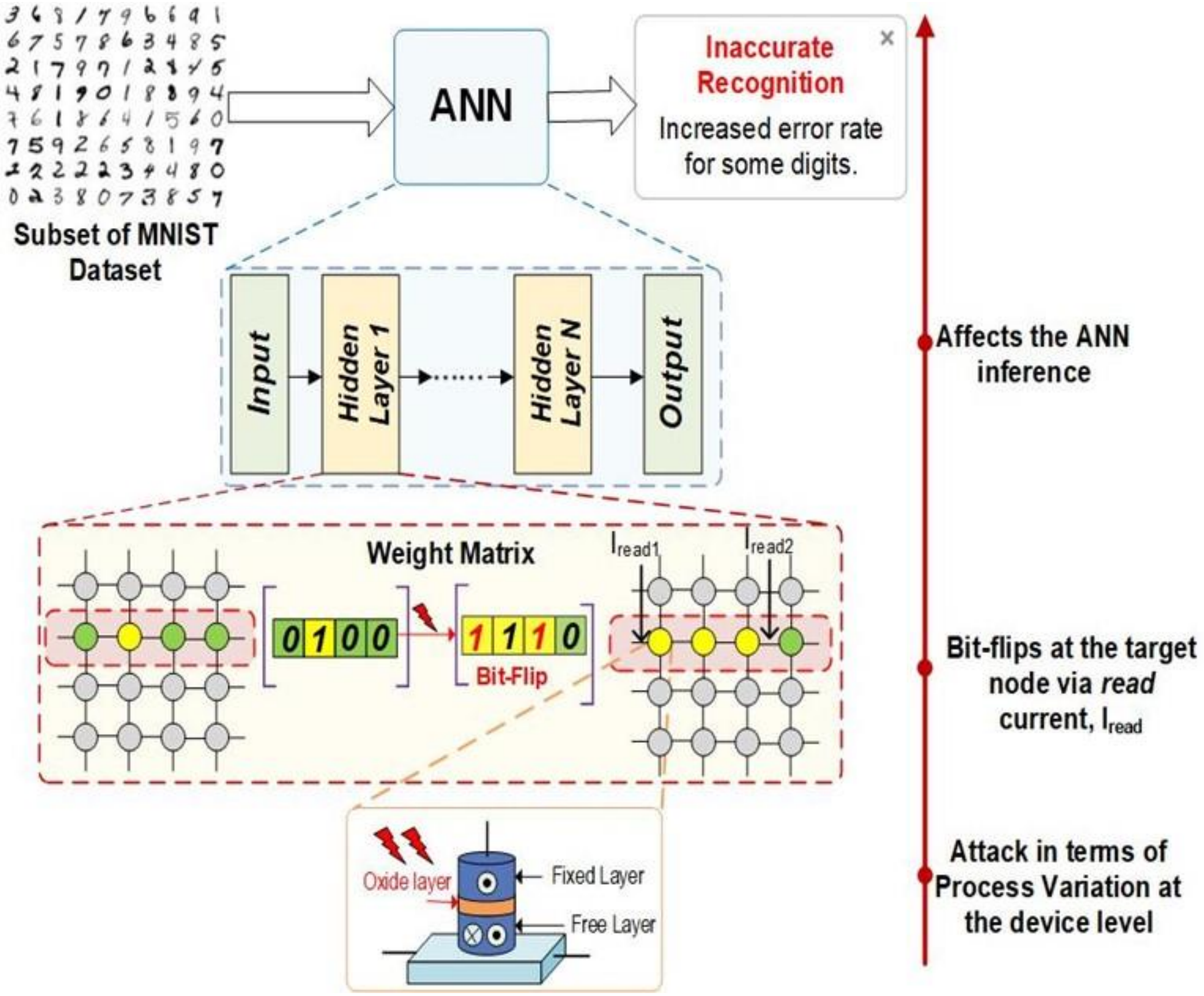

Subset of MNIST Dataset
ANN
Inaccurate Recognition
Increased error rate for some digits.
Input
Hidden Layer 1
Hidden Layer N
Output
Weight Matrix
0 1 0 0
1 1 1 0
Bit-Flip
Iread1
Iread2
Oxide layer
Fixed Layer
Free Layer
Affects the ANN inference
Bit-flips at the target node via read current, Iread
Attack in terms of Process Variation at the device level

Figure 24: Approach to Examine Sensitivity against Various Threats [4]

This study examines the impact of 10% isolated variations in oxide thickness, length, breadth, and thickness on the device resistance characteristics of the SOT-MTJ. The variations include the thickness, length, and width of the free layer and heavy metal layer. Furthermore, the combined effect of PV on all three factors, i.e., the free layer length, width, and the oxide thickness parameters, has been observed by performing Monte Carlo (MC) simulations, such that the combined total variation is limited to less than 10%. Exceeding this limit is avoided, since beyond this the variations in physical dimensions of the device could be detectable during the testing, violating the attack's purpose of remaining stealthy. After careful analysis, the effects of PV on the switching behavior of a single device are studied via HSPICE simulations. Finally, the impact of such variations on the performance of a hand-written digit recognition application is analyzed.

6.2 Experimental Setup

The proposed evaluation framework and process flow are depicted in Fig. 25. We used the approach given in paper [131] to simulate the behavior of SOT-MTJ devices in this paper, in which a Verilog-AMS model is built utilizing the physics equations provided in [132], [133]. The model is then used in the SPICE circuit simulator to test the functionality of the constructed circuits. To analyze the effect of the physical variations on device performance, we utilize a HSPICE model of the SOT-MTJ device with parameters in Table 18 [3], [133] along with the resistance values, i.e., high (anti-parallel (AP), and low, i.e., parallel (P), resistive states and tunnel magnetoresistance (TMR) as obtained from MATLAB simulations.

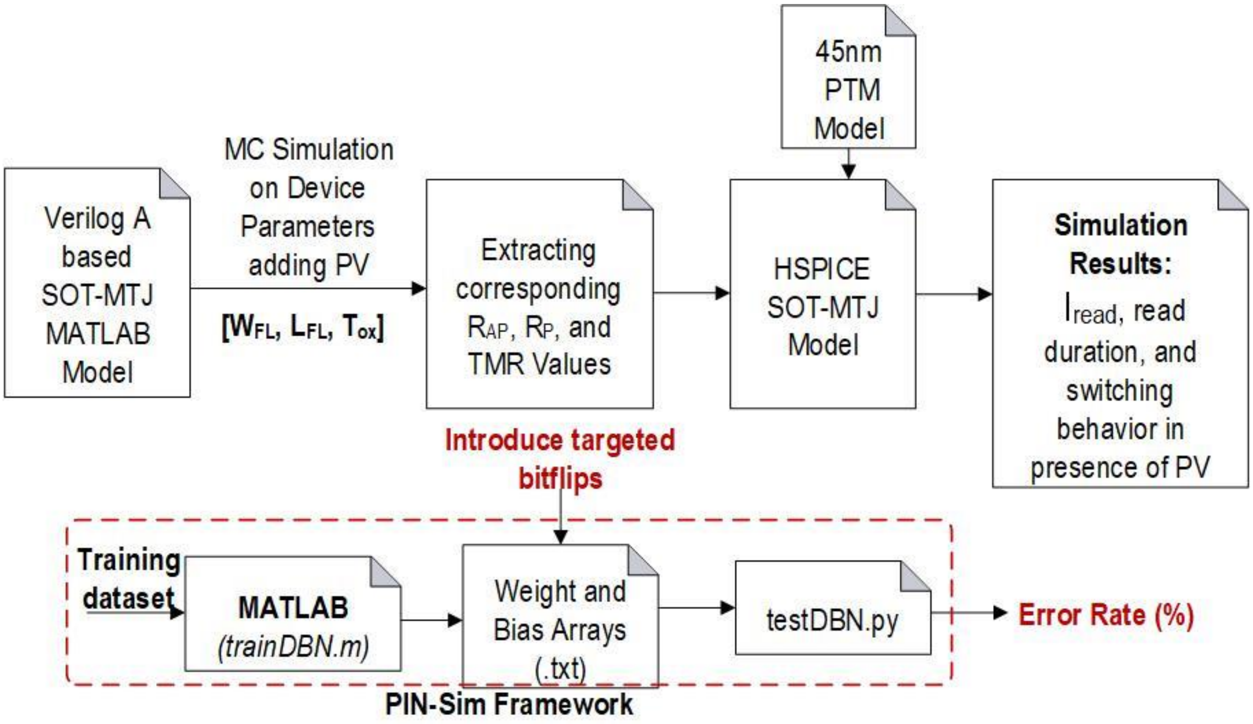

45nm PTM Model
Verilog A based SOT-MTJ MATLAB Model
MC Simulation on Device Parameters adding PV
[WFL, LFL, Tox]
Extracting corresponding RAP, RP, and TMR Values
HSPICE SOT-MTJ Model
Simulation Results:
Iread, read duration, and switching behavior in presence of PV
Introduce targeted bitflips
Training dataset
MATLAB (trainDBN.m)
Weight and Bias Arrays (.txt)
testDBN.py
Error Rate (%)
PIN-Sim Framework

Figure 25: Simulation Framework and Process Flow [4]

Table 18: HSPICE Device Simulation Parameters [4]

Symbol	Parameter	Value
α	Damping Coefficient	0.02
T	Temperature	300K
P	Polarization	0.73
TMR	Tunnel Magnetic Ratio	100%
T_{ox}	Thickness of oxide layer	1nm
RAp	Resistance Area Product	5Ω.µm^2
M_s	Saturation Magnetization	1185 A.m^{-1}
$\hbar$	Reduced Planck's Constant	6.626e-34/2π J.s
H_k	Anisotropy field	80 Oe
MTJ Volume	L × W × T × $\pi/4$	(60×45×0.07×$\pi/4$) nm^3
HM Volume	L × W × T	(60 nm×80 nm×2) nm^3

The read current (I_{read}) and the corresponding read duration of the SOT- MTJ are measured. It has also been studied whether due to PV, the same read current can end up causing the device to switch its state within the measured read duration. Moreover, we designed a 786x200x10 ANN using the PIN-Sim framework and introduced multiple targeted bitflips in the weights and bias arrays of the ANN to study the impact on in-memory applications targeted for ML accelerators [3].

6.3 Simulation Results and Analysis

This section provides an overview of the simulation results obtained from both a single device and an in-memory computing crossbar array. Potential threat detection and error mitigation techniques have also been outlined.

6.3.1 Single Device Results

The resistance of MTJ in a SOT-MRAM is modeled using the following equations:

$$R_{MTJ} = \frac{T_{ox}}{f \times A \times \sqrt{\varphi}} \exp\left(1.025\ t_{ox}\sqrt{\varphi}\right) \tag{6.1}$$

$$TMR = \frac{TMR_0}{1+\left(\frac{V_{bias}}{V_h}\right)^2} \tag{6.2}$$

where R_P = R_{MTJ} and R_{AP} = $R_{MTJ}(1+TMR)$**,** T_{ox} is the oxide layer thickness, f is a material dependent parameter, A is the device surface area, φ is the height of the energy barrier of the oxide layer, V_{bias} is the bias voltage, and V_h is the bias voltage at which TMR drops to half of its initial value [3]. Figure 29 shows the research findings by performing MC simulations with 2,000 instances to observe the effect of isolated 10% PV of various device parameters on R_P, R_{AP}, and TMR. The effect of PV on the dimensions of the heavy metal is found to be negligible on the device resistances and the TMR and hence, not included herein. However, the length and width of the device and the thickness of oxide layer (T_{ox}) shows high dependency with the device resistive behavior and TMR, which has been explored further to investigate the proposed threat model.

Fig. 26(a) and (b) show that device resistive behavior has a linear proportional relation with the width (W_{FL}) and length (L_{FL}) of the free layer. It is found that the TMR, being a ratio of the device resistances, remains constant for both the variations. Fig. 26(c) depicts that the device resistance increases exponentially with increase in the oxide thickness (T_{ox}), especially beyond 1.15 nm, whereas TMR vs. T_{ox} has a linear relationship as per Fig. 26 (d). It can be observed that with the decrease in W_{FL} and L_{FL} as well as decrease in T_{ox}, the gap in resistances of R_P and R_{AP} states narrows down. This indicates the possibility of potential threats and reliability issues such as read failure, oxide breakdown, unwanted bitflips.

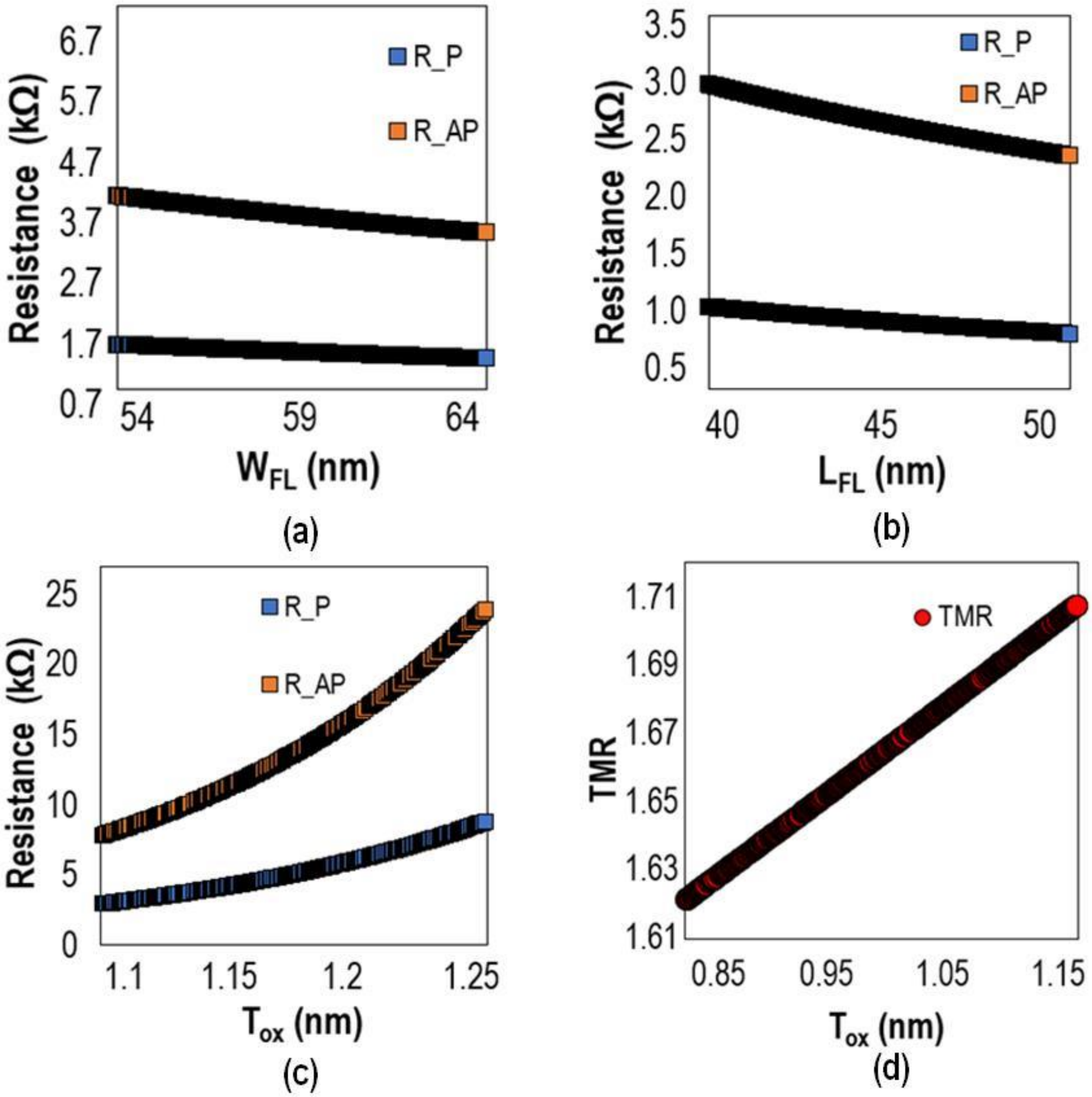


Figure 26: Individual PV Analysis of (a) Width of Free Layer, (b) Length of Free Layer, and (c) Oxide Thickness on R_P, and R_{AP}, and (d) Effect of Oxide Thickness on TMR [4]

Furthermore, in the scenario considering the combined effect of PV, amounting to a total 10% variation, on length, width, and thickness parameters, we observe from Fig. 27 (a) and (b) that R_P and R_{AP} device resistances exhibit comparable distributions with respect to the width (W_{FL}) and length (L_{FL}) of the free layer, with multiple overlapping samples, as well as a few unexpected anomaly points that do not fall within either cluster. Such anomalies may be of particular interest to a malicious attacker seeking to exploit the unanticipated device behavior to inject faults or cause device malfunction. Fig. 27(c) demonstrates the exponential dependence of device resistance with T_{ox}, in combination with variation in W_{FL} and L_{FL}. Moreover, Fig. 27 (d) demonstrates the linear dependence of TMR with respect to variations in oxide thickness, width, and length of free layer. For the range of oxide thickness, T_{ox}, between 0.8 nm and 1.15 nm, the R_P and R_{AP} values appear to be very close, as shown in Fig. 27(c). Based on these results, it is insightful to consider how a minor variation in oxide thickness may cause a change in device resistance from R_P to R_{AP}, and vice versa, making the devices prone to faults and bitflips from Logic '0' to Logic '1'. The experimental values of T_{ox}, W_{FL} and L_{FL} lie within 96% confidence intervals of 1nm ± 1.45e-3 nm, 60nm ± 8.85e-11 nm, and 45nm ± 6.65e-11 nm, respectively, for the 2,000 samples.

According to the SOT-MTJ model used [3], the device oxide thickness should be in the operating range of 0.85 nm to 1.15 nm. Thus, this has been applied as a limitation for our investigation considering variations within ~3% of 1nm, which is the baseline. First, the read duration and the read current that passes through the SOT-MTJ device is measured for different oxide thicknesses values, by modeling the device connected with simple read-write peripheral circuitry in HSPICE designed with CMOS PTM 45nm HP library, at V_{DD} = 0.8V [133].

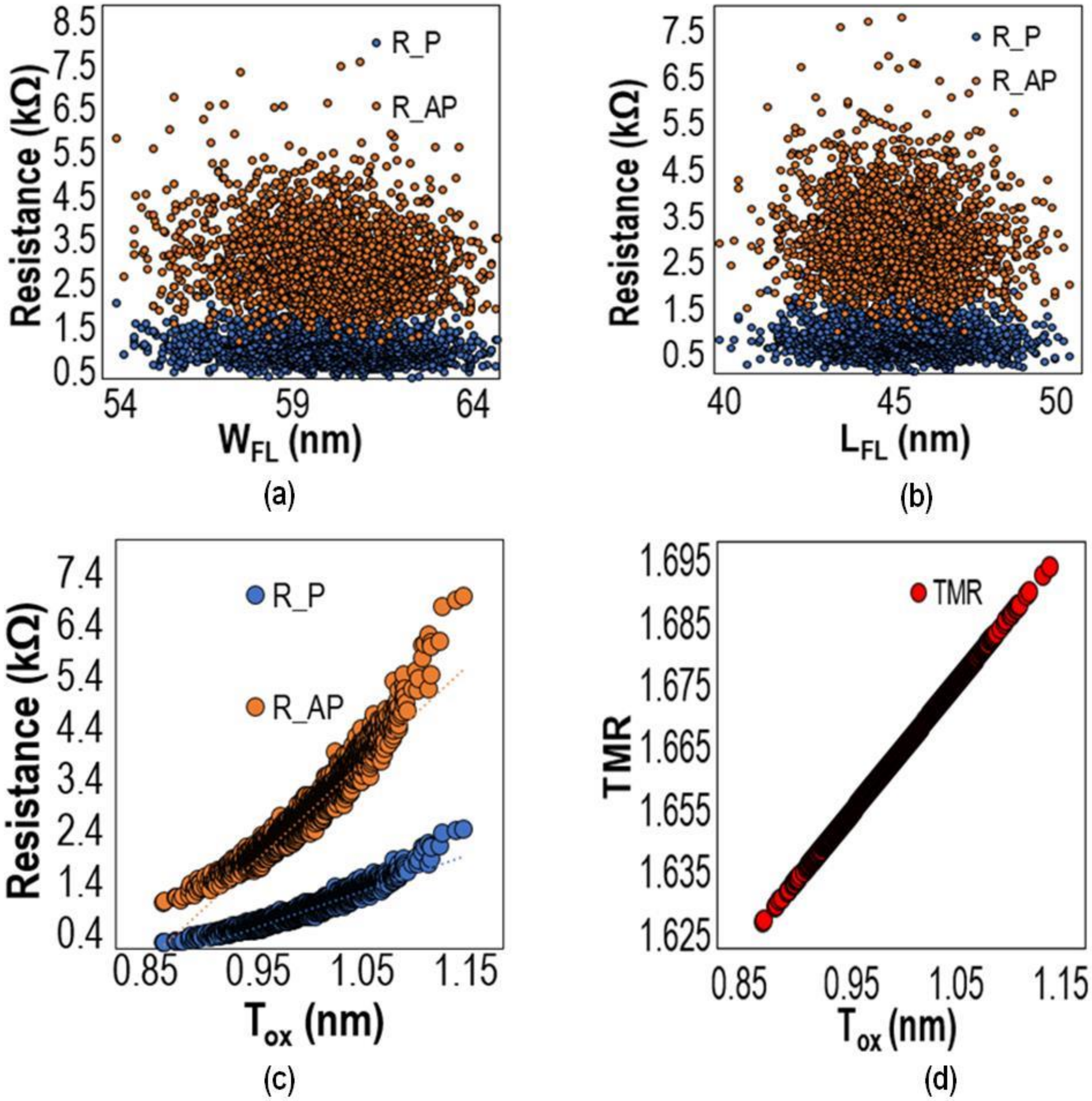


Figure 27: Graphs obtained by applying Combined PV to Length, Width, and Thickness Parameters, showing the effect of Modification in (a) Width of Free Layer, (b) Length of Free Layer, and (c) Oxide Thickness on Device Resistance, and (d) Effect of T_{ox} Variation on TMR [4]

It is then evaluated whether the read current through a device is significant enough to cause bitflips in the devices affected through PV within the read duration (<5ns). In particular, the study emphasizes on studying if accumulated read currents from neighboring branches in the crossbar becomes higher than the critical switching current resulting in an undesirable switching of the device state from 'P' to 'AP', or vice versa during the read operation. The switching current for the device model is calculated based on Eq. (6.3), where q is the electron charge, $\hbar$ is the Reduced Planck constant, α is the Gilbert damping coefficient, H_k is the anisotropy field, M_s is the saturation magnetization, and V is the volume of the nanomagnet [134].

$$I_{crit\_sw} = 2\frac{q}{\hbar}\alpha H_k M_s V\left[1+\frac{2\pi M_s}{H_k}\right] \tag{6.3}$$

Fig. 27 shows the read current values (I_{read}) that cause successful switching of initial state of MTJ for the values of T_{ox}, and are marked in red. These represent the targeted bitflips by the attacker via introduction of PV into the MTJ device. Specifically, for T_{ox} =< 1.2nm in case of 'P' to 'AP' switching, as I_{read} flowing through the device is above I_{crit_sw}, targeted bitflips occur. Similarly, for T_{ox} >1.2nm, the I_{read} is insufficient to cause bitflips and hence, represents safe limit of T_{ox} for such bitflip attacks through read current fluctuations. Likewise, this safe range for the 'AP' to 'P' switching is found to be T_{ox} >1.05nm, as the device can hold its initial stable resistance state and remains immune to switching. The values of critical switching current, I_{crit_sw}, for 'AP' to 'P', and 'P' to 'AP' switching of the MTJ device is shown by the dotted lines in Fig. 28 as observed in HSPICE simulation and found from Eq. (6 3). It aligns with the critical switching current of the device in literature and the asymmetric switching characteristics of such devices [134].

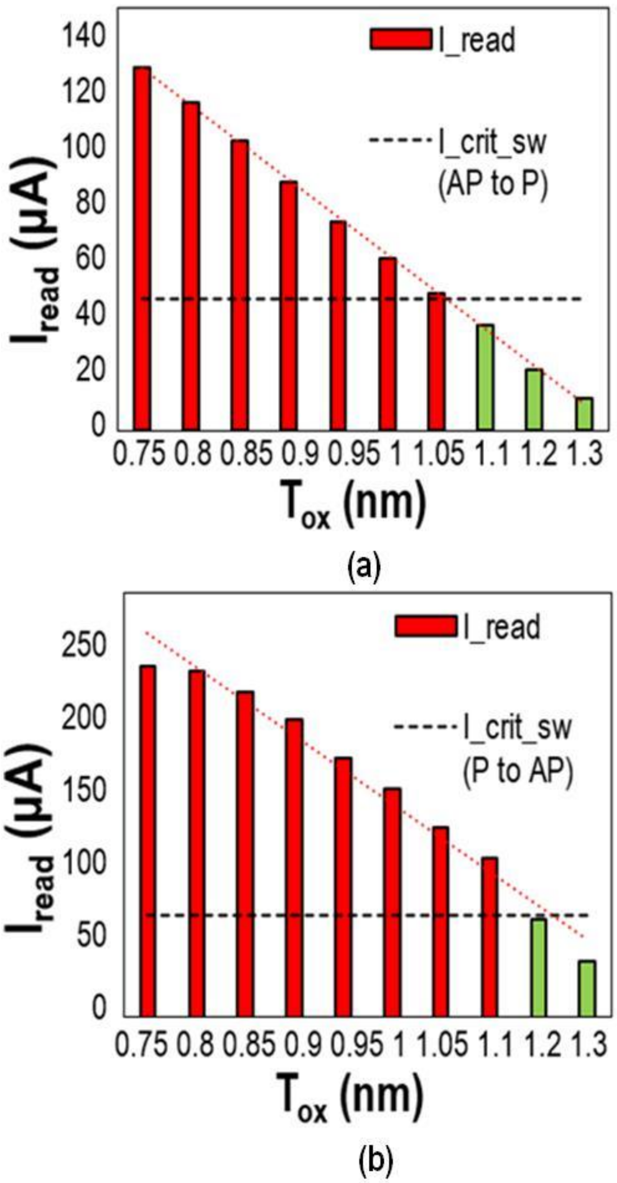


Figure 28: SOT-MTJ Device Read Current (I_{read}) Variation with Change in Oxide Layer Thickness during (a) 'AP' to 'P', and (b) 'P' to 'AP' switching [4]

6.3.2 Crossbar Array Analysis Results

The PIN-Sim consolidated framework developed in MATLAB, Python, and HSPICE has been utilized for evaluating the performance for large scale applications. A 784x200x10 ANN is designed and trained on 3,000 training samples in MATLAB and the testing results are presented via running 100 test samples in HSPICE, containing a mixture of the ten different digits from 0-9.

The training weights and biases extracted from the MATLAB-based model are translated to their corresponding memristive values in HSPICE. A python-based module is utilized to implement the memristive crossbar and a low-energy/-footprint spin-based neuron with sigmoidal activation function [3]. The overall error rate achieved for the 100 test samples along with individual error rates for each digit recognition are listed in Table 19, where *Error Rate = (# of incorrect recognitions of a digit) / (# of samples of that digit)×100%*. Initially, with weights and biases ranging from 1KΩ to 5KΩ, the overall error rate achieved for the 100 test samples is *41%,* which can be attributed to the small network size with only one hidden layer. Herein, the research focuses on the effect of PV-caused bitflips, applied to a single row of the weights in the crossbar array. In order to analyze the performance within the target T_{ox} confidence interval mentioned before, the weights and biases resistances are modified to two discrete levels, 2.5KΩ and 5KΩ, which results in an overall error rate jump to *58%*. It is observed, if only 0.05% of the overall weights are affected by bitflips, the resulting overall error rate increases by another 2%. Among the digits, digits '0', '4' and '9' show an increase in error rates due to bitflips, whereas digits '1', '5', '6', and '7' show a decrease in error rates due to implemented bitflips. With 0.05% of bits in crossbar having a flipped resistance state, digits '4', and '5' show highest overall error rates and digit '9' the lowest. The recognition accuracy of digits '2', '3', and '8' remain unaffected by bitflip attacks. These findings can be tactically exploited by an attacker to affect certain digit recognition more than others, thereby influencing the performance of other embedded applications interfacing with this digit recognition for further processing.

Table 19: Crossbar Array Analysis Results- Effect of Bitflips on Error Rates (%), and Accuracy= (100-Error Rate) % [4]

Test conditions	**Weights and biases ranging 1KΩ - 5KΩ**	**2 discrete weight levels 2.5KΩ & 5KΩ**	**Bitflips in 0.05% nodes of overall weight matrix**	**Impact on Accuracy**
Digit 0	0%	42.85%	57.14%	**Moderate**
Digit 1	66.67%	80%	73.33%	**Significant**
Digit 2	25%	50%	50%	**Minimum**
Digit 3	45%	36.36%	36.36%	**Minimum**
Digit 4	42.8%	64.28%	85.71%	**Significant**
Digit 5	42.8%	100%	85.71%	**Significant**
Digit 6	20%	60%	20%	**Moderate**
Digit 7	64.28%	85.71%	78.57%	**Moderate**
Digit 8	0%	50%	50%	**Minimum**
Digit 9	33.33%	0%	11.11%	**Minimum**
Overall	41%	58%	60%	**Moderate**

6.4 Potential of Threat Mitigation

Various strategies exist to safeguard ICs against threats like logic locking system, deep-learning power side-channel attack mitigation, neuromorphic computing modules for IoT, etc. [25], [135], [136], [69]. The work in [135] introduces an innovative approach to generate hardware watermarks by utilizing SOT-MTJ devices, which aims to secure intellectual property (IP) cores in system-on-chips (SoCs). Beyond manufacturing variations, the sense amplifier circuit which is utilized to read the state of MTJ is highly susceptible to aging-related degradation of the threshold voltage of its constituent transistors. Thus, a lifetime mitigation strategy should consider Bias Temperature Instability (BTI)-induced variations which may mask or otherwise interfere with an effective vulnerability mitigation strategy [136]. Alternatively, a self-organizing mitigation approach based on output discrepancy awareness demonstrated for CMOS-based arrays could be extended for crossbar configurations [69]. The various detection and mitigation strategies in literature can be classified into two broad categories:

a. **Detection and mitigation of hardware trojan attacks:** Reverse-engineering is the enabler of HT attacks and some countermeasures to mitigate reverse engineering attacks are proposed in the literature [137]. Some process variation mitigation techniques for spintronic and memristive devices have also been researched, such as tunable stochasticity using feedback mechanism, radiation hardening [138]. etc. Finally, in [28], the authors propose Symmetrical MRAM-LUT (SyM-LUT) by using the LOCK & ROLL approach to eliminate the reverse engineering and side-channel attack using a defense-in-depth mechanism.

b. **Detection and mitigation of fault injection attacks:** In [139], the authors present a dynamic task remapping using a built-in self-test (BIST) based technique fault detection method to determine the fault density of crossbars to guide the dynamic remapping technique. Rearranging tasks with lower fault tolerance from crossbars with high fault density to ones with lower fault for training can result in an average accuracy drop of only 0.85%. Another approach to detect and mitigate the FIA is proposed in [140]. The authors developed a Fault-to-Time Converter (FTC). To be precise, the effect of faults injected by an FI attack method is transformed into quantifiable “time” by use of the FTC sensor.

6.5 Summary and Discussion

Implementing SOT-MRAMs in ML accelerators in recent times obviates the growing necessity for awareness and, ultimately, reasonable mitigation of security threats associated with the underlying devices' manufacturing process. It has been explained how a maliciously modified SOT-MRAM can change the behavior of AI hardware performing critical decision-making tasks. The research presented in this chapter demonstrates how global changes to a single manufacturing aspect of a SOT-MRAM device, such as T_{ox}, can reveal bitflip vulnerability of memristive values. The simulation results illustrate a change in the oxide layers can cause unwanted switching of the operational state of the MTJ device. Beyond the simulation results, it is warranted to examine the actual physical parameters of maliciously fabricated MTJ device to demonstrate more vulnerabilities than the current simulation results. The potential weaknesses of the manufactured MTJ device may surpass those identified in the present simulation outcomes. Therefore, the future goal can focus on physically fabricating a maliciously modified MTJ device to differentially-execute operations. Doing so can advance hardware security at securing such emerging technology-based intelligent edge applications from manufacturing threats.

CHAPTER 7: LOW POWER NN ARCHITECTURES FOR RECONFIGURABLE EDGE COMPUTING APPLICATIONS[7]

7.1 Context and Background

Conventional hardware platforms for video and image processing are predominantly dependent on TPUs and GPUs, which demand substantial amounts of power and space. Numerous edge-IoT applications might not find this feasible. Therefore, reconfigurable, energy-efficient, and high-speed platforms such as FPGAs have gained considerable research traction in recent years for numerous ML and NN-based real-time object/pattern recognition tasks for edge computing applications. Consequently, this dissertation also presents my investigation on the hardware implementation of machine learning and object detection algorithms on reconfigurable CMOS-based FPGA platforms at the application level. I did so by contributing to a project that aimed to identify neural network accelerator frameworks that are both energy-efficient and resource-efficient, with the purpose of enabling effective video predictions.

A multitude of autonomous systems are swiftly integrating themselves into modern urban environments, whether they be maritime, terrestrial, or aerial. Robotic systems like unmanned aerial drones, while exhibiting remarkable competence in data collection, are limited in their capacity to execute in-situ learning and generate impromptu decisions due to operational resource allocation restrictions, processing power limitations, and storage capacity deficiencies. Amidst the tremendous volume of data processed by these gadgets, there is a

[7] ©IEEE. Part of this chapter is reprinted, with permission, from [141] and [142].

pressing need for novel computational theories, and efficient computational models. Therefore, the present research trajectory in this domain encompasses the development of a range of efficient ML and object detection algorithms, NN accelerators, and implementation of rapid prototypes on FPGAs. In addition, there have been research endeavors to minimize the resource and storage overhead of NN implementations on hardware, as they have large dimensions in terms of weights, biases, model setup, and computing requirements. For efficient NN-based pattern recognition, numerous optimization algorithms exist, including quantization, pruning, and others. However, most of these optimizations are tailored to work on a particular NN model specification that the user begins with. The choice of NN structure and the application of proper optimization schemes play a vital role in the overall task performance. Recent research efforts have been devoted to the development of automated platforms capable of identifying and executing the most effective combination of various optimization algorithms to generate the most suitable NN structure for the targeted application.

This chapter provides an overview of some of the past and contemporary efforts in the field of various NN implementations and their optimization strategies. Specifically, the focus has been targeted towards FPGA-based implementations of object and image detection algorithms and NN accelerators along two recent and trending frameworks: Vivado High Level Synthesis (HLS) and Vitis AI.

7.2 Implementation of NN Accelerators and Object Detection Models on Vivado HLS Framework

In the embedded applications domain, there is a growing research interest in utilizing Deep Neural Networks (DNN) in edge-devices. This can be predominantly attributed to the adaptability, versatility, and extensive applicability in the field of computer vision, which implements quantized deconvolution of various cutting-edge algorithms such as the generative adversarial network [143], among others. The Convolutional Neural Network (CNN) is a widely used DNN in the field of image and/or video processing, which utilizes the efficacy of convolution in filtering image matrices.

Although CNNs are becoming increasingly popular for computer vision tasks like object detection, identification, and classification, the frameworks used to implement these systems are primarily intended for usage with CPUs and GPUs, with GPUs being the more hardware-optimized choice. Further development of this application-optimized hardware strategy can be achieved through investigation of CNN implementations on low-power and embedded devices utilizing FPGAs. FPGAs offer prospective advantages over GPUs in terms of energy consumption, thereby posing additional benefits for embedded hardware acceleration of convolutional neural nets. However, GPUs continue to be the preferred target device for CNNs and therefore, a plethora of libraries and tools exist, which can be used to create custom architectures in programming languages such as Python and C++. In contrast, this is not the case for FPGA development.

Moreover, an existing obstacle in CNN framework research is that they require more energy than is feasible on a platform with limited resources, despite maintaining satisfactory levels of precision and throughput. Over the past few decades, various approaches have been proposed to address these limitations. One such approach is the one-step architecture [144], which consists of balancing accuracy and latency requirements to meet energy restrictions. You Only Look Once (YOLO) [145] is one such set of architectures that has attracted considerable interest due to its efficient inference and training processing. Our work in [141] extends real-time demonstrated FPGA platforms to prevailing non-symbolic AI processing approaches [146]. In [141], a detailed procedure for transforming CNNs from a high-level programming language implementation into a bitstream format has been presented. This bitstream can then be programmed onto an FPGA device and used as a hardware accelerator for image and video processing tasks. In this approach, the data is quantized to reduce memory usage. The C++ code is synthesized to Verilog using Vivado HLS tools. The resulting hardware module is integrated with the ZYNQ SOC processor. Finally, the accuracy of the final implementation is tested.

7.3 Implementation of NN Accelerators and Object Detection Algorithms on Vitis AI Framework

In the past few decades, object detection has emerged as a highly researched field of interest, specifically due to its varied applications ranging from image processing, face detection, Autonomous Driver Assistance Systems (ADAS), and Advanced Video Surveillance Systems (AVSS). Traditional object detection methods, such as sliding window and region-based algorithms, are plagued by low accuracy, but deep learning based convolutional neural networks

(CNNs) have emerged as a suitable choice due to their superior performance and higher accuracy. Approaches for designing FPGA accelerators have also been widely studied and expanded beyond creating custom Register Transfer Level (RTL) designs using a hardware description language (e.g., Verilog, VHDL) or utilizing High Level Synthesis (HLS) using imperative languages (e.g., C or C++). Acceleration frameworks, with more advanced levels of abstraction, such as Vitis-AI, have recently become available, which do not require users to have high levels of expertise in RTL design or hardware languages. Vitis AI offers increased ease-of-use and simplicity of implementation, along with an available suite of standard model pruning and static quantization optimizations, making it a popular acceleration framework. In our collaborative work [142], Framework for Accelerating YOLO-Based ML on Edge-devices (FAYME) was implemented as an approach that aims towards a transportable foundation utilizing the embedded ARM processor for control flow sequencing acceleration of Deep Learning Processor Units (DPUs) instantiated within a LUT-based FPGA fabric. Herein, we studied the various performance and design tradeoffs offered by accelerating our chosen YOLOv4 network using the AMD Xilinx Vitis AI toolchain.

Previous efforts towards object detection tasks on FPGAs has been outlined in this section. In 2009, for a facial recognition task on a low-cost robot, Farabet et al. implemented a Conv Net-based CNN processor on an FPGA [147]. This marked the earliest implementation of an FPGA-based neural network. Since then, numerous CNN-based hardware implementations for tasks spanning from voice recognition and self-driving autonomous vehicles to object detection have been proposed. In addition, many optimization strategies have been investigated to decrease the storage demands of the network on-chip (NoC) while still retaining an acceptable level of accuracy. [148] proposed a dynamic precision data quantization scheme for ImageNet classification on a deep VGG16 model utilizing a Xilinx Zynq FPGA. They demonstrated that this approach resulted in faster recognition speed without significantly sacrificing prediction accuracy. [149] explored a pruning-based approach by removing redundant synaptic connections between the DNN layers, leading to reduction in overall model size, resource demands, and run time. The impact of hyperparameter tuning and model compression on on-board inference on edge-based devices was investigated in [150]. A hybrid CPU/FPGA based approach was proposed and deployed on a Zynq MPSoC ZCU102 board, which was more efficient than PC-based inference in terms of recognition speed and power consumption. In [151], a comparative performance analysis of three types of DNN accelerators is presented: a course-grained custom accelerator implemented in System Verilog, a fine-grained accelerator implemented in the Xilinx FINN tool, and a sequential accelerator implemented in the Xilinx Vitis AI toolchain. The designs were tested on an Avnet Ultra96-V2 Xilinx development board and evaluated on Visual Object Tracking (VOT) and Visual Tracker Benchmark (VTB) datasets. The custom accelerator demonstrated the greatest throughput despite its increased design implementation time and resource consumption. The fine-grained accelerator achieved an acceptable throughput with a low resource utilization; however, the low resource utilization was attributed to the fact that only 4-bit quantization was deployable, while the 8- bit quantization ran out of available on-board resources. The accelerator based on Vitis AI showed constant resource consumption independent of the network depth, average throughput, with the benefit of shortest design time.

High prediction accuracy is achievable with object detection models such as Region-based CNN (R-CNN), Faster Region-based CNN (FR-CNN), and others. However, such models have low recognition speed due to more complex computations including region proposal followed by classification and therefore, are not suitable for real-time edge devices. Single-stage detection algorithms, such as YOLO and Single-Shot Multi-Box Detector (SSD), exhibit a good detection speed along with acceptable accuracy for detecting larger objects. Several incremental enhancements to the initial iteration of the YOLO (YOLOv1) algorithm have been suggested throughout the years [144]. The backbone network that was present in the initial iteration of YOLOv1 was substituted with DarkNet-19 in YOLOv2. In YOLOv3, objects were classified utilizing independent logical classifiers as opposed to the SoftMax function utilized in YOLOv2. [152] implemented a YOLOv3 FPGA accelerator on Xilinx ZCU104 quantized by Vitis AI, along with using model pruning and data preprocessing techniques, demonstrating lower energy consumption and higher throughput than a GeForce GTX1080 GPU at comparable recognition accuracy. YOLOv4 [153] incorporated network modifications through the addition of residual network layers, which were implemented in an effort to improve accuracy albeit at the expense of an increased network size. Additional optimizations were implemented, including Self-Adversarial Training (SAT), Cross-Stage-Partial connections (CSP), and Mish activation, to achieve a 10% improvement in average precision over YOLOv3 for MS COCO object detection dataset at 65 FPS on a Tesla V100. Meanwhile, [154] implements a YOLOv4 model using Vitis AI deployed on a ZCU102 board. The training and evaluation processes are conducted using a tableware dataset. The training iterations range from 1,000 to 30,000, and the model attains a mean average precision (mAP) of 96.2%. Alternatively, [155] takes an ensemble learning approach by executing training and evaluation on four versions of YOLO object detection models on the WIDER Face recognition dataset using the Darknet framework on an Nvidia K80 GPU. Results were combined using both the Non-Maximum Weighted (NMW) and Weighted-Boxes-Fusion (WBF) methods, where WBF was found to produce the better mAP. Alternate beyond-CMOS-based approaches for hardware-based acceleration are also being currently explored [43]. Fig. 29 presents a taxonomy of these related works based on their deployment platforms.

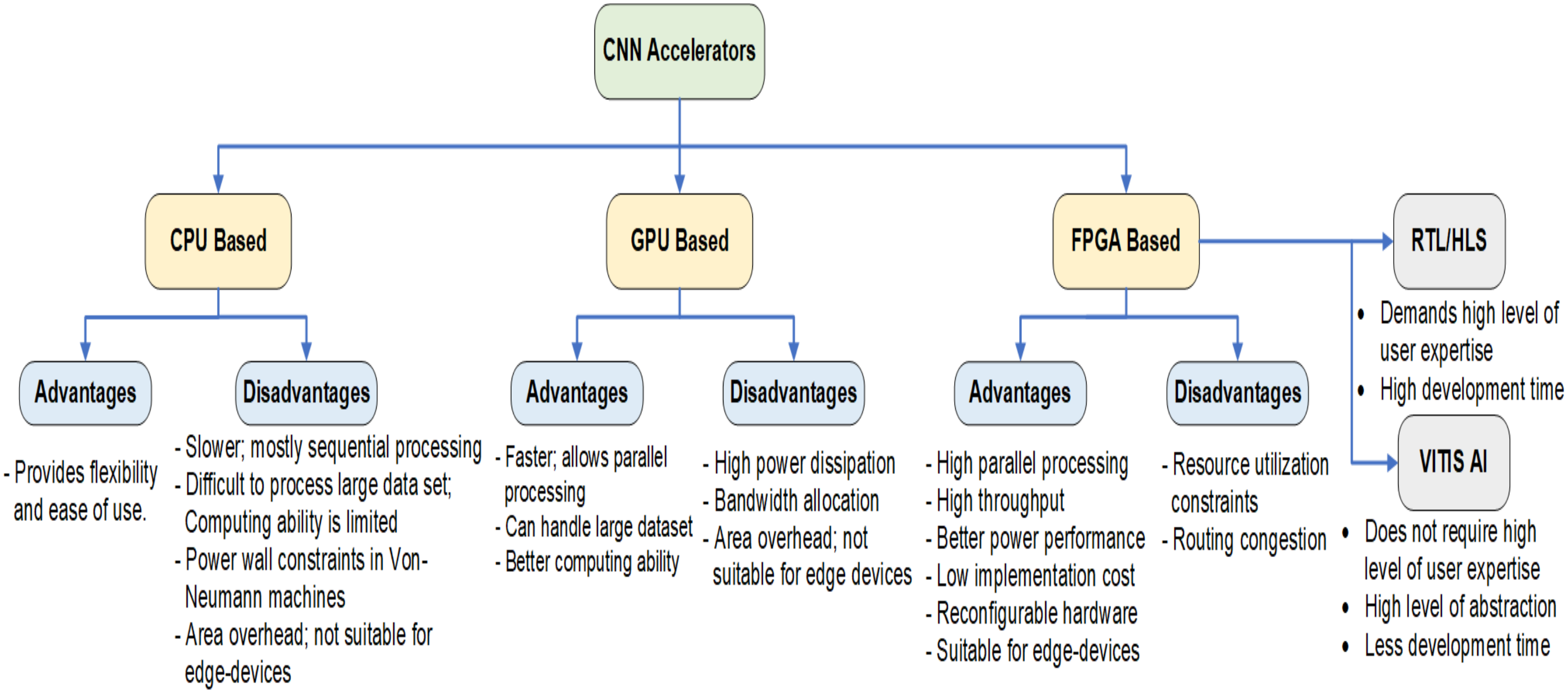

CNN Accelerators
CPU Based
GPU Based
FPGA Based
RTL/HLS
Advantages
Disadvantages
Advantages
Disadvantages
Advantages
Disadvantages
• Demands high level of user expertise
• High development time
VITIS AI
- Provides flexibility and ease of use.
- Slower; mostly sequential processing
- Difficult to process large data set; Computing ability is limited
- Power wall constraints in Von-Neumann machines
- Area overhead; not suitable for edge-devices
- Faster; allows parallel processing
- Can handle large dataset
- Better computing ability
- High power dissipation
- Bandwidth allocation
- Area overhead; not suitable for edge devices
- High parallel processing
- High throughput
- Better power performance
- Low implementation cost
- Reconfigurable hardware
- Suitable for edge-devices
- Resource utilization constraints
- Routing congestion
• Does not require high level of user expertise
• High level of abstraction
• Less development time

Figure 29: Taxonomy of CNN Hardware-based Accelerators Spanning Application, Energy, and Development Considerations [142]

In our collaborative work, as outlined in [142], we have put forward a method that seeks to create a transportable foundation by exploiting the embedded ARM processor to accelerate the control flow sequencing of Deep Learning Processor Units (DPUs) within a LUT-based FPGA fabric. Various performance and design tradeoffs offered by accelerating the proposed YOLOv4 network using the AMD Xilinx Vitis AI toolchain has been studied. Various levels of model bit-quantization was also tested and evaluated for performance and utilization of available memory and processing elements. A ResNet-50 model was also evaluated for additional comparisons. Our YOLO model was found to achieve a mAP of 0.581, and our ResNet model, a Top-5 accuracy of 0.950.

In addition, an attempt was made to determine the inflection point of the speedup offered considering Amdahl's law when deploying YOLOv4 on a ZCU102 board utilizing the Vitis-AI framework. Per Amdahl's law, the speedup achievable (*S*) is given by $S = 1/\left[(1-f)+\left(\frac{f}{N}\right)\right]$, where f is the fraction of execution time enhanced and denotes the fraction of workload that is parallelizable, and N is the number of cores employed. It is observed that even if the parallelizable fraction, f, is as high as 95%, the speedup achievable is approximately only ~20X when utilizing 512 parallel cores compared to execution on a single core, and the speedup drops sharply to only about 4X at $f = 0.75$ [156]. Based on the Vitis AI profiler results for execution times spent on the CPU vs. on the DPU when accelerating the 8-bit quantized YOLOv4 model, the commensurate ranges for f were anticipated to be approximately $0.83 < f < 0.94$. Assuming f is massively parallelizable, i.e., $N \rightarrow \infty$, then the overall speedup estimated by Amdahl's Law is limited to $S = 1/(1-f)$. Hence, for our case study, if we assume that N tends to infinity, although practically it does not, for $0.83 < f < 0.94$, we observe that a maximum speedup of 16.67-fold and a minimum of about 6-fold is achievable utilizing Vitis AI. The quantity N by which f can be sped-up depends on several factors in FPGAs, such as number of available DSP slices, off-chip I/O throughput, DMA transfer latency to move frames in-between the DSP and FPGA, available memory bandwidth, maximum data transfer rate in the memory control blocks within the FPGA, etc.

CHAPTER 8: CONCLUSIONS AND FUTURE DIRECTIONS

8.1 Dissertation Technical Summary

The Von-Neumann bottleneck, causing significant data transfer latency between the processor and main memory, is a key challenge in computer architecture. Crossbar arrays, based on emerging magnetoresistive devices such as Magnetic tunnel junctions, aim to address this bottleneck, offering substantial area and performance advantages, especially for applications requiring linear transformations and in-memory vector-matrix multiplication. Utilizing a hybrid analog-digital methodology enables intrinsic execution of specific computations, crucial for IoT sensors and embedded devices near the network edge, where energy and area are budgeted. This dissertation's primary goal is to design, implement, and evaluate adaptable computation platforms leveraging MRAM-based crossbar arrays and analog computation to support deep learning, error resilience, and trustworthiness of emerging technology applications. Major contributions are development of a workload driven analytical model of SRAM vs. MRAM for edge-of-network applications, development of a spin based PiM architecture for reasoning applications with generalizable activations, development of spin-based progressive redundancy techniques for efficient ANN-based inference applications, and a comprehensive sensitivity analysis of spin based ANNs from a hardware security perspective. Results obtained from functional and Monte Carlo simulations show considerable benefits in terms of area, energy, and resilience metrics evaluated for prominent benchmark datasets that are widely utilized and recognized in the fields of edge computing and IoT devices.

First, this dissertation extended recent research on Spin Torque Transfer MRAM (STT-MRAM) power dissipation, by developing a predictive power estimation model for hybrid CMOS/MTJ technology taking into account IoT energy profiles, determining metric thresholds to justify the emerging device lifecycle energy consumption. The model is developed and validated, with an $R^2>0.95$ coefficient of determination, along with establishing new metrics Mean Standby Duration (MSD), Mean Active Duration (MAD), and Power Dissipation Scaling Ratio (PDSR). Thresholds of MSD>0.995 and MAD<0.005 were determined to be inflection points for lifetime energy justification for considering MTJ devices in terms of total power. Results substantiate a transportable approach for the inclusion of emerging logic devices by considering the energy profile of some intermittently powered applications by parameterizing the workload using the metrics defined,

Next, the dissertation introduced the Spintronically Configurable Analog Processing in-memory Environment (SCAPE) architecture, which integrates hybrid analog/digital arithmetic, runtime reconfigurability of neuron activation function, and non-volatile devices within a selectable 2-D topology to implement different neural network components for deep belief networks to achieve machine learning inference on resource constrained embedded systems. An innovative GAAF based on spin-configurable activation function computes more expressive activation functions intrinsically in analog as per the target application and dataset. Simulation results show significant improvement in error rates, power consumption, and the power-error-product metric for real-world applications, including compressive sensing and machine learning at the network edge, along with process variation analysis. Results reflect that power consumption and error rate for MNIST dataset using sigmoidal square root activation of proposed GAAF based neuron shows up to 7% accuracy improvement versus baseline conventional sigmoidal activation at a comparable power dissipation. Realization of AMP signal processing algorithm show ~95% reduction in energy consumption at comparable accuracy.

Additionally, the dissertation explored incrementally applied redundancy techniques for more robust implementations of emerging spin-based PiM, for an ANN-based digit recognition use case. Results indicate that propsed progressive temporal modular redundancy, applied as required, can have lower footprint and reduced energy consumption at comparable or slightly reduced accuracy than more complex neural networks. This provides an alternative to binarization and other model compression options for intelligence at the edge of the network. The proposed Progressive Temporal Modular Redundancy approach using varied activations implemented on a 784×100×10 network shows a 3% improvement in accuracy compared to the baseline case of 784×500×500×10 network with sigmoidal activation, at 86.1% and 87% reduction in power and weighted crossbar normalized area overheads, respectively, and 87.5% reduction in power error product (PEP) at the cost of ~2.6x increased throughput latency.

Impact and mitigation approaches for malicious manufacturing interventions affecting emerging memristive device-based accelerators are also discussed from a hardware security perspective. Experimental analysis indicates ML recognition outputs can be significantly swayed via a global modification of oxide thickness (T_{ox}) resulting in bit-flips in the weight matrix of the crossbar array, thus corrupting the recognition of selected digits in MNIST dataset differentially, creating an opportunity for an adversary. With just 0.05% of bits in crossbar having a flipped resistance state, digits ‘4’ and ‘5’ show highest overall error rates and digit ‘9’ exhibit the lowest impact, with recognition accuracy of digits ‘2’, ‘3’, and ‘8’ unaffected by changing the oxide thickness of SOT-MTJs uniformly from 0.75 nm to 1.2 nm without modifying the netlist nor even having access to the circuit design itself.

Finally, the dissertation discussed some implementations for CMOS-based reconfigurable computing frameworks targeting AI/ML applications using state-of-the-art FPGAs.

8.2 Key Technical Insights from this Research

Below is a list of some of the key technical insights gathered during the course of this doctoral research:

- Emerging technology STT-MRAM demonstrates near-zero leakage power dissipation, in contrast to its more traditional counterpart, SRAM, which experiences substantial power dissipation during idle phases, primarily at scaled technology nodes. In addition, they provide area-efficiency via vertical integration, low read access time, and backend compatibility with existing CMOS fabrication processes. These attributes of STT-MRAM make them extremely suitable for intermittently powered IoT devices.
- Prior to implementing emerging technologies in any application, the lifespan activity profile of the target application should be used to determine the most important design considerations, particularly for embedded systems with limited power. To illustrate, despite the other inherent advantages they provide over conventional memory cells, STT-MRAMs suffer from asymmetric write power. This suggests that the excessive write power could potentially restrict their applicability to devices and applications that require frequent memory write operations. Thus, it is imperative that any predictive power estimation model for emerging technologies incorporates the appropriate factors.

- The reduction of the data-transfer bottleneck in Von-Neumann designs can provide substantial benefits to several application classes that can benefit from ML/NN implementations when hybrid spin-based computing architectures are utilized for in-memory processing. Several ML/NN functions can be better implemented via a combination of analog and digital computing, rather than only relying on digital blocks, due to the energy advantages offered by analog computations.
- Incorporating reconfigurable emerging technology capabilities into analog arithmetic computations can lead to generalized, application specific, tunable NN functionalities within in-memory computing paradigm, e.g., neuron activations, resulting in improved accuracy and/or energy utilization.
- Reliability and security aspects of spin based ANNs are expected to be an active area of research in the upcoming years, as these devices are becoming increasingly mainstream via commercialization as per the IRDS Beyond CMOS Roadmap.

8.3 Future Investigations

Subsequent investigations that may be undertaken in light of this dissertation center around a few primary research trajectories.

The execution mechanism for emerging technology designs in adaptive computing architectures for AI/ML requires additional software support with a focus on edge-of-network devices, particularly for software applications that utilize afaptive technology-based architectures, such as SCAPE. Therefore, it is necessary to develop appropriate ISA such as those outlined in Chapter 4, along with incorporating additional logic circuitry, processing elements, and datapath design, to enable the hardware co-designed with software support to effectively harness the speedup benefits offered by in-memory computing techniques and accelerator designs in hardware.

Furthermore, in order to enhance the energy efficiency of computations in the redundancy techniques described in Chapter 5, and to increase the resilience of DNN inference implementations to the inherent stochasticity of spin devices, the PTMR approach may be optimized further by capitalizing on potential opportunities for inference priming among the outputs of the intermediate layers. This can be done by sourcing outputs from the penultimate layer of the NN and halting execution of subsequent progressive evaluations of PTMR if the output is same as that of the previous iteration evaluation. This can result in further energy and latency improvements.

An additional avenue worthy of investigation based on this study is to secure the future semiconductor industry from manufacturing threats to the sensitivity of critical device parameters. This could be achieved by adopting methodologies akin to those described in [139], wherein the authors propose a dynamic task remapping technique, more precisely a built-in self-test (BIST) method for fault detection that uses crossbar fault density as guidance for the dynamic remapping process. Rearranging tasks with lower fault tolerance from crossbars with high fault density to those with lower fault is accomplished with minimal accuracy loss when training VGGs, ResNets, and SqueezeNet from inception using the ReRAM crossbar. Beyond the simulation results showed our work, there is an aspect called the actual physical model of these maliciously modified MTJ devices. The fabricated MTJ device may demonstrate more vulnerabilities than the current simulation results. So, the future goal could be to physically fabricate a maliciously modified MTJ device to differentially-execute operations and with the findings, the aim will be to secure it from such manufacturing threats to the sensitivity of critical device parameters.

Furthermore, formal verification has been an integral component within the synchronous IC design flow for the past three decades, predominantly after the Intel FDIV bug went undetected through extensive testing [157], [158]. Recently, even in the asynchronous domain, formal verification has been widely explored to make the domain more mainstream [159], [160], [161], [162], [163], [164], [165]. In the context of in-memory computing, especially for NN computing implementations, the mapping process, such as programming weights to the devices and/or configuring the neuron activation functions in the crossbar, can undergo variations that can significantly degrade the accuracy of the overall system. For the widespread implementation and commercial adoption of emerging technology based in-memory computing platforms, it is crucial to investigate the formal verification of such architectures to ensure the reliability and acceptability of the paradigm. Over the past few years, some efforts have been made to address this limitation [166], [167]. However, there are scopes for further advancements in this direction.

Fig. 30 gives a timeline of the resulted publications in this dissertation.

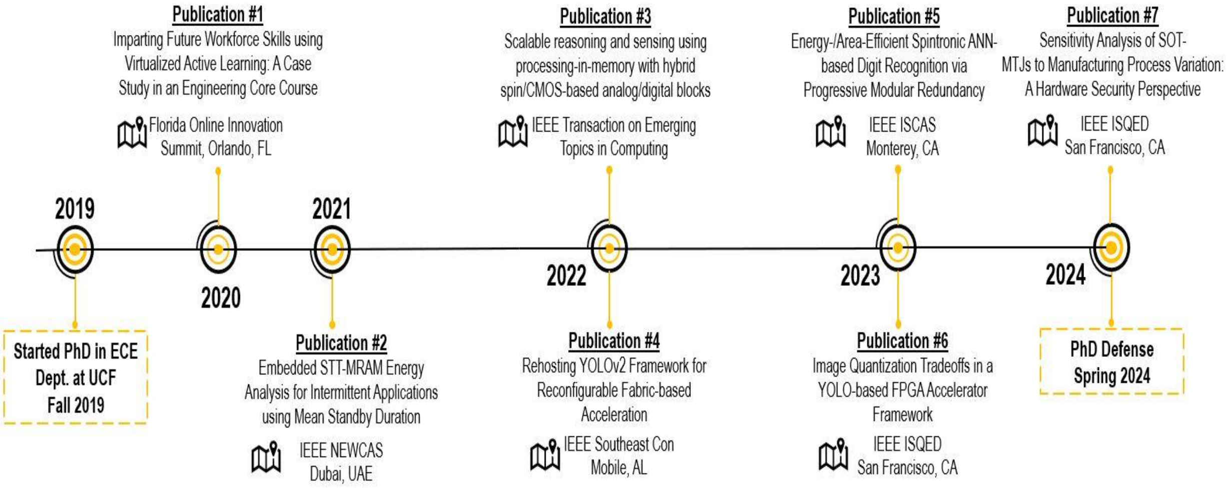
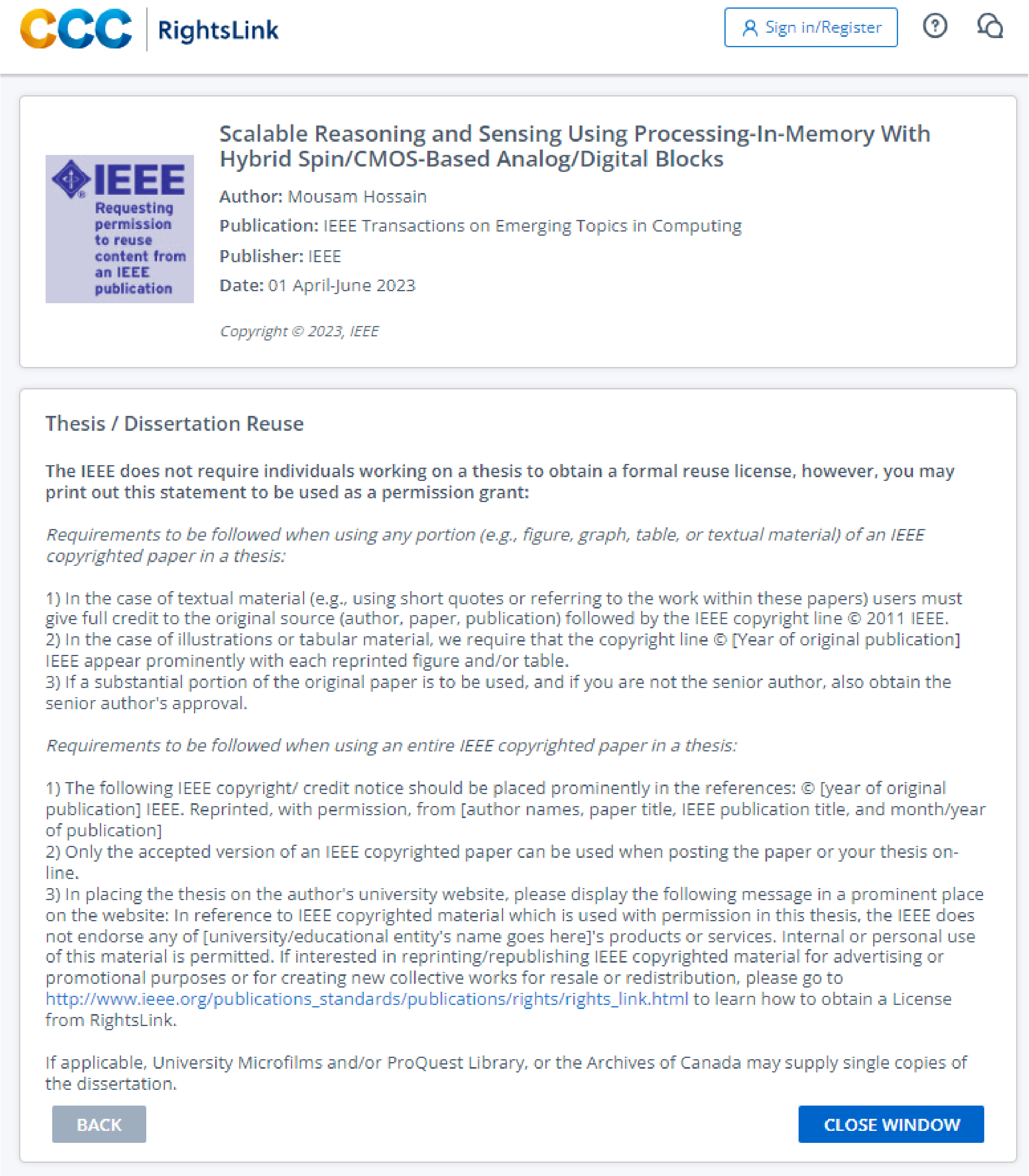

Publication #1
Imparting Future Workforce Skills using Virtualized Active Learning: A Case Study in an Engineering Core Course
Florida Online Innovation Summit, Orlando, FL
Publication #3
Scalable reasoning and sensing using processing-in-memory with hybrid spin/CMOS-based analog/digital blocks
IEEE Transaction on Emerging Topics in Computing
Publication #5
Energy-/Area-Efficient Spintronic ANN-based Digit Recognition via Progressive Modular Redundancy
IEEE ISCAS Monterey, CA
Publication #7
Sensitivity Analysis of SOT-MTJs to Manufacturing Process Variation: A Hardware Security Perspective
IEEE ISQED San Francisco, CA
2019
2020
2021
2022
2023
2024
Started PhD in ECE Dept. at UCF Fall 2019
Publication #2
Embedded STT-MRAM Energy Analysis for Intermittent Applications using Mean Standby Duration
IEEE NEWCAS Dubai, UAE
Publication #4
Rehosting YOLOv2 Framework for Reconfigurable Fabric-based Acceleration
IEEE Southeast Con Mobile, AL
Publication #6
Image Quantization Tradeoffs in a YOLO-based FPGA Accelerator Framework
IEEE ISQED San Francisco, CA
PhD Defense Spring 2024

Figure 30: Dissertation Progress and Selected Publications Timeline

APPENDIX: COPYRIGHT PERMISSIONS


CCC RightsLink
Sign in/Register
IEEE
Requesting permission to reuse content from an IEEE publication

Scalable Reasoning and Sensing Using Processing-In-Memory With Hybrid Spin/CMOS-Based Analog/Digital Blocks

Author: Mousam Hossain

Publication: IEEE Transactions on Emerging Topics in Computing

Publisher: IEEE

Date: 01 April-June 2023

Copyright © 2023, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

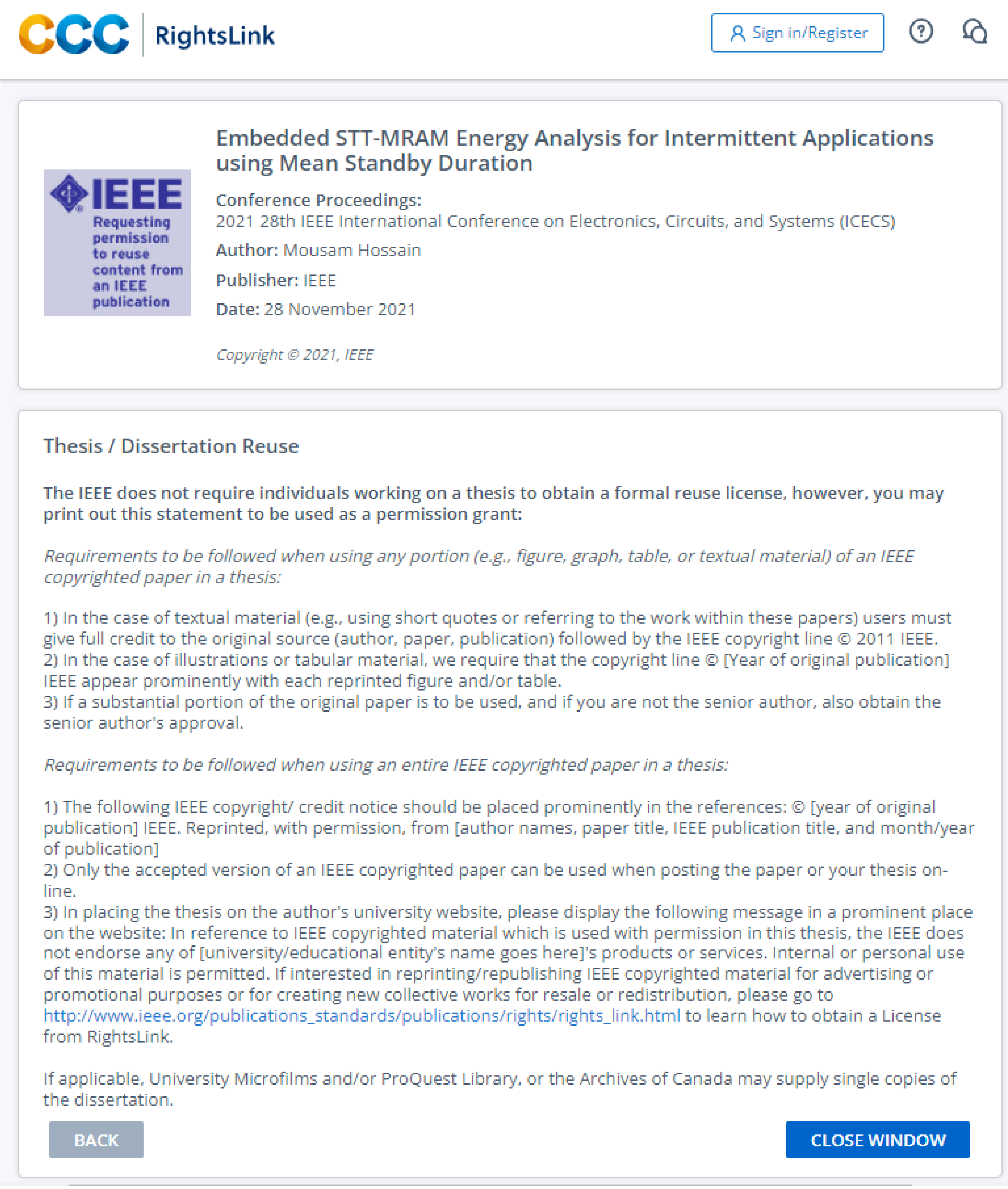
Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.


CCC
RightsLink
Sign in/Register
IEEE
Requesting permission to reuse content from an IEEE publication
BACK
CLOSE WINDOW

Embedded STT-MRAM Energy Analysis for Intermittent Applications using Mean Standby Duration

Conference Proceedings:
2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)

Author: Mousam Hossain

Publisher: IEEE

Date: 28 November 2021

Copyright © 2021, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

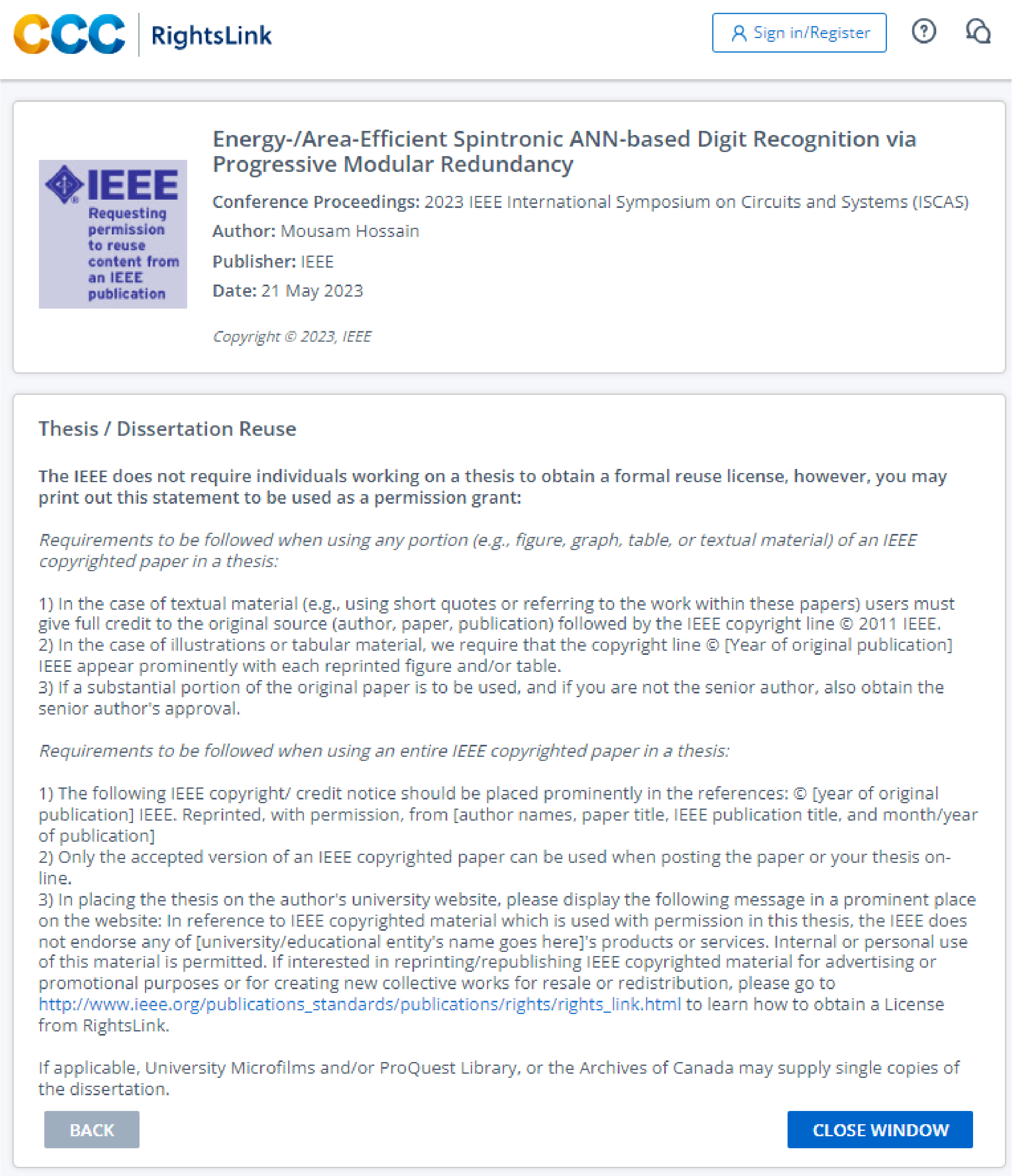
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

]


CCC
RightsLink
Sign in/Register
IEEE
Requesting permission to reuse content from an IEEE publication

Energy-/Area-Efficient Spintronic ANN-based Digit Recognition via Progressive Modular Redundancy

Conference Proceedings: 2023 IEEE International Symposium on Circuits and Systems (ISCAS)

Author: Mousam Hossain

Publisher: IEEE

Date: 21 May 2023

Copyright © 2023, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

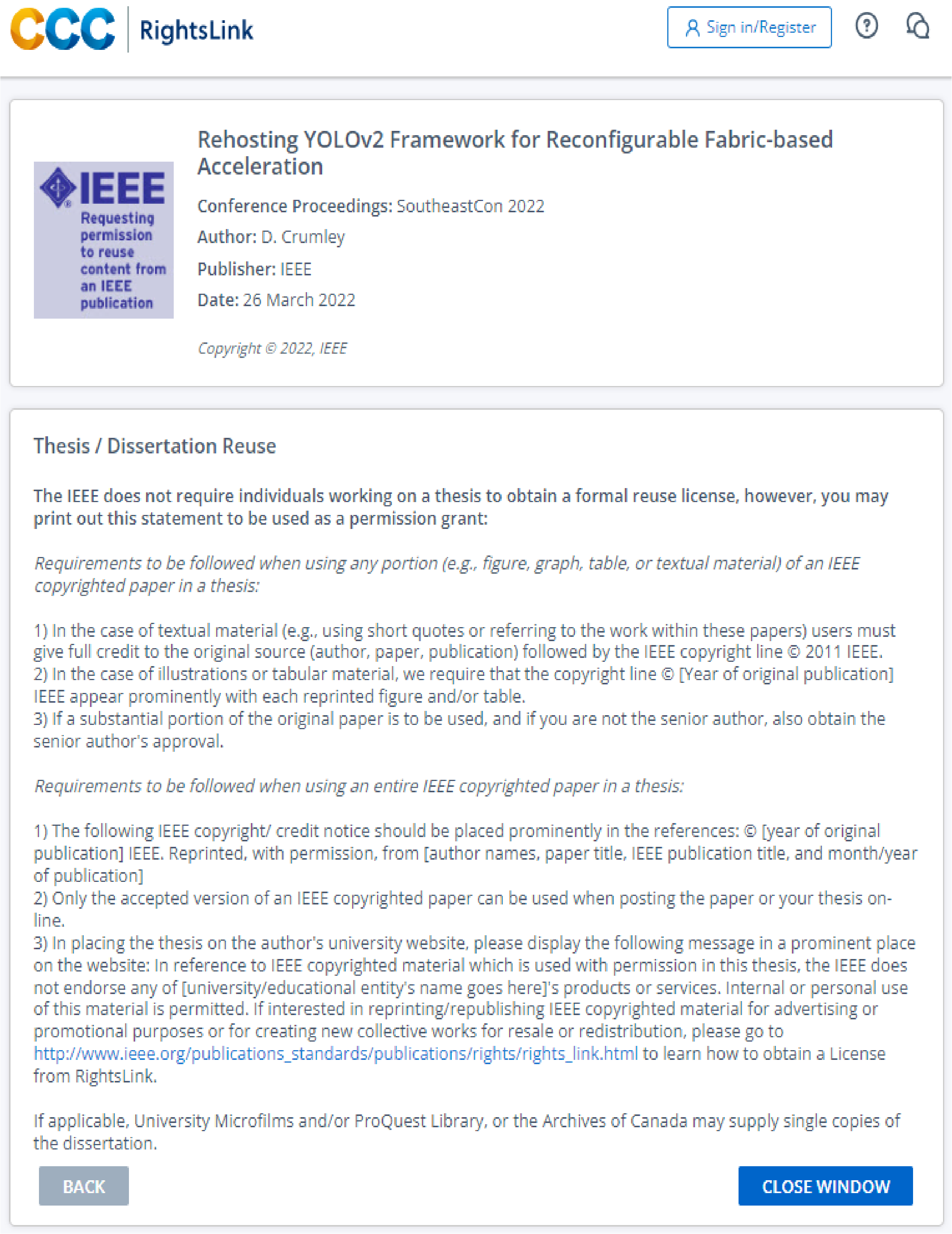
Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.


RightsLink
Sign in/Register
IEEE
Requesting permission to reuse content from an IEEE publication

Rehosting YOLOv2 Framework for Reconfigurable Fabric-based Acceleration

Conference Proceedings: SoutheastCon 2022
Author: D. Crumley
Publisher: IEEE
Date: 26 March 2022

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

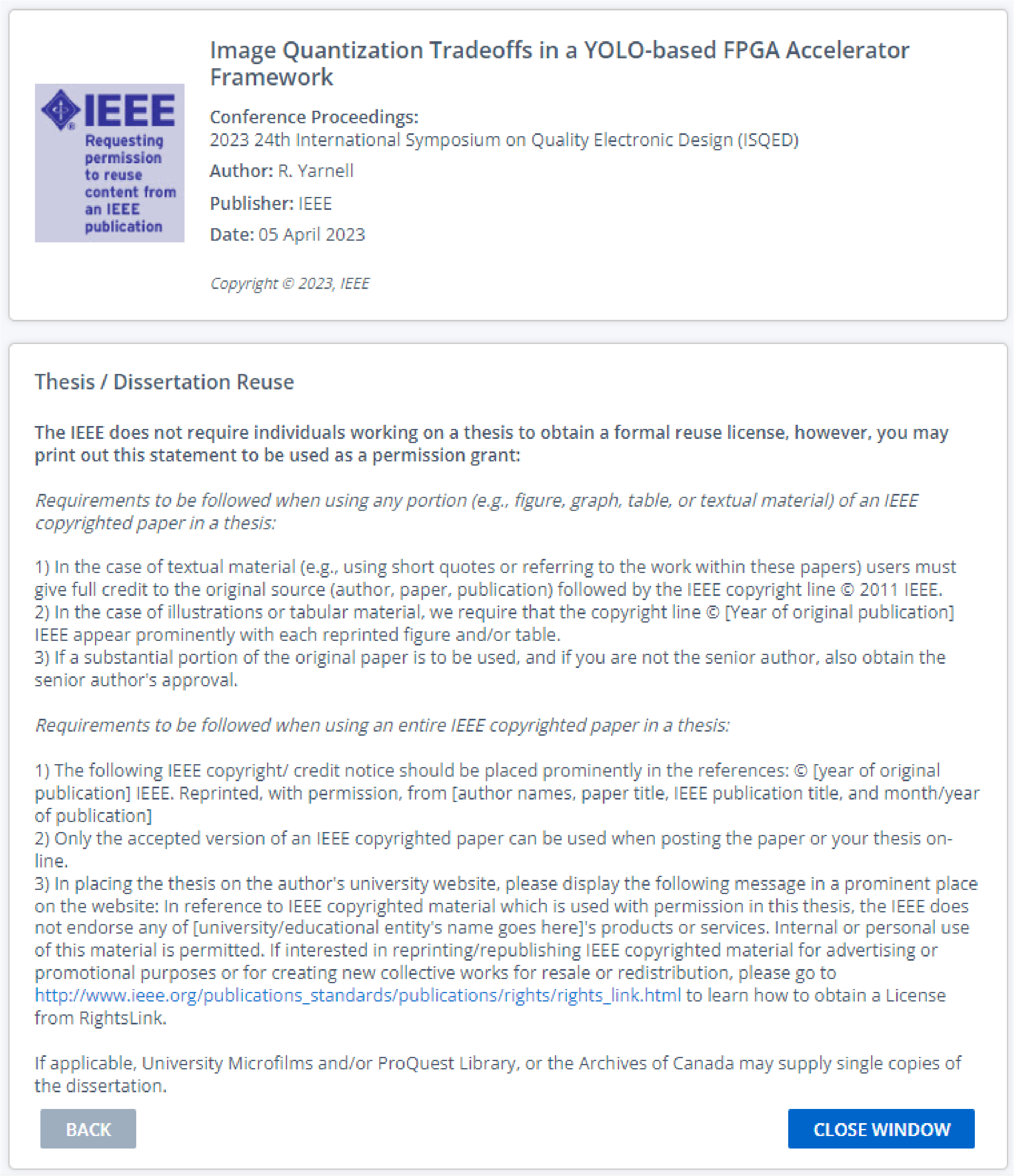
Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.



IEEE
Requesting permission to reuse content from an IEEE publication

Image Quantization Tradeoffs in a YOLO-based FPGA Accelerator Framework

Conference Proceedings:
2023 24th International Symposium on Quality Electronic Design (ISQED)

Author: R. Yarnell

Publisher: IEEE

Date: 05 April 2023

Copyright © 2023, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

LIST OF REFERENCES

[1] M. Hossain, A. Tatulian, S. Sheikhfaal, H. R. Thummala and R. F. DeMara, "Scalable Reasoning and Sensing Using Processing-In-Memory With Hybrid Spin/CMOS-Based Analog/Digital Blocks," in *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 2, pp. 343-357, Oct. 2022, doi: 10.1109/TETC.2022.3212341.

[2] M. Hossain, S. Salehi, D. Mulvaney and R. DeMara, "Embedded STT-MRAM Energy Analysis for Intermittent Applications using Mean Standby Duration," in *Proc. IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, Dubai, United Arab Emirates, 2021, pp. 1-6, doi: 10.1109/ICECS53924.2021.9665581.

[3] M. Hossain, A. Tatulian, H. R. Thummala, R. F. DeMara and S. Salehi, "Energy-/Area-Efficient Spintronic ANN-based Digit Recognition via Progressive Modular Redundancy," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Monterey, CA, USA, May 2023, pp. 1-5, doi: 10.1109/ISCAS46773.2023.10181529.

[4] M. Hossain, M. A. Chowdhury, R. F. DeMara, and S. Salehi, “Sensitivity Analysis of SOT-MTJs to Manufacturing Process Variation: A Hardware Security Perspective,” in *Proc. IEEE International Symposium Quality Electronic Design*, (ISQED’24), San Francisco, CA, April 3-5, 2024.

[5] S. Das, A. Chen and M. Marinella, "Beyond CMOS," 2021 IEEE International Roadmap for Devices and Systems Outbriefs, Santa Clara, CA, USA, 2021, pp. 1-129, doi: 10.1109/IRDS54852.2021.00011.

[6] E. Y. Vedmedenko, R. K. Kawakami, D. D. Sheka, P. Gambardella, A. Kirilyuk, A. Hirohata, C. Binek, O. Chubykalo-Fesenko, S. Sanvito, B. J. Kirby, and J. Grollier, “The 2020 magnetism roadmap,” Journal of Physics D: Applied Physics, vol. 53, pp.453001. doi: 10.1088/1361-6463/ab9d98.

[7] L. Wei, J. G. Alzate, U. Arslan, J. Brockman, N. Das, K. Fischer, T. Ghani, O. Golonzka, P. Hentges, R. Jahan, and P. Jain, “13.3 A 7Mb STTMRAM in 22FFL FinFET technology with 4ns read sensing time at 0.9V using Write-Verify-Write scheme and Offset-Cancellation sensing technique,” in *IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, pp. 214–216, Feb., 2019.

[8] A. Bansal, S. Mukhopadhyay and K. Roy, "Device-Optimization Technique for Robust and Low-Power FinFET SRAM Design in NanoScale Era," in IEEE Transactions on Electron Devices, vol. 54, no. 6, pp. 1409-1419, June 2007, doi: 10.1109/TED.2007.895879.

[9] D. Bhattacharya and N.K. Jha, “FinFETs: From Devices to Architectures”, in *Advances in Electronics*, 2014.

[10] P. Prakash, M. K. Sundaram, and M. A. Bennet, "A Review on Carbon Nanotube Field Effect Transistors (CNTFETs) for Ultra-low Power Applications," in *Renewable and Sustainable Energy Reviews* 89, pp. 194-203, 2018.

[11] B. Srinivasu and K. Sridharan, "Low-Power and High-Performance Ternary SRAM Designs with Application to CNTFET Technology," in *IEEE Transactions on Nanotechnology*, vol. 20, pp. 562-566, 2021, doi: 10.1109/TNANO.2021.3096123.

[12] A. A. Sakib, A. A. Akib and S. C. Smith, "Implementation of FinFET Based Static NCL Threshold Gates: An Analysis of Design Choice," in *IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Springfield, MA, USA, 2020, pp. 37-40, doi: 10.1109/MWSCAS48704.2020.9184629.

[13] D. Khodosevych and A. A. Sakib, "Evolution of NULL Convention Logic Based Asynchronous Paradigm: An Overview and Outlook," in *IEEE Access*, vol. 10, pp. 78650-78666, 2022, doi: 10.1109/ACCESS.2022.3194028.

[14] A. A. Sakib and S. C. Smith, "Implementation of Static NCL Threshold Gates using Emerging CNTFET Technology," *in 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Glasgow, UK, 2020, pp. 1-4, doi: 10.1109/ICECS49266.2020.9294823.

[15] J. Kao, S. Narendra, and A. Chandrakasan, "Subthreshold Leakage Modeling and Reduction Techniques," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2002, pp. 141–148.

[16] W. Zhao, and Y. Cao, "Predictive Technology Model", 2008. [Online]. Available: http://ptm.asu.edu.

[17] W. Wang, and M. Zhang, "Tensor deep learning model for heterogeneous data fusion in internet of things", in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 1, pp.32-41, 2018.

[18] S. Wen, H. Wei, Z. Zeng, and T. Huang, "Memristive fully convolutional network: an accurate hardware image segmentor in deep learning", in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 5, pp.324-334, 2018.

[19] Y. Kunpeng, H. Shan, T. Sun, R. Hu, Y. Wu, L. Yu, Z. Zhang, and T. Quek, “Reinforcement learning-based mobile edge computing and transmission scheduling for video surveillance”, in *IEEE Transactions on Emerging Topics in Computing*, doi: 10.1109/TETC.2021.3073744.

[20] M. Taghavi, and M. Shoaran, “Hardware complexity analysis of deep neural networks and decision tree ensembles for real-time neural data classification”, in *Proc. IEEE International Conference on Neural Engineering. (NER)*, pp. 407-410, Mar. 2019.

[21] K. Roy, I. Chakraborty, M. Ali, A. Ankit and A. Agrawal, "In-Memory Computing in Emerging Memory Technologies for Machine Learning: An Overview," in *Proc. ACM/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, 2020, pp. 1-6, doi: 10.1109/DAC18072.2020.9218505.

[22] K. Mishty and M. Sadi, "Designing Efficient and High-Performance AI Accelerators with Customized STT-MRAM," in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 29, no. 10, pp. 1730-1742, Oct. 2021, doi: 10.1109/TVLSI.2021.3105958.

[23] H. Salmani, “The Global Integrated Circuit Supply Chain Flow and the Hardware Trojan Attack,” in *Trusted Digital Circuits, Springer*, doi: /10.1007/978-3-319-79081-7_1.

[24] W. Hu, C. H. Chang, A. Sengupta, S. Bhunia, R. Kastner and H. Li, “An Overview of Hardware Security and Trust: Threats, Countermeasures, and Design Tools,” in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 6, pp. 1010-1038, Jun. 2021, doi: 10.1109/TCAD.2020.3047976.

[25] G. Kolhe, T. Sheaves, K. I. Gubbi, T. Kadale, S. Rafatirad, S. M. PD, A. Sasan, H. Mahmoodi, and H. Homayoun, “Silicon Validation of LUT-based Logic-Locked IP Cores,” in *Proc.*

ACM/IEEE Design Automation Conference, Jul. 2022, pp. 1189–1194, doi: /10.1145/3489517.3530606.

[26] K. I. Gubbi, B. S. Latibari, A. Srikanth, T. Sheaves, S. A. Beheshti-Shirazi, S. M. Pd, S. Rafatirad, A. Sasan, H. Homayoun, and S. Salehi, “Hardware Trojan Detection Using Machine Learning: A Tutorial,” in *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 46, pp 1-26, 2023, doi: /10.1145/3579823.

[27] T. Bryant, Y. Chen, D. S. Koblah, D. Forte, and N. Maghari, “A Brief Tutorial on Mixed Signal Approaches to Combat Electronic Counterfeiting,” in *IEEE Open Journal of Circuits and Systems*, vol. 4, pp. 99-114, Mar. 2023, doi: 10.1109/OJCAS.2023.3253144.

[28] G. Kolhe, T. Sheaves, K. I. Gubbi, S. Salehi, S. Rafatirad, S. M. PD, A. Sasan, and H. Homayoun, “LOCK&ROLL: Deep-Learning Power Side channel Attack Mitigation Using Emerging Reconfigurable Devices and Logic Locking,” in *Proc. ACM/IEEE Design Automation Conference*, Jul. 2022, pp. 85–90, doi: /10.1145/3489517.3530414.

[29] R. Zand, "Heterogeneous Reconfigurable Fabrics for In-circuit Training and Evaluation of Neuromorphic Architectures," *Doctoral Dissertation*., University of Central Florida, 2019.

[30] S. Bandyopadhyay and M. Cahay, “Introduction to Spintronics,” New York, NY, USA: Taylor & Francis, 2008.

[31] M. Julliere, ‘’Tunneling between ferromagnetic films,’’ in *Physics Letters A*, vol. 54, no. 3, pp. 225–226, Sep. 1975.

[32] V.K. Joshi, P. Barla, S. Bhat, and B.K. Kaushik, “From MTJ device to hybrid CMOS/MTJ circuits: A review”, in *IEEE Access*, vol. 8, pp.194105-194146, 2020.

[33] S. Salehi and R. F. DeMara, "SLIM-ADC: Spin-based logic-in-memory analog to digital converter leveraging she-enabled domain wall motion devices," *Microelectronics Journal*, vol. 81, pp. 137-143, 2018.

[34] S. S. P. Parkin, R. E. Fontana, and A. C. Marley, "Low-field magnetoresistance in magnetic tunnel junctions prepared by contact masks and lithography: 25% magnetoresistance at 295 K in mega-ohm micron-sized junctions." *Journal of Applied Physics*, vol. 81, no. 8, p. 5521, 1997.

[35] S. A. Wolf, D. D. Awschalom, R. A. Buhrman, J. M. Daughton, S. von Moln´ar, M. L. Roukes, A. Y. Chtchelkanova, and D. M. Treger, “Spintronics: A spin-based electronics vision for the future,” *Science*, vol. 294, no. 5546, pp. 1488–1495, 2001. [Online]. Available: http://science.sciencemag.org/content/294/5546/1488.

[36] I. L. Prejbeanu, M. Kerekes, R. C. Sousa, H. Sibuet, O. Redon, B. Dieny, and J. P. Nozires, “Thermally assisted MRAM,” *Journal of Physics: Condensed Matter*, vol. 19, no. 16, p. 165218, 2007. [Online]. Available: http://stacks.iop.org/0953-8984/19/i=16/a=165218.

[37] J. Slonczewski, “Current-driven excitation of magnetic multilayers,” *Journal of Magnetism and Magnetic Materials*, vol. 159, no. 1, pp. L1 – L7, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304885396000625.

[38] L. Liu, T. Moriyama, D. C. Ralph, and R. A. Buhrman, “Reduction of the spin-torque critical current by partially canceling the free layer demagnetization field,” *Applied Physics Letters*, vol. 94, no. 12, p. 122508, 2009. [Online]. Available: https://doi.org/10.1063/1.3107262.

[39] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno, “A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction,” in *Nature materials*, vol. 9, no. 9, p. 721, 2010.

[40] R. H. Koch, J. A. Katine, and J. Z. Sun, “Time-resolved reversal of spin-transfer switching in a nanomagnet,” Phys. Rev. Lett., vol. 92, p. 088302, Feb 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.92.088302.

[41] K. Y. Camsari, S. Salahuddin, and S. Datta, "Implementing p-bits with embedded MTJ," *IEEE Electron Device Letters*, vol. 38, no. 12, pp. 1767-1770, 2017.

[42] S. Datta, "p-Bits for probabilistic computing," in *Proc. Device Research Conference (DRC)*, pp. 35-36, IEEE, 2019.

[43] R. Zand, K. Y. Camsari, S. Datta, and R. F. DeMara, "Composable probabilistic inference networks using MRAM-based stochastic neurons," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 15, no. 2, pp. 1-22, 2019.

[44] M. Mamidipaka, K. Khouri, N. Dutta, and M. Abadir, “Leakage power estimation in SRAMs,” in *Technical Report Power*, University of California, Irvine, CA, USA. CECS, pp. 03-32, Oct. 2003. [Online]. Available: https://www.ics.uci.edu/~maheshmn/Pubs/TR03-32.pdf.

[45] 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 Solid-State Circuits*, vol. 48, no. 2, pp. 598–610, 2013.

[46] S. Salehi, and R.F. DeMara, “Process Variation immune and energy aware sense amplifiers for resistive non-volatile memories,” in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4, 2017, doi: 10.1109/ISCAS.2017.8050788.

[47] D. Younis, N. Madathumpadical, and H. Al-Nashash, “Modeling and simulation of static power dissipation in CMOS with SELBOX structure,” in *Proc. ICMSAO*, Sharjah, UAE, pp. 1–4, 2017.

[48] D. McGrath, “Intel says FinFET-based Embedded MRAM is production ready”, Feb. 2019. [Online]. Available: https://www.eetimes.com/intelsays-finfet-based-embedded-mram-is-production-ready.

[49] A. Jaiswal, X. Fong, and K. Roy, “Comprehensive scaling analysis of current induced switching in magnetic memories based on in-plane and perpendicular anisotropies,” in *IEEE Journal Emerging Selected Topics on Circuits and Systems*, vol. 6, no. 2, pp. 120–133, Jun., 2016.

[50] W. Zhao, S. Chaudhuri, C. Accoto, J. O. Klein, C. Chappert, and P. Mazoyer, “Cross-point architecture for spin-transfer torque magnetic random-access memory,” *IEEE Transactions on Nanotechnology*, vol. 11, no. 5, pp. 907–917, 2012.

[51] J. A. Butts and G. S. Sohi, “Static power model for architects,” in *Proc. Annual International Symposium on Microarchitecture*, pp. 191–201, 2000.

[52] L. Xiaoyao, K. Turgay, and D. Brooks, "Architectural power models for SRAM and CAM structures based on hybrid analytical/empirical techniques," *IEEE/ACM International Conference on Computer Aided Design (ICCAD),* pp. 824–830, 2007.

[53] A. Stillmaker and B. Baas, "Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm," *Integration*, vol. 58, no. May 2016, pp. 74–81, 2017.

[54] C. W. Smullen, A. Nigam, S. Gurumurthi, and M. R. Stan, "The STeTSiMS STT-RAM simulation and modeling system," in *Proc. IEEE/ACM International Conference Computer Aided Design (ICCAD)*, no. November, pp. 318–325, 2011.

[55] S. Togashi, T. Ohsawa, and T. Endoh, "Nonvolatile Low Power 16-bit/32-bit Magnetic Tunnel Junction Based Binary Counter and Its Scaling," in *Japanese Journal of Applied Physics*, vol. 51, no. 2, p. 02BE07, 2012.

[56] E. Garzón 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, Mar. 2020.

[57] 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, Dec., 2014. DOI: 10.1016/j.sse.2014.06.003.

[58] R. Patel, E. Ipek, and E.G. Friedman, "2T–1R STT-MRAM memory cells for enhanced on/off current ratio," *Microelectronics Journal*, vol. 45, no. 2, pp.133-143, Feb., 2014. DOI: 10.1016/j.mejo.2013.11.015.

[59] H. Kawasaki et al., “Embedded bulk FinFET SRAM cell technology with planar FET peripheral circuit for hp32 nm node and beyond,” in *Proc. Symposium on VLSI Technology*, Honolulu, HI, USA, 2006, vol. 48, no. 9, pp. 70–71.

[60] S.K. Gupta, S.P. Park, N.N. Mojumder and K. Roy, “Layout-aware optimization of STT MRAMs,” in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, Germany, 2012, pp. 1455-1458.

[61] X. Fong, Y. Kim, R. Venkatesan, S.H. Choday, A. Raghunathan, and K. Roy. “Spin-transfer torque memories: Devices, circuits, and systems,” in *Proceedings of the IEEE*, vol. 104, no. 7, pp. 1449-1488, Jul., 2016. DOI: 10.1109/JPROC.2016.2521712.

[62] E.I. Vatajelu, P. Prinetto, M. Taouil and S. Hamdioui, “Challenges and Solutions in Emerging Memory Testing,” in *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 3, pp. 493-506, Jul.-Sept., 2019. DOI: 10.1109/TETC.2017.2691263.

[63] S. Salehi, M.B. Mashhadi, A. Zaeemzadeh, N. Rahnavard, and R.F. DeMara, “Energy-aware adaptive rate and resolution sampling of spectrally sparse signals leveraging VCMA-MTJ devices,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 4, pp. 679–692, Dec. 2018.

[64] S. Senni, L. Torres, G. Sassatelli, and A. Gamatié, “Non-Volatile processor based on MRAM for ultra-low-power IoT Devices,” *ACM Journal on Emerging Technologies in Computing Systems*, Association for Computing Machinery, vol. 13, no. 2, pp.1-23, Dec, 2016.

[65] F. Goriawalla, “Advantages of embedded MTP non-volatile memory for IoT SoC designs,” NVM Product Line Manager, Synopsys , DesignWare Technical Bulletin. [Online]. Available: https://www.synopsys.com/designware-ip/technical-bulletin/advantagesof-mtv.html.

[66] A. Shukla, S. Chaturvedi, and Y. Simmhan, “RIoTBench: an IoT benchmark for distributed stream processing systems,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 21, pp. E4257, Nov., 2017. DOI: 10.1002/cpe.4257.

[67] R. A. Ashraf and R. F. DeMara, "Scalable FPGA Refurbishment Using Netlist-Driven Evolutionary Algorithms," *IEEE Transactions on Computers*, vol. 62, no. 8, pp. 1526-1541, Aug. 2013, doi: 10.1109/TC.2013.58.

[68] R. S. Oreifej, C. A. Sharma and R. F. DeMara, "Expediting GA-Based Evolution Using Group Testing Techniques for Reconfigurable Hardware," in *Proceedings of IEEE International Conference on Reconfigurable Computing and FPGA*'s, 2006, pp. 1-8, doi: 10.1109/RECONF.2006.307760.

[69] N. Imran, R. F. DeMara, J. Lee, et al, "Self-Adapting Resource Escalation for Resilient Signal Processing Architectures," *Journal of Signal Processing Systems*, vol. 77, 2014, pp. 257–280, doi: https://doi.org/10.1007/s11265-013-0811-x.

[70] G.H. Barnes, R.M. Brown, M. Kato, D.J. Kuck, D.L. Slotnick, and R.A. Stokes, “The ILLIAC IV Computer,” *IEEE Transactions on Computers*, 100(8), pp.746-757, 1968.

[71] C.C. Foster, “Content Addressable Parallel Processors”, John Wiley & Sons, Inc., 1976.

[72] R.F. DeMara, and D.I. Moldovan, “The SNAP-1 Parallel AI Prototype,” *IEEE Trans. on Parallel and Distributed Systems*, 4(8), pp.841-854, 1993.

[73] D. Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Thomas, R. and K. Yelick, “A case for Intelligent RAM,” *IEEE Micro*, 17(2), pp.34-44, 1997.

[74] D.G. Elliott, M. Stumm, W.M. Snelgrove, C. Cojocaru, and R. McKenzie, “Computational RAM: implementing processors in memory,” *IEEE Design & Test of Computers*, 16(1), pp.32-41, 1999.

[75] F. Alibart, E. Zamanidoost, and D. Strukov, “Pattern classification by memristive crossbar circuits using ex situ and in situ training,” *Nature Communications*, vol. 4, no. 2073,2013. DOI: doi.org/10.1038/ncomms3072.

[76] H. Zhang, W. Kang, K. Cao, B. Wu, Y. Zhang, & W. Zhao, “Spintronic processing unit in spin transfer torque magnetic random-access memory,” in *IEEE Trans. on Electron Devices*, vol. 66, part 4, pp. 2017-2022, 2019.

[77] H. Pourmeidani, S. Sheikhfaal, R. Zand, and R.F. DeMara, “Probabilistic interpolation recoder for energy-error-product efficient DBNs with p-bit devices,” *IEEE Transaction on Emerging Topics in Computing*, vol. 9 no. 4, pp. 2146-2157, 2020.

[78] A. Tatulian, and R.F. DeMara, “Generalized exponentiation using STT magnetic tunnel junctions: circuit design, performance, and application to neural network gradient decay”, *SN Computer Science*, 3(2), pp.1-14.

[79] S. Sheikhfaal, M. R. Vangala, A. Adepegba and R. F. DeMara, "Long short-term memory with spin-based binary and non-binary neurons," in *Proc. IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2021, pp. 317-320, doi: 10.1109/MWSCAS47672.2021.9531773.

[80] P. Chi, S. Li, Y. Cheng, Y. Lu, S. H. Kang and Y. Xie, "Architecture design with STT-RAM: opportunities and challenges," in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 109-114, 2016, doi: 10.1109/ASPDAC.2016.7427997.

[81] S. Sheikhfaal and R. F. Demara, "Short-Term Long-Term Compute-in-Memory Architecture: A Hybrid Spin/CMOS Approach Supporting Intrinsic Consolidation," in *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 6, no. 1, pp. 62-70, Jun. 2020, doi: 10.1109/JXCDC.2020.2983450.

[82] Shadi Sheikhfaal, "Energy-Efficient In-Memory Architectures Leveraging Intrinsic Behaviors of Embedded MRAM Devices," *Doctoral dissertation*, University of Central Florida Orlando, Florida, 2021.

[83] Cilingiroglu, ''A purely capacitive synaptic matrix for fixed-weight neural networks,'' in *IEEE Transactions on Circuits and Systems*, vol. 38, no. 2, pp. 210–217, Feb. 1991.

[84] D. Kwon and I.Y. Chung, ''Capacitive neural network using charge-stored memory cells for pattern recognition applications," in *IEEE Electron Device Letters*, vol. 41, no. 3, pp. 493–496, Mar. 2020.

[85] Z. Wang, M. Rao, J.W. Han, J. Zhang, P. Lin, Y. Li, C. Li, W. Song, S. Asapu, R. Midya, and Y. Zhuo, ‘’Capacitive Neural Network with Neuro-Transistors,’’ in *Nature Communications*, vol. 9, no. 1, pp. 1–10, Dec. 2018.

[86] S. Angizi and D. Fan, “ReDRAM: A Reconfigurable Processing-In-DRAM Platform for Accelerating Bulk Bit-Wise Operations,” in *Proc. IEEE/ACM International Conference on Computer Aided Design*, pp. 1–8, Nov. 2019.

[87] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, ‘’Prime: a Novel Processing-In-Memory Architecture for Neural Network Computation in ReRAM-based Main Memory’’, in *ACM SIG ARCH Computing Architectures News*, vol. 44, no. 3, pp. 27–39, 2016.

[88] Y. Long, E.M. Jung, J. Kung and S. Mukhopadhyay, “Re-RAM crossbar based recurrent neural network for human activity detection”, in *International Joint Conference on Neural Network*s, pp. 939-946, Jul. 2016.

[89] Y. Long, T. Na and S. Mukhopadhyay, “ReRAM-based processing-in-memory architecture for recurrent neural network acceleration,” in *IEEE Transactions on Very Larg- Scale Integration Systems*, vol. 26, no. 12, pp.2781-2794.

[90] A. Tatulian, “Leveraging Signal Transfer Characteristics and Parasitics of Spintronic Circuits for Area and Energy-Optimized Hybrid Digital and Analog Arithmetic,” *Doctoral dissertation*, University of Central Florida Orlando, Florida, 2023.

[91] R.J. D'Angelo and S.R. Sonkusale, “A time-mode translinear principle for nonlinear analog computation,” in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62 no. 9, pp. 2187-2195, 2015. DOI: 10.1109/TCSI.2015.2451912.

[92] V. Seshadri et al., “Gather-scatter DRAM: In-DRAM address translation to improve the spatial locality of non-unit strided accesses,” in *Proc. International Symposium in Microarchitecture*, 2015, pp. 267–280, doi: 10.1145/2830772.2830820.

[93] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, “Pinatubo: A Processing In-Memory Architecture for Bulk Bitwise Operations In Emerging Non-Volatile Memories,” in Proc. Des. Automat. Conf., 2016, Art. no. 173, doi: 10.1145/2897937.2898064.

[94] F. Qian, Y. Gong, G. Huang, M. Anwar, and L. Wang, “Exploiting memristors for compressive sampling of sensory signals,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 12, pp. 2737–2748, Dec. 2018.

[95] L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, “High-speed compressed sensing reconstruction on FPGA using OMP and AMP,” in *Proc. IEEE International Conference on Electronic Circuits and Systems*, 2012, pp. 53–56.

[96] P. Maechler et al., “VLSI design of approximate message passing for signal restoration and compressive sensing,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 569–590, Sep. 2012.

[97] H. Jiang, C. Liu, F. Lombardi, and J. Han, “Low-power approximate unsigned multipliers with configurable error recovery,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 189–202, Jan. 2018, doi: 10.1109/TCSI.2018.2856245.

[98] N. Arya, T. Soni, M. Pattanaik, and G. K. Sharma, “Area and energy efficient approximate square rooters for error resilient applications,” in *Proc. IEEE International Conference on VLSI*

Design, 19th Int. Conf. Embedded Syst., 2020, pp. 90–95, doi: 10.1109/VLSID49098.2020.00033.

[99] M. T. Abuelma'Atti and A. M. Abuelmaatti, "A new current-mode CMOS analog programmable arbitrary nonlinear function synthesizer," *Microelectronics Journal*, vol. 43, no. 11, pp. 802–808, 2012, doi: 10.1016/j.mejo.2012.07.003.

[100] B. R. Fernando, Y. Qi, C. Yakopcic, and T. M. Taha, "3D memristor crossbar architecture for a multicore neuromorphic system," in *Proc. IEEE International Joint Conference Neural Networks*, 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9206929.

[101] K. N. S. Batta and I. Chakrabarti, "VLSI architecture for enhanced approximate message passing algorithm," *IEEE Transaction on Circuits and Systems on Video Technology*, vol. 30, no. 9, pp. 3253–3267, Sep. 2020.

[102] N. D. P. Avirneni and A. Somani, "Low overhead soft error mitigation techniques for high-performance and aggressive designs," *IEEE Transactions on Computers*, vol. 61, no. 4, pp. 488–501, Apr. 2012.

[103] W. Zhao et al., "A radiation hardened hybrid spintronic/CMOS nonvolatile unit using magnetic tunnel junctions," *Journal of Physics D: Applied Physics*, vol. 47, no. 40, Art. no. 405003.

[104] R. A. Ashraf, O. Mouri, R. Jadaa and R. F. Demara, "Design-for-Diversity for Improved Fault-Tolerance of TMR Systems on FPGAs," in Proc. *International Conference on Reconfigurable Computing and FPGAs*, pp. 99-104, Cancun, Mexico, 2011, doi: 10.1109/ReConFig.2011.26.

[105] N. Imran, and R.F. DeMara, “Heterogeneous concurrent error detection (hCED) based on output anticipation,” in *Proc. International Conference on Reconfigurable Computing and FPGAs*, (pp. 61-66), Nov. 2011.

[106] K. Zhang, G. Bedette and R. F. DeMara, "Triple Modular Redundancy with Standby (TMRSB) Supporting Dynamic Resource Reconfiguration," in *Proc. IEEE Autotestcon*, pp. 690-696, 2006, doi: 10.1109/AUTEST.2006.283750.

[107] R. Al-Haddad, R. Oreifej, R.A. Ashraf, and R.F. DeMara, “Sustainable Modular Adaptive Redundancy Technique Emphasizing Partial Reconfiguration for Reduced Power Consumption,” in *International Journal of Reconfigurable Computing*, 2011.

[108] R.S. Oreifej, R.N. Al-Haddad, H. Tan, and R.F. DeMara, “Layered Approach to Intrinsic Evolvable Hardware Using Direct Bitstream Manipulation of Virtex II Pro devices,” in *Proc. IEEE International Conference on Field Programmable Logic and Applications*, pp. 299-304, 2007.

[109] N. Imran, J. Lee, and R.F. DeMara, “Fault Demotion Using Reconfigurable Slack (FaDReS),” in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 21 no. 7, pp.1364-1368, 2012.

[110] F.S. Alghareb, R.A. Ashraf, and R.F. DeMara, “Designing and Evaluating Redundancy-Based Soft-Error Masking on a Continuum of Energy Versus Robustness,” in *IEEE Transactions on Sustainable Computing*, vol. 3 no. 3, pp.139-152, 2017.

[111] M.G. Parris, C.A. Sharma, and R.F. Demara, “Progress in Autonomous Fault Recovery of Field Programmable Gate Arrays,” in *ACM Computing Surveys (CSUR)*, vol. 43 no. 4, pp.1-30, 2011.

[112] J. Lohn, G. Larchev, and R. DeMara, “A Genetic Representation For Evolutionary Fault Recovery in Virtex FPGAs,” in *Proc. Evolvable Systems: From Biology to Hardware: International Conference (ICES)*, pp. 47-56, Trondheim, Norway, March 17–20, 2003.

[113] J. Lohn, G, Larchev, and R. DeMara, “Evolutionary fault recovery in a Virtex FPGA using a Representation that Incorporates Routing,” in *Proc. IEEE International Parallel and Distributed Processing Symposium*, Apr. 2003, doi: 10.1109/IPDPS.2003.1213316.

[114] R.S. Oreifej, and R.F. DeMara “Intrinsic Evolvable Hardware Platform for Digital Circuit Design And Repair Using Genetic Algorithms,” in *Applied Soft Computing*, vol. 12, no. 8, pp.2470-2480, doi: 10.1016/j.asoc.2012.03.032.

[115] W. Kuang, P. Zhao, J. S. Yuan and R. F. DeMara, "Design of Asynchronous Circuits for High Soft Error Tolerance in Deep Submicrometer CMOS Circuits," in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 18, no. 3, pp. 410-422, 2010, doi: 10.1109/TVLSI.2008.2011554.

[116] A. A. Sakib, "Soft Error Tolerant Quasi-Delay Insensitive Asynchronous Circuits: Advancements and Challenges," in *34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI)*, Campinas, Brazil, 2021, pp. 1-6, doi: 10.1109/SBCCI53441.2021.9530001

[117] L. Zhou, S. C. Smith, and J. Di, "Radiation Hardened Null Convention Logic Asynchronous Circuit Design", in *Journal of Low Power Electronics and Applications*, vol. 5, no. 4, pp. 216-233, 2015.

[118] W. Jang and A. J. Martin, "SEU-Tolerant QDI Circuits [quasi delay-insensitive asynchronous circuits]", in *11th IEEE International Symposium on Asynchronous Circuits and Systems*, pp. 156-165, 2005.

[119] W. Jang and A. J. Martin, "A Soft-Error-Tolerant Asynchronous Microcontroller", in *13th NASA Symposium on VLSI Design*, 2007.

[120] M. Datta, A. Bodoh and A. A. Sakib, "Error Resilient Sleep Convention Logic Asynchronous Circuit Design," *2023 21st IEEE Interregional NEWCAS Conference (NEWCAS)*, Edinburgh, United Kingdom, 2023, pp. 1-5, doi: 10.1109/NEWCAS57931.2023.10198041.

[121] D. Mazumder, M. Datta, A. C. Bodoh, and A. A. Sakib, "A Scalable Formal Framework for the Verification and Vulnerability Analysis of Redundancy-Based Error-Resilient Null Convention Logic Asynchronous Circuits", in *J. Low Power Electron. Appl.* 2024, vol. 14, no. 5. https://doi.org/10.3390/jlpea14010005.

[122] D.A. Medler, and M.R Dawson, "Using redundancy to improve the performance of artificial neural networks," in Proc. of *the Biennial Conference-Canadian Society for Computational Studies of Intelligence*, Canadian Information Processing Society, (pp. 131-138), 1994.

[123] D.S. Phatak, and I. Koren, "Complete and partial fault tolerance of feedforward neural nets," in *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp.446-456, 1995.

[124] S. Angizi, H. Jiang, R. F. DeMara, J. Han and D. Fan, "Majority-Based Spin-CMOS Primitives for Approximate Computing," in *IEEE Transactions on Nanotechnology*, vol. 17, no. 4, pp. 795-806, Jul. 2018, doi: 10.1109/TNANO.2018.2836918.

[125] A. Roohi, S. Sheikhfaal, S. Angizi, D. Fan and R. F. DeMara, "ApGAN: Approximate GAN for Robust Low Energy Learning From Imprecise Components," in *IEEE Transactions on Computers*, vol. 69, no. 3, pp. 349-360, Mar. 2020, doi: 10.1109/TC.2019.2949042.

[126] L. Zheng, H. Liu, Y. Huang, D. Chen, C. Liu, H. He, X. Liao, H. Jin,and J. Xue, “A Flexible Yet Efficient DNN Pruning Approach for Crossbar-based Processing-In-Memory Architectures,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 11, pp. 3745-3756, Nov. 2022, doi: 10.1109/TCAD.2022.3197510.

[127] C. Liu, M. Hu, J. P. Strachan and H. Li, "Rescuing memristor-based neuromorphic design with high defects," in *Proc. ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1-6, doi: 10.1145/3061639.3062310.

[128] R. F. DeMara, K. Zhang, C. A. Sharma, “Autonomic fault-handling and refurbishment using throughput-driven assessment,” in *Applied Soft Computing*, Elsevier, vol. 11, no. 2, pp. 1588-1599, Mar. 2010.

[129] G. T. Tchendjou, K. Danouchi, G. Prenat and L. Anghel, "Spintronic Memristor based Binarized Ensemble Convolutional Neural Network Architectures," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022, doi: 10.1109/TCAD.2022.3213612.

[130] P. Barla, V.K. Joshi, and S. Bhat, “Design and analysis of SHE assisted STT MTJ/CMOS logic gates,” in *Journal of Computational Electronics*, Springer, vol. 20, part 5, pp.1964-1976, Aug. 2021.

[131] R. Zand, A. Roohi, D. Fan and R. F. DeMara, "Energy-Efficient Nonvolatile Reconfigurable Logic Using Spin Hall Effect-Based Lookup Tables," in *IEEE Transactions on Nanotechnology*, vol. 16, no. 1, pp. 32-43, Jan. 2017, doi: 10.1109/TNANO.2016.2625749.

[132] S. Manipatruni, D. E. Nikonov, and I. A. Young, “Energy-delay performance of giant spin Hall effect switching for dense magnetic memory,” in *Applied Physics Express*, vol. 7, no. 10, 2014, doi: 10.7567/APEX.7.103001.

[133] I. Ahmed, Z. Zhao, M. G. Mankalale, S. S. Sapatnekar, J. -P. Wang and C. H. Kim, "A Comparative Study Between Spin-Transfer-Torque and Spin-Hall-Effect Switching Mechanisms in PMTJ Using SPICE," in IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, vol. 3, pp. 74-82, Dec. 2017, doi: 10.1109/JXCDC.2017.2762699

[134] S. Rakheja, and A. Naeemi, “Graphene Nanoribbon Spin Interconnects for Nonlocal Spin-Torque Circuits: Comparison of Performance and Energy Per Bit with CMOS Interconnects,” in *IEEE Transactions on Electronic Devices*, vol. 59, no. 1, pp. 51–59, Oct. 2011, doi: 10.1109/TED.2011.2171186.

[135] D. Divyanshu, R. Kumar, D. Khan, S. Amara and Y. Massoud, "FSM Inspired Unconventional Hardware Watermark Using Field-Assisted SOT-MTJ," in *IEEE Access*, vol. 11, pp. 8150-8158, 2023, doi: 10.1109/ACCESS.2023.3238807.

[136] N. Khoshavi, R. A. Ashraf, and R. F. DeMara, “Applicability of power-gating strategies for aging mitigation of CMOS logic paths,” in *Proc. IEEE Midwest Symposium on Circuits and Systems* (MWSCAS), pp. 929-932, doi: 10.1109/MWSCAS.2014.6908568.

[137] K. I. Gubbi et al., “Securing AI Hardware: Challenges in Detecting and Mitigating Hardware Trojans in ML Accelerators,” in Proc. *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS),* Tempe, AZ, USA, Aug. 2023, pp. 821-825, doi: 10.1109/MWSCAS57524.2023.10406065.

[138] W. Zhao et al., “A Radiation Hardened Hybrid Spintronic/CMOS Nonvolatile Unit Using Magnetic Tunnel Junctions,” in *Journal of Physics D: Applies Physics*, vol. 47, no. 40, September 2014, doi: 10.1088/0022-3727/47/40/405003.

[139] C. H. Tung, B. K. Joardar, P. P. Pande, J. R. Doppa, H. H. Li and K. Chakrabarty, "Dynamic Task Remapping for Reliable CNN Training on ReRAM Crossbars," in *Proceedings ACM Design, Automation & Test in Europe (DATE)*, Antwerp, Belgium, April 2023, pp. 1-6, doi: 10.23919/DATE56975.2023.10137238.

[140] M. R. Muttaki, T. Zhang, M. Tehranipoor and F. Farahmandi, “FTC: A Universal Sensor for Fault Injection Attack Detection,” in *Proceedings IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, McLean, VA, USA, June 2022, pp. 117-120, doi: 10.1109/HOST54066.2022.9840177.

[141] D. Crumley, M. Hossain, K. Martin, F. Ivey, Richard Yarnell, R. F. DeMara, and Y. Bai "Rehosting YOLOv2 Framework for Reconfigurable Fabric-based acceleration," in *Proc. Of*

IEEE SoutheastCon, Mobile, AL, USA, Apr. 2022, pp. 445-446, doi: 10.1109/SoutheastCon48659.2022.9763979.

[142] R. Yarnell, M. Hossain and R. F. DeMara, "Image Quantization Tradeoffs in a YOLO-based FPGA Accelerator Framework," in *Proc. IEEE International Symposium on Quality Electronic Design (ISQED)*, San Francisco, CA, USA, Apr. 2023, pp. 1-7, doi: 10.1109/ISQED57927.2023.10129324.

[143] A. Alhussain, and M. Lin, "Hardware-Efficient Deconvolution-Based GAN for Edge Computing," in *Proc. IEEE Conference on Information Sciences and Systems (CISS)*, Mar. 2022, pp.1-5, Accessed on: Jan. 19, 2023.

[144] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[145] Y. Cai, H. Li, G. Yuan, W. Niu, Y. Li, X. Tang, B. Ren, and Y. Wang, "Yolobile: Real-time object detection on mobile devices via compression compilation co-design," in *Proc. of the AAAI Conference on Artificial Intelligence*, May 2021, vol. 35, no. 2, pp. 955-963.

[146] K. Zhang, R. F. DeMara, C. A. Sharma, "Consensus-based Evaluation for Fault Isolation and On-line Evolutionary Regeneration," in Proceedings of International Conference in Evolvable Systems (ICES'05), pp. 12 – 24, Barcelona, Spain, Sep. 12 – 14, 2005, doi:10.1007/11549703_2.

[147] C. Farabet, C. Poulet, J.Y. Han, and Y. LeCun, "CNP: An FPGA-based processor for convolutional networks," in *Proc. IEEE International Conference on Field Programmable Logic and Applications*, pp. 32-37, Aug. 2009.

[148] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, "Going deeper with embedded FPGA platform for convolutional neural network," in *Proc. ACM/SIGDA International Symposium on FPGAs*, pp. 26-35, 2016.

[149] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M.A. Horowitz, and W.J. Dally, "EIE: Efficient inference engine on compressed deep neural network", in *Proceedings ACM SIGARCH Computer Architecture News*, vol. 44, no.3, pp.243-254, 2016.

[150] A. Silva, D. Fernandes, R. Névoa, J. Monteiro, P. Novais, P. Girão, T. Afonso, and P. Melo-Pinto, "Resource-Constrained Onboard Inference of 3D Object Detection and Localization in Point Clouds Targeting Self-Driving Applications," *Sensors*, vol. 21, no.23, p.7933, 2021.

[151] M. Machura, M. Danilowicz, and T. Kryjak, "Embedded Object Detection with Custom LittleNet, FINN and Vitis AI DCNN Accelerators," *Journal of Low Power Electronics and Applications*, vol. 12, no. 2, pp. 30, 2022, doi: 10.3390/jlpea12020030.

[152] J. Wang, and S. Gu, "FPGA implementation of object detection accelerator based on Vitis-AI," in *Proc. IEEE International Conference on Information Science and Tech. (ICIST)*, pp. 571-577, May 2021.

[153] A. Bochkovskiy, C. Y. Wang, and H.Y.M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," Online [Available]: arXiv preprint arXiv:2004.10934, Apr. 2020.

[154] Z. Wang, Z, H. Li, X. Yue, and L. Meng, "Briefly Analysis about CNN Accelerator based on FPGA," *Procedia Computer Science*, vol. 202, pp.277-282, 2022.

[155] S. Khalili, and A. Shakiba, “A face detection method via ensemble of four versions of YOLOs,”, in *Proc. IEEE International Conf. on Machine Vision and Image Processing*, pp. 1-4, Feb. 2022.

[156] T. Hill, “Advancing the use of FPGA co-processors through platforms and high-level design flows,” Xilinx white paper, pp.1-14.

[157] C. Kern, and M. R. Greenstreet, "Formal Verification in Hardware Design: A Survey", in *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. *4, no.* 2, pp.123-193, 1999.

[158] A. Gupta, "Formal Hardware Verification Methods: A Survey," in *Formal Methods in System Design*, vol. *1*, pp.151-238, 1992.

[159] A. A. Sakib, S. Le, S. C. Smith, and S. Srinivasan, "Formal Verification of NCL Circuits", in *Asynchronous Circuit Applications,* Chap. 15, Edison, NJ, USA:IET, pp. 309-338, 2019. doi: 10.1049/PBCS061E_ch15.

[160] A. A. Sakib, S. C. Smith and S. K. Srinivasan, "Formal Modeling and Verification of PCHB Asynchronous Circuits", in *IEEE Trans. Very Large Scale Integr. (TVLSI) Syst.*, vol. 27, no. 12, pp. 2911-2924, Dec. 2019.

[161] M. Hossain, A. A. Sakib, S. K. Srinivasan and S. C. Smith, "An Equivalence Verification Methodology for Asynchronous Sleep Convention Logic Circuits", in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 1-5, May 2019.

[162] S. J. Longfield and R. Manohar, "Inverting Martin Synthesis for Verification", in *Proc. IEEE ASYNC*, pp. 150-157, May 2013.

[163] A. Saifhashemi, H. H. Huang, P. Bhalerao and P. A. Beerel, "Logical Equivalence Checking of Asynchronous Circuits Using Commercial Tools", in *Proc. DATE*, pp. 1563-1566, 2015.

[164] A. A. Sakib, S. C. Smith and S. K. Srinivasan, "Formal Modeling And Verification for Pre-Charge Half Buffer Gates and Circuits", *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, pp. 519-522, Aug. 2017.

[165] A. A. Sakib, S. C. Smith and S. K. Srinivasan, "An equivalence verification methodology for combinational asynchronous PCHB circuits", *Proc. IEEE 61st Int. Midwest Symp. Circuits Syst. (MWSCAS)*, pp. 767-770, Aug. 2018.

[166] M. Golmohamadi *et al.*, "Verification and Testing Considerations of an In-Memory AI Chip," *2020 IEEE 29th North Atlantic Test Workshop (NATW)*, Albany, NY, USA, 2020, pp. 1-6, doi: 10.1109/NATW49237.2020.9153079.

[167] Z. Yan, X. S. Hu, and Y. Shi, "Swim: Selective Write-Verify for Computing-in-Memory Neural Accelerators," in *Proceedings of the 59th ACM/IEEE Design Automation Conference,* pp. 277-282, Jul. 2022.