vi ENERGY-EFFICIENT MEMRISTOR-BASED NEUROMORPHIC COMPUTING CIRCUITS AND SYSTEMS FOR RADIATION DETECTION APPLICATIONS

Jorge Iván Canales Verdial

Follow this and additional works at: https://digitalrepository.unm.edu/ece_etds

Part of the Computational Engineering Commons, Computational Neuroscience Commons, Computer and Systems Architecture Commons, Electrical and Electronics Commons, Nanoscience and Nanotechnology Commons, Nuclear Engineering Commons, and the VLSI and Circuits, Embedded and Hardware Systems Commons

Recommended Citation

This Dissertation is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.
Jorge Iván Canales Verdial
Candidate

Electrical & Computer Engineering
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Payman Zarkesh-Ha, Chairperson

Dr. Miguel Figueroa Toro, Co-Chair

Dr. Marek Osiński

Dr. Adam Hecht

Dr. Ramiro Jordan
ENERGY-EFFICIENT MEMRISTOR-BASED
NEUROMORPHIC COMPUTING CIRCUITS AND SYSTEMS
FOR RADIATION DETECTION APPLICATIONS

by

JORGE IVÁN CANALES VERDIAL
BS, Biomedical Engineering, Universidad Tecnológica Centroamericana, 2016
BS, Mechatronics Engineering, Universidad Tecnológica Centroamericana, 2017
MS, Electrical Engineering, The University of New Mexico, 2020

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
Engineering

The University of New Mexico
Albuquerque, New Mexico

May 2023
DEDICATION

To all those people, of whom I have had too many in my life, who constantly told me that I could not do something.

For all the nay-sayers, the haters, and the nonbelievers:

guess what…

I could.
ACKNOWLEDGEMENTS

My graduate student life in University of New Mexico was one of the most exciting and rewarding experiences in my life. Hereby, I would like to show my best gratitude and appreciation to the people who helped me to have this wonderful time and made an impression in my life:

Foremost, I would like to express my deepest appreciation to Prof. Zarkesh-Ha whose support and guidance is beyond the limits of this research. Without his persistent encouragement and help, this dissertation simply would not have been possible. I thank him for his continuous advice, suggestions, and support throughout my amazing journey of knowledge. It was a great, exciting and pleasant opportunity to be a member of his team, to learn from him and to do research on so many emerging topics in electrical engineering. He provided me with every bit of guidance, assistance, and expertise that I needed during my learning. I quite simply cannot imagine better shoulders to stand on.

Similarly, I acknowledge Prof. Osiński, for his contributions to help me improve my academic writing. Without his thorough and detailed constructive critiques of my articles, I would not have been able to become the researcher that I am today. Thanks for pressuring me to go the extra mile in publishing my research work. Thanks for all the times you were able to catch the tiniest problematic issues in my papers. It frankly made me more comfortable with my writing, especially during the first years when I still struggled with the written language.

Equally, I would also like to acknowledge my other committee members Prof. Hecht, Prof. Jordan and Prof. Figueroa for their time and valuable input,
discussions, and accessibility. Your critiques and comments really made my dissertation better. Prof. Hecht’s knowledge of radionuclide spectra was instrumental for this study. Prof. Jordan’s imparted MATLAB lectures were key to my success. Prof. Figueroa in a brief time became a career role model.

I would also like to mention my research teammates and colleagues at the Electrical and Computer Engineering department: Nishchay, Zahra, Landon, and Mona. Their abundant knowledge and experience always helped me when I met any difficulty in my study and research. Thanks for sharing the office with me!

I am also deeply thankful to my family, especially my parents, and my brothers for their love, support, and sacrifices. They were always on my side in my good and tough times. Their encouragement was always my greatest motivation. Thanks for all the phone calls in which you gave me support when I was overwhelmed with the stress of graduate school. Likewise, I thank Luis for all his support throughout this journey, for taking care of me during those hardest middle years of my doctorate, for supporting me in my academic journey, and for believing in me when I did not believe in myself.

The completion of this work could also not have been possible without the participation and assistance of my support network at LH and TEC, as well as many more friends who would be too hard to list. Here, I acknowledge every one of them for their contribution. I owe you so much.

Finally, I also acknowledge that this work was supported by the generous funding from Defense Threat Reduction Agency HDTRA1-18-1-0009.
ENERGY-EFFICIENT MEMRISTOR-BASED
NEUROMORPHIC COMPUTING CIRCUITS AND SYSTEMS
FOR RADIATION DETECTION APPLICATIONS

by

Jorge Iván Canales Verdial

BS, Biomedical Engineering, Universidad Tecnológica Centroamericana, 2016
BS, Mechatronics Engineering, Universidad Tecnológica Centroamericana, 2017
MS, Electrical Engineering, The University of New Mexico, 2020
PhD, Engineering, The University of New Mexico, 2023

ABSTRACT

Radionuclide spectroscopic sensor data is analyzed with minimal power consumption through the use of neuromorphic computing architectures. Memristor crossbars are harnessed as the computational substrate in this non-conventional computing platform and integrated with CMOS-based neurons to mimic the computational dynamics observed in the mammalian brain's visual cortex. Functional prototypes using spiking sparse locally competitive approximations are presented. The architectures are evaluated for classification accuracy and energy efficiency. The proposed systems achieve a 90% true positive accuracy with a high-resolution detector and 86% with a low-resolution detector.
Contents
DEDICATION ......................................................................................................................... iii
ACKNOWLEDGEMENTS ....................................................................................................... iv
ABSTRACT ............................................................................................................................ vi
LIST OF FIGURES ............................................................................................................. xii
LIST OF TABLES ................................................................................................................ xvii
PREFACE ........................................................................................................................... xix
I. INTRODUCTION ............................................................................................................. 1
  1.1. PROBLEM STATEMENT ..................................................................................... 3
  1.2. RESEARCH QUESTIONS .................................................................................... 4
  1.3. OBJECTIVES ....................................................................................................... 5
    1.3.1. GENERAL GOAL ....................................................................................... 5
    1.3.2. SPECIFIC GOALS ................................................................................... 5
  1.4. JUSTIFICATION ................................................................................................. 6
  1.5. STRUCTURE ......................................................................................................... 7
II. NEUROMORPHIC COMPUTING ARCHITECTURES ...................................................... 9
  2.1. MOTIVATION FOR NC ARCHITECTURE DEVELOPMENT ................................ 10
  2.2. PREVIOUS WORK ON NC ARCHITECTURES ............................................... 13
    2.2.1. DIGITAL NC ARCHITECTURES ............................................................ 14
    2.2.2. ANALOG NC ARCHITECTURES ........................................................... 15
    2.2.3. MIXED ANALOG/DIGITAL NC ARCHITECTURES ............................... 16
    2.2.4. CIRCUIT-LEVEL NC DEVICES .............................................................. 16
  2.3. METHODS FOR MODELLING NEUROPHYSIOLOGY .................................... 20
    2.3.1. RESTING POTENTIAL PHYSIOLOGY ..................................................... 22
2.3.2. GRADED POTENTIAL PHYSIOLOGY ........................................ 24
2.3.3. ACTION POTENTIAL PHYSIOLOGY ................................. 25
2.3.4. SYNAPTIC CONNECTION PHYSIOLOGY ......................... 28
2.4. DEVELOPED NC ARCHITECTURES ................................. 29
  2.4.1. MODELLING POWER CONSUMPTION USING RASPBERRY PI ................................................................. 29
  2.4.2. NC ARCHITECTURE SCHEMATICS ................................. 31
  2.4.3. ENERGY EFFICIENT HARDWARE SCALABILITY .............. 34
  2.4.4. DEVELOPED CMOS NEURONS FUNCTIONALITY ............. 35
  2.4.5 NC ARCHITECTURE LAYOUTS ........................................ 40
2.5. CHAPTER SUMMARY ......................................................... 54
III. SYSTEM ACCURACY & RELIABILITY ANALYSIS ......................... 55
  3.1. MOTIVATION FOR SYSTEM FAULT-TOLERANCE ASSESSMENT 56
  3.2. PREVIOUS WORK ON NC TRAINING ALGORITHMS ............... 57
    3.2.1. SUPERVISED LEARNING ALGORITHMS .......................... 57
    3.2.2. UNSUPERVISED LEARNING ALGORITHMS ..................... 59
    3.2.3. LEARNING ALGORITHMS FOR SPIKING NEURAL NETWORKS ............................................................................ 59
    3.2.4. WHICH TRAINING ALGORITHM TO CHOOSE? .................. 60
  3.3. METHODS FOR TESTING THE SYSTEM’S FAULT-TOLERANCE 61
    3.3.1. MODELING MEMRISTOR FAULTS ................................... 62
    3.3.2. MODELLING THE COMPUTATION ALGORITHM .............. 63
3.3.3. DEFECT INJECTION SCHEMES ............................................ 65
3.3.4. COMPUTATIONAL DYNAMICS IMPLEMENTATION .......... 66
3.4. FAULT-TOLERANCE TEST RESULTS ........................................ 67
  3.4.1. DISCRETIZATION EFFECTS ........................................... 68
  3.4.2. OPEN CIRCUIT DEFECTS .............................................. 70
  3.4.3. SHORT CIRCUIT DEFECTS ............................................. 71
  3.4.3. BIT FAILURE DEFECTS ................................................. 72
3.5. DISCUSSION ON SYSTEM’S PERFORMANCE ....................... 73
  3.5.3. PERFORMANCE EFFECTS OF OPEN CIRCUIT DEFECTS ...... 73
  3.5.4. PERFORMANCE EFFECTS OF SHORT CIRCUIT DEFECTS ... 74
  3.5.5. OVERALL PERFORMANCE EFFECT OF BIT FAILURE .......... 75
  3.5.6. SYSTEM SENSITIVITY TO MULTIPLE RADIONUCLIDES ....... 75
3.6. CHAPTER SUMMARY ............................................................ 76
IV. NC SYSTEM ENERGY EFFICIENCY ANALYSIS ............................. 77
  4.1. MODELLING METHODS FOR ENERGY EFFICIENCY ANALYSIS .. 77
    4.1.1. STATIC POWER DISSIPATION IN CMOS NEURONS .......... 77
    4.1.2. DYNAMIC POWER DISSIPATION IN CMOS NEURONS .......... 78
  4.2. DYNAMIC POWER DISSIPATION IN THE PRESYNAPTIC CELLS .. 83
    4.2.1. INHIBITION CAPACITOR CHARGING PMOS M₀ TRANSISTOR 83
    4.2.2. INHIBITION CAPACITOR DISCHARGING NMOS M₁ TRANSISTOR 85
    4.2.3. INHIBITION CAPACITOR DISCHARGING NMOS M₂ TRANSISTOR 86
A1. NC VECTOR MATRIX MULTIPLICATION CONFIGURATIONS .................. 112
A2. 0.5-µm TRANSISTOR MODELS USED FOR SPICE SIMULATIONS ...... 115
A3. PRESYNAPTIC NEURON SIMULATION SPICE CODE ....................... 119
A4. POSTSYNAPTIC NEURON SIMULATION SPICE CODE ..................... 121
A5. NEUROMORPHIC COMPUTING MODELS TAXONOMY ...................... 124
A6. RADIONUCLIDE DETECTOR INTERFACE .................................... 125
A7. CMOS PROCESS LAYERS IN L-EDIT LAYOUT TOOL ...................... 127
REFERENCES .................................................................................... 128
LIST OF FIGURES

Figure 1. Aerial Euler diagram describing the disciplines involved in neuromorphic computing. .......................................................................................................................... xxii

Figure 2. High-level diagram of the Computing Approach. Through training the reservoir output layer, a device network’s intrinsic dynamics can be harnessed to perform a desired computation. ........................................................................... 2

Figure 3. Memristor Network .................................................................................. 3

Figure 4. Abstract computing setup using a network reservoir ......................... 3

Figure 5. Motivation Factors for NC Architecture Development ..................... 10

Figure 6. Network-Level NC Architecture Categories .................................... 14

Figure 7. Device-Level Components used as Computational Substrate in NC Architectures ............................................................................................................. 16

Figure 8. Biological Neurons and their Connections to NC Neurons .............. 22

Figure 9. High-Level Schematic of the NC Interneuron Connection .............. 22

Figure 10. Connection approach implemented to measure power consumption of NC classifying algorithm ................................................................. 30

Figure 11. Data acquisition NI LabVIEW program that monitors the power consumption of the Raspberry Pi ................................................................. 31

Figure 12. Memristor crossbar implemented as an NC computing core ......... 32

Figure 13. Basic memristor crossbar schematic ................................................. 32

Figure 14. High-level schematic of developed NC analog architecture .......... 32

Figure 15. High-level schematic of developed NC mixed signal architecture 34
Figure 16. Simulation results for developed proof-of-concept analog NC architecture

Figure 17. Simulation results for developed proof-of-concept mixed signal NC Architecture

Figure 18. Layout of developed NC analog architecture with the excitation ($\Phi_{cb}$) and inhibition ($G_{m,n}$) arrays are shown in a box.

Figure 19. L-Edit layout of developed NC mixed signal architecture showing the control circuit, the presynaptic and postsynaptic cells in a box.

Figure 20. CMOS presynaptic neuron with adaptation module for memristor potentiation or depression.

Figure 21. CMOS postsynaptic neuron with adaptation module for memristor potentiation or depression.

Figure 22. Energy consumption per spike in the presynaptic and postsynaptic neurons.

Figure 23. Prototypes implemented an ON Semiconductor B5 0.5-micron process and on a PCB with off-the-shelf components.

Figure 24. IC design containing the individual LCA & SSLCA Module: transmission gate (A2-A4), axon module (A5-A6), thresholding module (A7-A8), firing module (A8-A10), capacitor (A11-A12) operational amplifier (A13-A16), transceiver (A18-A23), presynaptic cell (A24-A26), inhibition module (A27-A30), and postsynaptic cell (A31-A32).

Figure 25. IC design containing a 5x6 analog LCA classifier with I/P: B2-B6, O/P: B7-B12, bias: B1 & B15-B17; and a 5x6 spiking SSLCA classifier with I/P:
B18-B22, O/P: B23-B28, bias: B1 & B17, capacitor (B13-B14), resistor (B29-B30),
and amplifier (B31-33)..................................................................................................................46

Figure 26. ASIC L-Edit layout containing individual test modules. ..............48

Figure 27. ASIC L-Edit layout containing analog & mixed signal NC
architectures. .............................................................................................................................48

**Figure 28.** Gate-level schematic of developed presynaptic input neuron for the
NC mixed-signal architecture...................................................................................................49

Figure 29. L-Edit layout of developed presynaptic input neuron for the NC
mixed-signal architecture showing the inhibition resistor and capacitor in boxes. 
.............................................................................................................................................49

Figure 30. Gate-level schematic of developed postsynaptic output neuron for
the NC mixed-signal architecture..........................................................................................50

Figure 31. L-Edit layout of developed postsynaptic output neuron for the NC
mixed-signal architecture........................................................................................................51

Figure 32. Device-Level schematic of developed thresholding modules for the
analog NC architecture (left) and the mixed signal NC architecture (right).........52

Figure 33. L-Edit layout of developed thresholding modules for the mixed signal
NC architecture..........................................................................................................................52

Figure 34. VTC results for unbalanced inverters implemented in thresholding
module.........................................................................................................................................52

Figure 35. Device-level schematic of axon module for the mixed signal NC
architecture..................................................................................................................................53
Figure 36. Device-level schematic of the low-power operational amplifier designed for the analog NC architecture. ........................................................... 54

Figure 37. L-Edit layout of the low-power operational amplifier designed for the analog NC architecture. .............................................................................................. 54

Figure 38. Training-approach NC algorithm categories. .............................................. 57

Figure 39. Circuit schematic for the implemented 2,048×27 memristor array. ........................................................................................................................... 62

Figure 40. Memristor fault models. ................................................................ 62

Figure 41. Memristor discretization levels. In the presence of small inference voltages, the resistance state remains constant. At high training voltages the memristor changes state. ........................................................................... 63

Figure 42. A 10×10 section of the required memristor array showing (left) the ideal normalized dictionary weights and (right) the same dictionary subsection with 30% randomly injected single-bit memristor defects. The open defects are shown in black and the short defects are shown in white .............................................. 66

Figure 43. Block diagram of LCA circuit implementation. .............................. 67

Figure 44. Classification accuracy with a weight density distribution of N=1. 68

Figure 45. Classification accuracy with a weight density distribution of N=2. ........................................................................................................................... 69

Figure 46. Classification accuracy with a Weight Density Distribution of N=0.5. ........................................................................................................................... 69

Figure 47. Classification accuracy with a weight density distribution of N=4. 70
Figure 48. Classification accuracy with a weight density distribution of N=0.25.
........................................................................................................................... 70

Figure 49. True positive classification accuracy vs. % of open circuit defects.
........................................................................................................................... 71

Figure 50. True positive classification accuracy vs. % of short circuit defects.
........................................................................................................................... 72

Figure 51. True positive classification accuracy with both open and short defects.............................................................................................................................. 73

Figure 52. Sensitivity to Multiple Radionuclides ............................................ 76

Figure 53. VMM Configurations used for NC Applications ....................... 112

Figure 54. VMM Configurations implementing bias column difference to compensate for the non-zero conductance of memristors in the high resistance state.............................................................................................................................. 113

Figure 55. Neuromorphic Computing Model Categories. ......................... 124

Figure 56. Frequency Neural Encoding of Eu-152 ..................................... 126
LIST OF TABLES

Table 1. Classification results for ................................................................. 39

Table 1. Energy Consumption of Presynaptic CMOS Neurons .................. 45

Table 2. Energy Consumption of Postsynaptic CMOS Neurons ............... 45

Table 3. Pinout Developed ASIC Prototype A ........................................... 47

Table 4. Pinout Developed ASIC Prototype B ........................................... 47

Table 5. Training Algorithm Decision Matrix ............................................ 61

Table 6. Channel Capacitance Approximation According to the Region of Operation ................................................................. 81

Table 7. Capacitances Contributing to Dynamic Power Dissipation in the Inhibition Modules’ M₀ Transistor ....................................................... 84

Table 8. Capacitances Contributing to Dynamic Power Dissipation in the Inhibition Modules’ M₁ Transistor ....................................................... 86

Table 9. Capacitances Contributing to Dynamic Power Dissipation in the Inhibition Modules’ M₂ Transistor ....................................................... 87

Table 10. Capacitances Contributing to Dynamic Power Dissipation in the Inhibition Modules’ NAND logic gate ........................................... 88

Table 11. Capacitances Contributing to Dynamic Power Dissipation in the Inhibition Modules’ NAND logic gate. .......................................... 89

Table 12. Capacitances Contributing to Dynamic Power Dissipation in the Crossbar Bilateral Analog Switch .............................................. 91

Table 13. Capacitances Contributing to Dynamic Power Dissipation in the 1X Inverter. ......................................................................................... 92
Table 14. Capacitances Contributing to Dynamic Power Dissipation in the Firing PMOS................................................................. 94

Table 15. Capacitances Contributing to Dynamic Power Dissipation in the Postsynaptic Cell's Transmission Gate........................................... 96

Table 16. Capacitances Contributing to Dynamic Power Dissipation in the M4 NMOS.................................................................................. 98

Table 17. Capacitances Contributing to Dynamic Power Dissipation in the Thresholding Unbalanced Inverter.............................................. 99

Table 18. Capacitances Contributing to Dynamic Power Dissipation in the Thresholding Balanced Inverter............................................... 100

Table 19. Capacitances Contributing to Dynamic Power Dissipation in the Firing Circuits M11 Transistor....................................................... 101

Table 20. Capacitances Contributing to Dynamic Power Dissipation in Inverter Chain Gates................................................................... 102
PREFACE

Neuromorphic computing (NC) architectures are inspired by neuronal processes in mammalian brains. In contrast to traditional von Neuman computers, NC architectures implement energy-efficient in-memory computing paradigms that imitate neuronal signal processing observed in the visual (V1) cortex. NC platforms offer potential energy-efficient information processing by addressing conventional computers' organization and structural limitations. Mimicking the brain, NC architectures allow lower power, smaller footprint and improved computing processing capabilities.

Upon its inception in the 1980s by Carver Mead, the term "neuromorphic computing" originally referred solely to subthreshold analog VLSI architectures that mimicked biological neural systems [1]. However, over the past 40 years, the field has expanded to include bioinspired neuronal modelling, brain-inspired learning algorithm development, biomimetic material and device fabrication, and neuromorphic computing system integration. Furthermore, the development of supporting systems and practical computing applications also interests the wider NC community.

The NC future offers significant growth potential. Mature NC architectures will process complex calculations faster. Moreover, the NC system's inherent ability for rapid learning and smaller footprint renders them highly compatible for neural network (NN) hardware implementations. NC parallel in-memory processing enables the development of highly connected low-power architectures.
Interest in the NC field has increased due to Moore's law's unavoidable slowdown. Moreover, NC's potential to remediate the von Neuman bottleneck and overcome Dennard's scaling energy constraints, facilitates their application for intrinsically adaptable machine learning (ML) computing algorithms [2]. Evidence shows [3] that NC platforms are the preeminent choice for NN algorithm implementation, because NC architectures harness ML and NN learning algorithms combined. Moreover, NC architectures offer the possibility of on-line/real-time learning, which has become more important with the ever-growing data collection applications of today's computing systems.

NC advances requires professional cooperation from several disciplines (as shown in Figure 1: neuroscience (NS), computer science (CS) electrical engineering (EE), computer engineering (CE), and material science (MS). NS studies neuronal structures and behavior in biological networks, to create significant models for mimicking a neural network. CS focuses on writing neural learning schemes, processing algorithms and supporting software that imitate the biological models developed by the neuroscientist. MS investigates novel materials that exhibit the biomimetic properties required by NC circuits. Finally, hardware-oriented EE and CE professionals develop analog, digital and mixed signal computing circuits that combine the novel biomimetic material and devices into fully functional NC architectures. NC platform development is only possible through this multidisciplinary professional interaction.

Significant challenges are still ahead for the NC community. Neurobiologists need accurate bio realistic neuronal models to model human brain processes.
These model advancements would aid the understanding of human brain conditions like Alzheimer’s or brain injury. Current biomimetic material and device constraints are yet another challenge. Moreover, current technology is still unreliable in accurately replicating neuronal behavior, because fabrication processes are still under development. Hence, material scientists must work along circuit designers and computer scientists during an NC platform’s development.

Figure 1. Aerial Euler diagram describing the disciplines involved in neuromorphic computing.
I. INTRODUCTION

Radionuclides can be identified statistically by analyzing the features of their gamma ray emission spectra. However, conventional computing systems must collect and process large quantities of data to learn the latent features from each isotope’s spectra. This results in significant power consumption and a high degree of user involvement required for system training. Hence, new computing avenues must be explored to address this issue and provide a more efficient computation.

Neural network architectures developed from biological nervous systems can be more efficient than conventional computer architectures in some circumstances. In the past few decades, neuromorphic computing architectures have evolved into highly energy-efficient, and robust systems based on brain-inspired architectures induced transients and permanent defects.

Unconventional NC architectures provide fast, energy-efficient, and cost-effective platforms by harnessing their inherent computing dynamics [8], [9]. An arbitrary or partially randomized assembly of a computational substrate is necessary to produce such architectures. An example of random assembly molecular switches has shown logical functions [10], and a general methodology for programming random assemblies has been presented in [11]. The use of nanodevices, such as ReRAM or memristors [12], [13], is described in a number of unconventional computing architectures. The random assembly of memristor devices into large networks has also been demonstrated in [14]–[16]. The use of random assemblies for tasks such as high harmonic generation illustrates computing that is based on intrinsic properties of dynamical systems, as
demonstrated by Reservoir Computing (RC) in [4] or other temporal information processing techniques [17].

A fault-tolerant and multi-tasking substrate is suitable for unconventional computing implementations [18]–[20]. Memristor-based NC can solve simple pattern-classification problems [7], [21] using random or ordered networks.

Figure 2 illustrates the non-conventional computing approach. Memristors accelerate and improve a wide range of computations using this scheme, because training is only required for the output layer, as opposed to the internal layer training used in traditional neural networks. Moreover, this systems are able to tolerate variations, defects, and flaws [7]. These memristor-based architectures can also be applied to the processing of temporal signals, which is the case for radiation detection applications.

![Graphical representation of the computing approach](image)

**Figure 2** High-level diagram of the computing approach. Through training the reservoir output layer, a device network’s intrinsic dynamics can be harnessed to perform a desired computation.

Figure 3 shows a detailed view of a memristor network. Figure 4 shows their implementation as an NC computing core. The memristor array acting as the reservoir “computing core” provides the characteristic high connectivity degree and parallelism required to achieve in-memory processing. The system further establishes promising unconventional paradigms for NC architectures.
1.1. PROBLEM STATEMENT

Radionuclides can be identified by using statistical inference to analyze gamma-ray spectra features. However, to learn latent features from the isotope spectra, conventional platforms need to collect and process large quantities of input data. As a result, existing detection systems consume a significant amount of power and require a high degree of user involvement. Hence, a necessity exists for a real-time energy-efficient alternative system capable of radiation detection, nuclide identification, and localization in mobile scenarios.
1.2. RESEARCH QUESTIONS

Below is the list of questions addressed in this research work:

- Are memristor-based neuromorphic computing architectures a better alternative than conventional computing technology currently used for radionuclide detection, identification and localization?

- To what degree does the memristor nondeterministic fabrication process and their resulting omnipresent manufacturing defects degrade computation accuracy in a radionuclide detection computation application?

- How can the neuroplasticity self-healing characteristics of neurobiological system be mimicked to overcome memristor defects such as bit and line failure, discretization, and resistivity range and threshold variability?

- How does the energy efficiency of the system compare to other computation schemes, other NC implementations, or other non-conventional computing schemes implemented for detection applications?
1.3. OBJECTIVES

1.3.1. GENERAL GOAL

The general goal of this research is to develop energy-efficient neuromorphic computing algorithms, architectures, and components for analyzing and identifying isotopes from mobile platforms.

1.3.2. SPECIFIC GOALS

Below are the specific goals of this research effort:

- Propose a neuromorphic computing system for radionuclide detection and use device-level circuit simulation for its validation.
- Analyze the performance of the proposed neuromorphic computing architecture using industry standard simulation tools.
- Optimize the architectures to find the best design choices and parameters for low-power and high robustness.
- Perform circuit-level simulations and tests using real-world spectra patterns for common radionuclides.
- Create a prototype neuromorphic computing architecture that operates in classification mode to identify radionuclide isotopes.
- Produce and test hardware proof-of-concept prototypes for the developed neuromorphic computing architecture using commercial off-the-shelf components and standard microcontrollers.
1.4. JUSTIFICATION

In recent years, advances in neuromorphic computing have enabled the creation of massive brain-like parallel neural networks (NN) computing systems. NC architectures, mimicking mammalian neuronal processes, allow classification tasks to be performed with less energy consumption than conventional computers [4], [5]. Various complex image and signal processing tasks have already been accomplished by NC in the past. NC technology can also be used for navigation, voice processing, and robot control. However, despite its significant energy optimizations, NC radionuclide detection has rarely been explored for its potential benefits. Algorithms for detecting radiation are described in [22], [23]. However, it remains a challenge to develop compelling NC architectures, despite the memristor-based computing platform being the most suitable choice for such a radiation detection hardware device.

Moreover, NC architectures can enhance detection platforms by improving their speed, cost, and energy efficiency[8], [9]. Several studies have shown that memristors can be used for the implementation of large random assembly networks [24], [25]. An ordered or random network can also be used by memristor-based systems to solve simple pattern classification problems. Radionuclide detection can also be accomplished using memristors because of their flexibility and parallel processing capabilities.
1.5. STRUCTURE

This dissertation is organized into five main chapters as follows:

- Chapter 1 introduces the research problem, explains the problem context and background, provides the problem’s definition and approach, lists the research questions and objectives, provides a justification for the study and describes the document’s structure.

- Chapter 2 introduces the developed neuromorphic computing (NC) architectures and discusses the motivations for their implementation in a radionuclide detection platform. This chapter initially provides a background summary of the existing NC architectures that have been used for other classification applications and compares their characteristics. The methods used for circuit design, functionality, testing and layout development are presented. Clear connections to the biological neural networks (BNN) are described. The biological inspiration of the designed NC circuits is discussed. Then the finalized developed architectures are presented schematically and as IC layouts. This chapter’s discussion focuses on the circuits simulation results and provides interpretations for the layout vs schematic results and the optimization possibilities for circuit performance improvement.

- Chapter 3 explores the developed system’s accuracy and throughput. The nonidealities of memristor manufacturing are discussed. An analysis is conducted to assess the effects of device variability in the proposed NC hardware implementations for radionuclide detection. The
chapter’s background section presents an array of NC algorithms and their fault tolerance capabilities. The methods section describes the implementation of one of these algorithms in MATLAB and presents the testing approach for memristor nonideality effects. The construction of the radionuclide dictionary is described, and the defect-injection schemes are explained. Defects studied include bit and line defects, discretization and resistance level and threshold variability issues, and thermal signal noise.

- Chapter 4 provides an estimation analysis method to assess the energy efficiency of an NC architecture. Despite the extremely varied design process of NC circuits, there are common elements that are ubiquitous across platforms that implement CMOS neurons. The basic circuit element parameters of the proposed NC design are analyzed in this chapter. Specifically, the elements present across the architecture that implement: the signal encoding scheme, the memory storage synaptic devices, the cell charge accumulators, the thresholding nonlinearity, and the firing subcircuits were studied.

- Chapter 5 presents an overall synopsis of the main dissertation contributions and research outcomes. A list of the final concluding remarks about the developed architectures is presented. Recommendations for NC circuit optimization are provided based on this work’s results. Finally, future research work questions are addressed.
II. NEUROMORPHIC COMPUTING ARCHITECTURES

A number of electronic and optoelectronic devices have been used to implement plasticity in artificial neurons as discussed in Section 2.2.3. Some examples include MOSFET neurons operating at subthreshold currents [1], three-dimensional stacking of neurons with through silicon-vias [26], resistance random access memories (memristors) with crossbar interconnects [27], as well as lasers and photodiodes with optical interconnects [28]. Image processing, speech recognition, controlling robots, and navigation are among the many applications of neuromorphic computing [21]. However, radionuclide detection applications using a memristor-based computation substrate have been seldom studied despite this potential NC platform’s inherent reconfigurability. Moreover, the self-healing characteristics of neuromorphic computation could be used to apply rescuing schemes that overcome radiation-induced transient failures and permanent system damage. Furthermore, neural architectures can generalize and classify input signals that they have not been trained for, as would be the case in mobile radionuclide detection applications.
2.1. MOTIVATION FOR NC ARCHITECTURE DEVELOPMENT

Several enhancing factors promote NC architecture development as shown in Figure 5. In neurobiology, the need for bio-realistic neuronal models is an important research driver. In contrast, for the computing community the intrinsically high parallelism, real-time performance, low-power potential, small footprint, fault tolerance, and online capabilities of NC architectures are the key motivators.

The earliest NC literature primarily emphasized on-chip parallel processing as the most popular research incentive [29]–[33]. NC systems’ inherent parallelism, coupled with on-chip learning potential, impulse circuit designers to develop brain-inspired architectures that spur neural processing techniques with highly dense synaptic interconnections for energy-efficient computing. Neural Network (NN) algorithm developers also emphasize computing speed as a primary motivator for NC system development [34]–[36]. When compared with conventional computers, NC systems harness neural computation as a machine learning accelerator. NN parallelism and speed have further inspired circuit designers to produce real-time high-performance NC architectures capable of smoothly running NN algorithms.

Unlike conventional computers, NC applications can process faster because they overcome the von Neuman bottleneck, making them ideal for autonomous robot control [37], digital image processing [38], and other energy intensive data...
classification tasks. Since NC systems are usually application specific, they are more energy efficient than equivalent processes run in a conventional computer. Bioinspired learning systems also have potential adaptability properties that can be harnessed by NC architectures to exploit the NN’s parallelism. NC platforms develop single on-chip applications, emphasizing their simple processing neurons and dense synaptic interconnections.

Another motivator to develop architectures that exploit neural computation advantages is the potential computing speeds. ASIC NC platforms[39] achieve faster speeds than conventional computers under the same task. Additionally, the inherent fault tolerance of NN learning algorithms justifies their application for NC implementations despite device errors. Fault tolerance is an important stimulating factor justifying NC research [29], [40] because variation yield and device imperfections are the primary cause for computing errors seen in current NC architectures. Undeniably, the variability of current biomimetic materials and devices makes fault tolerance an important characteristic for NC platforms.

Other key motivators for NC advances come from the neuroscience field, where NC architectures are needed for accurate human brain simulation. Currently, there are neuroscientist lead projects that use NC models for biological neural behavior research, e.g., the European “Human Brain Project” [41].

By far, today’s predominant NC research driver is the potential development of energy-efficient architectures. The human brain is an outstanding biological computer presenting remarkable processing capabilities on an extremely low power budget of 20W. Since their inception, NC architectures attempt to obtain
similar efficiency levels [42], [43]. In recent years, novel material and device developments have also further increased the potential energy consumption improvements. These novel material discoveries also coincide with microprocessor and embedded system improvements, rendering small-footprint energy-efficient architectures feasible. This is related to NC architectures’ scalability potential which will allow promising energy density and efficiency capacities. Furthermore, cascaded NC systems allow for a large number of neurons and synapses which may be several orders of magnitude more efficient than single layer architectures.

The unavoidable end of Moore’s law and Dennard’s scaling is currently a barrier preventing the improvement of conventional architecture’s computing processing capabilities. This catalyzed the interest in NC as a means to build better post-Moore’s Law computers. Although NC computers will not totally replace the ubiquitous von Neuman architecture in the near future, further NC development will address current computing issues. Namely, conventional computing architectures separate memory allocation from the processing unit, creating a computing inefficiency. This problem, known as the von Neumann bottleneck [44], is frequently cited as a reason for NC development. Since memory and processing are collocated in an NC architecture, their implementation would mitigate this issue.

A final reason for interest in NC architecture development is their inherent capability for online learning. NC architectures adapt to changes by implementing learning algorithms. With the ever-increasing big data collection that computing
systems manage today, online learning is indispensable for novel computing platforms. Moreover, increased brain processing understanding driven by the neuroscience community will further lead to better online learning platforms.

2.2. PREVIOUS WORK ON NC ARCHITECTURES

NC architectures can be categorized at different abstraction levels. At the network-level, NC architectures are classified according to their signal characteristics [45]. In contrast, at the device-level these architectures are distinguished by the technology exploited as computational substrate [46].

Figure 6 shows the network-level architecture taxonomy used throughout this work. In fully analog NC computers, electronic devices can be exploited for analog computation [47]. In fully digital computers, bit state logic gates achieve the computation. Analog architectures are noisier than digital systems, however, asynchrony is their remarkable advantage. Digital systems rely on synchronous clock-based operations. Nonetheless, exceptions exist like asynchronous event-driven digital systems or clock based analog synchronic systems.

Analog-based neuromorphic computing architectures are better suited to emulate the biological brain’s operation since mammalian brains rely on analog physical characteristics, rather than Boolean computations as conventional computers. Furthermore, analog NC systems tend to be more fault-tolerant and noise-tolerant because of their neural network plasticity and robustness.
2.2.1. DIGITAL NC ARCHITECTURES

FPGA platforms are preferred over ASICs for digital applications due to their design advantages [48]. NC FPGA-based architectures bridge the design gap between SPICE-based simulation and final ASIC chips. Nonetheless, the FPGA implementation is sometimes the final product, because even though the FPGA’s programmability is solely used for device programming, the reconfigurability inherent to FPGAs can be harnessed to explore several network topologies, models & algorithms.

2.2.1.1. FPGA NC ARCHITECTURES

In digital neuromorphic computing architectures, FPGAs are ubiquitous due to their accessibility, short design and fabrication times, reconfigurability, and reusability. Through simple host interfaces, hardware description languages like VHDL and Verilog facilitate platform customization and optimization. A major advantage of FPGA platforms is that they offer a faster verification process than ASIC. FPGA-based NC platforms have the disadvantage that they are not ideal for
systems that have a small footprint and are energy efficient when compared to ASIC applications.

2.2.1.2. DIGITAL ASIC NC ARCHITECTURES

   Neuromorphic computing architectures also employ digital ASICs, although not as frequently as analog ASICs. TrueNorth [49] and SpiNNaker [50] are the most examples of this group. In terms of energy consumption, these platforms are the most energy-efficient among digital NC systems.

2.2.2. ANALOG NC ARCHITECTURES

   Unlike digital NC systems, analog NC systems are fully programmable. Originally, Carver Mead described the concept of neuromorphic computing in terms of analog bioinspired ASICs running at low voltages for energy efficiency. In contrast to digital systems, Analog NC architectures are primarily based on ASICs, except for Field Programmable Analog Arrays (FPAA).

2.2.2.1. FPAA NC ARCHITECTURES

   The NC platform’s inherent reconfigurability justifies FPAA NC applications similarly to FPGAs in digital systems. Despite this, they are less appealing than their digital counterparts due to limited resources and low application programmability. There are some NC-specific FPAAAs, such as Neuro FPAA [51] and Field Programmable Neural Array (FPNA) [52]. NC-specific platforms include programmable neurons, synapses, and other parameters, unlike generic FPAAAs.
2.2.2.2. ANALOG ASIC NC ARCHITECTURES

Analog NC ASICs employ charge conservation, amplification, thresholding, and charge integration characteristics, which are comparable to neurons in biological networks. As a result, analog systems must overcome asynchronous circuitry, noise, and unreliable components. The bioinspired nature of NC systems’ plasticity, however, allows analog ASIC to surpass these barriers, and provide self-healing properties as would be required by a radionuclide NC platform.

2.2.3. MIXED ANALOG/DIGITAL NC ARCHITECTURES

Mixed systems combine the advantages of digital and analog circuitry. The analog components of a mixed system resemble a biological neuron or synapse. The digital elements overcome problems associated with noise, asynchrony, and unreliability commonly encountered in fully analog systems. In mixed NC systems, synaptic weights are usually stored in digitally based memory components. Analog memories are unreliable and noisy, contributing to this preference. Mixed NC applications using spike encoding protocols are primarily analog, while other mixed NC applications use digital inter chip communication. The retraining and learning of synapses are also performed using digital circuits in mixed systems.

2.2.4. CIRCUIT-LEVEL NC DEVICES

Figure 7. Device-level components used as computational substrate in NC architectures.
2.2.4.1. MEMRISTOR-BASED NC SYSTEMS

The resistive random-access memory (also known as memory resistor or memristor), is a dominant computational substrate device in neuromorphic computing research. The memristor’s low power operation model biological synaptic functions with extreme precision despite inherent fabrication nonidealities. Nonetheless, any neural network’s inherent fault tolerance mitigates the RRAM transients, noises, or device variations. Moreover, RRAM technology mimics the biological neuron’s synaptic plasticity and weight storage capabilities, exhibiting spike-time dependent plasticity (STDP) or Hebbian learning capabilities. NC applications of the memristor require crossbar architectures. These configurations are more efficient and have smaller footprints. Multiple memristor synapses can also achieve negative weights when employed as memristor bridge synapses or double element configurations.

Other specific requirements of NC architectures are met by the memristor, e.g., weight-forgetting properties. The memristors characteristics must be addressed by any memristor-oriented training algorithms. Memristor-based neurons models simulate synaptic connections with precision, e.g., the Hodgkin-Huxley neuronal models. NC circuits based on memristors simulate biological spiking systems, i.e., strengthen the synapse harnessing the RRAM’s stochastic behavior.

Memristor application in Neuromorphic Computing does exhibit disadvantages, e.g., sneak paths and geometrical imperfections that detriment neuronal STDP. Furthermore, RRAM increases neuronal models’ stochastic behavior. However,
stabilization/synchronization modifications and passive rescuing schemes could maneuver these concerns, since ideal RRAMs fabrication may be unfeasible.

2.2.4.2. SPIN-BASED NC SYSTEMS

A spintronic architecture develops NC architectures based on spin-based devices, a nanoscale CMOS-compatible technology. There are a number of components used in these NC systems, including magnetic devices, domain walls, spin wave devices, and spin-transfer torque devices. As a result of spintronic technology, devices can be packed into a small space and can be customized for optimal performance. In addition, this beyond-CMOS technology performs online learning synapses that display STDP in full networks or modules of NC.

2.2.4.3 FLOATING-GATE TRANSISTOR NC SYSTEMS

Digital storage memories are created using floating gate transistors (FGTs), a technology based on CMOS. These synaptic connections display STDP learning. As well as analog amplifiers, FGTs are suitable for mixed analog/digital circuits. Analog memory cells, such as FGTs, can also be used to store synaptic weights and parameters. Neuronal linear threshold elements, fill neurons, dendrite models, and even fire rate estimation can all be implemented with FGTs in a NC system.

2.2.4.4. PHASE CHANGE MEMORY NC SYSTEMS.

The Phase Change Memory (PCM) device is a fourth common component used in NC systems. By implementing PCM synapsing components, NC systems at high density can achieve STDP. Synaptic weight storage is the primary application of PCM devices within neurons.
2.2.4.5. OPTICAL NC SYSTEMS

It was common practice to use optical NC systems in early literature because of their parallelism. Due to the challenges associated with optical NC storage, popularity has decreased since then. Even though optical and photonic components have been less popular for several years, their potential ultrafast operation has renewed interest in NC systems. There is a moderate level of complexity and programmability in current optical NC systems. Novel optical material and optical/optoelectronic component synapses have recently been incorporated into optical platforms.

2.2.4.6. CONDUCTIVE BRIDGING RANDOM ACCESS MEMORY

In NC systems, CBRAM can be used for the implementation of synapse and neuron structures. Synaptic connections are formed and dissolved using electrochemical properties. This fast, nonvolatile, and low-power technology is nanosized and nonvolatile.

2.2.4.7. ATOMIC SWITCHES

Atomic switches are another nanodevice technology that has low popularity but still makes a significant contribution to NC systems. In atomic switches, metal ions are diffused to create atomic bridging. Synaptic switches are used as synaptic devices in NC systems and provide memristor like plasticity.


2.3. METHODS FOR MODELLING NEUROPHYSIOLOGY

Excitable biological cells harness their characteristic depolarizations and repolarizations functions for signal processing through synaptic communication dynamics. In this section, this biological communication scheme is explored to present circuit design techniques and NC material characteristics that were exploited to emulate neuronal physiology. Emphasis is given on the circuit modelling capabilities that mimic the biological neuron counterpart. For further information about the biological neuron’s dynamics please refer to an appropriate neuroscience or neurophysiology text [53]–[55].

Cells of the nervous, musculoskeletal, and endocrine systems all leverage transmembrane potential attributes for their physiological functions. Modelling each of these differentiated cells could have potential applications in the NC computing field. However, in this work, biological inspiration was taken from visual cortex processing neurons. The characteristic multipolar neurons in this brain region contain various ramified dendritic interneural connections, and one unique cell body (soma) connected to a single signal transmitting axon structure.

The dendrite and soma structures of a biological neuron are connected with up to 10,000 presynaptic neurons through as many as 1,000 trillion synapses [56]. This biological hyperconnectivity permits the neuron to implement its characteristic computational dynamics by balancing incoming excitatory and inhibitory signals. Modelling this synaptic hyperconnectivity through nonconventional computational substrates, such as memristor crossbars, remains a challenge for NC.
In this work’s developed architectures, hyperconnectivity was achieved through an implemented network of 55,296 memristors. Furthermore, the characteristic processing of excitation and inhibition signals in the biological neurons were achieved by balancing feed forward and feed backward crossbar currents.

In the biological neuron, the transmembrane charge accumulation analogical behavior is achieved by weighing the neurotransmitter contributions to the cell’s potential. These neurotransmitters are summed together to create an electric potential, which is fired digitally through the axon. This mixed signal processing approach of the biological cell is best mimicked by mixed-signal circuitry. An analogic capacitor-based charging/discharging accumulation circuit in a CMOS neuron was used to model this dendrite/soma transmembrane potential behavior. Conversely, a digitally firing circuit was used to model the all-or-nothing signal transfer behavior present in the biological axon.

Figure 8 shows a simple synaptic connection between two multipolar biological neurons where the main behaviors to be mimicked in the NC circuit are highlighted. A high-level schematic of how these NC circuit components were connected is presented in Figure 9. The operation of these circuit components and how they modeled the transmembrane potential of the biological neuron is detailed in the next three sections. First, the resting potential is described, including the circuit components that model cell polarization during the refractory period. Then the graded potential based on the stimulus that the neuron received is discussed, including the thresholding behavior that the developed circuit models. Finally, the action potential transporting of a digital axon signal is reviewed.
2.3.1. RESTING POTENTIAL PHYSIOLOGY

In the biological neuron, the resting potential is the electrical voltage across the membrane when the cell is in equilibrium. This potential results from the cell’s sodium-potassium pumps and its transmembrane ionic permeability. In CMOS-based neurons this potential was stored across the plates in an accumulation capacitor. The permeability of the biological membrane was modelled by varying the resistivity values of any conducting path transporting charge in or out of the accumulator. The sodium-potassium pumps, which play a major role in resetting the biological neuron to the resting potential, were achieved through a CMOS logic circuit that shorts the accumulator when the potential must be reset.
In the biological neuron, the resting potential, i.e., the separation of the charges that happens across the membrane, is present on the surface of the neuron. Human neurons have a resting potential of -70 mV. With CMOS neurons the accumulator’s potential must be a value between $V_{DD}$ and $V_{SS}$ due to technology limitations. Hence, in the CMOS neuron design a middle value of $V_{DD}/2$ was used to explore these transmembrane potential dynamics.

The resting potential occurs biologically when ion concentrations differ between the outside and inside of the neuron. The concentration of potassium ions ($K^+$) inside the cell is higher than the negative charge concentration outside the cell due to the sodium ($Na^+$) and chloride ($Cl^-$) ions [53, p. 353]. In the CMOS neuron the concentration of positive ions corresponds to the density of charges due to the presence of electron holes in the accumulating capacitor’s charge.

The flow of electrons into or out of the accumulation capacitor emulates the potential balancing dynamics of the biological sodium-potassium pumps. The charging time constant of the CMOS-neuron’s accumulator was adjusted according to the design requirements of the cell’s refractory period. In the biological neuron, the refractory period lasts 1 millisecond.

Across the biological neuron’s membrane, leak channels allow potassium or sodium to enter or exit the cell to reestablish the resting potential. This behavior was modeled in the CMOS neuron by controlling leak conducting paths between the accumulator and $V_{DD}$ or $V_{SS}$. These charge leak channels were exploited for modeling the inhibition signal dynamics present in the biological neuron.
2.3.2. GRADED POTENTIAL PHYSIOLOGY

The graded potential is the transmembrane voltage that the neuron holds based on the stimulus that the cell has received. Chemically gated channels are responsible for controlling the graded potential in the biological neuron. There is an all-or-nothing rule that governs all action potentials. Hence, a transmitted spike event will either occur completely or not at all based on the graded potential level.

For the biological neuron to generate an action potential, the threshold must be reached, regardless of the stimulus’ magnitude. In the developed CMOS-neurons, energy-efficient thresholding circuits were implemented to model the axon hillock’s behavior. These circuits exploit the voltage transferer characteristics of balanced and unbalanced CMOS inverter gates for their operation. The CMOS neuron’s thresholding circuit was directly connected to the accumulating capacitor of the cell’s body. The switching activation of the thresholding circuit is dependent on the electric potential accumulated in the cell body capacitor.

In both the biological and CMOS neurons, as soon as the threshold is crossed, the neurons fire and an action potential occur. Neurons are able to accumulate potential either upwards or downwards, but neuronal firing will always be triggered once the threshold voltage has been reached. In the biological neuron, the thresholding transmembrane voltage is -55 mV, so any stimulus signal that raises the graded potential above this level causes an action potential. In NC neurons, the thresholding level is application dependent and must be optimized for each case. In this work, thresholding levels varying from 10% to 30% of the $V_{DD}$ swing were explored. Adaptive thresholding was also proposed as a plasticity scheme.
2.3.3. ACTION POTENTIAL PHYSIOLOGY

Biological neurons emit action potentials through their axons when the transmembrane potential has exceeded the threshold level required by the axon hillock. Consequently, voltage-gated channels control the action potential transmission through the biological axon. The spikes of every action potential are identical in accordance with their all-or-nothing firing behavior. This physiology is best modeled through a fully digital circuit module in the CMOS-neuron.

In the biological neuron, reaching the thresholding voltage of -55 mV activates the sodium (Na\(^+\)) voltage-gated channels in the axon. Through these channels an enormous amount of Na\(^+\) enters the cell, depolarizing it. These voltage-gated channels generate an increasing transmembrane potential which eventually inactivates the sodium influx at +30 mV. At this point, voltage-gated potassium (K\(^+\)) channels activate, allowing potassium ions to escape the neuron, repolarizing it. As potassium rushes out of the neuron, the transmembrane potential surpasses the resting potential level of -70 mV before the voltage-gated K\(^+\) channels are deactivated. This causes a refractory period during which the neuron is unable to fire an action potential. The leak channels and membrane permeability, restore the transmembrane voltage to its resting potential level of -70mV [55, p. 4].

In the CMOS neuron, the characteristic rapid depolarization and repolarization of the biological counterpart was achieved through the development of a firing circuit module. The voltage accumulation properties of the axon were modeled with a capacitor similar to the one used for the cell body circuit. The voltage-gated sodium channel activation was modeled by a CMOS logic circuit that shorted the
accumulator when an action potential must be fired. The repolarization properties of the biological axon were implemented in the CMOS neuron by a limiting resistor controlling the accumulator’s charge time.

The action potential duration can be controlled in the CMOS neuron by varying the passive elements in the firing module. In a typical nervous cell, the action potential has a duration of 1 millisecond. In the CMOS neuron, this spike time duration is application specific and should be optimized for circuit efficiency.

In the biological neuron, the depolarization of one section of the axon due to the rapid sodium influx causes each subsequent leading segment to transmit the action potential. Upon influx of Na\(^+\), the signal is transmitted away from each section through ionic charge repulsion. This results in a change in voltage in the leading section of the axon. Therefore, the next sodium channel will be opened as soon as the leading segment hits the threshold level.

Signal backpropagation along the axon is not possible, due to ongoing repolarization on the trailing axon cross-section. Hence, the action potential signal travels only along the axon away from the soma but never backwards. This behavior happens because the trailing segment is undergoing repolarization by the active potassium voltage-gated channels. As a result, there is a positive feedback loop in the leading cross section where more sodium comes out and activates the next sodium channels. There is, however, an undershoot in the trailing segments, reestablishing the resting potential and allowing the signal to travel down the axon.

Mammalian neurons with exceptionally long axons utilize myelination sheaths provided by the Schwann cells as a mechanism for axonal signal acceleration.
These myelination structures are analogous to the protective coating of a conductor. Myelination allows the action potential to travel faster and further without experiencing dampening. There are, however, exposed neuron sections along the biological axon called Ravinder nodes.

Located between the myelin sheaths of a long axon, the Ravinder nodes improve conduction by permitting an optimized signal transmission mechanism called saltatory conduction. Rather than transmitting the action potential to each subsequent axon cross-section, a myelinated axon transmits the signal from one node to the next. This happens because voltage-gated channels are found only at the nodes, but not under the myelinated sections of the axon. This alternation of myelin sheaths and nodes facilitates the transmission of the electrochemical gradient along the axon.

The signal propagation mechanics of the biological axon echoes the operational dynamics of the CMOS inverter chains used for digital VLSI design. Each inverter gate in the chain has an intrinsic input capacitance that must be charged before triggering operation. The input capacitance of an inverter is a result of all the capacitances connected to the gate, i.e., drain/source overlap, junction, and nonlinear gate-oxide channel capacitances. Therefore, by adjusting transistor sizes along the chain, an optimized CMOS-axon can be designed. Moreover, through proper gate sizing optimization techniques, delay propagation along the inverter chain can be minimized to improve circuit response times. This emulates the action potential transmission optimization that the myelination permits in the biological neuron counterpart through saltatory conduction.
2.3.4. SYNAPTIC CONNECTION PHYSIOLOGY

In the biological neuron, the graded potential occurs due to information gathered from other nerves across synaptic connections. In the postsynaptic neuron, chemically gated channels in the dendrites open up when the presynaptic neuron releases neurotransmitters from its axonal buttons. These neurotransmitters bind to proteins on the postsynaptic side and start a flow of ions across the membrane that modifies the accumulated graded potential.

The neurotransmitters present in the biological synapse can either be excitatory or inhibitory, i.e., move the postsynaptic cell’s graded potential toward or away from the threshold. Sodium is allowed into the neuron through the dendrite when an excitatory neurotransmitter opens the chemically gated channel. Upon entering the neuron, sodium causes depolarization towards the threshold. In contrast, inhibitory neurotransmitters and receptors move the graded potential away from the threshold. Potassium rushing out and chloride entering the cell through chemically gated channels, cause postsynaptic neuronal inhibition.

In the biological neuron, synaptic plasticity is achieved by modifying the amount of neurotransmitter receptors in the dendrites. When neurons fire sequentially, synaptic connections are strengthened by increasing excitatory neurotransmitter receptors and decreasing inhibition receptors. Conversely, during synaptic weakening, excitatory receptors are decreased, and inhibition receptors are increased. This is analogous to the long-term potentiation or depression that can be achieved by adjusting the conductance state in a memristor-based synapse. A dendritic circuit was designed to control this synaptic adaptation in the NC neuron.
2.4. DEVELOPED NC ARCHITECTURES

The complete design of a small-scale proof-of-concept neuromorphic computing radionuclide classification system prototype is presented in this section. Their design schematics are discussed with an operational descriptive emphasis. SPICE-based system simulation optimization results are shown. A simulation result interpretation for the developed architectures is given. A contrast analysis between the architectures’ schematic and finally a layout-based modelling is also demonstrated in this section.

The optimized neuromorphic computing architectures were submitted for fabrication through MOSIS, using their 0.5 µm technology node. Testing results of these fabricated architectures are discussed. Based on the classifying NC architectures’ characterization result, a localization prototype has been proposed. The localization architecture’s schematic representation is shown and initial SPICE simulation is performed.

2.4.1. MODELLING POWER CONSUMPTION USING RASPBERRY PI

The proposed nuclide identification was evaluated by using a small single-board Raspberry Pi microcontroller to get a baseline of the power consumption. A radionuclide from a library of 27 common nuclide from the NNDC’s Nuclear Wallet Card’s document was used as test signal.
2.4.1.1 MEASUREMENT METHOD

The average power consumption was measured by interrupting the Raspberry Pi’s USB connection and inserting a shunt in the power line as shown in Figure 10. Using a 16-bit Data Acquisition module the Raspberry Pi’s voltage values were constantly monitored. The data acquisition program monitoring the power consumption parameters was written in NI LabView and is shown in Figure 11. The power consumption of the Raspberry Pi was calculated by:

\[ P = \frac{V_1 \cdot (5.2V - V_1)}{R} \]  \hspace{1cm} (1)

Idle measurements were also taken to account for the background power consumption of the Raspberry Pi.

Figure 10. Connection approach implemented to measure power consumption of NC classifying algorithm.
The system’s mean time for nuclide identification with the Raspberry Pi was measured to be 49ms, with a mean energy consumption of 96.2mJ and a standard deviation of 21.6mJ. It was measured that the Raspberry Pi demands an average power of 1.5W.

2.4.2. NC ARCHITECTURE SCHEMATICS

Figure 12 shows a high-level illustration diagram of the memristor network implemented as the “computing core” in the developed architectures. This network is applied to identify radionuclide gamma ray spectra. The developed architectures implementing this memristor crossbar further establish promising unconventional paradigms for non-conventional computation approaches. Figure 13 shows the schematic representation of the developed memristor crossbar, which is used in higher level diagram representations throughout the rest of this work. The cross-point bit elements represent a memristor element. The row (→) and column (↑) nanowires are represented by lines.
2.4.2.1. ANALOG NC CIRCUIT SCHEMATICS

A high-level diagram representation of the developed small-scale proof-of-concept fully analog NC system is shown in Figure 14. This architecture was designed to analyze 5x6 systems, i.e., 5 presynaptic neurons connected to 6 postsynaptic neurons for a total of 30 synaptic connections.

This analog signal system uses two crossbars. The first crossbar stores the dictionary weights, and the second transports inhibitory signals. In this architecture, the inhibition connection weights ($G_{m,n}$) are a function of the crossbar
dictionary (Φ) weights. The relationship between this two memristor crossbars is given by the dot product of the m\textsuperscript{th} neuron’s weights (Φ\textsubscript{m}) and the n\textsuperscript{th} neuron’s weights (Φ\textsubscript{n}).

\[ G_{m,n} = <\Phi_m, \Phi_n> \] (2)

The input s(t) is a vector containing all the detector channel energy bin analog voltage signals. The crossbar contains the neuronal receptive fields weights Φ, mapped into each memristor state. The VMM determines the initial activation b(t). An inverting amplifier operating acts as a virtual ground adding (Σ) the current contributions from each memristor. The internal state capacitances are given by u(t). A thresholding module containing a differential amplifier provides the circuit output \( a(t) \), which is then fed back through the interneuron connections G to determine the inhibition signals.

2.4.2.2. MIXED SIGNAL NC CIRCUIT SCHEMATICS

The developed mixed-signal architecture shown in Figure 15 has part of its running time dedicated to calculating inhibitory G\textsubscript{m,n} forces for hardware optimization. Whenever an output neuron fires and a spike is generated, that spike passes feedback currents from the corresponding firing column neuron back through the same dictionary crossbar (Φ), charging the inhibition capacitors in the input row neurons. Intuitively, the charges on these capacitors indicate how well represented the corresponding input signal is in the current reconstruction. Overrepresented input signals are suppressed.
The input $\mathbf{s}(t)$ is a vector containing all the detector channel energy bin voltage signals. Unlike the fully analog architecture, this circuit has its input signal $\mathbf{s}(t)$ mapped using spike-frequency-encoding as described in Appendix 6.

The single crossbar in this architecture contains the neuronal receptive fields weights $\Phi$, mapped into each memristor state. The VMM determines the initial activation $b(t)$. The internal state capacitances are given by $u(t)$. A thresholding module based on unbalanced inverters provides the circuit output $a(t)$, which is then fed back through the same crossbar to determine the inhibition signals.

![Figure 15](image)

Figure 15. High-level schematic of developed NC mixed signal architecture.

### 2.4.3. ENERGY EFFICIENCY HARDWARE SCALABILITY

A power consumption reduction was achieved in the mixed-signal architecture by implementing a system adaptation permitting linear scalability. The fully analog NC circuit has quadratic hardware scalability, because an additional memristor is required to establish feed backward inhibition paths for each neuron pair. This hardware scalability complexity was resolved in the mixed-signal implementation by using a fraction of the computation time to calculate inhibitory signals through feedback currents from the memristor crossbar.
When a postsynaptic neuron in the mixed-signal architecture has charged above its firing threshold, inhibition signals are sent back to the presynaptic cell through the same memristor cross point. Hence, the inhibition signal’s magnitude depends on the synaptic strength of each neuron pair. This inhibition current will therefore charge the accumulation capacitors in the presynaptic cells. This mechanism permits system conversion by suppressing overrepresented inputs.

2.4.4. DEVELOPED CMOS NEURONS FUNCTIONALITY

The main cells within the developed spiking mixed signal neuromorphic detection system are the presynaptic input and the postsynaptic output neurons.

The presynaptic neurons are connected to the memristor crossbar’s rows and serve as the system’s inputs. They receive the spike-train encoded radionuclide spectral data as their input. These signal receiving neurons are analogous to a sensory neuron in a biological system.

The postsynaptic neurons are connected to the memristor crossbar’s columns and serve as the system’s outputs. These neurons contain the processing logic units of the system. Since the switching activity required to implement the neuronal logic is higher, the power demand of postsynaptic neurons is more significant than that of the presynaptic neurons.

In biological neurons, an action potential is fired down the axon only when the accumulated graded potential has surpassed a given threshold value. This is modeled in both the CMOS neuron designs by implementing accumulation state capacitors. Once a certain threshold voltage is accumulated in the state capacitor, the firing module is activated. This sends a signal down the CMOS neuron’s axon.
In the biological cell, the thresholding is done by the axon hillock. In the CMOS neurons, the thresholding is implemented through an inverter-based nonlinear thresholding circuit.

The postsynaptic circuit cell also contains a Leaky-Integrate-and-Fire (LIF) neuron setup, with the state capacitor connected directly through a transmission gate to the nanowire crossbar’s column. The transmission gate is analogous to the cell membrane in the biological neuron’s dendrite or soma.

The cell membrane in the biological neuron controls the flow of ionic charges by activating chemically gated channels when receiving a neurotransmitter. In the CMOS neuron, transmission gates regulate the amount of electric charge entering the cell’s accumulator.

A Schmitt trigger setup allows enough firing time, such that the output state capacitor drains sufficiently when the neuron fires and resets all the accumulated potentials in all the system neurons. Additionally, the firing neuron sends an inhibition signal back through its axon, this is modeled by a pull-up transistor that sends the signal back to the crossbar.

During each neuron firing, an inhibition current flows back through the dictionary into the presynaptic circuit cells. Since this charging occurs through a feed backward path, the inhibition current flowing back into each presynaptic circuit is proportional to how well that row activated the firing column neuron.

The row circuit cell, which acts as the sensing neuron, has a more intricate processing logic. It too has a cell body (soma) to charge, which is also modelled by a capacitor. However, in this case, the cell is actively transmitting a signal until
it charges up with the feed backward inhibition signal and blocks transmission. These row input cells discharge whenever an input spike signal arrives and charge when an inhibition output spike is occurring, i.e., the inhibition capacitor charge models how much the input is represented in the output. This capacitor increases its inhibition voltage with the actual activity coming from the output column cells through the crossbar junctions. The capacitor discharges through an inhibition resistor to restore the resting potential.

2.4.2.3. SPICE BASED SIMULTATIONS RESULTS

As demonstration of the NC architecture, a simulation of a simplified system was obtained using SPICE. For circuit-level simulation, a simple LCA system containing 6 neurons with 5-element receptive fields (RFs) was constructed, where the implemented dictionary was a mathematical representation of 6 classes, including 10000, 01000, 00100, 00010, 00001, and 11111, numbered 1 through 6, respectively. It is expected that, e.g., if 01001 is given as input, the LCA should identify both classes 2 and 5 as the input. Both architectures were assessed with this task. In both architectures, the circuit successfully identified both classes. Figure 16 and Figure 17 show the time response diagrams of each system.
Figure 16. Simulation results for developed proof-of-concept analog NC architecture

Figure 16 shows SPICE simulation result for the analog architecture. The two representative outputs (2 and 5), shown in red, successfully converge into -1 within 30 ns. The other outputs, shown in green and blue, converge into 0. The output is negative due to the inverting amplifiers in the column cells.
Figure 17. Simulation results for developed proof-of-concept mixed signal NC architecture

Figure 17 shows SPICE simulation of the spiking architecture. The two representative outputs (2 and 5) are detected. The other outputs (1, 3, 4, and 6) are inhibited. Other combination of input signals and results are shown in Table 1.

Table 1. Classification results showing likelihood of the input to be closest to the class 1-6

<table>
<thead>
<tr>
<th>Input Signal</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Class 5</th>
<th>Class 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,0,0,0]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[1,0,0,0]</td>
<td>1.0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.2000</td>
</tr>
<tr>
<td>[0,1,0,0]</td>
<td>0</td>
<td>1.0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.2000</td>
</tr>
<tr>
<td>[0,0,1,0]</td>
<td>0</td>
<td>0</td>
<td>1.0000</td>
<td>0</td>
<td>0</td>
<td>0.2000</td>
</tr>
<tr>
<td>[0,1,1,0]</td>
<td>0</td>
<td>0.9907</td>
<td>0.9907</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[0,1,1,1]</td>
<td>0</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[1,1,0,0]</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[1,1,1,0]</td>
<td>0.9978</td>
<td>0.9978</td>
<td>0.9978</td>
<td>0.9978</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[1,1,1,1]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1022</td>
</tr>
</tbody>
</table>
2.4.5 NC ARCHITECTURE LAYOUTS

The following section assumes familiarity with the CMOS process layers in Layout Tools such as Tanner L-Edit or Cadence. Readers unexperienced in this basic VLSI design concept please refer to Appendix 7 for a color key handout to aid in the interpretation of all designs provided henceforth.

2.4.5.1. ANALOG NC CIRCUIT LAYOUT

Figure 18 shows the layout representation of the developed small-scale proof-of-concept fully-analog neuromorphic architecture’s layout. Both architectures were fabricated to analyze 5x6 systems, i.e., 5 presynaptic neurons connected to 6 postsynaptic neurons for a total of 30 synaptic connections. This developed analog signal system uses two crossbars, one stores the dictionary weights, and a second one transports inhibitory signals characteristic of a Locally Competitive Algorithm (LCA) architecture. In this system, the inhibition Connection crossbar \( G_{m,n} \) weights are a function of the crossbar dictionary \( \Phi \) weights. Their relationship is given by:

\[
G_{m,n} = \langle \Phi_m, \Phi_n \rangle \quad (3)
\]
Figure 18 shows LCA analog-circuit layout implementation. This circuit integrates a system of nonlinear ordinary differential equations loosely based on [81]. This circuit dynamics are used to calculate the oscillation dynamics of each neuron in the circuit.

\[
\dot{u}_m(t) = \frac{1}{\tau} [b_m(t) - u_m(t) - \sum_{n} G_{m,n} \alpha_n] \quad (4)
\]

The input \( s(t) \) is a vector containing all the detector channel analog voltage signals. The crossbar contains the neuronal receptive fields weights \( \Phi \), mapped into each memristor state. The VMM determines the initial activation \( b(t) \):

\[
b_m(t) = \langle \Phi_m, s(t) \rangle \quad (5)
\]

An inverting amplifier operating as a virtual ground sum (\( \Sigma \)) the current contributions from each memristor. The internal state capacitances are given by \( u(t) \). A thresholding module containing a differential amplifier provides the circuit output \( a(t) \), which is then fed back through the inhibitory connections \( G \) to determine the inhibition signals.
2.4.5.2. MIXED SIGNAL NC CIRCUIT LAYOUT

In Figure 19 a mixed-signal architecture has part of its running time dedicated to calculating inhibitory $G_{m,n}$ forces for hardware optimization. Whenever an output neuron fires and a spike is generated, that spike passes feedback currents from the corresponding firing column neuron back through the same dictionary crossbar ($\Phi$), charging the inhibition capacitors in the input row neurons. Intuitively, the charges on these capacitors indicate how well represented the corresponding input signal is in the current reconstruction; overrepresented input signals will be suppressed.

Figure 19 shows the layouts of the neurons implemented in the mixed-signal architecture and Figure 20 and Figure 21 show the main modules that compose the CMOS neurons developed in the architectures.

![Figure 19. L-Edit layout of developed NC mixed signal architecture showing the control circuit, the presynaptic and postsynaptic cells in a box.](image-url)
2.4.5.3. ENERGY EFFICIENCY RESULTS

The energy and area efficiency of the mixed-signal CMOS neurons was optimized and is shown in Figure 20 and Figure 21. Namely, these CMOS integrate-and-fire neurons have a minimum number of transistors. The main energy consumption of the neurons occurs due to the charge/discharge of the capacitors. Hence, it aimed to minimize the capacitor dimensions to optimize the energy efficiency of the neurons. The spiking mixed-signal neurons have accumulation, firing and idle modes of operation. Accumulation occurs when the neuron receives and integrates the charge carried by a spike signal through the memristor synaptic weight. The postsynaptic column neuron operates primarily in accumulation when a signal is sent by the presynaptic neuron through the crossbar. Presynaptic row neurons only operate in accumulation when receiving inhibition signal feedback through the memristor crossbar.

Figure 20. CMOS presynaptic neuron with adaptation module for memristor potentiation or depression.
Firing occurs when the neuron produces a spike. The generation of the post-neuron output spike determines the energy consumption. The output spike characteristics determine the energy consumption. The presynaptic row cell operates primarily in firing mode, transmitting the spike signals when the row is not inhibited. The postsynaptic column cell fires only when the neuron has accumulated charge above the threshold level, generating an inhibition spike signal output that is feedback through the crossbar.

The idle energy is dominated by CMOS leakage in the 0.5µm process. The CMOS neuron does not consume energy during learning unlike the memristor synapse. Table 2, Table 3 and Figure 22 show the per spike energy consumption of the input row and output column neurons.
Table 2. Energy consumption of presynaptic CMOS neurons.

<table>
<thead>
<tr>
<th>Neuron Phase</th>
<th>Energy per Spike</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulation</td>
<td>1.1 pJ</td>
</tr>
<tr>
<td>Idle</td>
<td>0 pJ</td>
</tr>
<tr>
<td>Firing</td>
<td>17.2 pJ</td>
</tr>
</tbody>
</table>

Table 3. Energy consumption of postsynaptic CMOS neurons.

<table>
<thead>
<tr>
<th>Neuron Phase</th>
<th>Energy per Spike</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulation</td>
<td>21.8 pJ</td>
</tr>
<tr>
<td>Idle</td>
<td>3.5 pJ</td>
</tr>
<tr>
<td>Firing</td>
<td>140.6 pJ</td>
</tr>
</tbody>
</table>

Figure 22. Energy consumption per spike in the presynaptic and postsynaptic neurons.
2.4.5.4. NC ARCHITECTURE FABRICATION

The fabricated prototypes are shown in Figure 23, Figure 26, and Figure 27.
The pinouts are given in Figure 24, Figure 25, Table 4 and Table 5.

**Figure 23.** Prototypes implemented an ON Semiconductor B5 0.5-micron process and on a PCB with off-the-shelf components.

**Figure 24.** IC design containing the individual LCA & SSLCA Module: transmission gate (A2-A4), axon module (A5-A6), thresholding module (A7-A8), firing module (A8-A10), capacitor (A11-A12) operational amplifier (A13-A16), transceiver (A18-A23), presynaptic cell (A24-A26), inhibition module (A27-A30), and postsynaptic cell (A31-A32).

**Figure 25.** IC design containing a 5x6 analog LCA classifier with I/P: B2-B6, O/P: B7-B12, bias: B1 & B15-B17; and a 5x6 spiking SSLCA classifier with I/P: B18-B22, O/P: B23-B28, bias: B1 & B17, capacitor (B13-B14), resistor (B29-B30), and amplifier (B31-33).
| A1 – GND | A17 – VDD |
| A2 – Transmission Gate In/Out | A18 – Transceiver 1 Rx |
| A3 – Transmission Gate Ctrl | A19 – Transceiver 2 Rx |
| A4 – Transmission Gate In/Out | A20 – Transceiver 2 Vx |
| A5 – 5-stage Inverter Chain Input | A21 – Transceiver 2 Tx |
| A6 – 5-stage Inverter Chain Output | A22 – Transceiver 1 Tx |
| A7 – Thresholding Module Input | A23 – Transceiver 1 Vx |
| A8 – Thresholding Output/Firing Input | A24 – Row Cell Output |
| A9 – Not Connected | A25 – Row Cell Spike Input |
| A10 – Firing Module Output | A26 – Row Cell Firing Input |
| A11 – Capacitor Cathode | A27 – Inhibition Module Spike Input |
| A12 – Capacitor Anode | A28 – Inhibition Module Firing Input |
| A13 – Amplifier’s Output | A29 – Inhibition Module Charge Output |
| A14 – Amplifier’s Inverting Input | A30 – Inhibition Module Crossbar Output |
| A15 – Amplifier’s Non-Inverting Input | A31 – Column Cell Output |
| A16 – Amplifier’s Biasing Input | A32 – Column Cell Input |

| B1 – GND | B17 – VDD |
| B2 – LCA ROW INPUT 1 | B18 – SSLCA ROW INPUT 1 |
| B3 – LCA ROW INPUT 2 | B19 – SSLCA ROW INPUT 2 |
| B4 – LCA ROW INPUT 3 | B20 – SSLCA ROW INPUT 3 |
| B5 – LCA ROW INPUT 4 | B21 – SSLCA ROW INPUT 4 |
| B6 – LCA ROW INPUT 5 | B22 – SSLCA ROW INPUT 5 |
| B7 – LCA COLUMN OUTPUT 1 | B23 – SSLCA COLUMN OUTPUT 1 |
| B8 – LCA COLUMN OUTPUT 2 | B24 – SSLCA COLUMN OUTPUT 2 |
| B9 – LCA COLUMN OUTPUT 3 | B25 – SSLCA COLUMN OUTPUT 3 |
| B10 – LCA COLUMN OUTPUT 4 | B26 – SSLCA COLUMN OUTPUT 4 |
| B11 – LCA COLUMN OUTPUT 5 | B27 – SSLCA COLUMN OUTPUT 5 |
| B12 – LCA COLUMN OUTPUT 6 | B28 – SSLCA COLUMN OUTPUT 6 |
| B13 – Capacitor Anode | B29 – Resistor |
| B14 – Capacitor Cathode | B30 – Resistor |
| B15 – VDD/2 | B31 – Amplifier Non-Inverting Input |
| B16 – Amplifier Biasing Terminal | B32 – Amplifier Inverting Input |
Figure 26. ASIC L-Edit layout containing individual test modules.

Figure 27. ASIC L-Edit layout containing analog & mixed signal NC architectures.
2.4.5.5. NC ARCHITECTURE TESTING

Testing of the NC architectures is performed modularly by assessing the characteristics of each component in the system.

![Figure 28. Gate-level schematic of developed presynaptic input neuron for the NC mixed-signal architecture.](image)

**Figure 28.** Gate-level schematic of developed presynaptic input neuron for the NC mixed-signal architecture.

![Figure 29. L-Edit layout of developed presynaptic input neuron for the NC mixed-signal architecture showing the inhibition resistor and capacitor in boxes.](image)

**Figure 29.** L-Edit layout of developed presynaptic input neuron for the NC mixed-signal architecture showing the inhibition resistor and capacitor in boxes.

Figure 28 and Figure 29 show the presynaptic input cell schematic and its layout. This sensory neuron receives the rate-encoded spike signal and the control input $\overline{\text{Firing Any}}$, where the bar means negation. The neuron contains an inhibition capacitor $C_{\text{inh}}$, which charges when the cell’s signal is accurately represented in the system’s output. When any of the output neurons is firing, the
control signal permits the charging of the inhibition capacitor through a feed backward path across the memristor array.

**Figure 30.** Gate-level schematic of developed postsynaptic output neuron for the NC mixed-signal architecture.
Figure 31. L-Edit layout of developed postsynaptic output neuron for the NC mixed-signal architecture.

Figure 30 shows the column circuit output cell schematic and Figure 31 shows its layout. Incoming charge from the crossbar input (Cb) charges the accumulating internal state capacitor $C_{st}$ through the transmission gate. When the capacitor's charge surpasses a predetermined level, the operational thresholding circuit activates the cell and fires for a time interval determined by the firing resistor $R_f$ and capacitor $C_f$. The Firing Self outputs of all the system’s column neurons are sent to a NOR circuit that determines the $\neg$Firing Any Control signal, which determines if the memristors are conducting in feed forward (i.e., row cells to column cells) or a feedback direction.
Figure 32. Device-Level schematic of developed thresholding modules for the analog NC architecture (left) and the mixed signal NC architecture (right).

Figure 33. L-Edit layout of developed thresholding modules for the mixed signal NC architecture.

Figure 34. VTC results for unbalanced inverters implemented in thresholding module.
Some of the main modules that compose the implemented CMOS neurons in the NC architecture are shown in Figure 32 through Figure 37. Figure 32 shows blowup versions of the thresholding circuits $T_{\lambda}(\cdot)$ implemented in the circuits. Figure 33 shows an adapting thresholding circuit that exploits the balance between the p-channel and n-channel mosfets in an inverter gate to change its VTC according to the thresholding requirement of the architecture. Figure 34 shows a testing result for this thresholding circuit showcasing the functionality of the circuit.

Figure 35 shows the inverted chain used in the CMOS neuron to speed up the signal; its behavior mimics the saltatory signal conduction shown. Figure 36 and Figure 37 show an operational amplifier used in the system to integrate the contribution charges from every synapse and to operate the crossbar with the double element or bias column weight mapping approach.
Figure 36. Device-level schematic of the low-power operational amplifier designed for the analog NC architecture.

Figure 37. L-Edit layout of the low-power operational amplifier designed for the analog NC architecture.

2.5. CHAPTER SUMMARY

CMOS-based neurons for a radionuclide detection neuromorphic computing system were designed and tested using 0.5µm technology. The developed neuromorphic architectures are capable of performing simple classification tasks with minimal energy consumption.
III. SYSTEM ACCURACY & RELIABILITY ANALYSIS

Memristor-based neuromorphic computing architectures’ online-learning potential promises a triumphant post-Moore era computer. Memristors’ ability to modify and store weights like a biological synapse [57]–[59], allows them to perform certain computational tasks while consuming remarkably less power than conventional computing architectures [5], [60]. Moreover, memristor-based systems are especially well suited for certain signal, image, and pattern detection algorithm hardware implementations [21], since they can generalize and classify input signals, for which they have not been trained. In addition, a memristor NC platforms’ reconfigurability offers powerful self-healing properties that can provide protection against radiation-induced upsets or transients, making them ideal for radionuclide detection tasks.

A neuromorphic computing architecture’s overall performance is determined by vector matrix multiplications (VMMs) [61]. The NC systems’ energy consumption and computation speed are heavily influenced by the VMM operation [62]. The use of memristor arrays for VMM dynamics presents a promising possibility for reducing energy consumption [63]. Previous implementations have demonstrated significant energy efficiency reduction [64]. Memristor-based NC architectures can consume several orders of magnitude less energy than traditional CMOS ASIC [65]–[67], particularly when larger arrays are employed. Nevertheless, when performing VMMs with memristor arrays, certain circuit issues may occur due to fabrication errors and parameter variations in the nanoscale fabrication process.
3.1. MOTIVATION FOR SYSTEM FAULT-TOLERANCE ASSESSMENT

Harnessing memristors for memory-processing collocation is favorable for NC architectures. However, any computing platform employing strong variation devices such as memristors requires substantial fault tolerance [68]–[70]. Hence, NN hardware implementations with device nonideality immunity would be more effective at leveraging memristor characteristics. Through neuronal biomimicry, spiking neural networks (SNN) have achieve enhanced energy efficiency [6].

The NC community is particularly enthusiastic about nano-CMOS device integration circuits for their neuron/synapse modeling faculties. CMOS-exclusive applications do exist. However, these architectures can only implement a limited number of synapses due to transistor density [47], [71]. Nanodevice integration in neuromorphic circuits ameliorates this problem by wielding the memristor’s memory-processing collocation characteristic. Memristors have emerged in with functions, such as resistive RAMs [13], [68], and adaptive transistors [69], [72].

Biological neuronal learning is governed by spike timing dependent plasticity (STDP) [73]–[75]. Computer systems with nanoscale synapses and CMOS neurons could be developed to efficiently implement STDP or other brain-inspired learning algorithms. Currently, this concept is getting an in-depth look [76]. It must, however, show its sustainability, especially in the context of the material variability problem that affects all memristor technologies [77], [78]. Electronic constraints differ greatly from biological constraints, so biologically-accurate STDP-learning may not be appropriate to address memristor faults. Hence, a suitable learning algorithm must be selected to maneuver nonidealities.
3.2. PREVIOUS WORK ON NC TRAINING ALGORITHMS

A neuromorphic computing architecture must balance material/device constraints and its implemented NC learning algorithm. Accordingly, a certain training scheme may be preferable for a given particular NC application. Hence, a background discussion of the available NN algorithms is necessary [79].

An important consideration for algorithm selection is the specific application's requirements. Is offline training sufficient or is online learning necessary? Will computational substrate devices exercise on-chip or off-chip adaptation? Should the training scheme be supervised, unsupervised, or a combination of both?

NN hardware training algorithms with device constraint limitation adaptation is an active research area. For radionuclide detection applications, on-chip learning algorithms are preferable, because these training approaches may help to overcome radiation induced transients in the computational substrate. Furthermore, chip-in-the-loop and hardware-oriented learning algorithms must also be considered. Figure 38 shows algorithms that were considered for the NC architecture training scheme requirements.

![Figure 38](image)

**Figure 38.** Training-approach NC algorithm categories.

3.2.1. SUPERVISED LEARNING ALGORITHMS

Offline supervised on-chip learning schemes are commonly applied to train NC hardware. The learning algorithms considered for training implementation in this radionuclide detection NC application were back-propagation and evolutionary.
3.2.1.1. BACKPROPAGATION LEARNING ALGORITHMS

The most popular supervised training algorithm used in NC architectures is Back-Propagation (BP) Learning. NC architectures implementing BP learning must consider constraints such as memristor non-idealities. Other gradient descent optimization methods based on simplified BP algorithms can address memristor faults [80]–[84]. Moreover, there are also chip-in-the-loop back propagation training methods where most learning occurs off-chip but inference is on-chip to address the device and material characteristic variability [85]–[88]. BP algorithms confront issues that make difficult their hardware implementation. However, hardware BP implementations are costly and restrictive on neuron models, network and topology.

3.2.1.2. NATURE-INSPIRED LEARNING ALGORITHMS

Nature-Inspired approaches such as evolutionary (EV) and genetic algorithms [89] rely on the evaluation of the current network and use the chip during the training process through chip-in-the-loop implementations. The current network evaluations optimize the NC architecture for the specific peculiarity of their devices’ characteristics. Evolutionary inspired HW architectures are popular because they are independent of the chosen models and device characteristics. Furthermore, these architectures implement off-chip learning keeping the NC devices in the loop.
3.2.2. UNSUPERVISED LEARNING ALGORITHMS

On-chip architectures with unsupervised learning harness the full potential of an NC system. Early implementations of unsupervised learning algorithms were based on self-organizing maps, Hebbian learning and STDP [90]–[93].

3.2.2.1. HEBBIAN LEARNING

Hebbian learning rules are a popular learning on-line approach. This learning rule explained in a single statement is: “Neurons wire together if they fire together” An extension of Hebbian approach is STDP which considers the causation aspects of learning and requires temporal precedence for long temporal potentiation [94].

3.2.2.2. SPIKE-TIMING DEPENDENT PLASTICITY

Spike-Time-Dependent-Plasticity (STDP) is the learning algorithm seen in biological neurons. STDP is also the most common unsupervised learning mechanism implemented in NC. If the presynaptic neuron fires shortly before or after the postsynaptic neuron, then the synapses weight increases or decreases respectively with a magnitude change inversely proportional to time between fires. Full STDP applicability has yet to be demonstrated in NC architectures, but there exist some custom STDP potentiation and depression circuits of NC platforms [95]–[99].

3.2.3. LEARNING ALGORITHMS FOR SPIKING NEURAL NETWORKS

Spiking Neural Networks (SNN) have gained most popularity among the NC architectures because they are easy to train and utilize for computation. SNNs tailor for low power/energy efficient NC applications. Moreover, SNN demonstrates
an event-driven computation that is closer to the biological neural network than other architectures[94]. SNN architecture implementations have demonstrated STDP learning but their full potential is yet to be realized. Current neuromorphic spiking system still use non-spiking network models and learning algorithm which are unreliable to harness the full potential of an SNN architecture.

3.2.4. WHICH TRAINING ALGORITHM TO CHOOSE?

When choosing a learning algorithm to implement an NC system, several factors must be considered. Regarding the algorithm’s hardware applicability, NC systems using Hebbian and STDP learning in their architecture are not well developed whereas supervised approaches like BP and EV have a variety of demonstrated applications. Unsupervised Hebbian and STDP approaches are bio realistic whereas BP learning is not and EV algorithms are only sometimes related to BNNs. Both unsupervised learning mechanisms do provide the characteristic fault tolerance inherent to NC systems. However, supervised algorithms are more restrictive with respect to how they approach material or device constraints and require adaptation. Regarding model options, both Hebbian and STDP learning are restrictive, whereas for supervised learning only BP is model limiting. On-chip learning is possible with evolutionary and Hebbian algorithms, and requires adaptation for BP implementations, but STDP is significantly complex to implement. Supervised learning occurs primarily offline with the use of feedback connections, whereas unsupervised mechanisms implement online training. BP has a faster convergence time than evolutionary algorithms but for Hebbian and
STDP the training time varies. Table 5 summarizes the decision criteria between the four learning approaches considered.

**Table 6. Training algorithm decision matrix.**

<table>
<thead>
<tr>
<th>Training Schemes</th>
<th>Back-Propagation</th>
<th>Nature-Inspired</th>
<th>Hebbian</th>
<th>STDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applicability</td>
<td>Broad</td>
<td>Broad</td>
<td>Narrow</td>
<td>Narrow</td>
</tr>
<tr>
<td>Biological Realism</td>
<td>Unrealistic</td>
<td>Moderate Bio realism</td>
<td>Bio realistic</td>
<td>Bio realistic</td>
</tr>
<tr>
<td>Material/Device Constraints</td>
<td>Requires Adaptation</td>
<td>Requires Adaptation</td>
<td>Fault-Tolerant</td>
<td>Fault-Tolerant</td>
</tr>
<tr>
<td>Model Options</td>
<td>Restricted</td>
<td>Unrestricted</td>
<td>Restricted</td>
<td>Restricted</td>
</tr>
<tr>
<td>On/Off-Chip Training</td>
<td>On-Chip w/ adaptation</td>
<td>On-Chip</td>
<td>On-Chip</td>
<td>Off-Chip</td>
</tr>
<tr>
<td>Online/Offline Training</td>
<td>Offline</td>
<td>Offline</td>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>Training Time</td>
<td>Fast Convergence</td>
<td>Slower Convergence</td>
<td>Application Specific</td>
<td>Application Specific</td>
</tr>
</tbody>
</table>

### 3.3. METHODS FOR TESTING THE SYSTEM’S FAULT-TOLERANCE

Due to the fact that the manufacturing quality of memristors and their yield is poor, it is indispensable to analyze faults and model neuromorphic computing architectures based on memristors. Non-ideal conditions have been studied for neuromorphic systems [100], [101]. This chapter examines the impact of memristor defects on neuromorphic radiation detection computing systems by using a winner-take-all spiking neuron readout. Classification accuracy is an important parameter for any NC platform, however, this study’s focus is on radionuclide identification applications. Figure 39 shows a design for the proposed crossbar array.
3.3.1. MODELING MEMRISTOR FAULTS

As it is the case with other nanoscale devices, there are high defects rates and manufacturing variability in memristors [102]. Moreover, as a result of excessive training cycles, memristors can deteriorate over time due to fatigue.

To ensure that memristor-based architectures are dependable, an analysis of the system’s fault tolerance is necessary to account for imperfect synaptic weights.

The spectrum of faults that could disturb a memristor includes single-bit open or stuck devices, and the variability of both the resistivity range and the threshold [21]. There are several defects, as illustrated in Figure 40. The resistivity of a memristor is considered defective when it remains at either the highest or lowest extreme as shown on Figure 40 (a) and (b). An effective open state (a) is achieved when the memristor is stuck in its highest resistivity state. In (b), the memristor is
stuck in the lowest resistivity state (effectively shorted). Resistivity swings between devices within the same array are impacted by the third fault category (c), variability of the resistivity range. In addition, threshold voltage (d) varies with time, resulting in a reduction in training accuracy. Defects resulting from effective open and effective short devices are examined as the primary focus of the chapter. However, other nonidealities such as memristor state discretization also affect computation accuracy in NC architectures and have been studied. Moreover signal-to-noise ratios have been studied to access accuracy and throughput of the proposed NC architectures.

![Figure 41. Memristor discretization levels. In the presence of small inference voltages, the resistance state remains constant. At high training voltages the memristor changes state.](image)

### 3.3.2. MODELLING THE COMPUTATION ALGORITHM

NNs are implemented in a wide range of applications [103]. With their linear weights and nonlinear activation functions, NNs are able to approximate arbitrary functions. The reservoir substrate in NC platforms exploits NN characteristics by using vector matrix multiplications (VMMs). NN inference is applied with a single-layer thresholding activation function for rectified linear units (ReLUs) which acts
Researchers have implemented multilayer neural networks with memristors in the past [104]. Additionally, Bala et al. demonstrate memristor approximation results for ReLU functions using memristors [81].

The local-competitive algorithm (LCA) [105] is the basis of the sparse coding system's NN. LCA implements a sparse coding computation principle that mimics the dense neuronal activity of the mammalian primary visual cortex (V1). Neurons are connected in a local inhibition manner through the LCA. When NC hardware implementations are implemented with inhibitory interneural connections, the learning algorithm is enhanced.

LCA dynamics maps dictionary class weights into m\textsuperscript{th} neurons' receptive fields, which are sent as columns of data. The dictionary classes in the system are assigned to specific radionuclide spectrums. Row elements in \( \Phi_m \) determine the sensitivity to each interacting presynaptic neuron. This vector contains sparse nonzero values and represents the input excitation signal generated by presynaptic neurons. The m\textsuperscript{th} postsynaptic neurons are excited by \( b_m(t) \), which is given by \( \langle \Phi_m, s(t) \rangle \) for each post-synaptic neuron. This means that the strength of \( b_m(t) \) is related to how similar \( s(t) \) is to the neuron's receptive field \( \Phi_m \). The artificial neurons charge up before firing, just as in biological neural networks. Throughout the neuron's life, the amount of charge accumulated is represented by an internal state variable \( u_m(t) \). In turn, a thresholding module monitors whether \( u_m(t) \) exceeds the threshold level \( T_\lambda \) to activate that neuron and produce an output signal \( a_m(t) \).
Competition arising from inhibition signals between active neurons is proportional to their activity levels \(a(t)\) and their similarity in receptive fields \(G_{m,n} = \langle \Phi_m, \Phi_n \rangle\).

An integrated system of nonlinear ordinary differential equations is utilized to calculate the oscillation dynamics of each neuron [105]:

\[
\dot{u}_m(t) = \frac{1}{r} \left[ b_m(t) - u_m(t) - \sum_{m \neq n} G_{m,n} a_n \right] \tag{6}
\]

Input signals will be represented by neurons that have fast-charging internal states \(u(t)\), which will activate sooner and inhibit slower-charging neurons. It is possible for all output neurons to be sparsely active owing to the inhibitory connections across them.

### 3.3.3. DEFECT INJECTION SCHEMES

When a memristor deviates from its desirable resistive range, a single-bit failure can occur. An extreme value may cause the device to become extremely inaccurate, particularly when it gets stuck. These two types of defects are therefore the focus of this section.

Radionuclides were used as ideal normalized weight patterns to analyze the neuromorphic system under memristor defects. The lowest weight in the system's dictionary effectively models memristors with the highest resistivity value in the simulation. As with resistance, the highest weight is associated with the lowest resistance state of a memristor. In this way, weighting can help determine how well an algorithm performs when dealing with open and short circuit defects. Thus, weights corresponding to the lowest and highest memristor states were substituted in the dictionary to randomly inject single-bit failures. Throughout the evaluation process, faulty devices were swept from 0% to 30%, with 0.25% steps between
each. This scheme produces 30%, single-bit defects in Figure 42(b), which has the same subsection as Figure 42(a). The randomly distributed defects in this dictionary subsection significantly blur the required weight pattern, as observed by comparing these two images.

![Figure 42. A 10×10 section of the required memristor array showing (left) the ideal normalized dictionary weights and (right) the same dictionary subsection with 30% randomly injected single-bit memristor defects. The open defects are shown in black and the short defects are shown in white](image)

### 3.3.4. COMPUTATIONAL DYNAMICS IMPLEMENTATION

The system accuracy was assessed using $s(t)$, an excitatory signal chosen from the spectrum dictionary after defect injection. Based on the partially defective dictionary previously generated, LCA dynamics were conducted for evaluation. In order to determine the accuracy of the output of the system, the sparse representations decoded for each classification task were recorded. Normalizing $s(t)$ results in normalizing $a_m(t)$.

The LCA dynamics are depicted in figure 4 as a simplified block diagram. Each memristor represented a synaptic connection to the excitatory input signal in the crossbar dictionary. In this main crossbar, the VMM was used to determine each neuron's initial excitation $b_m(t)$. A correlation matrix containing weights proportional to the similarity between each pair of receptive fields $G_{mn}$ was used to implement
inhibitory connections across output neurons. The correlation matrix memristors had the same resistivity range as the crossbar dictionary memristors after normalizing all dictionary neuron receptive fields. Simulated correlation matrix memristors were not subjected to faults for simplicity.

![Block diagram of LCA circuit implementation.](image)

By accumulating and scaling the incoming currents in each neuron, the LCA achieves its leaky-integrate-and-fire behavior. In each column header subcircuit, amplification stages summed and scaled the internal state $u_m(t)$ of the algorithm, and thresholding circuits determined the activation function $a_m(t)$. A second VMM was used to obtain the inhibition signal between the correlation matrix and the column headers.

### 3.4. FAULT-TOLERANCE TEST RESULTS

The overall computation accuracy of an array of neuromorphic devices can be severely degraded by a single-bit stochastic failure across a memristor. As a result, defect analysis is essential for the detection of resistive open and short defects. A fault tolerance analysis was also conducted to assess the computational
algorithm’s fault tolerance levels for different memristor nonidealities, such as noise, discretization and bit and line defects.

3.4.1. DISCRETIZATION EFFECTS

Memristor state discretization was studied, and results are shown for each weight density distribution assessed. Weight distributions with N=1 are equally spaced states. Distributions with N>1 have a higher density of states in the low conductivity range, and N<1 a higher density of high conductive states.

In general, the weights mapped by the memristor into are given by:

\[ R_n = \left[ \frac{x(R_{max} - R_{min})}{n-1} \right]^N + R_{min}, \text{ where } x \in [0, \ldots, n-1] \] (7)

Figure 44. Classification accuracy with a weight density distribution of N=1.
Figure 45. Classification accuracy with a weight density distribution of $N=2$.

Figure 46. Classification accuracy with a Weight Density Distribution of $N=0.5$. 
3.4.2. OPEN CIRCUIT DEFECTS

Using only memristors in the high resistivity state, an effective open circuit for the first analysis, crossbar dictionaries with single-bit faults were simulated. According to the test radionuclide signal used for each evaluation, this figure shows the normalized classification output for the neuron. A scatter plot shows the results of each test as dots.
3.4.3. SHORT CIRCUIT DEFECTS

The second analysis used only memristors stuck on the lowest resistivity (the short defect) to model partially faulty arrays. A normalized classification output is shown in Figure 50 for the neuron corresponding to the test radionuclide. There is a visible exponential decay in the classification performance in the output.
3.4.3. BIT FAILURE DEFECTS

Due to the significant impact that extreme resistance faults have on the system's performance, this work focuses on these cases. However, in the final hardware implementation, defective memristors could actually be stuck at any resistance value within their hysteresis swings. Consequently, combinations of both open and short defects were injected in different amounts in order to produce a more realistic model. Figure 51 shows the contour plot of the results.
3.5. DISCUSSION ON SYSTEM’S PERFORMANCE

With the focus on static faults, this work proposes and assesses a memristor-based radiation detection neuromorphic system for robustness. Open circuit defects, where the resistance is stuck at its maximum state, are more tolerable to the system than short defects, when the resistance is stuck at its minimum state. Up to 15% random open defects were found to maintain close to the system's maximum performance. Nevertheless, only 1% of short defects are capable of degrading the system accuracy, resulting in normalized classification output levels below 50%.

3.5.3. PERFORMANCE EFFECTS OF OPEN CIRCUIT DEFECTS

There is no sign of damage to the system when memristors are stuck in higher resistivity. Additionally, up to 15% random open defects are injected into the

Figure 51. True positive classification accuracy with both open and short defects.
correct neuron, and its output level is above 75%. Due to the high defect density, the system is still capable of successfully identifying radionuclide signals. In the case of incoming radionuclides with more than 15% defects, the system loses its ability to accurately classify them, as no neuron is active enough. As a result of the sparse nature of the implemented radionuclide spectrum dictionary, this behavior makes sense. Moreover, there is a significant gap between evaluations with 90% and 100% final output activity. There is a gap in the LCA dynamics because of the thresholding characteristics. Inhibition causes other neurons to become inactive as their activation decreases, allowing the neuron that corresponds to the test signal to fully charge. A 10% threshold level is implemented for a neuron to activate, and this gap accurately corresponds to that threshold level.

3.5.4. PERFORMANCE EFFECTS OF SHORT CIRCUIT DEFECTS

The system exhibits extremely high sensitivity to short defects in contrast to Figure 5. Due to receptive fields that diverge from their ideal normalized representation, this type of fault increases initial activation of neurons. In addition, premature neuron inactivation severely affects classification accuracy because all other neurons are highly inhibited initially. Moreover, notice that the output ranges from 10% to 0%. Again, this is a result of thresholding implemented in the LCA. By reducing the thresholding level, the circuit's performance could be improved. Nevertheless, the system's energy efficiency would decrease as a result of a longer time to converge to an answer.
3.5.5. OVERALL PERFORMANCE EFFECT OF BIT FAILURE

A low number of injected defects results in the best performance, shown in blue. It also indicates that effective short defects are more sensitive to the neuromorphic system than effective open defects.

3.5.6. SYSTEM SENSITIVITY TO MULTIPLE RADIONUCLIDES

The radionuclide classifying system’s sensitivity was tested for reliability in the presence of multiple radionuclides detected by the spectroscopic sensor. Figure 52 shows the true positive rate of executing the classification in the presence of multiple radionuclides combined in the sample. This test was performed assuming no noise incoming from the detector.

The test shows that for a system mapping the dictionary with only two memristor states the classification accuracy would be above 80% with up to 6 radionuclides in the test sample. Moreover, for a system mapping the radionuclide dictionary with 6 state memristors, the system’s sensitivity remains above 90% with up to 10 radionuclides present in the test samples. This shows the system’s reliability since it is possible to have more than one source detected but it is an unusual situation for all of them to be strong sources.
3.6. CHAPTER SUMMARY

Neuromorphic systems suffer from significant computation accuracy degradation when single-bit stochastic failures occur across a memristor array. It is therefore imperative to analyze defects resulting from resistive open and short circuits. Using defect-oriented analysis and modelling with a focus on static faults, a memristor-based radiation detection neuromorphic system was proposed and assessed for robustness. It was discovered that the system is more tolerant of open circuit defects than short defects, where the resistance is at its minimum value. In an array containing 15% random open defects, the system performs close to its full potential. In spite of this, only 1% of short defects can have a detrimental effect on the system's accuracy, resulting in less than 50% normalized classification output levels. The system's sensitivity was also shown to be reliable in the presence of multiple radionuclides present in the sample.

Figure 52. Sensitivity to Multiple Radionuclides
IV. NC SYSTEM ENERGY EFFICIENCY ANALYSIS

The in-memory processing capability of neuromorphic architectures provides energy efficiency benefits by performing the computations in the reservoir substrate. This allows a vastly versatile design process for NC architectures. Despite this extremely varied design process, there are common elements that are ubiquitous across NC platforms that implement CMOS neurons. These basic circuit element parameters of an NC design are analyzed in this chapter. Specifically, the elements that are present across the architecture implementing: the signal encoding scheme, the memory storage synaptic devices, the cell charge accumulators, the thresholding nonlinearity, and the firing subcircuits were studied.

4.1. MODELLING METHODS FOR ENERGY EFFICIENCY ANALYSIS

In CMOS-based circuits, such as the neurons in this work, the power dissipation occurs due to leakage and logic gates switching from one state to another by charging and discharging load and intrinsic capacitances in the design. Understanding the sources of power consumption in any NC circuit becomes crucial for energy-efficient circuit improvements.

4.1.1. STATIC POWER DISSIPATION IN CMOS NEURONS

CMOS static power dissipation occurs in the design’s neurons due to leakage when the logic gates are not switching state. In any CMOS-based design, the static dissipation is primarily technology node dependent, hence, beyond the designer’s control. However, static power dissipation is typically several orders of magnitude
lower than dynamic dissipation. Therefore, it will not be considered for this analysis.

4.1.2. DYNAMIC POWER DISSIPATION IN CMOS NEURONS

Dynamic power dissipation is the principal reason for power consumption in any VLSI circuit design. It is a function of the switching frequency $f$, voltage operation swing $V_{DD}$, and load capacitance $C_L$. For every logic gate in the design, the power dissipation is given by:

$$P_{dyn} = \alpha f C_L V_{DD}^2 \tag{8}$$

For this energy-efficiency analysis discussion load capacitances will be determined based on process parameters for the 0.5µm technology that was used for fabrication. All constants are taken from the NMOS & PMOS models used throughout this work for transistor SPICE simulations given in Appendix 2.

The intrinsic capacitances of the MOSFET devices to be considered in this estimation are of three types: (1) Overlap Capacitances due to the topology created by the lateral diffusion of the source and drain, (2) Channel Capacitance contributing to the channel charge and dependent on the operation region, and (3) Junction Capacitances, which are caused by reverse biased pn-junctions.

4.1.2.1. DYNAMIC POWER DISSIPATION IN OVERLAP CAPACITANCES

The overlap capacitances, also called structure capacitances, are dependent on how much the lateral diffusion causes an overlap region between the source and drain regions with the gate. In technology files, since the overlap distance is a
fabrication parameter, a value of per unit channel width capacitance is given to
determine this type of parasitic capacitance.

For the implemented transistors, the following per unit channel width
 capacitance is applied:

\[
\text{Gate-Drain Overlap: } C_{gdo_n} = 1.83 \times 10^{-10} \frac{F}{m} \quad \text{&} \quad C_{gdo_p} = 2.3 \times 10^{-10} \frac{F}{m} \quad (9)
\]
\[
\text{Gate-Source Overlap: } C_{gsn} = 1.83 \times 10^{-10} \frac{F}{m} \quad \text{&} \quad C_{gps} = 2.3 \times 10^{-10} \frac{F}{m} \quad (10)
\]

4.1.2.2. DYNAMIC POWER DISSIPATION IN JUNCTION CAPACITANCES

The Junction Capacitance, also sometimes called Diffusion Capacitance, is
more intricated to model. Junction Capacitances are dependent on their geometric
characteristics, on the region in which they operate, and the abruptness of the
junction. Hence, several more parameters are needed for estimation. Since the
reversed biased pn-junctions are formed all throughout the source and drain
contacts, junction capacitances occur both on the bottom plate and the sidewalls.
Further complicating the analysis of these capacitances, the level of abruptness in
doping level changes vary between the bottom plate and the sidewalls, hence
distinct grading coefficients must be analyzed. The doping level change in the
bottom plate is completely abrupt, hence the grading coefficient is closer to \( m_j = 0.5 \),
which is characteristic of an ideal abrupt change. On the contrary for the sidewalls,
the doping level changes more gradually and the grading coefficient is therefore
much smaller. The parameters for the bottom plate are, for simplicity of the
analysis, based on the source and drain area:

\[
\text{Junction Capacitance: } C_{in} = 4.147437 \times 10^{-4} \frac{F}{m^2} \quad \text{&} \quad C_{ip} = 7.209632 \times 10^{-4} \frac{F}{m^2} \quad (11)
\]
Built-in Junction Potential: $\phi_{bn} = 0.834387\, V \& \phi_{bp} = 0.870762\, V$  \hfill (12)

Grading Coefficient: $m_{jn} = 0.4259021 \& m_{jp} = 0.4909036$  \hfill (13)

The side walls junction capacitance parameters are perimeter based.

Junction Capacitance: \[ C_{jsw_n} = 3.339021 \times 10^{-10} \frac{F}{m} \] \hfill (14)
\[ C_{jsw_p} = 2.088154 \times 10^{-10} \frac{F}{m} \] \hfill (15)

Built-in Junction Potential: $\phi_{bwn} = 0.8055683\, V \& \phi_{bw} = 0.8\, V$  \hfill (16)

Grading Coefficient: $m_{jswn} = 0.1887251 \& m_{jswp} = 0.1735655$  \hfill (17)

Note that the model has different parameters for the channel side wall, but for effects of this estimation they have been ignored as their effects are considered within the analysis of the channel capacitance in the next section.

Since Junction Capacitances depend heavily on the applied voltage, they are nonlinear. Nonetheless, to simplify the analysis a multiplication factor $K_{eq}$ can be used to relate an equivalent linearized capacitor to the value of the junction capacitance under zero-bias conditions $C_{eq} = K_{eq}C_{j0}$

Where the multiplication factor $K_{eq}$ is a function of the junction capacitances, the build in potential and the grading coefficient.

\[ K_{eq} = \frac{-\phi_0^m}{(V_{high} - V_{low})(1-m)}[(\phi_0 - V_{high})^{1-m} - (\phi_0 - V_{low})^{1-m}] \] \hfill (18)

With $V_{high}=-5\, V$ and $V_{low}=-2.5\, V$ for the high-to-low transition (thl), and $V_{high}=-2.5\, V$ and $V_{low}=0\, V$ during the low-to-high transition (tlh). Note that this linearization applies for both the bottom plate and side wall junction capacitance using their respective built-in potentials and grading coefficients.

For the NMOS transistors during high-to-low transitions:
Bottom Plate: $K_{eq} (m=0.4259021, \phi_0=0.834387V) = 0.487786399$ (19)

Side Walls: $K_{eqsw} (m=0.1887251, \phi_0=0.8055683V) = 0.723185742$ (20)

For the NMOS transistors during low-to-high transitions:

Bottom Plate $K_{eq} (m=0.4259021, \phi_0=0.834387V) = 0.706439661$ (21)

Sidewalls: $K_{eqsw} (m=0.1887251, \phi_0=0.8055683V) = 0.851411166$ (22)

For the PMOS transistors during high-to-low transitions:

Bottom Plate: $K_{eq} (m=0.4909036, \phi_0=0.870762V) = 0.44808913$ (23)

Sidewalls: $K_{eqsw} (m=0.1735655, \phi_0=0.8V) = 0.741503304$ (24)

For the PMOS transistors during low-to-high transitions:

Bottom Plate: $K_{eq} (m=0.4909036, \phi_0=0.870762V) = 0.678601914$ (25)

Sidewalls: $K_{eqsw} (m=0.1735655, \phi_0=0.8V) = 0.861756197$ (26)

### 4.1.2.3. DYNAMIC POWER DISSIPATION IN THE CHANNEL CAPACITANCES

The channel capacitance for every transistor in the developed CMOS neurons is highly non-linear. This channel capacitance is dependent on the voltage of operation and on the channel length modulation of the transistors.

<table>
<thead>
<tr>
<th>Operation Region</th>
<th>$C_{GBCCH}$</th>
<th>$C_{GSCCH}$</th>
<th>$C_{GDCCH}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cutoff</td>
<td>$C_{ox}WL_{eff}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Linear</td>
<td>0</td>
<td>$\frac{1}{2} C_{ox}WL_{eff}$</td>
<td>$\frac{1}{2} C_{ox}WL_{eff}$</td>
</tr>
<tr>
<td>Saturation</td>
<td>0</td>
<td>$\frac{2}{3} C_{ox}WL_{eff}$</td>
<td>0</td>
</tr>
</tbody>
</table>

For simplification purposes, the channel capacitance can be approximated according to Table 7. This channel capacitance modelling approach is also dependent on the operation region of the transistor. In cutoff mode the channel capacitance has only a gate-bulk component. In linear mode there is both a gate-
source and gate-drain channel capacitance. In saturation mode only a gate-source capacitance term affects the dynamic power consumption.

Before proceeding with the calculation of the channel capacitance other assumptions regarding technology parameters must be defined in this methods section. For example, the gate oxide capacitance per unit area $C_{ox}$, is not explicitly given by the transistor model use. However, it can be inferred from the oxide thickness as follows:

The permittivity of silicon dioxide is given by:

$$\varepsilon_{ox} = 3.9\varepsilon_0 = 3.9 \times 8.854 \times 10^{-12} \frac{F}{m} = 3.45 \times 10^{-11} \frac{F}{m} \quad (27)$$

From the SPICE device model, the Gate Oxide Thickness ($t_{ox}$) is $1.39 \times 10^{-8} m$ for both NMOS & PMOS Therefore, the Oxide Capacitance:

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} = 2.48 \times 10^{-3} \frac{F}{m^2} = 2.48 \frac{F}{\mu m^2} \quad (28)$$

Likewise, other technological parameters like the carrier mobility can be approximated from the technology model files. For this analysis purposes the carrier mobility for both electrons and holes can be approximated to the low field mobility ($U_0$), by ignoring mobility degradation, hence:

$$\mu_n = 449.4127448 \frac{cm^2}{Vs} \quad \& \quad \mu_p = 201.3603195 \frac{cm^2}{Vs} \quad (29)$$

With this assumption, the process transconductance parameter can be obtained:

$$k'_n = \mu_n C_{ox} = 1.116 \times 10^{-4} \frac{A}{V^2} \quad \& \quad k'_p = \mu_p C_{ox} = 5.002225 \times 10^{-5} \frac{A}{V^2} \quad (30)$$

Then the threshold voltage at zero bias can also be obtained from the SPICE device model: For NMOS, this value is 0.628 V and for PMOS it is −0.915 V. Note
that these threshold voltages are for large L. Short channel effects are neglected for this power consumption analysis discussion.

Finally, note that to account for all the loads in the output node of each stage, it is assumed that all capacitances are lumped together into a single $C_L$, that can be broken down into 1) the output capacitance of that stage, 2) the input capacitance of the next stages, and 3) the capacitance from the interconnect wiring. However, also note that due to the short lengths of the wires, the power dissipation effects of the interconnect capacitances are neglected.

### 4.2. Dynamic Power Dissipation in the Presynaptic Cells

This analysis of the NC system was possible by implementing several assumptions and simplifications to address the many nonlinear capacitances in the MOS transistor model. Hence, this power estimation modelling is performed assuming that all capacitances are lumped together in a single capacitor $C_L$. For all cells, power consumption is calculated individually per gate. An assumption is made that the input voltages of each individual gate are driven by an ideal voltage source, i.e., there are no rise or fall times.

#### 4.2.1. Inhibition Capacitor Charging PMOS M0 Transistor

This p-type enhancement transistor is part of the transmission logic in each of the row circuit cells. This transistor controls the connection from the memristor array receptive field to the inhibition capacitor in the presynaptic cell. Its gate is connected to the $\text{notFiringAny}$ bus line, such that the transistor only closes/conducts when one of the output neurons is firing. The source of this
transistor is connected to one of the crossbar’s row nanowires, and the drain is connected to the inhibition capacitor. Conduction always occurs from the crossbar to the capacitor, charging the latter with the inhibition current signals coming from the firing neurons. Since this transistor only conducts when a neuron is firing, then the dynamic power consumption is only dependent on the firing frequency \( f_{fire} \).

The load capacitances seen by this PMOS are: its own Gate-to-Drain Overlap Output Capacitance \( (C_{gd,M_0}) \), its own Drain-to-Bulk Junction Output Capacitances \( (C_{db,M_0}) \) the Gate capacitances of the transistors in the driven not gate \( (C_{gM_3} \& C_{gM_4}) \), the lumped capacitances connected across the inhibition resistor \( (C_{gd,M_1} \& C_{db,M_1}) \) and the dominant capacitor from the inhibition logic module \( (C_{inh}) \).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (23), as well as the more detailed expressions for each load in Table 8. Therefore, 

\[
P_{M_0} = \alpha f_{fire} V_{DD}^2 \left( C_{gd,M_0} + C_{db,M_0} + C_{gM_3} + C_{gM_4} + C_{gd,M_1} + C_{db,M_1} + C_{inh} \right)
\]

**Table 8.** Capacitances contributing to dynamic power dissipation in the inhibition modules’ M_0 transistor

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{gd,M_0} )</td>
<td>( 2 * C_{gd0P} * W_P )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{db,M_0} )</td>
<td>( K_{eqp} * AD_P * C_j + K_{eqswp} * PD_P * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gM_3} )</td>
<td>( W_P * \left( C_{gd0P} + C_{gsoP} \right) + C_{ox} * W_P * L_P )</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>( C_{gM_4} )</td>
<td>( W_N * \left( C_{gd0N} + C_{gsoN} \right) + C_{ox} * W_N * L_N )</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>( C_{gd,M_1} )</td>
<td>( 2 * C_{gd0N} * W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{db,M_1} )</td>
<td>( K_{eqN} * AD_N * C_j + K_{eqswN} * PD_N * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{inh} )</td>
<td>( C_{inh} )</td>
<td>Dominant Capacitance</td>
</tr>
</tbody>
</table>
4.2.2. INHIBITION CAPACITOR DISCHARGING NMOS M₁ TRANSISTOR

When a row input circuit is sufficiently charged by inhibition currents (proceeding from firing output neurons), it becomes dormant and stops transmitting the signal. This occurs to mimic a biological neuron’s refractory period. Like in the biological counterpart, the artificial neuron stops sending a signal but only for a period of recovery time. The duration of this period is controlled by the inhibition module’s capacitor and resistor. Transistor M₁ is used to mimic how the cell membrane brings back the neuron to its resting state where it can start transmitting signals again. M₁ discharges the inhibition capacitor whenever the row cell is receiving a signal. This transistor’s gate is connected to the incoming excitatory input spike train signal, the drain to the inhibition capacitor, and the source to the second discharging capacitor M₂. Since this transistor switches with the same frequency as the incoming signal, then the dynamic power consumption is dependent on the signal frequency ($f_{sig}$).

The load capacitances seen by this NMOS are its own Gate-to-Drain Overlap Output Capacitance ($C_{gdM₁}$), its own Drain-to-Bulk Junction Output Capacitance ($C_{dbM₁}$), the lumped gate capacitances of the transistors in the driven inverter gate ($C_{gM₃}$ & $C_{gM₄}$), the lumped capacitances connected across the inhibition resistor ($C_{gdM₀}$ & $C_{dbM₀}$), and the inhibition logic module dominant capacitor ($C_{inh}$).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (24), as well as the detailed expressions for each load capacitance in Table 9. Therefore,
\[ P_{M_1} = \alpha f_{sl} V_{DD}^2 (C_{gdM1} + C_{dbM1} + C_{gM3} + C_{gM4} + C_{gdM0} + C_{dbM0} + C_{inh}) \] (32)

Table 9. Capacitances contributing to dynamic power dissipation in the inhibition modules' M1 transistor

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{gdM1} )</td>
<td>( 2 * C_{gd0N} * W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbM1} )</td>
<td>( K_{eqN} * A_DN * C_j + K_{eqSW}N * P_{DN} * C_{jSW} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gM3} )</td>
<td>( W_P \ast \left( C_{gd0_P} + C_{gs0_P} \right) + C_{ox} \ast W_P \ast L_P )</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>( C_{gM4} )</td>
<td>( W_N \ast \left( C_{gd0_N} + C_{gs0_N} \right) + C_{ox} \ast W_N \ast L_N )</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>( C_{gdM0} )</td>
<td>( 2 \ast C_{gd0P} \ast W_P )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbM0} )</td>
<td>( K_{eqP} \ast A_DP \ast C_j + K_{eqSW}P \ast P_{DP} \ast C_{jSW} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{inh} )</td>
<td>( C_{inh} )</td>
<td>Dominant Capacitance</td>
</tr>
</tbody>
</table>

4.2.3. INHIBITION CAPACITOR DISCHARGING NMOS M2 TRANSISTOR

Transistor M2 is also used to discharge the inhibition capacitor. This transistor's gate is connected to the notFiringAny bus line, the drain to transistor M1, and the source to ground. Its function is to stop the discharge path when an output neuron is firing, such that the inhibition capacitor can charge completely with the inhibition current being fed backwards. Since this transistor switches only when a neuron is firing, then the dynamic power consumption is dependent on the firing frequency.

The load capacitances seen by this NMOS are its own Gate-to-Drain Overlap Output Capacitance \( (C_{gdM2}) \), its own Drain-to-Bulk Junction Output Capacitances \( (C_{gdM2}) \), and the Gate-to-Source Overlap Capacitance of Transistor M1 \( (C_{gSM1}) \).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (25), as well as detailed specific expressions for each load in Table 10. Therefore,
\[ P_{M_2} = \alpha f_{\text{fire}} V_{DD}^2 (C_{gdM_2} + C_{dbM_2} + C_{gsM_1}) \]  

(33)

**Table 10.** Capacitances contributing to dynamic power dissipation in the inhibition modules’ M2 transistor.

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{gdM_2} )</td>
<td>( 2 * C_{gd0N} * W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbM_2} )</td>
<td>( K_{eqN} * AD_N * C_j + K_{eqswN} * PD_N * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gsM_1} )</td>
<td>( C_{gd0P} * W_P )</td>
<td>Overlap Capacitance</td>
</tr>
</tbody>
</table>

4.2.4. INHIBITION LOGIC CIRCUIT 1X INVERTER LOGIC GATE

This inverter in the inhibition logic circuit is responsible for letting the row circuit cell transmit the input signal only when the inhibition capacitor is not charged, i.e., when the signal coming from that neuron is underrepresented in the system. This gate switches only when a firing neuron has inhibited the cell, hence the dynamic power consumption is dependent on \( f_{\text{fire}} \).

The load capacitances seen by this Inverter are its own Gate-to-Drain Overlap Output Capacitances \( (C_{gdM_3} \& C_{gdM_4}) \), its own Drain-to-Bulk Junction Output Capacitances \( (C_{dbM_3} \& C_{dbM_4}) \), and the Gate Capacitances of the driven transistors in the NAND Gate \( (C_{gM_6} \& C_{gM_8}) \).

With all these capacitances already identified, a general power consumption expression for this component can be now defined in equation (26), as well as detailed expressions for each load capacitor in Table 11. Therefore,

\[ P_{-1} = \alpha f_{\text{fire}} V_{DD}^2 (C_{out_{-1}} + C_{inA_{-1}}) = f_{\text{fire}} V_{DD}^2 \left( C_{gdM_3} + C_{dbM_3} + C_{gdM_4} + \cdots \right) \]

(34)
Table 11. Capacitances contributing to dynamic power dissipation in the inhibition modules’ NAND logic gate.

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{gdM3}$</td>
<td>$2 \times C_{gd0p} \times W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM3}$</td>
<td>$K_{eqp} \times AD_p \times C_J + K_{eqswp} \times PD_p \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM4}$</td>
<td>$2 \times C_{gd0N} \times W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM4}$</td>
<td>$K_{eqN} \times AD_N \times C_J + K_{eqswN} \times PD_N \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gM6}$</td>
<td>$W_P \times (C_{gd0p} + C_{gs0p}) + C_{oxw} \times W_P \times L_P$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{gM8}$</td>
<td>$W_N \times (C_{gd0N} + C_{gs0N}) + C_{oxw} \times W_N \times L_N$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
</tbody>
</table>

4.2.5. INHIBITION LOGIC CIRCUIT 1X NAND LOGIC GATE

This NAND is part of the inhibition logic circuit. Its function is to control the transmission of the input signal into the receptive field. It switches with about the same frequency as the input signal. However, it does pause its switching when the cell is inhibited. Since the actual switching activity is hard to estimate here, the frequency for the dynamic power consumption estimation has been approximated to $f_{sig}$. The load capacitances seen by this NAND are its own Gate-to-Drain Overlap Output Capacitances ($C_{gdM5}, C_{gdM6}, C_{gdM7} & C_{gdM8}$), its own Drain-to-Bulk Junction Output Capacitances ($C_{dbM5}, C_{dbM6}, C_{dbM7} & C_{dbM8}$), and the Gate Capacitances of the transistors in the driven inverter gate ($C_{gM9} & C_{gM10}$).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (27), as well as detailed expressions for each load capacitance in Table 12. Therefore:

$$P_x = \alpha f_{sig} V_{DD}^2 \left( C_{gdM5} + C_{dbM5} + C_{gdM6} + C_{dbM6} + C_{gdM7} + C_{dbM7} + \cdots \right)$$

(35)
<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{gdM5}$</td>
<td>$2 \cdot C_{gd0p} \cdot W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM5}$</td>
<td>$K_{eqp} \cdot AD_p \cdot C_j + K_{eqsw_p} \cdot PD_p \cdot C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM6}$</td>
<td>$2 \cdot C_{gd0p} \cdot W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM6}$</td>
<td>$K_{eqp} \cdot AD_p \cdot C_j + K_{eqsw_p} \cdot PD_p \cdot C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM7}$</td>
<td>$2 \cdot C_{gd0N} \cdot W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM7}$</td>
<td>$K_{eqN} \cdot AD_N \cdot C_j + K_{eqsw_N} \cdot PD_N \cdot C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM8}$</td>
<td>$2 \cdot C_{gd0N} \cdot W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM8}$</td>
<td>$K_{eqN} \cdot AD_N \cdot C_j + K_{eqsw_N} \cdot PD_N \cdot C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gm9}$</td>
<td>$W_p \cdot (C_{gd0p} + C_{gs0p}) + C_{ox} \cdot W_P \cdot L_P$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{gm10}$</td>
<td>$W_N \cdot (C_{gd0N} + C_{gs0N}) + C_{ox} \cdot W_N \cdot L_N$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
</tbody>
</table>

4.2.6. TRANSMISSION 1X LOGIC GATE

This bilateral analog switch is part of the transmission logic in each of the row circuit cells. This transmission gate controls the connection in the row cell allowing signal to pass to the memristor array receptive field. The control terminal is connected to the notFiringAny bus line, such that the row neuron only fires and sends a signal from the neuron to the cell when none of the output cells are firing. Since the signal that passes through this gate is a spike train, the switching happening on the transmitting NMOS and PMOS in the gate has the same frequency as the signal. However, the control logic in the transmission gate circuit switches only when one of the neurons fires, so the power consumption has two different frequency dependencies. The transmitting part of the gate switches with
the same activity as the signal \( f_{\text{sig}} \), and the controlling transistors fire with the same frequency that the neurons fire \( f_{\text{fire}} \).

The load capacitances seen by this transmission gate are: (a) The transmitting transistors’ Gate-to-Drain Overlap Capacitances \( C_{gd_{TG,M1}} \) & \( C_{gd_{TG,M2}} \), (b) the transmitting transistors’ Drain-to-Bulk Junction Capacitances \( C_{db_{TG,M1}} \) & \( C_{db_{TG,M2}} \), (c) the controlling transistors’ Gate-to-Drain Overlap Capacitances \( C_{gd_{TG,M3}} \) & \( C_{gd_{TG,M4}} \), (d) the controlling transistors’ Drain-to-Bulk Junction Capacitances \( C_{db_{TG,M3}} \) & \( C_{db_{TG,M4}} \), (e) the Gate-to-Drain Capacitance of the Charging \( M_0 \) Transistor \( C_{gd_{M0}} \), (f) the Drain-to-Bulk Capacitance of the Charging \( M_0 \) Transistor \( C_{db_{M0}} \), (g) the transmitting PMOS transistor’s Gate Capacitances \( C_{g_{TG,M1}} \), and (h) the memristor crossbar’s lumped capacitances \( C_{cb} \).

Note that the capacitance connected through the crossbar \( C_{cb} \), is hard to estimate because it is dependent on the number of rows \( M \), and columns \( N \), as well as the row cell’s output capacitance \( C_{\text{rowOut}} \) and the column cell’s input capacitance \( C_{\text{columnIn}} \).

With these approximations, a general power consumption expression for this component can be defined by equation (29) and the detailed expressions for each capacitance Table 13. Therefore,
\[ P_{TG} = \alpha f_{\text{sig}} V_{DD}^2 (C_{gd_{TGM1}} + C_{db_{TGM1}} + C_{gd_{TGM2}} + C_{db_{TGM2}} + C_{gd_{M0}} + C_{db_{M0}} + C_{cb}) \]

\[ \ldots f_{\text{ freq}} V_{DD}^2 (C_{gd_{TGM3}} + C_{db_{TGM3}} + C_{gd_{TGM4}} + C_{db_{TGM4}} + C_{gd_{M1}}) \]

(36)

Table 13. Capacitances contributing to dynamic power dissipation in the crossbar bilateral analog switch

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{gd_{TGM1}} )</td>
<td>( 2 * C_{gd_{0P}} * W_P )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{db_{TGM1}} )</td>
<td>( K_{eqP} * AD_P * C_j + K_{eqswP} * PD_P * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gd_{TGM2}} )</td>
<td>( 2 * C_{gd_{0N}} * W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{db_{TGM2}} )</td>
<td>( K_{eqN} * AD_N * C_j + K_{eqswN} * PD_N * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gd_{M0}} )</td>
<td>( 2 * C_{gd_{0P}} * W_P )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{db_{M0}} )</td>
<td>( K_{eqP} * AD_P * C_j + K_{eqswP} * PD_P * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gd_{TGM3}} )</td>
<td>( 2 * C_{gd_{0P}} * W_P )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{db_{TGM3}} )</td>
<td>( K_{eqN} * AD_N * C_j + K_{eqswN} * PD_N * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gd_{TGM4}} )</td>
<td>( 2 * C_{gd_{0N}} * W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{db_{TGM4}} )</td>
<td>( K_{eqN} * AD_N * C_j + K_{eqswN} * PD_N * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gd_{M1}} )</td>
<td>( W_P * \left( C_{gd_{0P}} + C_{gs_{0P}} \right) + C_{ox} * W_P * L_P )</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>( C_{cb} )</td>
<td>( (M - 1) * C_{row_out} + N * C_{column_n} )</td>
<td>Lumped Capacitors through Crossbar</td>
</tr>
</tbody>
</table>

4.2.7. INHIBITION LOGIC CIRCUIT 1X INVERTER LOGIC GATE

This other inverter gate also controls the transmission of the input signal into the receptive field. Like the NAND, it switches with the same frequency of the input signal, pausing only when inhibited. Since the actual switching activity is hard to estimate here, the frequency for the dynamic power consumption estimation can also be approximated to \( f_{\text{sig}} \).
The load capacitances seen by this Inverter are its own gate-to-drain overlap capacitances \( (C_{gdM9} \& C_{gdM10}) \), the drain-to-bulk junction capacitances \( (C_{dbM9} \& C_{dbM10}) \), the transmission gate’s gate-to-drain overlap capacitances \( (C_{gdTG,M1} \& C_{gdTG,M2}) \), and the transmission gate’s drain-to-bulk junction capacitances \( (C_{dbTG,M1} \& C_{dbTG,M2}) \).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (30), as well as detailed expressions for each load capacitor in. Therefore,

\[
P_{in} = \alpha f_{sig} V_{DD}^2 \left( C_{gdM9} + C_{dbM9} + C_{gdM10} + C_{dbM10} + \cdots \right)
\]

\[
\text{Table 14. Capacitances contributing to dynamic power dissipation in the 1X inverter.}
\]

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{gdM9} )</td>
<td>( 2 \times C_{gd0p} \times W_p )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbM9} )</td>
<td>( K_{eq} \times AD_p \times C_j + K_{eqsw} \times PD_p \times C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gdM10} )</td>
<td>( 2 \times C_{gd0N} \times W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbM10} )</td>
<td>( K_{eq} \times AD_N \times C_j + K_{eqsw} \times PD_N \times C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gdTG,M1} )</td>
<td>( 2 \times C_{gd0p} \times W_p )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbTG,M1} )</td>
<td>( K_{eq} \times AD_p \times C_j + K_{eqsw} \times PD_p \times C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gdTG,M2} )</td>
<td>( 2 \times C_{gd0N} \times W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbTG,M2} )</td>
<td>( K_{eq} \times AD_N \times C_j + K_{eqsw} \times PD_N \times C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
</tbody>
</table>

4.2.8. INHIBITION LOGIC OHMIC POWER CONSUMPTION

The inhibition logic subcircuit in the row cell has a resistor which controls the refractory period \( \tau \), during which the row cell stops transmitting the signal towards the receptive field. Like any ohmic component, some power dissipation will occur due to the Joule effect. This is given by:

\[
P_R = V_{DD}^2 / R_{inh}
\]
4.3. DYNAMIC POWER DISSIPATION IN THE POSTSYNAPTIC CELLS

Like the calculations in the previous section, for the postsynaptic circuit cell, the power consumption can also be broken up by dissipation in each gate. All input voltages in each gate are still assumed ideal.

4.3.1. FIRING ACTIVATION SIGNAL PMOS

This p-type enhancement transistor is part of the firing subcircuit in each of the column circuits cells. This transistor acts as the firing axon in its biological counterpart; it sends back the inhibition signals to the row cells through the memristor synapses. Its gate is connected to the SelfFiring subcircuit, such that the transistor only closes/conducts when the neuron is firing. The source of this transistor is connected to $V_{DD}$, and the drain is connected to the crossbar column. Since this transistor only conducts when the neuron is firing, then the dynamic power consumption is only dependent on the firing frequency $f_{fire}$.

The load capacitances seen by this PMOS are its own gate-to-drain overlap output capacitance ($C_{gdM3}$). Its own Drain-to-Bulk Junction Output Capacitances ($C_{dbM3}$), the transmission gate’s gate-to-Drain overlap capacitances ($C_{gdTG, M1}$ & $C_{gdTG, M2}$), the transmission gate’s drain-to-bulk junction capacitances ($C_{dbTG, M1}$ & $C_{dbTG, M2}$), and the crossbar’s lumped capacitances ($C_{cb}$).

Note that the capacitance connected through the crossbar ($C_{cb}$), is dependent on the number of rows M, and columns N, as well as the row cell’s output capacitance ($C_{rowOut}$) and the column cell’s input capacitance ($C_{columnIn}$).
With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (32), as well as detailed expressions for each load capacitor in Table 15. Therefore, 

\[
P_{M3} = \alpha f_{\text{fire}} V_D^2 (C_{gdM3} + C_{dbM3} + C_{cb}) \tag{39}
\]

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_{gdM3})</td>
<td>(2 \times C_{gd0P} \times W_P)</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>(C_{dbM3})</td>
<td>(K_{eqP} \times AD_P \times C_j + K_{eqswP} \times PD_P \times C_{jsw})</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>(C_{gdTG,M1})</td>
<td>(2 \times C_{gd0P} \times W_P)</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>(C_{dbTG,M1})</td>
<td>(K_{eqP} \times AD_P \times C_j + K_{eqswP} \times PD_P \times C_{jsw})</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>(C_{gdTG,M2})</td>
<td>(2 \times C_{gd0N} \times W_N)</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>(C_{dbTG,M2})</td>
<td>(K_{eqN} \times AD_N \times C_j + K_{eqswN} \times PD_N \times C_{jsw})</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>(C_{cb})</td>
<td>(M \times C_{rowOut} + (N - 1) \times C_{columnIn})</td>
<td>Lumped Capacitors through Crossbar</td>
</tr>
</tbody>
</table>

### 4.3.2. TRANSMISSION 1X GATE LOGIC

This bilateral analog switch is part of the transmission logic in each of the column output circuit cells. This transmission gate controls the influx of excitatory signals coming from the receptive field into the column cell. The control terminal is connected to the `notFiringAny` bus line, such that the neuron only accepts an input signal when none of the output cells are firing. Since the signal that passes through this gate is a spike train, the switching happening on the transmitting NMOS and PMOS in the gate has the same frequency as the signal. However, the control logic in the transmission gate circuit switches only when one of the output neurons fires, so the power consumption has two different frequency dependencies. The transmitting part of the gate switches with the same activity as the signal, \(f_{sig}\), and
the controlling transistors fire with the same frequency that the neurons fire, i.e., $f_{fire}$. The load capacitances seen by this gate are its own gate-to-drain overlap capacitances ($C_{gd,M1} \cdot C_{gd,M2} \cdot C_{gd,M3} \cdot C_{gd,M4}$), its own drain-to-bulk junction capacitances ($C_{db,M1} \cdot C_{db,M2} \cdot C_{db,M3} \cdot C_{db,M4}$), the load capacitances from the resetting M4 transistor ($C_{gd,M4} \cdot C_{db,M4}$), the driven unbalanced inverter’s gate capacitances ($C_{gm7} \cdot C_{gm8}$), the transmitting PMOS transistor’s gate capacitances ($C_{g}, C_{gm}$), and the dominant capacitor from the internal state capacitor ($C_{st}$).

With these capacitances now identified, a general power consumption expression for this component can be defined in equation (33), as well as detailed expressions for each load capacitance in Table 16. Therefore,

$$P_{MTG} = \alpha_f s_{Ig} V_{DD}^2 (C_{gd,M1} + C_{db,M1} + C_{gd,M2} + C_{db,M2} + C_{gd,M3} + C_{db,M3} + C_{gd,M4} + C_{db,M4} + C_{gm7} + C_{gm8} +$$
$$C_{st}) + f_{fire} V_{DD}^2 (C_{gd,M3} + C_{db,M3} + C_{gd,M4} + C_{db,M4} + C_{g,M1}) \quad (40)$$
Table 16. Capacitances contributing to dynamic power dissipation in the postsynaptic cell’s transmission gate

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{gd_{TM1}}$</td>
<td>$2 \times C_{gd0} \times W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{db_{TM1}}$</td>
<td>$K_{eq} \times AD_P \times C_j + K_{eqsw_p} \times PD_P \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gd_{TM2}}$</td>
<td>$2 \times C_{gd0_N} \times W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{db_{TM2}}$</td>
<td>$K_{eq} \times AD_N \times C_j + K_{eqsw_N} \times PD_N \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gd_{M4}}$</td>
<td>$2 \times C_{gd0} \times W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{db_{M4}}$</td>
<td>$K_{eq} \times AD_N \times C_j + K_{eqsw_N} \times PD_N \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gm7}$</td>
<td>$W_P \times (C_{gd0_p} + C_{gs0_p}) + C_{ox} \times W_P \times L_P$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{gm8}$</td>
<td>$W_N \times (C_{gd0_N} + C_{gs0_N}) + C_{ox} \times W_N \times L_N$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{gd_{TM3}}$</td>
<td>$2 \times C_{gd0} \times W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{db_{TM3}}$</td>
<td>$K_{eq} \times AD_P \times C_j + K_{eqsw_p} \times PD_P \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gd_{TM4}}$</td>
<td>$2 \times C_{gd0_N} \times W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{db_{TM4}}$</td>
<td>$K_{eq} \times AD_N \times C_j + K_{eqsw_N} \times PD_N \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gt_{M1}}$</td>
<td>$W_P \times (C_{gd0_p} + C_{gs0_p}) + C_{ox} \times W_P \times L_P$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{st}$</td>
<td>$C_{st}$</td>
<td>Dominant Capacitance</td>
</tr>
</tbody>
</table>
4.3.3. INTERNAL STATE CIRCUIT DISCHARGING NMOS M₄

When anyone of the output column cells is firing, a Schmitt trigger controls the resetting of the neurons internal state capacitor \((C_{st})\). This is done through the discharging NMOS connected in parallel to \((C_{st})\). The discharge mimics the resetting done by a biological neuron’s cell membrane. Like in the biological counterpart, the artificial neuron must also go back to its resting state before becoming available. The duration of the discharge period is controlled by the firing circuit’s capacitor and resistor. The gate is pulled-up when any of the output neurons are firing. The drain is connected to \((C_{st})\), and the source to ground. Since this transistor switches only when a neuron fires, then the dynamic power consumption is dependent on the firing frequency \((f_{fire})\).

The load capacitances seen by this NMOS are its own Gate-to-Drain Overlap Output Capacitance \((C_{gdM₄})\), its own drain-to-bulk junction output capacitances \((C_{dbM₄})\), the overlap capacitances from the transmission gate \((C_{gdTG,M₁} \& C_{gdTG,M₂})\), the junction capacitances from the transmission gate \((C_{dbTG,M₁} \& C_{dbTG,M₂})\), the driven unbalanced inverter’s gate capacitances \((C_{gM₇} \& C_{gM₈})\), the dominant capacitor from the internal state capacitor \((C_{st})\).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (34), and detailed expressions for each load capacitance in Table 17. Therefore,

\[
P_{M₄} = \alpha f_{fire} V_{DD}^2 \left( C_{gdM₄} + C_{dbM₄} + C_{gdTG,M₁} + C_{dbTG,M₁} + C_{gdTG,M₂} + C_{dbTG,M₂} + C_{gM₇} + C_{gM₈} + C_{st} \right)
\]

(41)
Table 17. Capacitances contributing to dynamic power dissipation in the M4 NMOS

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{gdM4}$</td>
<td>$2 \times C_{gd0N} \times W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM4}$</td>
<td>$K_{eq} \times AD_N \times C_j + K_{eqsw} \times P_DN \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM1}$</td>
<td>$2 \times C_{gd0P} \times W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM1}$</td>
<td>$K_{eq} \times AD_P \times C_j + K_{eqsw} \times P_DP \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM2}$</td>
<td>$2 \times C_{gd0N} \times W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM2}$</td>
<td>$K_{eq} \times AD_N \times C_j + K_{eqsw} \times P_DN \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gm7}$</td>
<td>$W_P \times \left( C_{gd0P} + C_{gs0P} \right) + C_{ox} \times W_P \times L_P$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{gm8}$</td>
<td>$W_N \times \left( C_{gd0P} + C_{gs0N} \right) + C_{ox} \times W_N \times L_N$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{st}$</td>
<td>$C_{st}$</td>
<td>Dominant Capacitance</td>
</tr>
</tbody>
</table>

4.3.4. THRESHOLDING CIRCUIT’S UNBALANCED INVERTER

This unbalanced inverter controls the thresholding level $V_M$ at which the neuron fires. This gate switches only when the neuron’s $C_{st}$ internal charge has surpassed the threshold, hence the dynamic power consumption is dependent on $f_{fire}$.

The load capacitances seen by this Inverter are its own gate-to-drain overlap output capacitances ($C_{gdM7} \& C_{gdM8}$), its own Drain-to-Bulk Junction Output Capacitances ($C_{dbM7} \& C_{dbM8}$), and the gate capacitances of the driven transistors in the next inverter ($C_{gm9} \& C_{gm10}$).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (42), and the detailed expressions for each load are given in Table 18. Therefore,

$$P_{\text{fire}} = \alpha f_{fire} V_{DD}^2 \left( C_{gdM7} + C_{dbM7} + C_{gdM8} + C_{dbM8} + C_{gm9} + C_{gm10} \right) \quad (42)$$
Table 18. Capacitances contributing to dynamic power dissipation in the thresholding unbalanced inverter

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{gdM7}$</td>
<td>$2 * C_{gd0p} * W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM7}$</td>
<td>$K_{eq} * ADp * C_j + K_{eqsw} * PDp * C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM8}$</td>
<td>$2 * C_{gd0N} * W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM8}$</td>
<td>$K_{eq} * ADN * C_j + K_{eqsw} * PDN * C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gM9}$</td>
<td>$W_P * (C_{gd0p} + C_{gs0p}) + C_{ox} * W_P * L_P$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>$C_{gM10}$</td>
<td>$W_N * (C_{gd0N} + C_{gs0N}) + C_{ox} * W_N * L_N$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
</tbody>
</table>

4.3.5. THRESHOLDING CIRCUIT’S BALANCED INVERTER

This inverter is also part of the thresholding sub circuit. It drives the firing circuit’s M11 transistor. It switches with the same frequency as the previous inverter. Hence its power consumption is also dependent on $f_{fire}$. The load capacitances seen by this inverter are its own gate-to-drain overlap output capacitances ($C_{gdM9}$ & $C_{gdM10}$), its own drain-to-bulk junction output capacitances ($C_{dbM9}$ & $C_{dbM10}$), the gate capacitance of the driven M11 transistors ($C_{gM11}$).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (36), with the detailed expressions for each load capacitance in Table 19. Therefore,

$$P_{\text{L2}} = \alpha f_{fire} V_{DD}^2 \left( C_{gdM9} + C_{dbM9} + C_{gdM10} + C_{dbM10} + C_{gM11} \right)$$

(43)
Table 19. Capacitances contributing to dynamic power dissipation in the thresholding balanced inverter.

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{gdM9}$</td>
<td>$2 \times C_{gd0p} \times W_P$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM9}$</td>
<td>$K_{eq} \times AD_P \times C_J + K_{eqswp} \times PD_P \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gdM10}$</td>
<td>$2 \times C_{gd0N} \times W_N$</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>$C_{dbM10}$</td>
<td>$K_{eq} \times AD_N \times C_J + K_{eqswN} \times PD_N \times C_{jsw}$</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>$C_{gM11}$</td>
<td>$W_P \times (C_{gd0p} + C_{gs0p}) + C_{ox} \times W_P \times L_P$</td>
<td>Channel and Overlap Capacitances</td>
</tr>
</tbody>
</table>

4.3.6. NMOS M11 TRANSISTOR FOR FIRING CIRCUIT

When the charge accumulated in the internal state capacitor $C_{st}$ has surpassed the threshold level, transistor M11 starts conducting, activating the firing circuit. This is analogous to the action potential transmission in the biological neuron’s axon. The transistors’ function is to discharge the capacitor connected in parallel. This causes the neuron to fire sending inhibitory signals back to the memristor array and resetting the state capacitors $C_{st}$. Once this transistor stops conducting, the neuron continues firing until the firing capacitor has recharged. This transistor’s gate is driven by the thresholding subcircuit. The drain is connected to the firing capacitor, and the source to ground. Since this transistor switches only when the neuron fires, then the dynamic power consumption dependency is on $f_{fire}$.

The load capacitances seen by this NMOS are its own gate-to-drain overlap output capacitance ($C_{gdM11}$), its own drain-to-bulk junction output capacitances ($C_{dbM11}$), the gate capacitances of the transistors in the driven inverter gate ($C_{gM1}$ & $C_{gM2}$), the gate capacitances of the firing M3 PMOS ($C_{gM3}$) and the dominant firing capacitor connected in parallel $C_{f_{fire}}$. 
With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (37), and detailed expressions for each load capacitance is given in Table 20. Therefore, 

\[
P_{M_{11}} = \alpha_{\text{fire}} V_{DD}^2 (C_{gd_{M11}} + C_{db_{M11}} + C_{g_{M1}} + C_{g_{M2}} + C_{g_{M3}} + C_{f_{fire}})
\]

(44)

**Table 20.** Capacitances contributing to dynamic power dissipation in the firing circuits M11 transistor.

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_{gd_{M11}})</td>
<td>(2 * C_{gd0_{N}} * W_{N})</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>(C_{db_{M11}})</td>
<td>(K_{eq_{N}} * A_{D_{N}} * C_{j} + K_{eqsw_{N}} * P_{D_{N}} * C_{j_{sw}})</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>(C_{g_{M1}})</td>
<td>(W_{P} * (C_{gd0_{P}} + C_{gs0_{P}}) + C_{ox} * W_{P} * L_{P})</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>(C_{g_{M2}})</td>
<td>(W_{N} * (C_{gd0_{N}} + C_{gs0_{N}}) + C_{ox} * W_{N} * L_{N})</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>(C_{g_{M3}})</td>
<td>(W_{P} * (C_{gd0_{P}} + C_{gs0_{P}}) + C_{ox} * W_{P} * L_{P})</td>
<td>Channel and Overlap Capacitances</td>
</tr>
<tr>
<td>(C_{f_{fire}})</td>
<td>(C_{f_{fire}})</td>
<td>Dominant Capacitance</td>
</tr>
</tbody>
</table>

**4.3.7. INVERTER CHAIN’S 1X GATE**

This gate is part of the inverter chain in the firing subcircuit. The more inverters in the chain, the higher the power consumption of the cell. Biological neural systems operate at exceptionally low frequencies (~1kHz), this contributes significantly to the low power consumption of the neural processes. For this power estimation, since low energy is preferable to speed, only this one inverter gate will be considered for the chain. All the inverters in the chain switch only when the action potential firing has been triggered. Hence, the dynamic power consumption is dependent on \(f_{fire}\).

The load capacitances seen by this inverter are its own gate-to-drain overlap output capacitances \((C_{gd_{M1}} \& C_{gd_{M1}})\), its own drain-to-bulk junction output
capacitances \( (C_{dbM_1} \& C_{dbM_2}) \), the gate capacitances of the driven PMOS \( (C_{g\varphi_P}) \), the gate capacitances of the driven NMOS \( (C_{g\varphi_N}) \).

With all these capacitances already identified, a general power consumption expression for this component is now defined in equation (38), and detailed expressions for each load capacitance is given in Table 21. Therefore,

\[
P_{\text{delay}} = \alpha f_{\text{fire}} V_{DD}^2 \left( C_{gdM_1} + C_{db} + C_{gdM_2} + C_{dbM_2} + C_{g\varphi} \right)
\]

\( \text{(45)} \)

<table>
<thead>
<tr>
<th>Capacitor</th>
<th>Expression</th>
<th>Comments/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{gdM_1} )</td>
<td>( 2 * C_{gd0P} * W_P )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbM_1} )</td>
<td>( K_{eqP} * AD_P * C_J + K_{eqswP} * PD_P * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{gdM_2} )</td>
<td>( 2 * C_{gd0N} * W_N )</td>
<td>Miller Effect Doubling</td>
</tr>
<tr>
<td>( C_{dbM_2} )</td>
<td>( K_{eqN} * AD_N * C_J + K_{eqswN} * PD_N * C_{jsw} )</td>
<td>Bottom Plate &amp; Sidewall Capacitances</td>
</tr>
<tr>
<td>( C_{g\varphi_P} )</td>
<td>( W_P \left( C_{gd0P} + C_{gs0P} \right) + C_{ox} * W_P * L_P )</td>
<td>Increases in size with the # of columns</td>
</tr>
<tr>
<td>( C_{g\varphi_N} )</td>
<td>( W_N \left( C_{gd0N} + C_{gs0N} \right) + C_{ox} * W_N * L_N )</td>
<td>Channel and Overlap Capacitances</td>
</tr>
</tbody>
</table>

### 4.3.8. FIRING CIRCUIT’S OHMIC POWER CONSUMPTION

The firing subcircuit in the column output cell has a resistor which controls the firing period \( \tau \), during which the cell transmits inhibition signals towards the receptive field. Like any ohmic component, some power dissipation will occur due to the Joule effect. This is given by: \[ P_R = V_{DD}^2 / R_{fire} \] \( \text{(46)} \)

### 4.4. ENERGY ANALYSIS DISCUSSION

This chapter presented an energy efficiency analysis that predicts and characterizes the energy efficiency of the developed NC hardware. The models show that the rate of energy consumption is directly proportional to the sparsity of
the signal, the resistive values storing memory in the reservoir substrate, and the capacity of the cell’s accumulators. The model follows a simple approach using only a few primitive architecture parameters. Unlike other design analyses presented in the previous chapters, this energy efficiency modelling focused on the absolute minimum fundamental limits of the developed unconventional neuromorphic computing architectures. Based on parasitic capacitance modelling techniques, it is shown that the energy efficiency of the developed NC architecture can be accurately estimated using these models. Moreover, the presented energy efficiency analysis can be used in future design spaces to explore tradeoffs, key limitations, and opportunities for today’s memristor-based computing architectures. This energy efficiency modelling could be expanded to account for the presence of operational defects in the computational substrate.

4.5. CHAPTER SUMMARY

It has been shown that the dynamic power consumption is primarily due to parasitic capacitances in the peripheral circuitry of the NC system. However, the dominating load generating dissipation is given by the accumulation state capacitances. Also, a secondary power consumption source is the VMM operation itself. Hence, subthreshold operation of the architectures could allow for even further energy consumption reductions.
V. RESEARCH OUTCOMES

5.1. SYNOPSIS

The neuromorphic computing circuit development process in this work focused on CMOS neuron design and optimization that loosely mimics their biological counterpart cells. The architecture designs presented in Chapter 2 assumed operation with the ideal memristor model, for which a trained synapse remains in its selected state during inference calculations. This simplification allowed the design effort to focus on CMOS neuron optimization without memristor constraints due to device variability. With this assumption, two CMOS-based NC architectures were designed, verified, fabricated, and evaluated completing one whole VLSI design cycle. The developed in-silico neurons were proof-of-concept systems consisting of a limited number of neurons. These fabricated systems were evaluated systematically and modularly and demonstrated an acceptable bioinspired circuit operation. Full scale radionuclide identification system functionality was evaluated only using SPICE-based circuit simulation tools.

Then, in chapter 3, a study of the nonideal effects of the memristors' manufacturing process was introduced. The effects in computation accuracy and throughput of the proposed architectures were explored in detail by primarily relying on computer-based simulations. The nonidealities studied included synaptic variability due to bit and line defects in the crossbar, discretization effects of the memristor states, and effects of test signal-to-noise levels in the incoming radionuclide spectra inputs. These memristor nonideal characteristics are inherent to the manufacturing process. Hence, they remain an ongoing challenge for NC
circuit designs. However, through the device reconfigurability rescuing schemes proper of neuromorphic computing architectures the variability these nonidealities may be overcome. Circuit adaptation techniques and rescuing schemes to address this memristor problem were proposed but remain uncorroborated in-silico.

In Chapter 4 an energy analysis of the developed NC architectures was performed. It was shown that dynamic power consumption is primarily caused by parasitic capacitances in the peripheral circuitry of the developed NC systems. However, most of the load power dissipation is caused by accumulation state capacitances. In addition, the operation of the VMM itself is a secondary source of power consumption. The architectures could therefore be operated at subthreshold levels, thereby reducing energy consumption even further.
5.2. CONCLUSIONS

Below summarizes the list of contributions made in this thesis project:

• Two energy-efficient memristor-based neuromorphic computing systems for radionuclide isotope detection from mobile platforms have been developed, the first one being a fully analog implementation and the second a mixed-signal circuit. Both architectures have been assessed for accuracy implementing a 27-common radionuclide dictionary into a crossbar with 55,296 memristor devices. Each radionuclide class is mapped into a 2048-bit memristor column representing the isotopes spectra in equally weighted energy bins. A postsynaptic cell was used for each radionuclide class implemented in the dictionary. The full-scale analog system contains 27 low-power operational amplifiers functioning as virtual grounds. The mixed signal architecture also implements a presynaptic cell for each of the 2048 energy bins in the system.

• The proposed neuromorphic computing system has been validated using SPICE-based device-level circuit simulation.

• The system's performance has been analyzed using industry standard simulation tools to estimate power consumption due to intrinsic CMOS-layout characteristics.

• The developed architectures have been optimized for design parameters that allow low-power operation, high robustness, and maximum throughput.

• Circuit-level simulations have been performed to assess the system using real-world spectra patterns from common radionuclides.
• Two 5x6-neuron prototype neuromorphic computing architectures operating in classification mode have been created and evaluated using MOSIS 0.5 um technology-node.

• A prototype neuromorphic computing architecture that operates in search mode to localize radionuclide sources has been designed and presented schematically.

• Hardware proof-of-concept prototypes for the developed neuromorphic computing architectures have been fabricated and evaluated using commercial off-the-shelf components and a standard Raspberry Pi Microcontrollers.
5.3. RECOMMENDATIONS

Below is the list of recommendations for the NC architectures in a given project:

• Implement spiking-based VMM architectures to access the maximum power consumption reductions possible, when designing a memristor-based neuromorphic computing architecture. These computing systems are a great alternative to the conventional computing technology currently used for radionuclide detection applications. Moreover, the Spiking VMM configuration will yield a higher efficiency than an analog circuit approach when the spiking average is lower than the encoded analog signal.

• Integrate high-yield memristor crossbars with adaptable CMOS-neurons. The memristor nondeterministic fabrication process and their resulting omnipresent manufacturing defects can significantly degrade computation accuracy when device yield is low. The device's defect nature also determines computation accuracy; shorted bits generate a larger accuracy reduction than open devices.

• Exploit the inherent neuroplasticity and self-healing characteristics of neuromorphic systems by employing rescuing schemes to overcome computational substrate defects. In this work, the computation dynamics advantages of neuronal inhibition and excitation signals were studied in detail. Results demonstrate a reduction of memristor defect effects when inhibition signals are backpropagated across the crossbar.

• Consider neuromorphic computing architectures as an acceptable alternative to conventional von Neumann architectures for radionuclide detection applications when low power systems are required.
5.4. IMPLICATIONS

This research work has established a foundation for developing energy-efficient NC systems for radiation detection and isotope identification in mobile scenarios using emerging device technology and highly application- and hardware-specific algorithms. This approach has the potential for breakthroughs well beyond the current application, because the developed architectures have the capacity of integrating highly customized and novel approaches seamlessly. In comparison to other detection applications, implementation of this architecture may result drastically in higher energy efficiency. The architectures developed in this work permit energy savings by removing unnecessary abstraction layers and harnessing memristor dynamics more directly.

Detection and identification of radioactive materials in mobile situations application have been directly ameliorated by this research. This research will enable the development of novel, energy-efficient, autonomous sensing platforms that utilize embedded chip-scale hardware. Moreover, this NC architecture open avenues to reduce detection platform size and weight.

In situations where energy efficiency and reliability are crucial, the results of this dissertation will provide solutions. Different sensors could, for instance, be used on the platform to accommodate other applications. It is possible to extend this technology to a more generic "artificial nose", which would be capable of detecting a broad range of potentially hazardous chemicals. This research could also extend to detecting biological threats. Internet-of-Things (IoT) principles can be used to combine the platform into a large-scale smart sensor network.
5.5. FUTURE WORK

Future work effectuated during the next VLSI cycle process should incorporate the lessons learned from these first-generation neurons. To address memristor nonidealities, these potential second-generation neurons should incorporate neuronal adaptation schemes such as thresholding optimization, receptive field device redundancy, memristor retraining schemes, and fault diagnosis schemes.

Moreover, any device allowing for in situ modifiable resistances would suffice to implement these neuromorphic circuits. Hence, future work should explore the promising possibility of integrating the developed neurons with other memory processing devices, e.g., phase change memory and CBRAM.
APPENDICES

A1. NC VECTOR MATRIX MULTIPLICATION CONFIGURATIONS

A2. 0.5-µm TRANSISTOR MODELS USED FOR SPICE SIMULATION

A3. DEVELOPED NC PRESYNAPTIC CELL SPICE CODE

A4. DEVELOPED NC POSTSYNAPTIC CELL SPICE CODE

A5. NEUROMORPHIC COMPUTING MODELS TAXONOMY

A6. RADIONUCLIDE DETECTOR INTERFACE

A7. CMOS PROCESS LAYERS IN L-EDIT LAYOUT TOOL
A1. NC VECTOR MATRIX MULTIPLICATION CONFIGURATIONS

Energy Efficient Vector Matrix Multiplications (VMM) can be achieved by exploiting the substrate’s operational dynamics. To implement memristor-based NC architectures, the crossbar must be complemented with CMOS-based peripheral devices. Some common configurations are shown in Figure 53. See [58], [76] for a detailed comparison discussion of VMM architectures.

Spiking architectures will consume less power than a virtual ground or termination resistor architecture when the average spiking train rates of the input signals are less than the analog voltage values they represent. However, spiking architectures have slower conversion times, so the required throughput of the NC architecture must be considered for selection of a configuration. Termination
resistor-based architectures also may be more energy efficient than the power-hungry virtual ground configuration approach.

Contingent on the synaptic weight requirements of the NC application, a compensation circuit may be necessary to account for the non-zero conductance of the memristor’s high resistance state. A differential amplifier operation between the output column measurement and the bias memristor column accounts for this consideration. These VMM configurations are shown in Figure 54.

**Figure 54.** VMM configurations implementing bias column difference to compensate for the non-zero conductance of memristors in the high resistance state.
When negative weights are required for computation, using a double element memristor synapse is a widespread practice. This is shown on Figure 54.

Figure 44. VMM configurations using memristor pair synapses.
A2. 0.5-µm TRANSISTOR MODELS USED FOR SPICE SIMULATIONS

* DATE: Nov 9/09
* LOT: T97T WAF: 4101
* Temperature_parameters=Default

```
.MODEL CMOSN NMOS (LEVEL = 49
+VERSION = 3.1 T Nom = 27 TOX = 1.39E-8
+X J = 1.5E-7 N CH = 1.7E17 VTH0 = 0.6284838
+K1 = 0.941838 K2 = -0.1170233 K3 = 33.2097067
+K3B = -8.6556859 W0 = 1.014244E-8 NLX = 1E-9
+DVT0W = 0 DVT1W = 0 DVT2W = 0
+DVT0 = 0.7587781 DVT1 = 0.3471058 DVT2 = -0.5
+U0 = 449.4127448 UA = 1E-13 UB = 1.210026E-18
+UC = 3.758274E-14 VSAT = 1.971734E5 A0 = 0.5600365
+AGS = 0.1202672 B0 = 1.992867E-6 B1 = 5E-6
+KETA = -8.240253E-3 A1 = 8.889262E-5 A2 = 0.3
+RDSW = 1.051566E3 PRWG = 0.1240431 PRWB = 0.0265251
+WR = 1 WINT = 2.42907E-7 LINT = 8.825924E-8
+XL = 1E-7 XW = 0 DWG = -2.515965E-9
+DWB = 4.505667E-8 VOFF = 0 NFACTOR = 0.8372875
+CIT = 0 CDSC = 2.4E-4 CDSCD = 0
+CDSCB = 0 ETA0 = 1.341016E-3 ETA B = 0.2716776
+DSUB = 6.576602E-3 PCLM = 2.1514826 PDIBLC1 = 1.35594E-4
```
+PDIBLC2 = 1.069922E-3   PDIBLCB = -0.2754349   DROUT = 1.721849E-4

+PSCBE1 = 2.509279E9   PSCBE2 = 1E-3   PVAG = 0
+DELTA = 0.01   RSH = 84.7   MOBMOD = 1
+PRT = 0   UTE = -1.5   KT1 = -0.11
+KT1L = 0   KT2 = 0.022   UA1 = 4.31E-9
+UB1 = -7.61E-18   UC1 = -5.6E-11   AT = 3.3E4

+WL = 0   WLN = 1   WW = 0
+WVN = 1   WWL = 0   LL = 0
+LLN = 1   LW = 0   LWN = 1

+LWL = 0   CAPMOD = 2   XPART = 0.5
+CGDO = 1.83E-10   CGSO = 1.83E-10   CGBO = 1E-9
+CJ = 4.147437E-4   PB = 0.834387   MJ = 0.4259021
+CJSW = 3.339021E-10   PBSW = 0.8055683   MJSW = 0.1887251
+CJSWG = 1.64E-10   PBSWG = 0.8055683   MJSWG = 0.1887251
+CF = 0   PVTH0 = -0.0447187   PRDSW = 500

+PK2 = -0.0588965   WKETA = 0.0147128   LKETA = 9.862682E-4

*.

.MODEL CMOSP PMOS

+VERSION = 3.1   TNOM = 27   TOX = 1.39E-8
+XJ = 1.5E-7   NCH = 1.7E17   VTH0 = -0.9152268
+K1 = 0.553472   K2 = 7.871921E-3   K3 = 8.6654583
+K3B = -0.1807851   W0 = 1E-8   NLX = 5.090632E-8

116
+DVT0W = 0  DVT1W = 0  DVT2W = 0
+DVT0 = 0.6014393  DVT1 = 0.3483694  DVT2 = -0.3
+U0 = 201.3603195  UA = 2.408572E-9  UB = 1E-21
+UC = -1E-10  VSAT = 9.431124E4  A0 = 0.9163618
+AGS = 0.1287533  B0 = 4.884433E-7  B1 = 1.095657E-9
+KETA = -4.865785E-3  A1 = 1.34055E-4  A2 = 0.6420955
+RDSW = 3E3  PRWG = -0.0281455  PRWB = -0.0479095
+WR = 1  WINT = 2.753034E-7  LINT = 1.226641E-7
+XL = 1E-7  XW = 0  DWG = -1.199648E-9
+DWB = -1.443743E-8  VOFF = -0.0643708  NFACTOR = 1.1413154
+CIT = 0  CDSC = 2.4E-4  CDSCD = 0
+CDSCB = 0  ETA0 = 5.822694E-4  ETAB = -0.2
+DSUB = 1  PCLM = 2.3211743  PDIBLC1 = 0.0474373
+PDIBLC2 = 3.088998E-3  PDIBLCB = -0.0352776  DROUT = 0.258157
+PSCBE1 = 1E8  PSCBE2 = 3.356147E-9  PVAG = 3.064255E-3
+DELTA = 0.01  RSH = 107.5  MOBMOD = 1
+PRT = 0  UTE = -1.5  KT1 = -0.11
+KT1L = 0  KT2 = 0.022  UA1 = 4.31E-9
+UB1 = -7.61E-18  UC1 = -5.6E-11  AT = 3.3E4
+WL = 0  WLN = 1  WW = 0
+WWN = 1  WWL = 0  LL = 0
+LLN = 1  LW = 0  LWN = 1
+LWL = 0  CAPMOD = 2  XPART = 0.5
+CGDO  = 2.3E-10  CGSO  = 2.3E-10  CGBO  = 1E-9
+CJ    = 7.209632E-4  PB    = 0.870762  MJ    = 0.4909036
+CJSW  = 2.088154E-10  PBSW  = 0.8  MJSW  = 0.1735655
+CJSWG = 6.4E-11  PBSWG  = 0.8  MJSWG = 0.1735655
+CF    = 0  PVTH0  = 5.98016E-3  PRDSW = 14.8598424
+PK2   = 3.73981E-3  WKETA = 5.940243E-3  LKETA = -0.0120242
 )
A3. PRESYNAPTIC NEURON SIMULATIONS

SPICE CODE

*RowHeader

.subckt rowHeader spikeIn cbIn notFiringAny Vdd

*Discharging NMOS Transistors

M1 drain1 spikeIn drain2 Gnd CMOSN L=0.6u W=3.0u *Discharges Cinh when spikeIn & notFiringAny
M2 drain2 notFiringAny Gnd Gnd CMOSN L=0.6u W=3.0u *Discharges Cinh when notFiringAny & spikeIn

*Passive Elements of Row Header

Cinh charge Gnd 1.2p  *The Inhibition Capacitor to determine the tau
Rinh charge drainA 40k  *The Inhibition Resistance to determine the tau

*Not Gate

M3 notCharge charge Vdd Vdd CMOSP L=0.6u W=3.0u *PMOS for a 1x Inverter
M4 notCharge charge Gnd Gnd CMOSN L=0.6u W=1.5u *NMOS for a 1x Inverter
*Nand Gate for
M5 nandOut  spikeIn  Vdd Vdd CMOSP  \textbf{L}=0.6u  \textbf{W}=3.0u  *PMOS for a 1x NAND gate
M6 nandOut  notCharge  Vdd Vdd CMOSP  \textbf{L}=0.6u  \textbf{W}=3.0u  *PMOS for a 1x NAND gate
M7 betweenPMOS spikeIn  Gnd Gnd CMOSN  \textbf{L}=0.6u  \textbf{W}=3.0u  *NMOS for a 1x NAND gate
M8 nandOut  notCharge  betweenNMOS Vdd CMOSN  \textbf{L}=0.6u  \textbf{W}=3.0u  *NMOS for a 1x NAND gate

*Not Gate
M9  signal nandOut  Vdd Vdd CMOSP  \textbf{L}=0.6u  \textbf{W}=3.0u  *PMOS for a 1x Inverter
M10 signal nandOut Gnd Gnd CMOSN  \textbf{L}=0.6u  \textbf{W}=1.5u  *NMOS for a 1x Inverter

*Transmission Logic of Row Header
X1 signal cbIn notFiringAny Vdd transGate
M0 charge notFiringAny cbIn Vdd CMOSP  \textbf{L}=0.6u  \textbf{W}=37.5u  *Allows charging Cinh when a Firing Neuron

.ends rowHeader
A4. POSTSYNAPTIC NEURON SIMULATION SPICE CODE

*Column Header**************************************************************************************

.subckt columnHeader cbOut firingSelf notFiringAny Vdd

*Self Firing Transistors

M1  firingSelf notfiringSelf Vdd Vdd CMOSP L=0.6u W=3.0u *INVERTER PMOS for firingSelf negation

M2  firingSelf notfiringSelf Gnd Gnd CMOSN L=0.6u W=1.5u *INVERTER NMOS for firingSelf negation

M3  cbOut      notFiringSelf Vdd Vdd CMOSP L=0.6u W=36u *This PMOS feeds Vdd to cb when firing

*Transmission Gate

X1 cbOut intState notFiringAny Vdd transGate

*Internal State Circuit

M4  intState   firingAny Gnd Gnd CMOSN L=0.6u W=1.5u *This NMOS drains the capacitor when theres any neuron firing

M5  firingAny notFiringAny Vdd Vdd CMOSP L=0.6u W=3.0u *PMOS for the inverter of firingAny
M6  firingAny notFiringAny Gnd Gnd CMOSN \( L=0.6u \) \( W=1.5u \) *NMOS for the inverter of firingAny
Cst intState Gnd 1.2p

*ThresholdingCircuit Based on an Inbalanced First Inverter with \( V_m=1V \)
M7  threshold intState Vdd Vdd CMOSP \( L=0.6u \) \( W=1.5u \)  *First INVERTER PMOS for thresholding Circuit
M8  threshold intState Gnd Gnd CMOSN \( L=0.6u \) \( W=9.75u \)  *First INVERTER NMOS for thresholding Circuit
M9  firing  threshold Vdd Vdd CMOSP \( L=0.6u \) \( W=3.0u \)  *Second INVERTER PMOS for thresholding Circuit
M10 firing  threshold Gnd Gnd CMOSN \( L=0.6u \) \( W=1.5u \)  *Second INVERTER NMOS for thresholding Circuit

*Output Spike Circuit
M11 outSpike firing Gnd Gnd CMOSN \( L=0.6u \) \( W=1.5u \)  *This NMOS trigers the output Spike when firing
Rfire outSpike Vdd 40k  *Resistor for Output Spike in Firing Circuit
Cfire outSpike Gnd 1.2p  *Capacitor for Output Spike in Firing Circuit

*Inverter Chain
M12 outSpike1 outSpike Vdd Vdd CMOSP \( L=0.6u \) \( W=2.7u \)  *PMOS for first Not of Inverter Chain
M13 outSpike1  outSpike  Gnd Gnd CMOSN L=0.6u W=1.5u
*NMOS for first Not of Inverter Chain
M14 outSpike2  outSpike1 Vdd Vdd CMOSP L=0.6u W=13.65u
*PMOS for second Not of Inverter Chain
M15 outSpike2  outSpike1 Gnd Gnd CMOSN L=0.6u W=7.65u
*NMOS for second Not of Inverter Chain
M16 outSpike3  outSpike2 Vdd Vdd CMOSP L=0.6u W=68.25u
*PMOS for third Not of Inverter Chain
M17 outSpike3  outSpike2 Gnd Gnd CMOSN L=0.6u W=38.25u
*NMOS for third Not of Inverter Chain
M18 outSpike4  outSpike3 Vdd Vdd CMOSP L=0.6u W=344.7u
*PMOS for fourth Not of Inverter Chain
M19 outSpike4  outSpike3 Gnd Gnd CMOSN L=0.6u W=191.7u
*NMOS for fourth Not of Inverter Chain
M20 firingSelf outSpike4 Vdd Vdd CMOSP L=0.6u W=1.7235m
*PMOS for fifth Not of Inverter Chain
M21 firingSelf outSpike4 Gnd Gnd CMOSN L=0.6u W=958.5u
*NMOS for fifth Not of Inverter Chain

.ends columnHeader

***************************************************************************
A5. NEUROMORPHIC COMPUTING MODELS TAXONOMY

The theoretical neuroscience models that have been developed by the NC community can be loosely categorized according to the taxonomy system shown in Figure 55. The neuronal and synaptic models are categorized according to their fidelity to the biological neurons and their complexity. The network models are divided based on their connectivity level. The neurons developed in this work are hardware implementations of the analog-spiking integrate-and-fire model family. The system as a whole is a sparsely and locally connected recurrent network.

Figure 55. Neuromorphic computing model categories.
A6. RADIONUCLIDE DETECTOR INTERFACE

Radiation exposure was not required for this study, since all the radionuclide spectra data was acquired from the National Nuclear Data Center Nuclear Website [106] and the Nuclear Wallet Cards [107]. To evaluate the system’s performance, various gamma-ray spectra of different durations, distances, activities, and backgrounds were mapped into the memristor dictionary based on the collected data sets. Co-57, Cs-137, Eu-152, Co-60, Pu-239, and U-235 were among the test spectra used as control inputs. The test sets were created to quantify the effect of resolution, efficiency, and background on system accuracy.

In this Appendix the signal encoding scheme used to interface with the radionuclide detection hardware is presented. These schemes would be implemented in hardware via an interfacing pulse-frequency modulation encoding circuit [108] connecting the analog spectroscopic detector data with the input presynaptic neurons. The spike trains representing the signal enter the presynaptic cell via the \textit{SpikeIn} terminal of the CMOS neuron.

There are several encoding and decoding schemes that have been used to implement neuromorphic systems in the past. Among the most famous schemes are rate-encoding and temporal-encoding. Other less popular approaches include phase-encoding, stochastic encoding, and burst encoding. For this circuit application rate-encoded signals are used to represent the radionuclides.

Rate encoding is a popular encoding scheme because it is the simplest data representation encoding scheme available. Rate encoding is the spike encoding scheme present in biological neural networks (BNN).
In rate encoding the analog value to be encoded, i.e., the neuronal signal that is carrying the information is associated with the average frequency of spikes in a train of pulses. The neuron transmits the signals through its mean firing rate. For radionuclide applications the encoding circuit maps a high-count energy bin with a higher spike rate than an energy bin with less spikes. Figure 56 shows an example of frequency neural encoding of Eu-152.

Figure 56. Frequency neural encoding of Eu-152.
A7. CMOS PROCESS LAYERS IN L-EDIT LAYOUT TOOL

The following key to interpret layout schematics is added here for those readers who are unfamiliar with the VLSI design process.

<table>
<thead>
<tr>
<th>Layout</th>
<th>Color</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well (p,n)</td>
<td>Yellow</td>
<td><img src="image" alt="Yellow" /></td>
</tr>
<tr>
<td>Active Area (n+, p+)</td>
<td>Green</td>
<td><img src="image" alt="Green" /></td>
</tr>
<tr>
<td>Select (p+, n+)</td>
<td>Green</td>
<td><img src="image" alt="Green" /></td>
</tr>
<tr>
<td>Polysilicon</td>
<td>Red</td>
<td><img src="image" alt="Red" /></td>
</tr>
<tr>
<td>Metal 1</td>
<td>Blue</td>
<td><img src="image" alt="Blue" /></td>
</tr>
<tr>
<td>Metal 2</td>
<td>Grey</td>
<td><img src="image" alt="Grey" /></td>
</tr>
<tr>
<td>Contact to Poly</td>
<td>Black</td>
<td><img src="image" alt="Black" /></td>
</tr>
<tr>
<td>Contact to Diffusion</td>
<td>Black</td>
<td><img src="image" alt="Black" /></td>
</tr>
<tr>
<td>Via</td>
<td>Black</td>
<td><img src="image" alt="Black" /></td>
</tr>
</tbody>
</table>
REFERENCES


Architectures (NANOARCH), Jul. 2013, pp. 1–6, doi: 10.1109/NanoArch.2013.6623028.


143


