Hardware for Memristive Neuromorphic Systems with Reliable Programming and Online Learning

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Ryan John Weiss

December 2022

Dedicated to my family who have loved and supported me through this journey.

Acknowledgments

I would like to thank my advisor Dr. Garrett S. Rose for his dedication and support in the completion of my Phd at the University of Tennessee. I would also like to thank the members of my doctoral committee Dr. Nicole McFarlane, Dr. Andy Sarles, and Dr. Ahmedullah Aziz for serving on my committee and for their support and time. I could not have achieved this without the support of the TENN-lab research team. Thank you to everyone who has helped make this a possibility.

Abstract

Alternative computing technologies are highly sought after due to limitations on transistor fabrication improvements. Fabricated memristive technology allows for a non-volatile analog memory for neuromorphic computing. In an integrated CMOS process, the synapse circuits designed for a spiking neuromorphic system can use memristors to regulate accumulation in the neuron circuits. Testing the fabricated memristive devices composed of hafnium oxide and developing a model to represent the key device characteristics lead to specific design choices in implementing the analog memory core of the synapse circuit. The circuits I designed for neuromorphic computing in this process take advantage of the unique capabilities of the memristive device to store a programmable analog memory reliably and efficiently. I designed the peripheral circuitry required including the circuits for programming the memristor and for online learning capabilities.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The computer is a tool to ease human effort in solving difficult problems. The need for a device that allowed for consistent and quick computation led to the eventual invention of the modern computer. Advancements in computing have continued to improve their capabilities and increased the range of possible problems to solve. While the standard digital computer is capable of solving many problems and continues to improve, varying limitations are slowing the advancement [62]. The continued increase in the number of transistors on chip is going to cease as manufacturing smaller transistor feature sizes becomes increasingly more difficult [74]. The feature sizes of silicon transistors are reaching physical limitations as manufacturing closes in on the atomic size of silicon atoms. Photolithography, the technique used to produce transistors on silicon, attempts to continue to reduce feature sizes in silicon, but faces increasing challenges in engineering and cost [63]. Moore's law states the transistor count on chip doubling every two years [37]. The premise of this law implies improvements in computing technology are achieved through increased transistor count. While increasing available transistors can improve computing, the effective use of the available electronic devices can also be improved. Continued effort

in increasing device density will coincide with alternative computing paradigms to enhance future computing technology.

In conjunction with difficulties in increasing computing performance, potential power consumption is relatively increasing. Dennard scaling states the power consumption for smaller technology nodes stays equal to previous technology nodes due to equivalent decrease in necessary voltage and currents for new technology nodes [11]. This no longer holds as feature sizes continue to decrease. For continually decreasing transistor sizes, leakage power is increasing while supply voltage and clock frequency are stagnating. The increase leakage current is due to effects like reverse bias junction currents, band-to-band tunneling current, and tunneling into gate oxide [55]. Many of these sources of leakage become prevalent in deep sub micron processes due to the macro model of silicon transistors breaking down. The quantum effects of electrons allows currents to flow unintentionally. The increase in leakage current compounds negative effects due to increasing the temperature of the devices. Power limitations for advanced nodes have to thermal runaway into account. One option is slowing down operating frequency. With the increase in embedded battery powered processors, power is a key factor in improving computing systems.

The modern computer uses digital memory and logic to implement processing. The von Neumann architecture that dominates existing computers has a separation of main memory and the processing unit [82]. The separation of processing and memory requires a data bus to transfer the information required by the processor from the memory storage system. Improving computing technology with this architecture requires increased bus performance, otherwise the data sent across the bus will be a bottleneck for processing. To eliminate the potential von Neumann bottleneck, computing architectures can implement processing-in-memory computing. Processing-in-memory computing poses significant challenges in topological designs. This process of bringing computing to the memory adds additional difficulty in designing algorithms [64]. Another technique to improve computing capabilities beyond von Neumann architectures is quantum computing. Quantum computing consists

of an entirely different paradigm to computing that relies on quantum physics to produce the correct solution to a problem. Quantum computing requires a set of instructions using quantum bits to solve problems. The ability to design quantum computers posses challenges in resilience to environmental noise [16]. These alternatives to the standard computing architecture posses difficulties in both programming and circuit level design.

An architecture for improving computing technology through bio-mimicry is neuromorphic computing. Brain inspiration in computing considers biological processes for naturally accomplished computing and implements equivalent functions in artificial systems [39]. The biological brain efficiently uses high density of cells for computation [13]. Mimicking brain functionality has the potential to solve computing challenges with low power cost. Through the use of neurons and synapses, the brain is a computing system that builds in features to learn and improve functionality. The adaptability inspired by biological functions creates possibilities for this architecture to outperform traditional computing systems [22]. The biological functions performed by the brain are also analog in nature. Analog computing was overtaken by digital computing due to the lack of precision. The neuromorphic computing can take advantage of the brain's nature by build highly efficient analog circuits.

Alongside improvements to transistor technology, other devices have been developed for computing. The memristor is a two terminal element that can store information as a resistance on previous usage [15]. The memristor has potential as a memory storage element to fit into the neuromorphic framework as an adaptable analog memory unit [34]. The ideal device would have an infinite life cycle of a continuous analog range of resistive memory states. Practically, there are challenges with fabricated memristor devices [33]. As manufacturing technology improves the devices, the circuits that utilize them can be further optimized and tailored to the devices. For neuromorphic computing the memristor and similar devices show potential to implement the plasticity of biological functions. Two of the main components of neuromorphic computing, synapses and neurons, when implemented using analog circuits have the need for an adaptive analog memory

element. For synapses, the memristor can fit in as a direct representation. The synapse controls the efficiency of communication between neurons, and the memristor can be used as the representation of that connection. Changes in synaptic efficiency are then directly correlated with changes in the resistance of the memristor. To implement memristors as synapses, circuits needed to control and utilize them are needed. The characteristics of the memristor make them suitable for use in an analog neuromorphic system [79].

1.2 Research Goal

Neuromorphic computing can take advantage of the unique properties of the memristor to implement analog memory circuits. In this work, I study the key characteristics of the integration of a hybrid CMOS-memristor VLSI process to develop circuits for neuromorphic computing. The fabrication of on-chip memristors in-line with a CMOS process allows for large scale integration of memristive neuromorphic systems-on-chip (SOC). To implement the memristive neuromorphic SOC, all aspects of the fabrication of the memristors and their utilization are taken into account. The goal of this research is to build the circuits for a reliable and programmable integrated memristor-CMOS fabrication process. The design is grounded in tested results of the devices used.

This work explores the idea of current compliance for programming and implements an online learning technique that is commonly adapted for these circuits. Table 1.1 shows the common usage of memristors and similar devices as memory elements in neuromorphic computing. The methods to program the devices allow for the online learning functionality. This work utilizes circuitry required for the device's basic operations to implement the programming and online learning capabilities seen in this field.

1.3 Research Contribution

The research contributions are enumerated below:

Table 1.1: VLSI synapse implementations.

	Memory Type	Programming	Online Learning
This work	Memristor	Current Compliance	STDP
[1]	Memristor	Pulse Shaping	STDP
[77]	Memristor	Pulse Count	STDP
[25]	Memristor	Pulse Shaping	STDP
[30]	PCM	Pulse Shaping	STDP

- A simple compact model for the memristor that highlights the relationship of current and voltage on the device. The model takes into account the device response to signals that fully change the updated resistive state.
- A memristor circuit using fabricated on-chip memristors that can perform all necessary functions to use the device as an analog memory. The functions include: forming, set, reset, and read.
- A low-power, low-complexity read out operation based on current limitation for the memristor.
- A current control system for implementing current control for both programming and read out of the memristor.
- A current generating circuit for the memristor to act as a synapse to output current into a neuron.
- A current steering digital to analog conversion circuit to allow for current programming of the memristor.
- A reference generation circuit using a memristor to create a range of output currents for the synapse circuit.
- An online learning circuit to implement spike timing dependent plasticity using the current control programming of the memristor.

1.4 Dissertation Overview

In this dissertation, I will describe the design of analog neuromorphic circuits using memristors. The goal of this work is to show circuits that efficiently use the device technologies available while providing insight into other potential devices and techniques. The core analog circuits used here for neuromorphic computing are designed to utilize the

memristive devices. Testing results from the devices led to particular design choices for efficient plastic synapse and neuron circuits. I have implemented a particular flavor of synapse circuit to utilize the devices available. The synapse is designed to take advantage of the characteristics of the memristor device for simple and reliable programming and updating. It is also designed for low area and power cost. The programming technique stems from the physically tested results and is effectively modelled for simulation [36].

Chapter 2

Background

2.1 Nano Electronic Devices

The motivation of brain inspired computing leads to different technological needs than standard computing. The potential advantages of neuromorphic computing stems from a highly dense and interconnected memory system. One long term memory is the synapses ability to manage the communication of information between neurons. The activity of the neurons can alter their connected synapses to improve computing performance. The ideal memory for this system is fabricated at high density, representing the value of synaptic strength, and is easily adjustable. In implementing neuromorphic computing, stored values are not required to be digitally precise as neural networks can tolerate a degree of noise. This allows the use of analog circuits, and specifically analog memory elements to implement neural circuits. Analog circuits can implement some mathematical functions efficiently, for example adding to values represented as currents on a wire is accomplished by connecting the wires together. An analog value can be converted from a digital system using a digital to analog converter, but this will require area and power to represent and convert the value. Improving upon this, an analog memory can be created using a multitude of devices, including standard silicon. A standard silicon process is important due to the ease of integrating the memory with the processing circuits. Floating gate

transistors can be used to implement the analog memory in neuromorphic computing using a standard CMOS process [17]. The need for a highly dense analog memory has led to the development of alternative types of non-volatile memory devices. Such analog memory devices for neuromorphic computing include memristors, phase change materials, spintronics, and ferroelectric FETs [68].

Phase change material, or PCM, offers analog memory storage and updating. PCM are made of chalcogenide materials, which are made from one chalcogen and one electropositive element. Chalcogens are in the same family as oxygen in the periodic table of elements. An example PCM device is a stack of Ag, Ag_2Se, and Ge_2Se_3 [49]. The chalcogenide material used in these devices changes phase between amorphous and crystalline giving a continuous range of resistance values [48]. These devices can be used to mimic a synaptic learning rule, STDP [66]. The temperature effect on the memory of the device adds additional challenges to efficiently pack in a large amount of synapses on chip [51, 52]. Scaling up the number of devices on chip is a challenge due to high programming current. The resistance in the amorphous state is unstable and varies over time. The device endurance is limited by the number of times the resistance is increased. Systems using this device take in special considerations to reduce the number of resistance increases, control temperature and current levels.

Spintronic devices also called magnetic tunnel junctions, MTJ, use a three layer structure. The layers consist of two magnetic layers with an-oxide layer in between [69]. One magnetic layer is fixed, and the other is free. The free layer can be parallel or antiparallel magnetization relative to the fixed layer, with each orientation yielding a different resistance value. These devices have a programmable resistance range and can achieve synaptic plasticity like STDP. However, the resistance range can be as small as $5\,\Omega$ [32]. A small ratio between the highest and lowest resistance states requires more precise circuitry to detect the different states. These devices require low power to operate, but have high switching times. They have long endurance because the switching mechanism

does not require location changes to atoms. However, these devices require improvements in variability and scalability.

A ferroelectric field-effect transistor, FeFET, is a three terminal device similar to a metal-oxide-semiconductor field-effect transistor, MOSFET. Unlike a MOSFET, a FeFET uses a ferromagnetic layer instead of the highly resistive gate-oxide. The properties of FeFET devices are similar to MOSFET, but the ferromagnetic gate layer allows for a programmable channel conductivity [45]. These devices show a wide resistance range and fast programmablity. They have symmetric switching for increasing and decreasing channel conductivity. They require large area. Decreasing area and improving scalability generates problems with leakage current, reliability, manufacturing, energy consumption, and sensing capability. They are charge-based memory and suffer from leakage similar to DRAM [44].

The transition metal-oxide memristor, or TMO, is a two terminal device consisting of a switching layer made from a depleted-oxide layer sandwiched between two electrodes. The device stores analog memory states by growing and rupturing a metallic conductive filament through the metal-oxide layer. The composition of the switching layer and the electrodes play a key factor in determining the switching characteristics. A key concern with TMO devices is the variability due to filament growth and rupture [23, 24]. The devices are scalable with a high resistance ratio, but improve the variability to a tolerable level in neuromorphic computing can worsen the switching characteristics. This issue will be addressed in this work by implementing circuits to operate the device by controlling filament growth. The goal is to remove variability due to filament growth by actively controlling the process. The trade off is reducing the resistance range to only filament formed resistances.

These devices are all under active research. The best candidate for neuromorphic computing is yet to be determined. All the device types show the minimum requirements, which are the ability to store an analog memory. Table 2.1 shows the best reported values for key characteristics, not necessarily from the same exact device, adapted from [81].

Table 2.1: Nonvolatile memory devices

Device	Operating Voltage	Power	Size	Resistance Ratio	Switching times	Endurance cycles
Memristor	0.3V	55fj	$4nm^2$	10^6	300ps	10^12
PCM	0.9V	100fj	$100nm^2$	10^5	300ps	10^12
Ferroelectric	0.9V	1fj	$25000nm^2$	10^6	10ns	10^5
Spintronic	1V	1fj	$800nm^2$	<10^1	1ns	10^15

From this table, there is no clear winner. The devices suffer from different set backs, including temperature, resistance range, and leakage. To further explore these devices for neuromorphic computing they must be implemented with the rest of the system. This includes all the circuits for controlling the synapses, programming, plasticity, connectivity, and the neurons. The circuits designed for the neuromorphic processor using these devices take into account their specific device properties to best utilize them. This work focuses on the hafnium-oxide memristor. The circuit techniques used to create a neuromorphic component of the device can potentially be applied to other devices if they show similar properties.

2.2 Memristor Theory and History

The memristor has been termed the fourth basic circuit element among electronic devices [15]. Resistors, capacitors, and inductors link current and voltage, charge and voltage, and current and flux, respectively. The theoretical memristor device links flux and charge [15]. The resulting property from this definition creates a controllable state for the resistance of the device that is built on the history of device operation. The ability to hold information representing previous usage of the device is why the device is called a memristor, a term short for "memory resistor". The resulting phenomena of linking flux and charge gives way to variations of resistance from the history of the applied voltage or current. In a current versus voltage graph, this is characterized by a hysteresis loop seen in Figure 2.1. The high and low resistances states, HRS and LRS, go through the origin, and at high and low voltages, the resistance switches. This idea of a memory in resistance was only theoretical for many years. The theory of a resistance holding memory based on previously applied voltages or currents led to the possibility for other devices to exhibit similar properties. Aside from memristors that can vary in resistance based on previous usage, memcapacitors

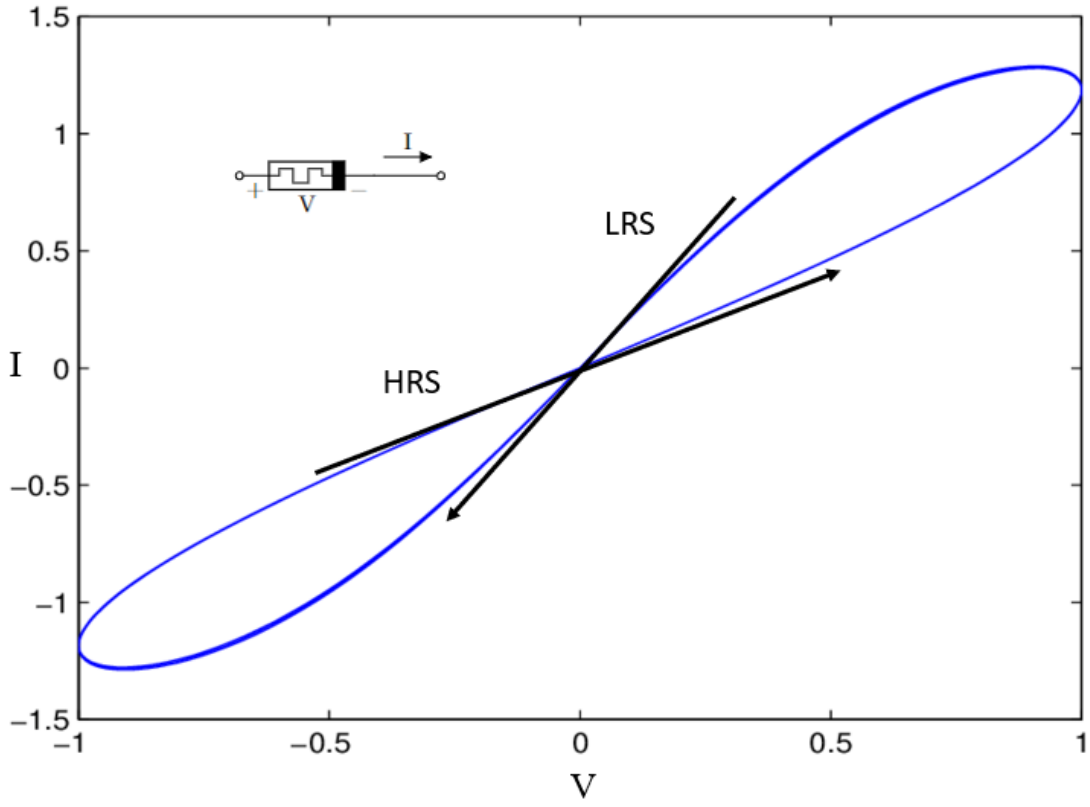

Figure 2.1: An ideal I-V sweep of a memristor showing device symbol, LRS and HRS.Adapted from [72]

and meminductors are other possible devices that story a state based on previous usage [76]. These memory devices pose challenges in fabrication but show potential for circuit applications.

2.3 Fabricated Hafnium-Oxide Memristors

The memristor was first fabricated in 2008 [65]. Different materials have been used with varying results for potential memory applications. The transition metal-oxide memristor, TMO, is one type of memristor which has shown great results and promise for inline fabrication in CMOS processes. The key characteristics for different applications vary slightly. For this work, an analog resistive memory is allows for analog computation, reducing area and power.

The fabricated device used in this process is a made of hafnium-oxide, a transition metal-oxide memristor. It has characteristics suitable for analog memory [10]. The resistance of the device can vary from below 10 kΩ to over 100 kΩ [36]. The common operations for using these devices fabricated in silicon includes forming, setting, and resetting. The forming operation is a one time requirement to initialize the device [7]. Before forming the device is in a very high resistive state. Forming generates the conductive filament by applying a high voltage on the device [21]. The set and reset operations are used to change the resistance value of an already formed device. The reset operation occurs when a voltage is applied with the opposite polarity to the forming voltage. The high resistance state, HRS, achieved through the reset operation has high variability, ranging from 10 kΩ to 1 MΩ in some devices [21]. The set process generates a low resistance state, LRS, ranging from 100 Ω to 1 kΩ in those devices [21]. The devices used in this work have shown HRS ranges from 10 kΩ to 300 kΩ and LRS range from 2 kΩ to 30 kΩ [53]. A binary application greatly benefits from separating the maximum possible resistance and minimum possible resistance, as it is detecting the correct state is more likely. Analog applications require programming into intermediate states, preferable a continuous range from the highest to

lowest resistances. Achieving specific resistance values and a continuous range between HRS and LRS is theoretically possible, but presents challenges in accurately hitting target resistances. The set and reset operations decrease and increase the resistance, respectively. The reset operation for the fabricated hafnium-oxide memristors can achieve varying resistance by varying the voltage applied to the device or by varying the time the voltage is applied [10]. The low resistance state can vary by controlling the current through the device [36]. The low resistance state is achieved in under 100 ps while the high resistance state is achieved in 100 ns [10]. The magnitude difference in time required to fully set and reset the device implies pulses with opposite polarity will not result in equal resistance changes. The greater time to fully reset allows for incremental pulse based switching, but those pulses will not work with the set operation. Also, the variability in change per pulse increases as the resistance increases. Pulse programming works for these hafnium memristors when in a low resistance state and staying near the low resistance state. The voltage applied to the memristor during reset also controls the maximum high resistance state achieved when fully reset. Due to the difference in the set and reset operations, equal increment and decrement in resistance with opposite polarity voltage pulses is not feasible. Therefore, the circuit is designed to use one repeatable and consistent resistance change method.

The method used for this design takes advantage of the low resistance state of the device and the properties of the set process. During forming and set a transistor is used to limit the current through the device. Figure 2.2 shows the one transistor one memristor, 1T1R, circuit using a n-type transistor. These memristor circuits sometimes refer to the device as a ReRAM, hence the R in 1T1R. The voltage V_{top} is positive for forming and set and negative for reset operations. The voltage V_g sets the saturation current, which limits the voltage through the memristor. For forming and setting the device, the current limiting transistor is necessary to achieve switching. Unlimited, the memristor can be permanently stuck in the low resistance state. Since the current limiting transistor is a necessary, the programming method reuses it to control the resistance. Changing the

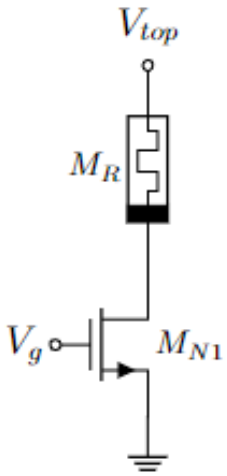

Figure 2.2: One transistor one memristor circuit adapted from [53].

resistance after forming is achieved by resetting the device into its high resistance state, and once in the high resistance state, the device is then set into the low resistance state. The low resistance state is controlled with V_g and can achieve a range of resistance values [36]. Using a transistor to limit the current, via the saturation current, shows the low resistance state achieved varies by a few kilo ohms. The current limitation from a transistor in saturation is accomplished by holding the gate at a specific voltage.

2.4 Alternative Memristors

Aside from hafnium-oxide, other types of transition metal-oxides can be used to create a memristor. Other TMO memristors include tantalum-oxide, titanium-oxide, and niobium-oxide [19]. These memristors have different values for properties such as their HRS, LRS, and switching voltages. One of the drawbacks of the hafnium-oxide memristor is the asymmetric switching characteristics. The titanium-oxide memristor does not exhibit the same difference in magnitude between the set and reset switching time [43]. However, these devices require a higher voltage for switching that is incompatible with deep sub-micron 65 nm CMOS process. These devices also have lower endurance [43]. Finding the best material for creating the memristor layer in a CMOS process is an ongoing area of fabrication research. Ideally the device has high endurance, is CMOS compatible, has a wide continuous, easily achievable resistance range, and is easily fabricated.

There are other devices to consider for different applications due to their different properties. Volatile memristors, or second order memristors, are a class of device that exhibit similar properties to the memristors considered in this work, in that they have definable states due to previously applied voltage and resistance, but these states decay fairly quickly over time. Practically, the volatile device can be programmed into a new state, but will shortly regress to its base condition. This has applications in timing specific circuits or thresholding circuits [6]. The insulator-metal transition memristor, IMT, is a type of memristor similar to the non-volatile TMO, except the resistance change is volatile.

The devices work by Joule heating where heating causes a drop in resistance followed by a quick cool down over time, resetting the change in resistance. This can be made from NbO_2 or VO_2 [80]. Research considering IMT devices show promise for theses devices as selectors [50] and potentially as neurons [75]. The devices switch from HRS to LRS when a high voltage is applied due to thermal energy. When the resistance is low, the temperature cools off and the device goes back to a high resistance state. This can work as a threshold for a neuron to activate, or a selector for a synaptic array, but is not equivalent to the nonvolatile TMO for a synaptic device that provides long term memory storage.

The dipole-induced bilayer, DIB, has exhibited properties similar to biological functions [47, 38]. This device is created by a lipid membrane formed when lipid filled water droplets are submerged in oil. The lipids have hydrophobic tails that coat the outside of the water droplet and cause two droplets to not coalesce when brought into contact. Instead the tails of the lipids form a layer that resembles a cell membrane. The droplets can then be doped with voltage controlled peptides. The droplet system without any additives shows a memcapcitive effect due to the change in geometry of the bilayer. The additives create memristive behavior due to the insertion mechanics of the peptides causing a change in resistance. These changes in capacitance and resistance are not permanent, but are also not entirely instantaneous voltage dependant. For both the change in capacitance and change in resistance properties, there is a timing element, were the devices returns to its original state after the voltage is removed. This can be used to store the timing information in synaptic plasticity functions such as spike rate dependent plasticity, SRDP, or spike timing dependent plasticity, STDP [47]. These devices show promise as low power synaptic mimics with dynamic resistance changes to stimuli. They have been connected to an artificial neuron circuit [71]. The circuits have to take into special consideration to not damage the device and fabrication is not inline CMOS compatible.

2.5 Neuromorphic Computing Framework

Neuromorphic computing architecture is one of the most effective alternatives to the existing von-Neumann architecture for certain applications. There are multiple neuromorphic computing frameworks including Intel Loihi and IBM TrueNorth [8, 18]. These computing architectures use biological inspired components to achieve high efficiency in implementing brain-like functions. The advantages of neuromorphic computing are not yet fully understood, but the systems show promising results for reducing power and resources needed to solve problems such as constraint satisfaction problems [3]. Neural networks, the key component to implementing neuromorphic computing, has many flavors. Artificial neural networks, ANN, such as multilayer perceptrons are a feed-forward network. An example single layer with a single output is shown in Figure 2.3. The layers consisting of weighted inputs sum together and activate the output. MLPs are trained with gradient descent. This type of neural network strays heavily from biology. Spiking neural networks, SNN, are event based. These neural networks use timing information of activity not present in ANNs. The computing functionality is realized with two brain inspired components, neurons and synapses. The functions of these basic cells as they relate to information processing is a significant unsolved problem. There are a multitude of properties relating to the biological functions of these cells that provide adaptability and efficient computing. Neuromorphic computing uses reduced mathematical models of these cells, limiting the functionality to core components. Biological neurons collect ions to charge their cell body and ultimately release the ions to the subsequent neurons via the synapse. The electrochemical nature of the cells can be interpolated with mathematical models that can then be implemented in digital or analog circuitry. A key factor in the computation is the efficiency with which neurons deliver electrochemical signals. The pathway between neurons is defined as the synapse for neuromorphic computing. The synapse holds a weight value that is a memory of the efficiency of the communication between two neurons. These two components inspired by biology have a plethora of other possible biological functions acting on them to implement the plasticity and learning capable of the brain. Hardware

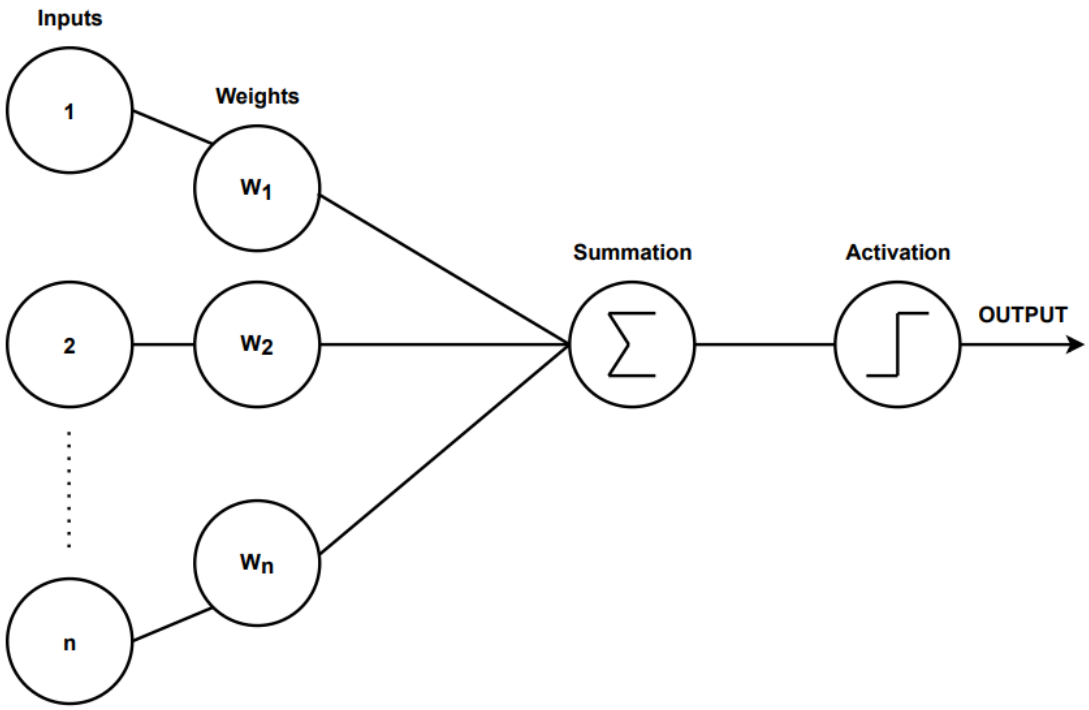

Figure 2.3: Single perceptron showing weighted inputs, summation, and output activation.

for neuromorphic computing builds these components with all the functionalities desired at a lower cost than running the functions through a general computer. The history of the framework for this project is NIDA, MrDANNA, and the current framework RAVENS[60, 14, 20, 54]. This framework uses an integrate and fire neuron model with connected synapses. The neuron has a threshold for accumulating inputs via synapses. The synapses hold weight information and delay the transmission of the information to the neuron. The setup for defining networks uses a genetic algorithm, evolutionary optimization for neuromorphic systems or EONS, to optimize the network [60, 61]. Optimization generally prioritizes accuracy or functionality, followed by low component count and low power cost. The circuits designed to implement the generated networks need to fulfill the roles given by the framework. The neuron accumulates inputs from the synapse, and the synapses transmit information between neurons with specific delay and strength. Other features, like synaptic plasticity, can be implemented in the framework, and needs to be reflected in the circuits.

2.6 Synapse and Neuron Circuits

The hardware designed for neuromorphic computing generally follows the biological inspiration. The inputs and outputs are connected to multiple synapses and neurons. While building these components out of purely digital circuits is possible, to truly match the biological nature analog circuits are designed [41]. The use of the term neuromorphic comes from Carver Mead's work on silicon VLSI neurons. The silicon neuron shown in Figure 2.4 is a biologically inspired circuit named after the axon hillock which is the part of the neuron that generates an action potential [40]. The circuit integrates an input on a capacitor, C_{mem}, creating a voltage, V_{mem}, and generates a pulse output upon crossing the buffer circuits threshold. Neuromorphic computing continually improved the relationship with circuits to their biological inspiration. The implementation of neuromorphic circuits is limited by the understanding of biology [42]. As the understanding of the principles of

neuromorphic computation improve, the circuits needed to implement the operations will be realized. Improvements to the computational models will elicit the need for the circuits.

In implementing neuromorphic systems, many analog neurons take an input current from the synapse [28]. The axon hillock neuron in Figure 2.4 contains the basis for a silicon neuron. The current into the neuron is integrated and accumulated with the result being a voltage stored on a capacitor. Analog neurons use the stored voltage with a comparator circuit to determine the output of the neuron. Some comparator circuit possibilities include an inverter or op-amp [40, 9]. The output generated when crossing the comparison threshold resets the voltage stored in the neuron and communicates with the next neuron through a synapse. Control circuitry is required to communicate and reset neurons. Synapse circuits respond to the neuron input information and output into the following neuron. The current generated from the synapse into the neuron is a function of the synapse's weight value. For a voltage applied to the synapse an output current can be generated with a resistance.

The memristor could fit into the synapse directly as the weight memory. However, memristors as synapses require additional control circuitry to achieve all the desired synaptic functions. The circuits implementing the neuromorphic functions should be both power and area efficient to outperform the same functions implemented on a traditional system. The benefits can be achieved by improving the circuit efficiency for the simplest models and improving the computing potential of the models used. Including more biological features can potentially improve the computing results and improving the circuitry designed to implement those functions will definitely help the neuromorphic system outperform a traditional counterpart.

Different neurons follow different mathematical models [28]. The integrate and fire neuron uses an integrator and a comparator to accumulate inputs and fire outputs [26]. The axon hillock neuron mimics the axon hillock of a biological neuron [40]. The axon hillock is the location where the output is first generated in the neuron. All neuron circuits have these functions: accumulate and store inputs and output upon crossing an accumulation

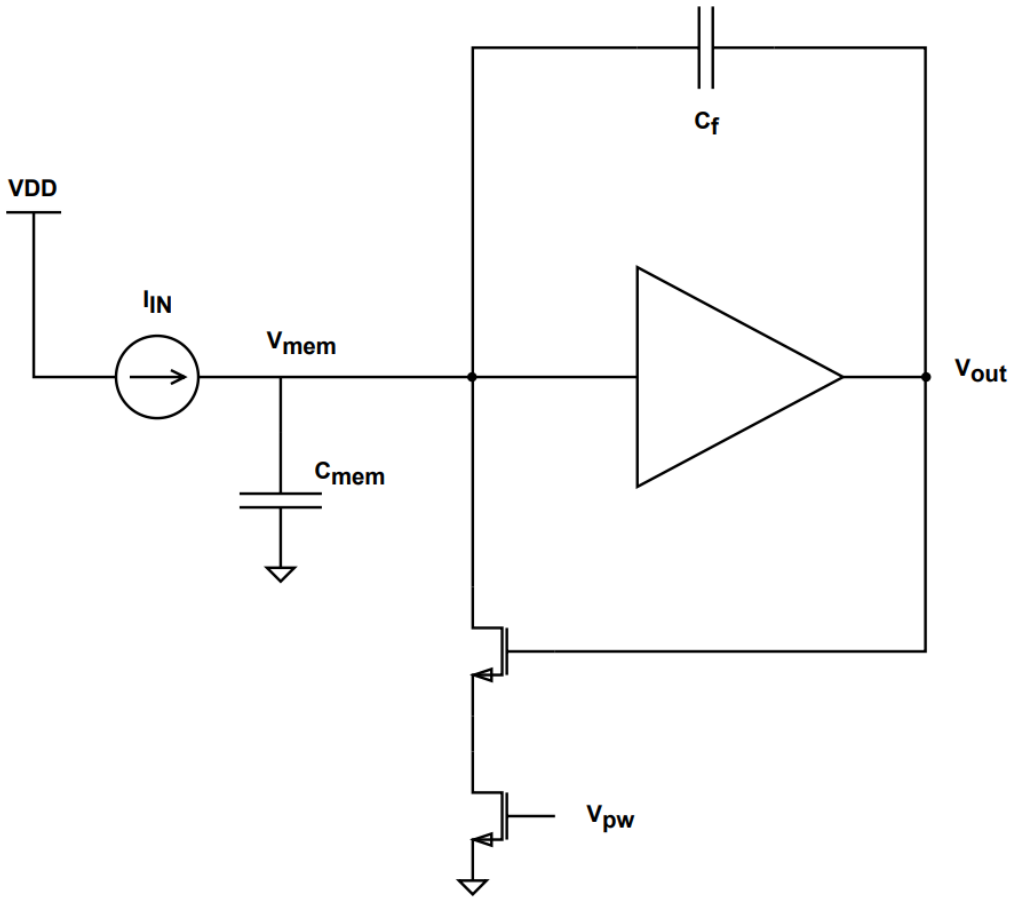

Figure 2.4: Original axon hillock neuron circuit [28, 40]

threshold. Some neuron circuits look to implement the intricacies of the ion channels in biological neurons [56]. Other neuron circuits have limited biological representation and are instead optimized for power consumption [31]. Neurons can also benefit from memristor. Neurons have been implemented making use of FeFET, IMT, PCM, and MTJ [35]. Mostly these are integrate and fire neurons.

The synapse can be implemented with transistors and capacitors [28]. These synapse design work by shaping the spike to achieve different accumulations at the input of the neuron. Synapses can consist of only a memristor or similar analog memory element. A common configuration for this is a two dimensional array, where the synapse is a memristor at the intersection of rows and columns. Rows are pre-synaptic neurons and columns are post-synaptic neurons. These configurations are crossbar arrays [9, 27, 33, 29]. The crossbar array has the benefit of a dense connectivity. Synapses connect neurons one to one, but are activated by row and column neurons. This configuration can have current flow through memristors not in use, called sneak paths. These memristor circuits use voltage control to implement both the programming and reading of the memristor. The neurons apply voltages which control what synapses are activated and for plasticity if implemented.

In earlier work, two memristors were used to hold the synaptic weight information [58, 57]. One synapse corresponded to the positive weight value and the other the negative weight value. Weights values came from the model, positive weights allowed for excitation and cause neurons to fire, while negative allowed for inhibition and stopped neurons from firing [59]. This synapse was designed to sink or source current from the neuron. The synapse had STDP functionality using voltage control. Similar to [46, 12] multiple memristors are used to help alleviate potential issues. For the hafnium-oxide memristors, one issue identified was asymmetric switching. Special circuit techniques were used to adjust for the difference in switching speeds.

This work takes into account the variability and asymmetric switching of the memristors fabricated in a hybrid CMOS process. I address the need for a consistent method of

programming and updating hafnium-oxide memristors with highly asymmetric switching properties. This required building all necessary components to implement a fully integrated system on a hybrid CMOS memristor wafer. The goal is implementing circuits that are low power and low area to take advantage of the current fabricated memristors. This is accomplished by using the low resistance state to program the device and taking special considerations to reduce power and simplify the required circuitry. The final value of the resistance achieved from the set operation is determined by the current. The current is controlled with one transistor that is reused to control current during a readout operation. This allows for a compact programming and memory storage.

Chapter 3

Memristor Model

To simulate circuits that use a memristive device, a model of the device is needed. The memristor's model should include the required characteristics to verify the circuits. The model accurately responds to the operations to update and read the memristor's resistance value. The hafnium-oxide memristor model used here is based on [5, 4]. The device models for hafnium-oxide memristors can be created from intrinsic characteristics of the materials used [78] or from fitting physically tested results of the device [53]. For this model, the characteristics are based on tested results and abstracts possible physiological phenomenon. The switching properties pertaining to the memristor need to be included in the model. Its ability to store a resistance and update that resistance state should match the tested results. The model I implemented for my circuit stems from the tested results of the device with a transistor and leads to my design choices for programming and updating the weight. I found in testing the device, updating the device's resistance is easily achieved with DC signals and created a model that reflects this idea.

3.1 Experimental Results for Hafnium-Oxide Memristor

The characteristics of the device are measured to understand and model device behavior. The tests for the memristors are usage-based, including forming voltage and switching

voltages, and usability of the device, including cycle to cycle variation and lifetime. The on chip device is part of a one transistor one memristor, 1T1R, test structure seen in Figure 2.2. The connected nodes of the device are the gate V_g and source of the n-type transistor and the top of the memristor V_top. The source of the transistor is held at ground and the gate is held at a DC voltage. At the top of the memristor, V_top is a specific voltage applied for different tests. The on chip fabricated devices starts as an unformed device. The unformed resistance is a high resistance and applying a high positive voltage at V_top causes it to form the conductive filament. The device is formed and the resulting voltage at which it formed is measured. Figure 3.1 shows a current response to voltage on the unformed device. The method of collection is further explained in Chapter 6. The forming voltage is temperature dependent, and at room temperature ranges from 1 V to 4 V, but is generally close to 2.5 V . The forming voltage is a high limiting factor for circuit design. The need for a high voltage on the unformed device also implies all circuits attached to those nodes need to be able to handle the high voltage. There are many potential solutions to this including preforming the device before it is connected to the silicon, changing the device to lower its forming voltage, or heating up the wafer to lower its forming voltage. In this work, the forming voltage is a key factor for device consideration at the circuit implementation and design level. The forming process coincides with the set process. In this way, after forming the device is in its low resistance state, which implies a filament is fully formed. A current limitation is needed during this process to reliably reset the device otherwise the device can be stuck in the low resistance state. Aside from the measured forming voltage, different applied current limitations are studied to determine the most reliable forming process.

After forming, the device is in a low resistance state. From here the device is cycled between high and low resistance states. The key characteristics for switching include set and reset switching voltages. For the reset process to occur a switching voltage above $-600\,\mathrm{mV}$ is applied at V_{top}. The maximum voltage applied in the reset process affects the high resistance state achieved, and a high negative voltage has the potential to adversely

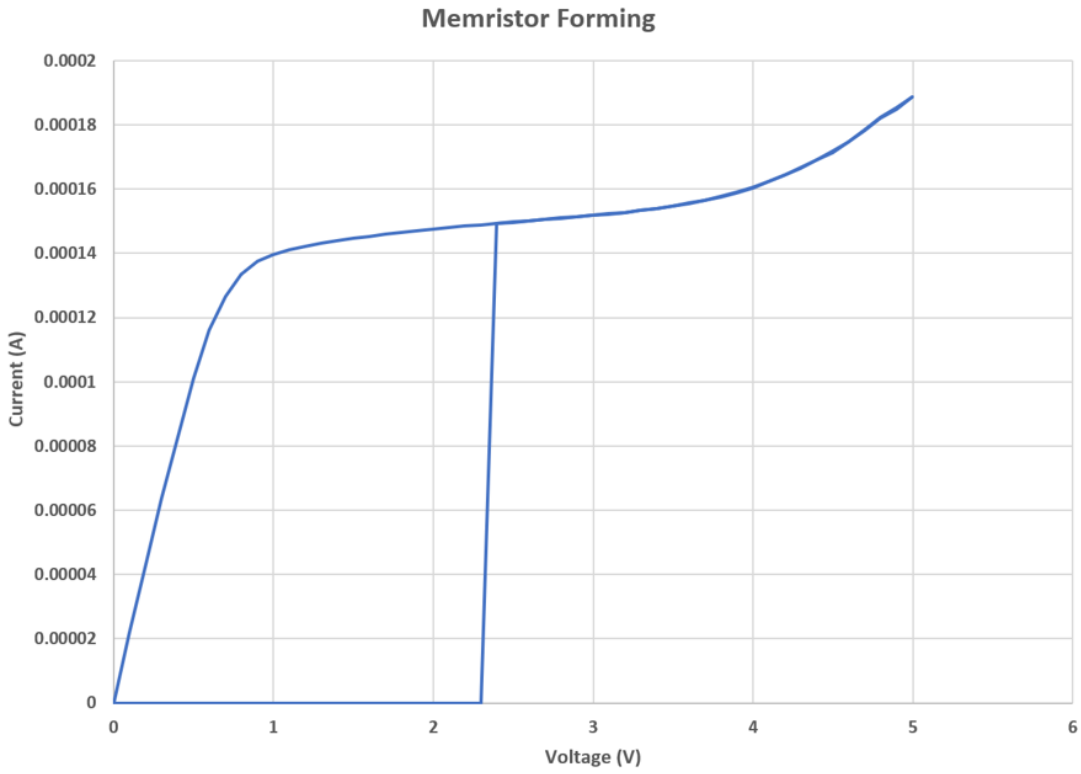

Figure 3.1: Voltage sweep of unformed hafnium-oxide memristor with conductive filament forming at 2.3 V.

affect the device . The reset process is not instantaneous and requires a high negative voltage to be applied for around 100 ns. Different high resistance values can be achieved by either applying varying negative voltages above the switching threshold at V_{top} or applying the voltage for different lengths of time. The pulses must be shorter than the time needed to fully reset the device to possibly see incremental switching. The pulse programming method for gradually increasing resistance shows good results during the initial resetting of the devices [10]. As the devices resistance gets higher, the possibility of the resistance jumping to a final high resistance value increases. The high resistance state is variable, and the relationship of pulse incriminating the resistance is consistent at lower resistances.

The set process is similar to the forming process. However the switching voltage for the set process is constant. It is around 600 mV. The time need for the forming and set process is almost instantaneous at under 100 ps [10]. Due to the fast switching speed, pulse based incremental switching is near impossible. However, this work takes advantage of the low resistance state due to its ability to be consistently programmed. This comes about from another mechanic of the filament forming process. The current limitation needed to reliably recreate the conductive filament has the effect of adjusting the low resistance state achieved. For different current limitations, around 100 µA, the low resistance state varies in a range around 5 kΩ. All these possible resistance values are achieved with the same voltage applied at V_{top}. Thus the applied voltage is not the dependent factor, and the current limitation is causing a different final resistance.

After testing the mechanisms for switching, the reliability of the device is tested. The main tests for this are the consistency of the device readout, and the longevity of the device. For the consistency of the readout, the device is programmed to a specific resistance value, and then the device is measured periodically. Ideally, the device will hold this resistance value indefinitely and each measurement will be identical. The device has a constant resistance value and thus is a non-volatile memory device. The readout is consistent over many read operations over a long time. The other reliability factor is the number of cycles of set and reset the device can go through and continue to achieve reliable resistance states.

For these devices, the reliability is a key factor with devices of resistance value changes with equivalent resistance changing processes.

3.2 Memristor Model

In my model I capture the the ability to reach different high and low resistance states through DC voltage signals. I do not concern myself with the intricacies of the switching and simply linearly increase or decrease at a definable specific rate. Figure 3.2 and 3.3 show state diagrams for the hafnium-oxide memristor. The four processes used in the device include forming, set, reset, and read. The device starts in an unformed state, and requires a high positive voltage to form the device. The forming voltage parameterized and is generally above 2 V at room temperature. After forming, the device is at a low resistance and requires a negative voltage with a greater magnitude than the negative switching threshold to increase resistance. The device's resistance decreases with a positive voltage above the positive switching threshold voltage. In any state, the device's current state can be determined with a read voltage that is a between the positive and negative switching threshold voltages. The model encapsulates these different switching capabilities. For a binary implementation, which only includes a singular high resistance state and a singular low resistance state, the model can simply jump from the low resistance to the high resistance state by using a constant positive and negative voltage at V_{top}. While the resistance during switching is intended to be outside the scope of my model, the final result from a varying voltage at V_{top} and V_g should achieve varying resistance levels. To accomplish this goal, I implemented a bounding system in the model that ends the set and reset processes. For the set, and similarly forming, the process begins with a positive voltage above the switching threshold and continues while above another specified voltage. For the reset process, the process starts with a highly negative voltage and continues while the current magnitude is above a specified current. This achieves variable low and high resistance states based on the applied stimuli.

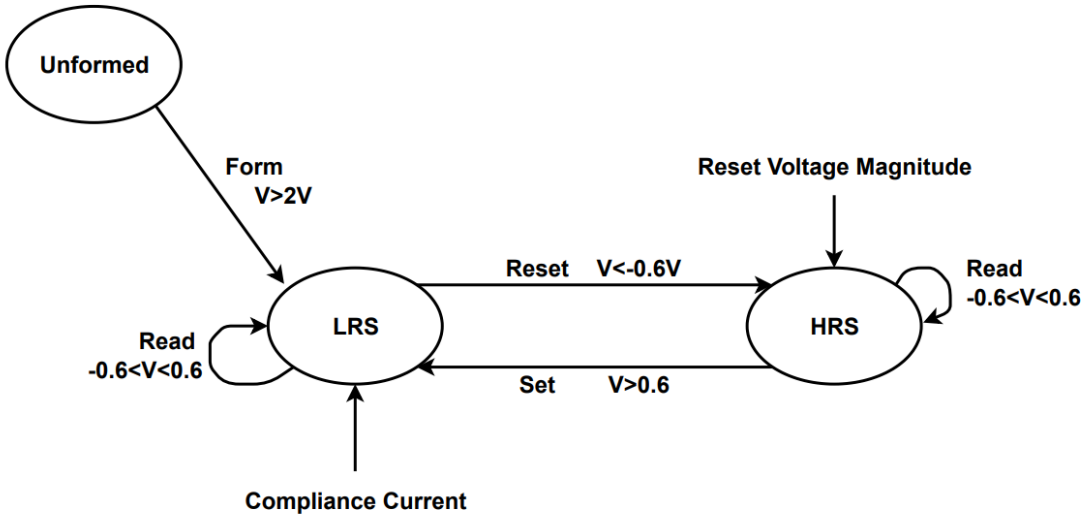

Figure 3.2: Flow state diagram of hafnium-oxide memristor.

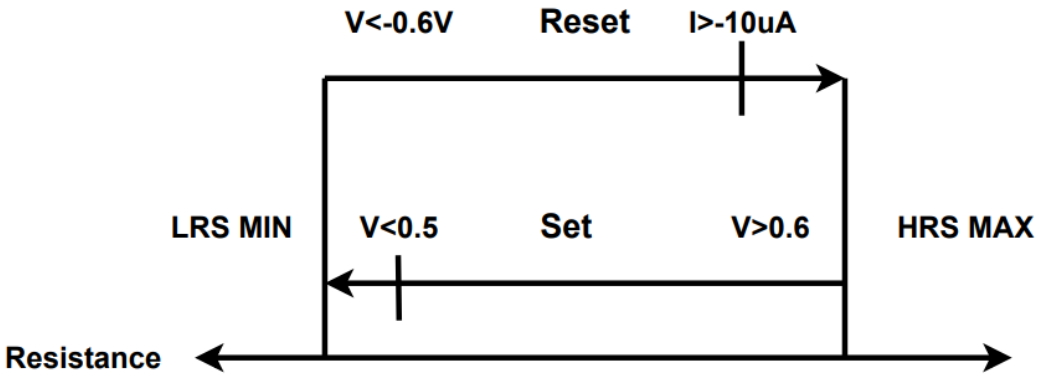

Figure 3.3: Diagram of hafnium-oxide memristor resistance movement.

Memristor's are well understood to have a threshold voltage for switching. A high positive voltage, 600 mV is approximately the positive switching voltage threshold Vtp for the set operation of the hafnium-oxide memristors I tested. I implemented my model with a second voltage threshold for the set operation. This voltage threshold lower bound $Vtpl$ can be approximated as a linear value, 400 mV. However, it is more accurately represented as a function of the device's current resistance value, and it increases as the device decreases. This causes a non-linearity in the relationship between current limitation and resistance achieved. The lower bound voltage allows for different low resistance values as a result of different current limitations due to varying V_g. As the resistance of the device lowers, the voltage across the current limiting transistor increases and the voltage across the device decreases. For higher transistor saturation currents of the current limiting transistor the voltage across the device will stay above the lower bound positive threshold at lower resistances. This coincides with the physical results of a higher gate voltage, or higher current limitation, resulting in a lower resistance. The memristor model uses the same idea for implementing the forming process, but uses different voltage thresholds Vf and Vfl that match the physically tested results.

3.2.1 Set Operation

The set operation reduces the resistance while above a specific voltage $Vtpl$. If the voltage applied stays above the lower bound, or has no current limiting device, the device will reach an absolute minimum resistance. While a linear threshold works to achieve the correct relationship between the current limiting transistor and the low resistance state, from device testing, a more accurate relationship can be calculated. The lower bound voltage can be calculated as the relationship between the current limiting transistors current multiplied by the final resistance achieved. From testing, the lower bound is calculated in the model as a function of the resistance. For the low resistance state, as the resistance decreases $Vtpl$ increases. The code snippet below shows the set process in the Verilog-A model. The lower bound voltage $Vtpl$ is calculated with a linear relationship

to the current resistance. The set process begins with a voltage across the device Vmr above the threshold Vtp and activates the start flag. The process then continues until the voltage is below the lower bound. In the process of switching, the resistance Rm is lowered linearly at a specified rate $td * dRmp$. The rate is calculated by the total range of resistances divided by the positive switching speed, which is 100 ps. If the resistance reaches the absolute minimum resistance the set process will also end.

```
1
2  //Define the function for the lower bound
3  //Vtpl = .6; //Constant approximation
4  Vtpl = -0.0001*Rm + 1;
5
6  //Activate flag for set process
7  if (Vmr >= Vtp && Rm > LRS_min) begin
8      Setstart = 1;
9  end
10
11 //Continue set process while flag, voltage, and resistance are high
12 if (Vmr > Vtpl && Setstart == 1) begin
13         Rm = Rm -  td * dRmp;
14         if (Rm_tmp <= LRS_min) begin
15                 Rm_tmp = LRS_min;
16                 Setstart = 0;
17         end
18 end
```

Simulation results for the set process show a close match to a physically tested device for the resulting resistance for different transistor saturation currents. Figure 3.4 shows the resulting resistance after the set process from different applied gate voltages to the current limiting transistor of a 1T1R circuit. The applied voltage was 1.5 V on the memristor and the gate voltage varied. The current through the device during the set process was

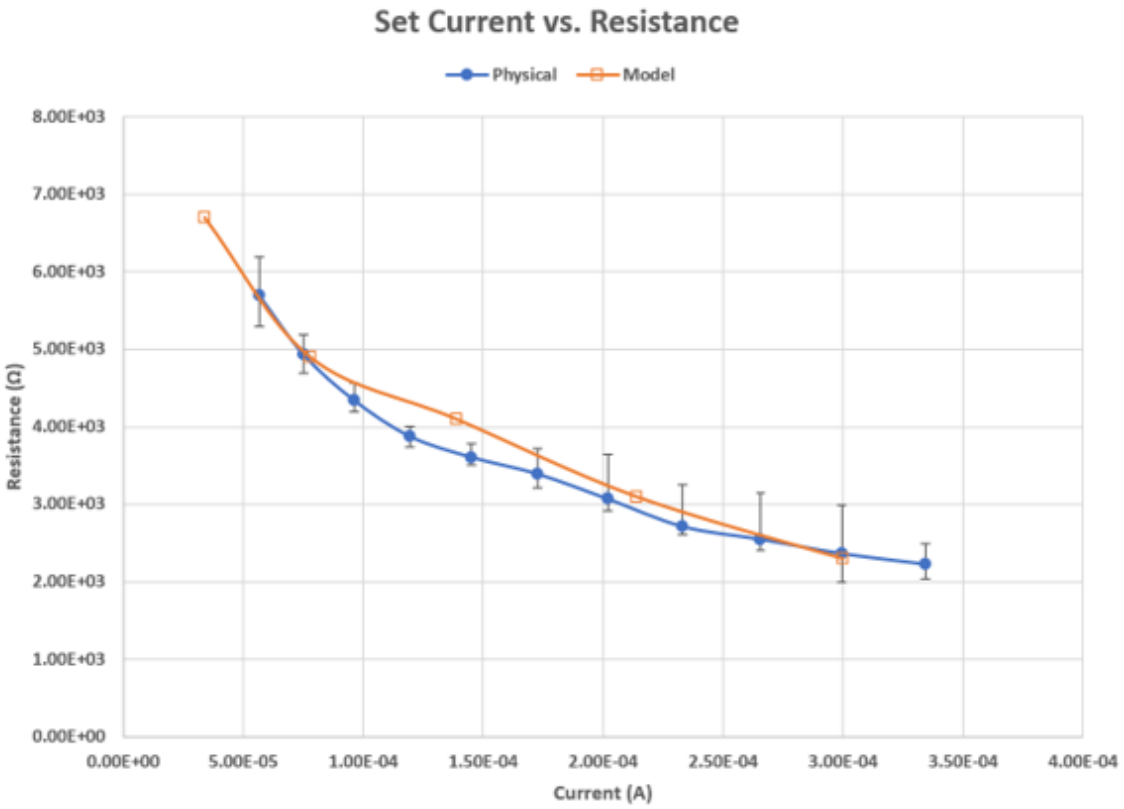

Figure 3.4: Model vs. Physical tested device set resistance at different currents.

directly proportional to the voltage at the gate of the transistor. The gate voltage on the transistor activated the transistor and applied the current limitation at its saturation current. While the resistance of the memristor is decreasing the voltage across the device is decreasing. The voltage across the drain to source of the transistor is increasing. Once the drain to source voltage is high enough, the current through the transistor's drain to source current saturates. The lower bound voltage for the set process is the voltage across the memristor for a given current limitation. This is calculated by multiplying the current limitation used in the set process and the final resulting resistance. Figure 3.5 shows the lower bound voltage $Vtpl$ for different low resistance states. In the simulation $Vtpl$ is fit to that specific device. While a constant voltage works, the lower bound voltage is fit with a function of resistance.

3.2.2 Reset Operation

The reset operation requires a negative voltage below the negative switching threshold. For the hafnium-oxide memristor, the negative voltage at V_{top} required to increase the resistance is approximately $-600\,\mathrm{mV}$. Like the set process, I implemented my model to start the reset process upon crossing this voltage threshold. However, unlike the set process, during the reset process the voltage magnitude across the device is not decreasing due to the change in resistance. To make an equivalent stopping mechanic in the model, I instead used the current through the device. I set a lower bound $Itpl$ of $10\,\mu\mathrm{A}$ as a required current to continue the reset process. This resulted in different high resistance states for the applied voltage. From testing, the lower bound is a function of the resistance. For the high resistance state, as the resistance increases $Itpl$ decreases. The code snippet below shows the reset process in the Verilog-A model. The lower bound voltage $Itnl$ is calculated with a linear relationship to resistance. The reset process begins with a voltage across the device Vmr below the threshold Vtn and activates the start flag. The process then continues until the current is above the lower bound. In the process of switching, the resistance Rm is lowered linearly at a specified rate $td * dRmn$. The rate is calculated by

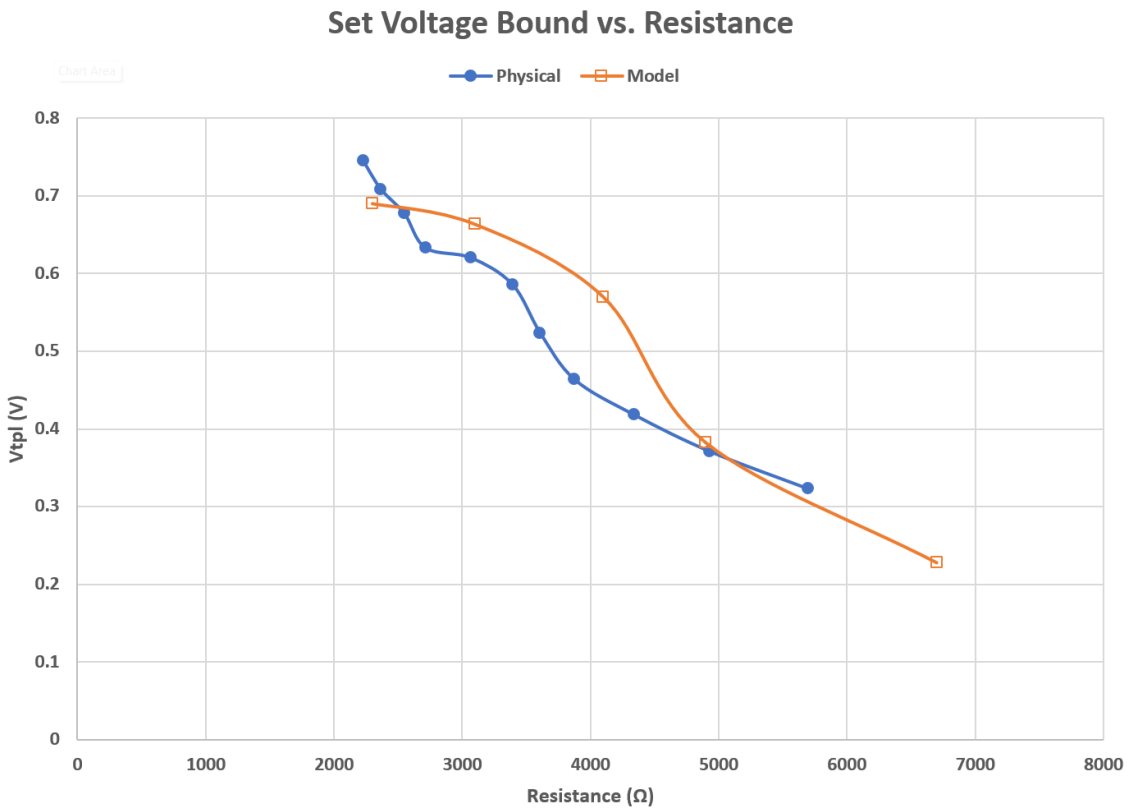

Figure 3.5: Model vs. Physical tested device calculated voltage across device during set at LRS.

the total range of resistances divided by the negative switching speed, which is 100 ns. If the resistance reaches the absolute maximum resistance the reset process will also end.

```
1
2  //Define the function for the lower bound
3  //Itnl = 1e-5; //Constant approximation
4  Itnl = 8e-11*Rm-2.19e-5;
5
6  //Activate flag for reset process
7  if (Vmr <= Vtn && Rm < HRS_max ) begin
8      Resetstart = 1;
9  end
10
11 //Continue set process while flag, voltage, and resistance are high
12 if (Imr <= Itnl && Resetstart == 1) begin
13          Rm = Rm +  td * dRmn;
14     if (Rm_tmp >= HRS_max) begin
15               Rm_tmp = HRS_max;
16                    Resetstart = 0;
17          end
18 end
```

Simulation results for the reset process show a close match to a physically tested device for the resulting resistance for different reset voltages applied. Figure 3.6 shows the resulting resistance after the reset process from different negative voltages on the memristor. The applied voltage was 1 V to 1.45 V on the memristor. The current through the device during the reset process reduces as the resistance increases. The varying applied voltages allow for a higher current at higher resistances. The lower bound current for the set process is the measured current. In the simulation, *Itnl* is fit to the recorded current measurements of that device. Figure 3.7 shows the lower bound current *Itnl* for different

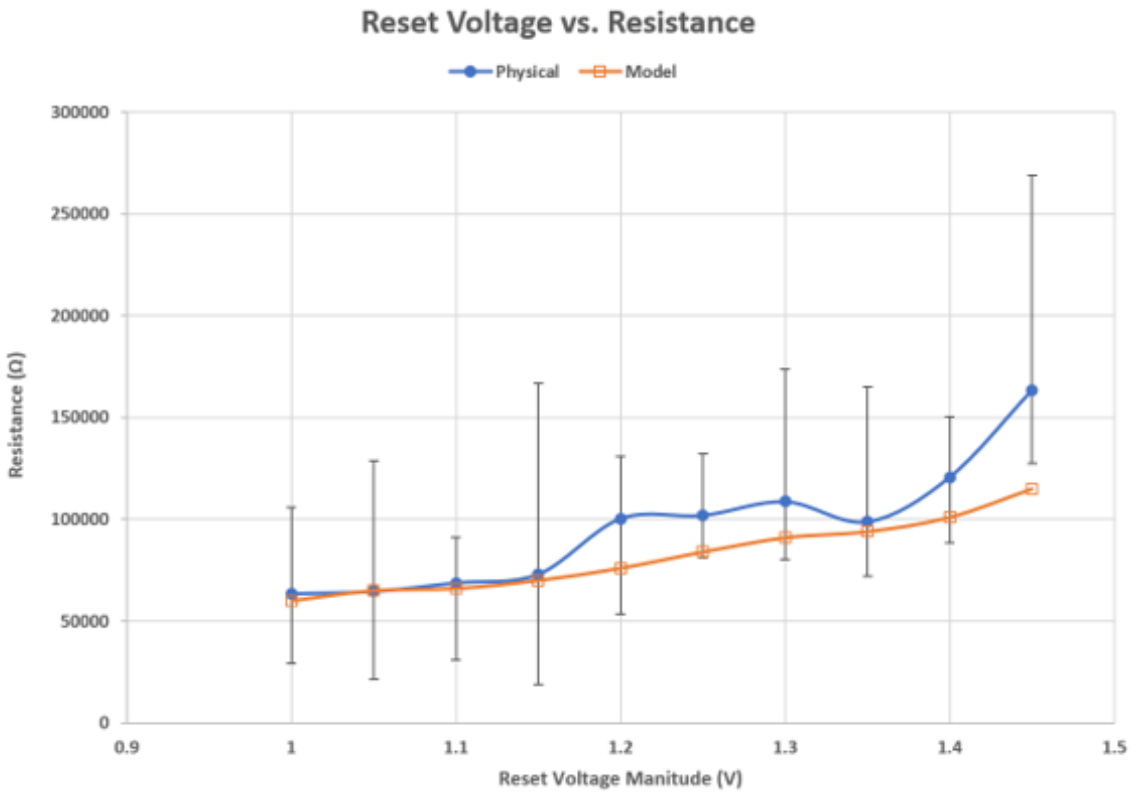

Figure 3.6: Model vs. Physical tested device reset resistance at different voltages.

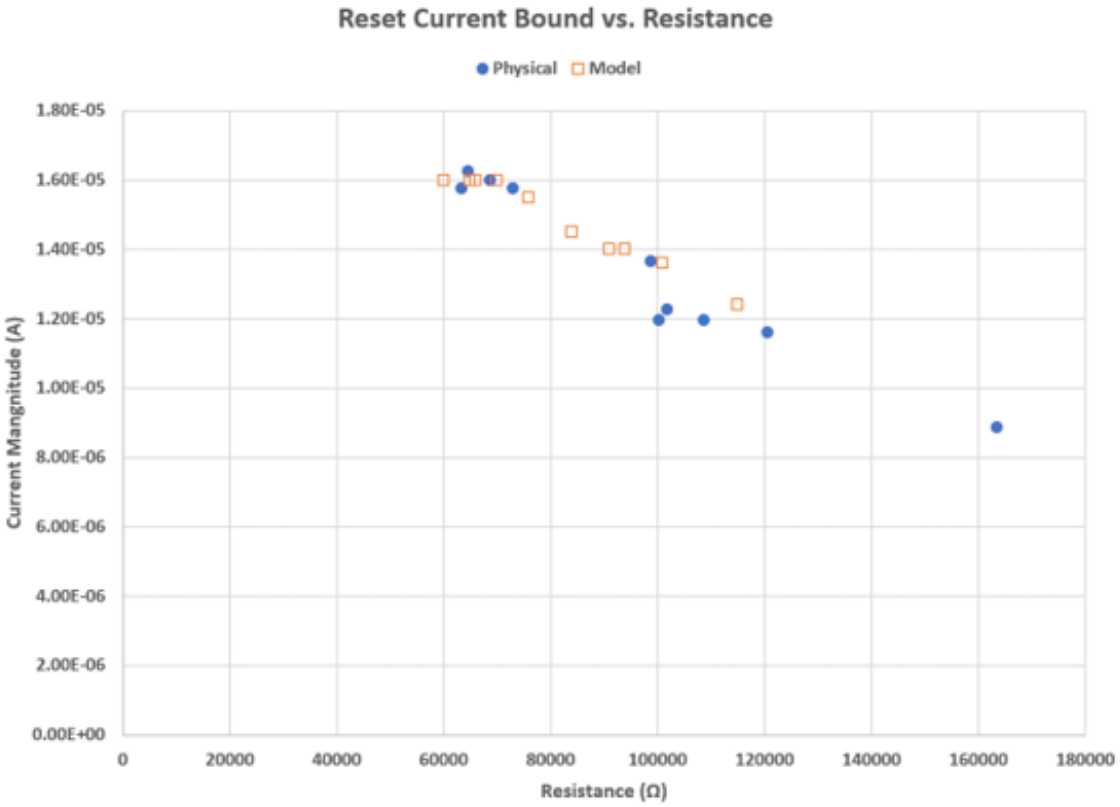

Figure 3.7: Model vs. Physical tested device calculated current through the device during reset at HRS.

low resistance states. While a constant current works, the lower bound voltage is fit with a function of resistance.

Simulation results for both set and reset for the model show the characteristic hysteresis loop when sweeping voltage. Figure 3.8 shows the transition between HRS and LRS when sweeping voltage from $-1.5\,\mathrm{V}$ to $1.5\,\mathrm{V}$ in both the model and a physically tested device. The model uses a linear relationship between the resistance and the switching bounding variables, $Itnl$ and $Vtpl$. Table 3.1 shows the fitting parameters to best approximate the relationship. This model is used in Chapter 6 to show the change in resistance due to change in current compliance for the set operation.

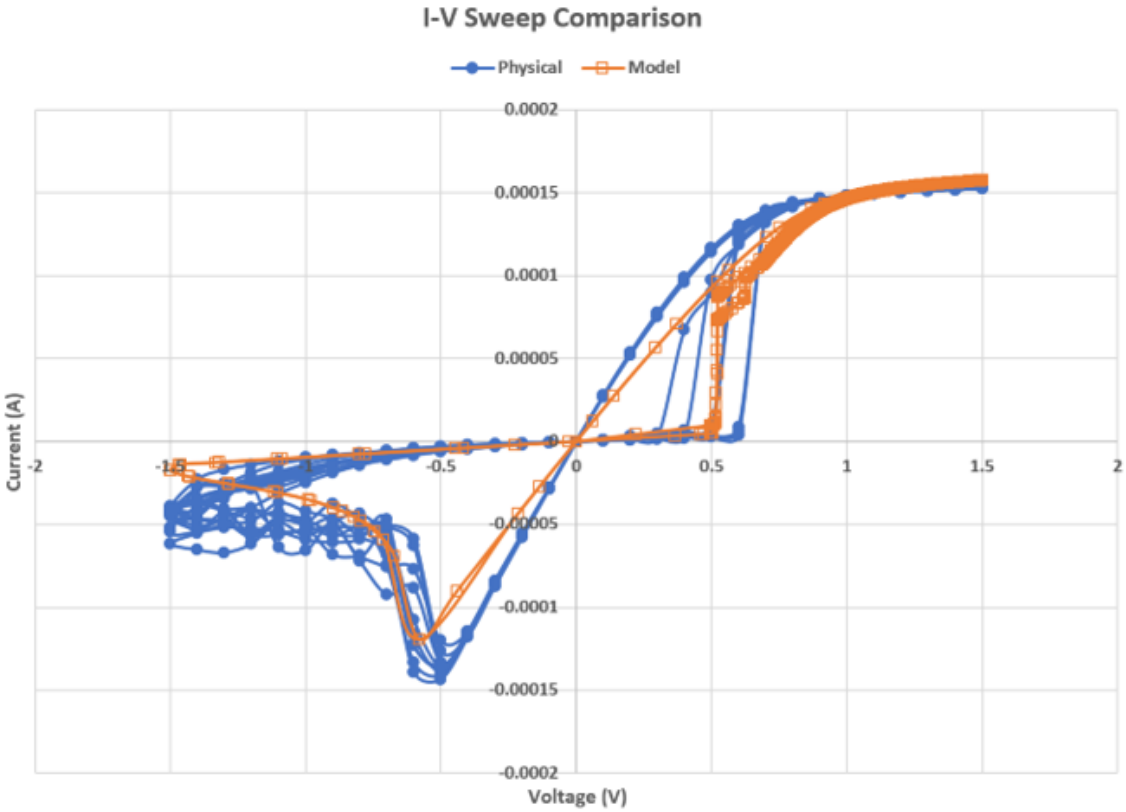

Figure 3.8: Model vs. Physical tested device I-V sweep results.

Table 3.1: Fitting parameters for memristor model.

	Slope	Offset
Vtnl	-0.0001	1
Itpl	8e-11	-2.19e-5

Chapter 4

Memristive Synapse Circuit

4.1 Analog Synapse with Memristor for Weight Storage

In a neuromorphic system the synapse is the connection between neurons. The neurons communicate information via the synapse by accumulating charge. The synaptic strength, referred to as the weight of a synapse, is the amount the synapse can charge or discharge its subsequent neuron. The network of neurons and synapses instantiate synaptic weights for all synapses used to solve a given problem. The neuromorphic system can implement weight changes beyond initial values chosen to improve the effectiveness of the neuromorphic system. As an electrical component, the synapse requires two terminals to connect the two neurons, the pre-neuron and post-neuron. Similar to a memristor, the synaptic weight depends on previous usage. The synapse like a memristor is a two terminal device that has memory. The memory functions provided by the synapse require the change and storage of the synaptic weight. Memristors used for the synapse can mimic the plasticity of the synapse by retaining information an past usage. Non-volatile memristors can store long term weight changes and follow learning rules for long term change. Volatile memristors can exhibit short term weight changes. To mimic biological learning rules, the weight change of the synapse needs complex resistance change based on the activity it receives. Building a synaptic circuit using solely memristors requires the memristors

to possess the equivalent resistance change criteria to biological synapses. For fabricated memristors, achieving similar weight change relationships to neuron activity requires circuitry surrounding the synapse. The circuitry provides adequate voltages to integrate the memristor. The synapse control circuitry is required to implement the analog memory. The control circuitry activates the synapse to read out the weight, programs the memristor to store a weight value, and implements the intended synaptic plasticity. The read out of a synapse is a current into the neuron. The synapse is designed as an analog memory cell using the memristor as the value storage. Implementing the synapse ideally uses low power and area while reliably outputting a stored value. The synapse circuit and all additional control circuitry presented here is intended to use the on chip fabricated hafnium oxide memristor efficiently.

4.2 Synapse Circuit

The block diagram in Figure 4.1 shows the components used in this work to implement a synapse. The goal of the synapse is to hold an analog memory that controls the flow of charge in the connected neuron. The current control circuitry used for programming is reused for read out. The memory is stored on the memristor device in the memristor cell circuit. The output current is generated with the synapse output current control circuit. The synapse output is a current intended to charge an integrate-and-fire neuron. The neuron takes the different synaptic currents due to different resistance values of the memristor and charges proportionally. The currents of all synapses connected to a neuron are summed together by directly connecting at the output node.

4.2.1 Memristor Cell

The circuit seen in Fig. 4.2 for the on chip hafnium oxide memristor shows the necessary transistors required for the four different modes of operation, which are *Forming*, *Set*, *Read*, and *Reset*. The size and expected gate voltages of each transistor is seen in

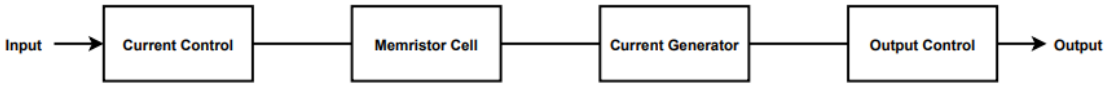

Figure 4.1: Block diagram of Synapse circuit.

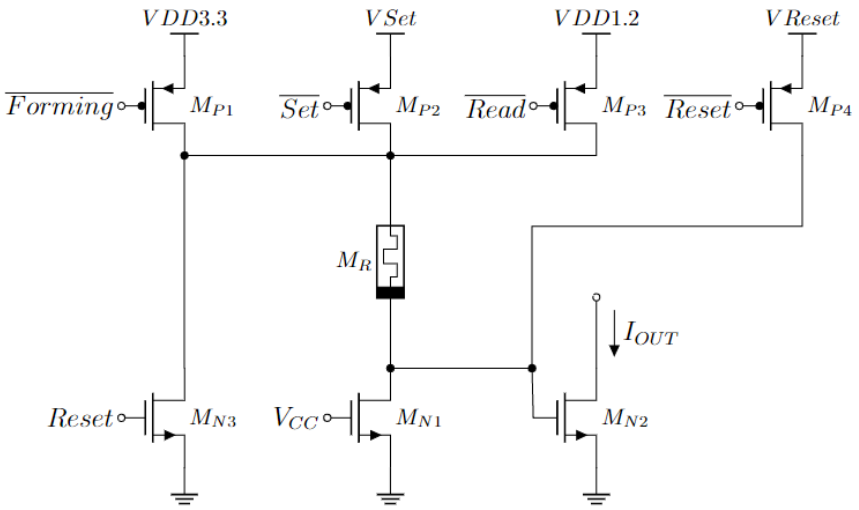

Figure 4.2: Circuit depicting the memristor and transistors directly connected to it.

Table 4.1. The first action is a one time operation to create a metallic filament by applying a high voltage across the unformed device called forming. Due to the need for a high voltage, all transistors that are directly connected to the hafnium oxide memristor will need to be large transistors able to operate in a high voltage region. These large transistors control the memristor through the various operations. For a single stand alone memristor there are seven high voltage transistors. They correspond to four p-type transistors and three n-type transistors. Each operation has a corresponding p-type transistor. The reset and read operations have their own unique n-type transistors. The last n-type transistor is shared for the forming, set, and read operation. For an array of these devices, the transistors for reading and set, M_{N1} M_{N2} M_{P1} are unique to each memristor, while the other transistors can be shared. The p-type transistors for set, reset and forming are shared across a column in a two by two matrix. The n-type transistor for reset is shared for every row.

The transistor M_{N1} has three modes of operation. During forming the gate voltage at M_{N1} is 1 V which corresponds to a 100 µA current in saturation. Due to the unformed device having a high resistance, applying this voltage on the gate of M_{N1} reduces the voltage below the memristor to near ground. In the forming process the corresponding forming transistor M_{P1} is digitally turned on by bringing the gate voltage from 3.3 V to 0 V. This causes the upper node of the memristor to be pulled up to 3.3 V. Once the filament forms, the voltage drop across the device reduces, and M_{N1} goes into saturation. The current of M_{N1} in saturation limits the current through the forming path.

After forming, the device is in a low resistance state. The synapse is designed to use the low resistance state of the device. In order to program the device to a new low resistance state, the device is first reset to a high resistance. The reset process is digitally controlled by transistors M_{P4} and M_{N3}. With all other transistors turned off, activating these two transistors will bring the top node of the memristor to ground and the bottom node of the memristor above 1.6 V. The voltage drop across the device when resetting should be above the switching threshold. This is adjustable with the voltage $VReset$. The opposite

Table 4.1: High voltage transistor connected to memristor size and voltage.

Device	Size (μm/μm)	Gate Voltages (V)
Mn1	5/0.5	0-1.2
Mn2	1/0.5	$\sim$0.675
Mn3	2/0.5	0-3.3
Mp1	1.5/1	0-3.3
Mp2	8/4	0-3.3
Mp3	0.5/0.5	0-3.3
Mp4	8/4	0-3.3

polarity of high voltage on the device breaks the filament created in the forming process. The time and voltage used for reset in the synapse design is long enough to fully reset the device into a high resistance state. Due to the variability in the high resistance state, it is not used in the read operation of the synapse.

Once the device is reset into its high resistance state. It is reprogrammed to a low resistance state with the set operation. The set operation applies a high voltage to the memristor to activate the switching process. This is adjustable with the voltage $VSet$. Similar to the forming operation the set operation uses M_{N1} to current limit the device. The current produced when M_{N1} is in saturation results in different resistances for the memristor. These low resistance states are current programmed via an adjustable gate voltage on M_{N1}. The synapse is designed to have a gate voltage range of 0.9 V to 1.1 V to allow the use of the low voltage transistors. These gate voltages equate to saturation currents of 70 µA to 200 µA. The p-type set transistor M_{P2} provides the high voltage necessary to recreate the filament which can be up to 2.5 V. In this method, the resistance of the device is reliably programmed via current control in the range of 3 kΩ to 13 kΩ.

The read operation uses M_{N2} to create an output current based on the voltage at the top node of the memristor. The memristor in its low resistance state is supplied a current through M_{N1}. This current is as low as possible while still able to generate a reliable output. The values chosen are 600 mV at the gate of M_{N1} resulting in 1 µA current. The p-type transistor M_{P3} is chosen for its resistance when $\overline{READ}$ is 0 V. The current from M_{N1} through the resistances of the memristor and M_{P3} sets the voltage at the gate of M_{N2}. The expected voltage is 600 mV to keep M_{N2} turned on in saturation. The expected change in voltage at the gate of M_{N2} is equal to the current set by M_{N1} and the resistance range of the memristor. For a 1 µA current and 3 kΩ to 13 kΩ range the voltage change is 10 mV. The output current ranges from 1.11 µA to 0.97 µA. This read method differs from the previous works [57] which relied on voltage control across the device. The power equation $P = IV$ or written as $P = I^2R$ gives the average power usage of the memristor in this design. Since the current is constant 1 µA the power is 5 nW for a resistance of 5 kΩ.

For the voltage controlled method, the power consumption is $P = V^2/R$ and to achieve a similar equivalent power requires a voltage control of 1 µV. While potentially possible to implement, the voltage control requires a difficult to produce reference relationship to achieve similar power at low resistances. Implementing the current read out guarantees consistent low power even when the resistance is a low value. Implementing the current limitation to reduce power during read aligns with the transistors used to control current during programming.

4.2.2 Synapse Output

The synapse outputs a current that is accumulated by a post-neuron when activated by a pre-neuron. This is handled by the current generator and the output control circuit in Figure 4.1. The current generator for the memristive device is intended for analog usage with an integrate and fire neuron. The output of the synapse is a current with a range of −20 nA to 20 nA in normal operating conditions. The output current from the synapse is generated by a current mirror circuit shown in figure 4.3 and expects an integrator holding the output node to a specific virtual DC voltage, 600 mV. The current into the current generator is around 1 µA and is mirrored one-to-one. The synapse uses 3 µA to generate the output, 1 µA through the memristor, 1 µA through the high voltage transistor output of the memristor, and 1 µA through the low voltage transistors.

The circuit converts the 3.3 V read transistor M_{N3} in Figure 4.2 to 1.2 V transistors. The usable current range for the neuron is minimized to reduce area in the neuron, thus an expected output of only −20 nA to 20 nA. Outputs based on the memristor's resistance value are transmitted as current to the connected neuron. This circuit takes a reference, $NCMref$, that helps define the midpoint, or zero output current point, which allows output currents to be both positive and negative. The current output uses the virtual ground set by the neuron's op amp integrating input to set the voltage between M_{N3} and M_{P3} to mid-rail. The midpoint operation is defined by the memristor characteristics. For a device with a low resistance range of 2 kΩ to 16 kΩ a midpoint at 7 kΩ is chosen. At a

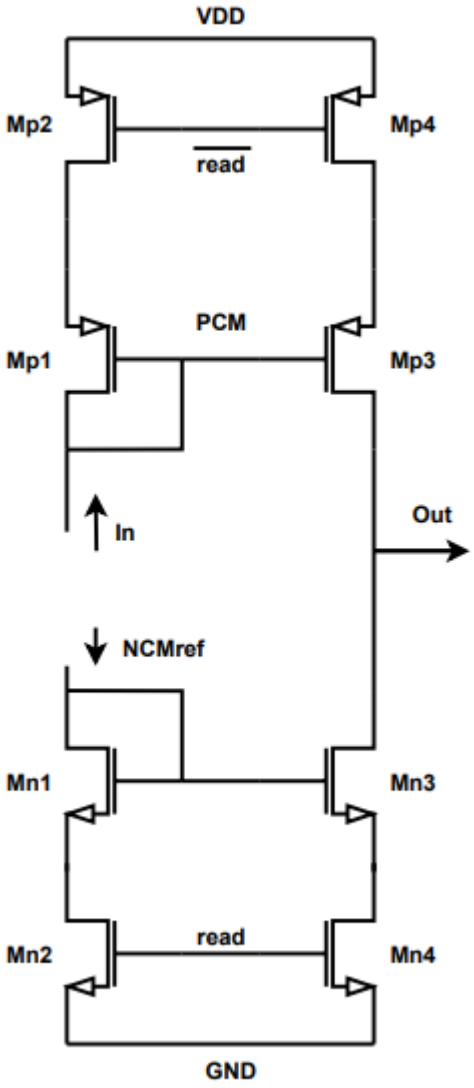

Figure 4.3: Synapse output buffer circuit.

device resistance of 7 kΩ, the output voltage for the current generator circuit is designed to produce no output. This results in an output current of 0 nA into the neuron. The range of outputs is adjusted by this circuit to produce 20 nA when the resistance is minimum, 2 kΩ, and −20 nA when the resistance is maximum, 16 kΩ. A lower resistance value increases the voltage at the gate of M_{N3} in Figure 4.2 which in turn increases the current through M_{N3}. The increased current lowers the voltage at the p-type gates in the current generator and increases the output current. The n-type transistor at the output has a constant gate voltage from the reference, and since the p-type transistor is supplying more current, the output current is positive. For higher resistance the opposite happens and the output p-type transistor has less output current. All transistors stay in the same operating region so the output range is approximately linear for all resistance values of the memristor.

The output of the current generator circuit passes through a transmission gate in the output control circuit before going to the neuron circuit shown in Figure 4.4. The transmission gate circuit allows a startup time for the current generator circuit. Due to the low current used in the read operation and the large high voltage transistor sizes, the capacitance of the transistors gates need time to charge. When the read process is initiated, the voltage levels for all the transistors must reach their steady state to have a valid output. If the output is active before the circuit reaches its DC state, the current into the neuron will be unintended. To overcome the start up time, the transmission gate blocks the output for the first half of the clock cycle the read operation is activated. This limits the speed at which a read needs to be reliable to half a clock cycle. The expected maximum clock period for this circuit from simulation is 400 ns. The output is only active for 200 ns. The output current from the read out buffer circuit only reaches the neuron when the read operation is active and the clock is low. At the maximum clock frequency the synapse uses under 5 pJ per spike. This is in line with other spiking synapse designs [2, 73, 25].

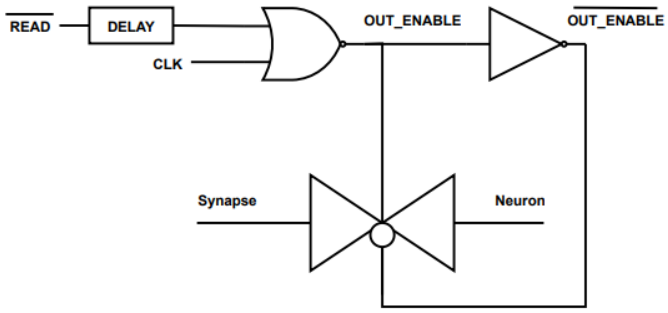

Figure 4.4: Block diagram of the output control circuit.

4.2.3 Current Control

The focal point for this memristive synapse design is the current control functionality. The synapse relies on current control for both programming the memristor and reading the memristor's resistance value. The current control circuit applies a voltage to the gate of M_{N1} in Figure 4.2. This gate voltage imposes a limit on the current through the device when the transistor is in saturation. The three possible voltage ranges required are a high voltage for forming or set that allows a high current in saturation, a low voltage for read allowing a low power and low probability of affecting the resistance stored, and no voltage when not in use. The digital control signals in the current control circuit are from the forming, set, and read processes.

Alongside the digital control signals, analog control signals are used to specify specific values for form, set, and read generated by the circuit in Figure 4.5. The gate voltage during read is 600 mV which turns on the transistor and limits the saturation current to 1 µA. The low current and low voltage used during the read process help alleviate the possibility of unintentionally adjusting the resistance of the device. When the device resistance is initial formed or set into a new value, the gate voltage is much higher. The gate voltage ranges from 900 mV to 1.1 V creating a higher current through the device. These specific gate voltages come from a read reference and a programming reference. Either the transistors for read are activated with the analog and digital signals, or the transistors for forming or setting the device are activated. If neither operation is in use the gate voltage is pulled down to ground. Chapter 6 shows the waveform operation of this circuit.

4.3 Weight Programming

The hafnium oxide memristor requires a current limit during the set operation. This is accomplished by a transistor in saturation. The current limitation is the maximum allowed current through the transistor in saturation. To program the memristor, the gate

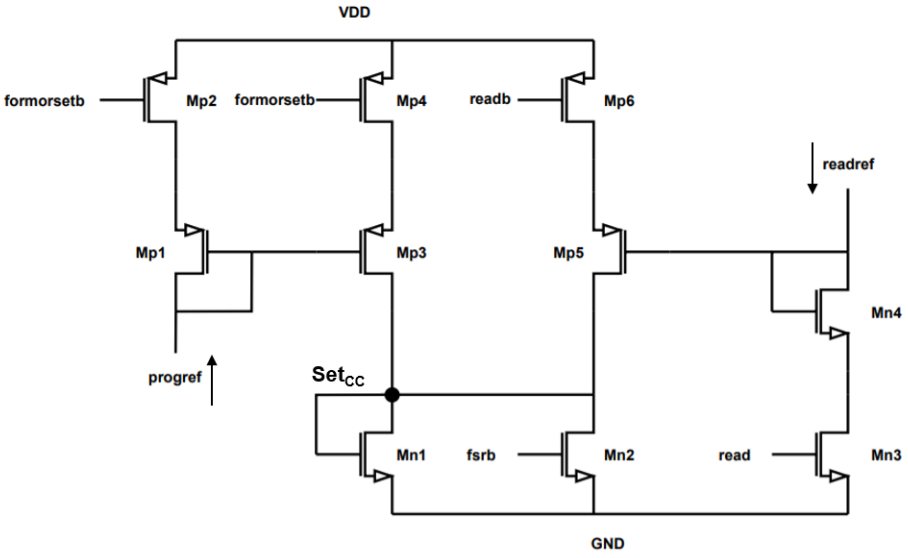

Figure 4.5: Current control circuit for read reference generation adjustment.

voltage used to limit the current through the device is adjusted. Setting the voltage used for programming the device is handled by a 3-bit DAC circuit shown in Figure 4.6. The initial current limit used during the forming process will be determined by the voltages that produces the best results for a specific run of the memristor. Likely a middle point of 100uA at 1V on the gate will be a sufficient for forming. After the device is formed, it is in a low resistance state. This will result in a resistance near 1 kΩ-10 kΩ. The circuit that applies the gate voltage is the current control circuit. The 3-bit DAC circuit uses a scaled digitally controlled current mirror to generate different reference voltages. The transistors $M_(p3)$, $M_(p5)$, and $M_(p7)$ each double in size. This creates a voltage at the node labeled $bitsum$. The output voltage $progref$ is the voltage reference that the current control circuit uses. This 3-bit DAC circuit can be a shared resource between multiple memristors. For multiple connections, the transistors M_{p9} and M_{p10} are instantiated for each device that needs a connection.

4.4 Reference Feedback Circuit

The output current from the current generator circuit takes a reference to set the midpoint of the memristor resistance range. This allows the output current to range between 20 nA when the resistance is minimum and −20 nA when the resistance is maximum. The reference can be externally supplied or internally generated. Since the reference is intended to determine the distance from the middle value of the resistance range the device can achieve in the low resistance state, a reference generation circuit using the memristor is desired. The simple design shown in Figure 4.7 uses a memristor set to a middle resistance value to generate an output reference. The reference used by each connected synapse sets the voltage at the current output transistor which relates the resistance of the memristor in the reference generating circuit to an output of 0 nA in the synapse. Using this scheme allows for an adjustable reference that is based on the devices that require the reference. Since the output reference is based on the same memristor devices and equivalent circuity,

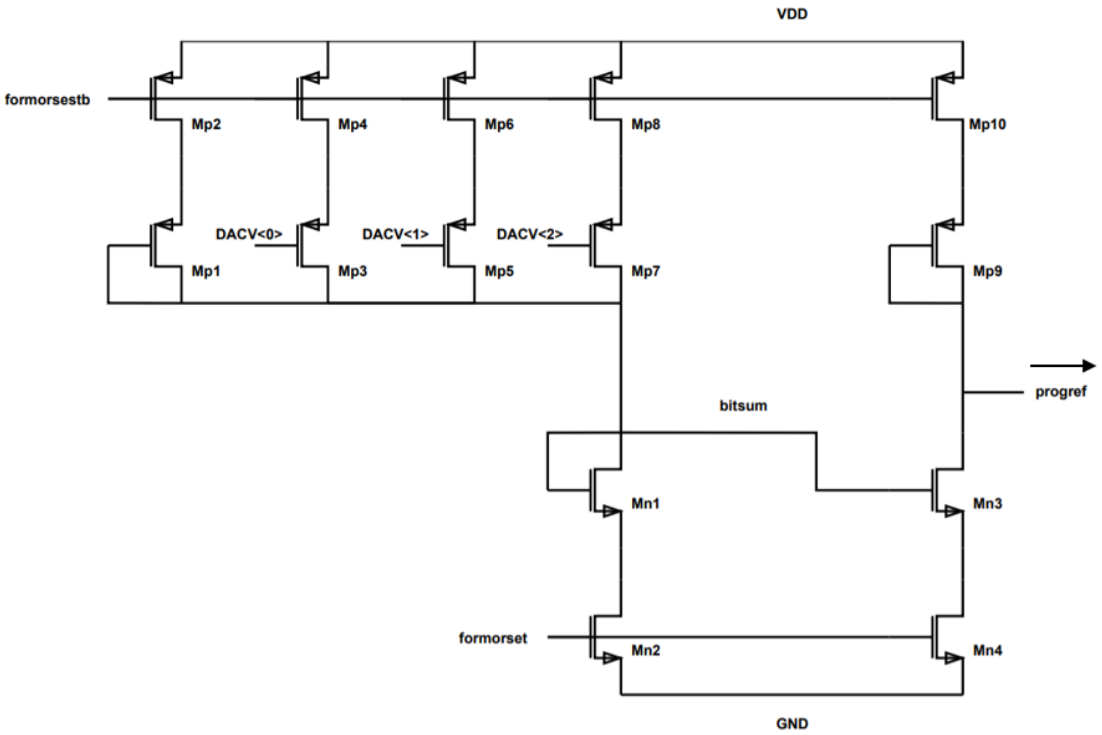

Figure 4.6: Current steering DAC for programming.

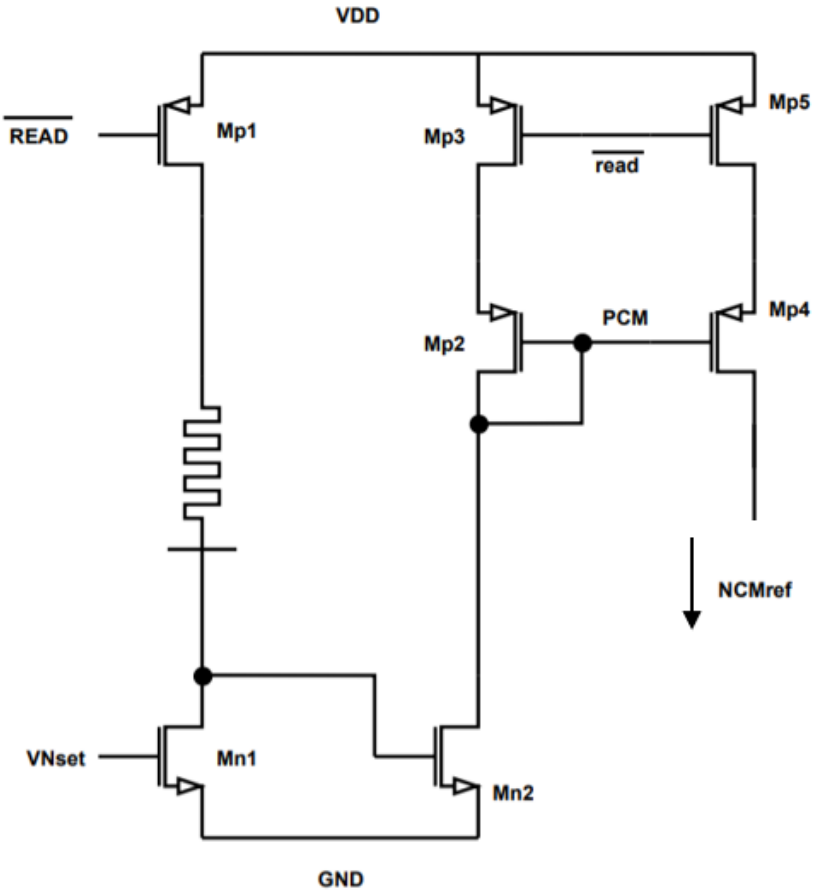

Figure 4.7: Memristor reference generator circuit.

the reference can readily match the synapses. The reference can also be used to adjust the output. If only a positive or negative output current is desired, the memristor in the reference can be set to a maximum or minimum low resistance state. The memristor reference circuit can be shared across multiple synapse circuits by duplicating the current output transistors M_{p4} and M_{p5}.

Building upon the memristor reference, a feedback loop can be instantiated in the circuitry to settle the read into an optimal midpoint for no output current. Figure 4.8 shows the additional circuitry to the memristor reference circuit required for the feedback loop variation. The memristor reference for the feedback loop generates the current expected from a synapse of the same memristor resistance called I_{ref} using the reference voltage used by the synapses NCM. This additional output is fed into a circuit shown in Figure 4.9. This circuit first takes the output current and converts it to a voltage relative to mid-rail. It is then compared against a mid-rail reference. Last it adjust the read reference seen in the current control circuit. Overall this creates a large feedback loop. If the output current I_{ref} is non-zero, the read reference voltage is adjusted. This works to reduce the current to zero. A transistor level view is seen in Figure 4.10. The output for this variation of the reference circuit is not only the reference for the current generator circuit, but also the read reference for the current control circuit. Like the simple memristor variation, the output of one reference generation circuit can be copied and distributed to multiple synapse circuits. Chapter 6 shows the waveform operation of the reference generation circuit.

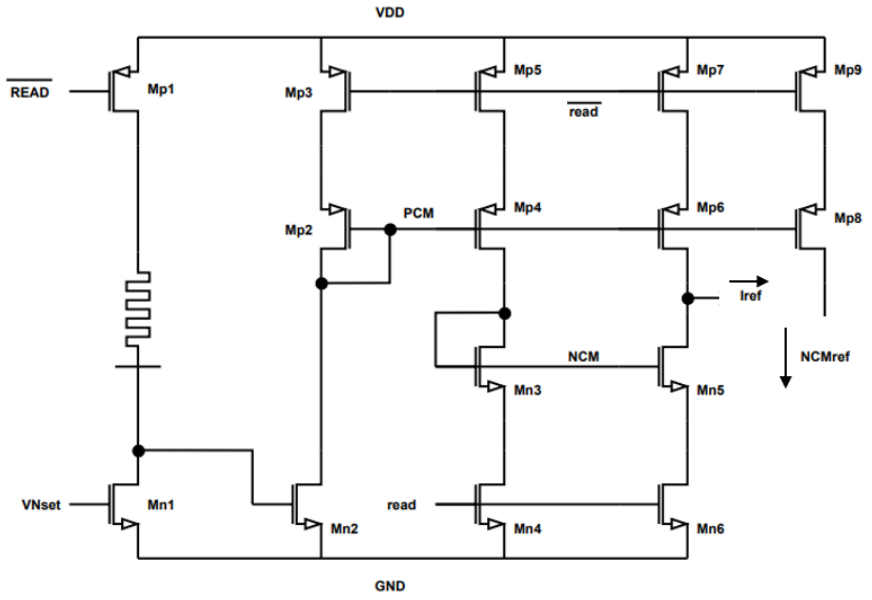

Figure 4.8: Memristor reference generator circuit.

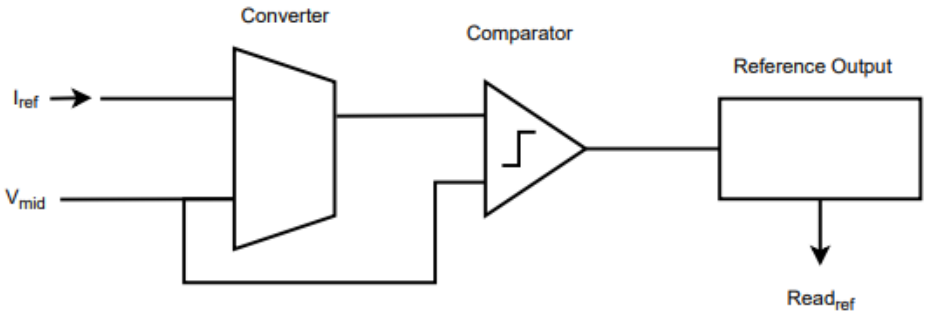

Figure 4.9: Current reference feedback control circuit block diagram.

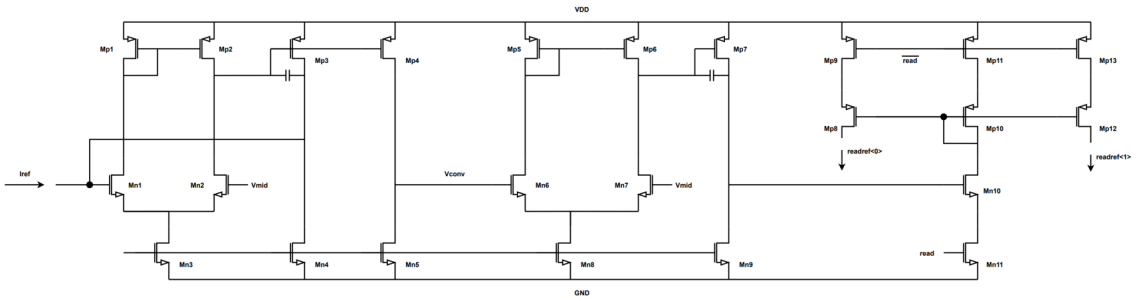

Figure 4.10: Current reference feedback control circuit.

4.5 Online Learning

The online learning for this memristive synapse circuit follows the same general procedure as the programming. The main difficulty with the online learning using only the set operation is that the resistance change during set is unidirectional. To allow for both an increases and a decreases in resistance, the weight update in an online learning system has three events. First the current value of the memristor is read out and temporarily stored, second the device is reset, and third the device is set into a new resistance value.

The example online learning rule implemented is the spike timing dependent plasticity, or STDP, learning rule. STDP uses the relative timing of the activity of the connected neurons to a synapse to adjust its weight. The pre-neuron is the neuron feeding into the synapse and the post-neuron is the neuron receiving information from the synapse. The weight of the synapse is increased if the pre-neuron fires before the post neuron, called a potentiation, and the weight decreases if the post-neuron fires before the pre-neuron, called a depression. The magnitude of weight change is an exponential decay as time between pre-neuron and post-neuron activity increases.

The block diagram for the STDP circuit is shown in Figure 4.11. At the top of the block diagram, the digital control circuitry takes the memristive synapse through the three stages used to update the memristor. The read operation, reset operation, and set operation all occur sequentially after both a pre-neuron and post-neuron activate. Before the flip flops used to control the synapse operations, there is a digital latch that activates on a pre-neuron or post-neuron input. The inputs, Pre and Post come from their respective neurons, and learn is a toggle for the circuit. The two latches for Pre and Post generate PreH and PostH respectively and both are required to begin the memristor update sequence. The held values are stored until the process ends and resets the latches with the signal END.

The degree of the weight update is determined by the blocks in the lower half of Figure 4.11. The internal working of the sample, hold, decay and sum blocks are in Figure 4.12 and 4.13. The circuit determines the weight update by first reading the memristor's

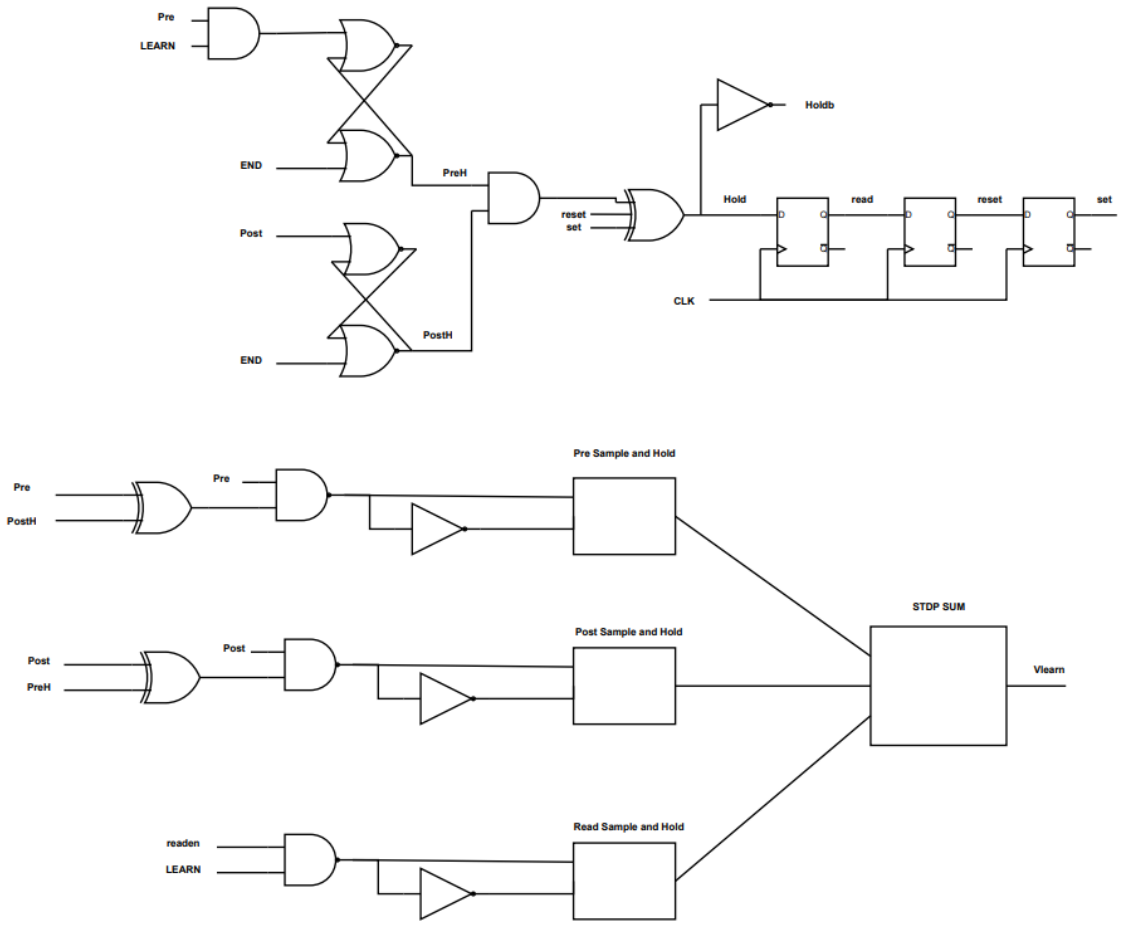

Figure 4.11: Block diagram for STDP circuit

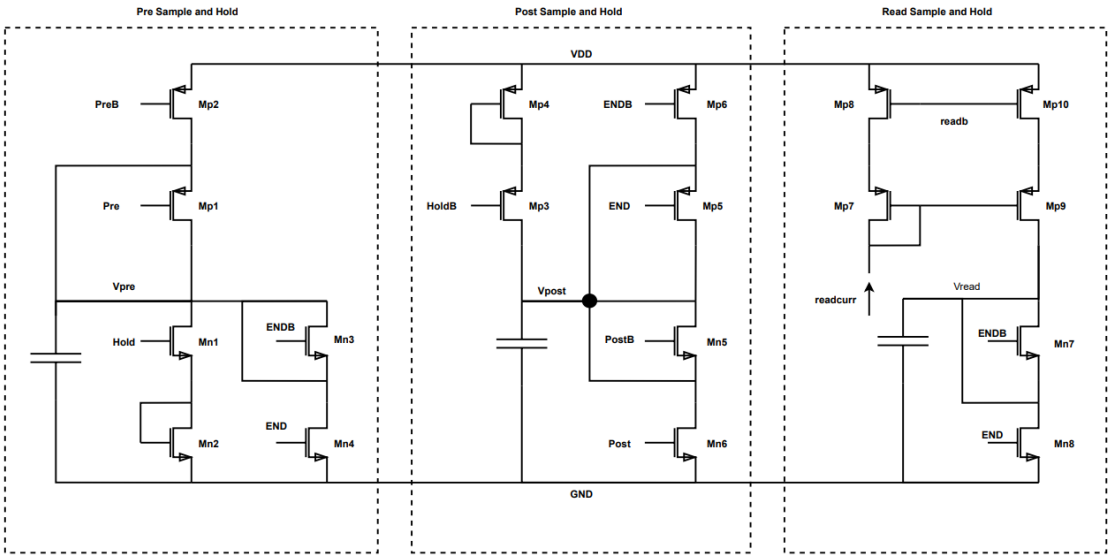

Figure 4.12: Sample and hold circuit.

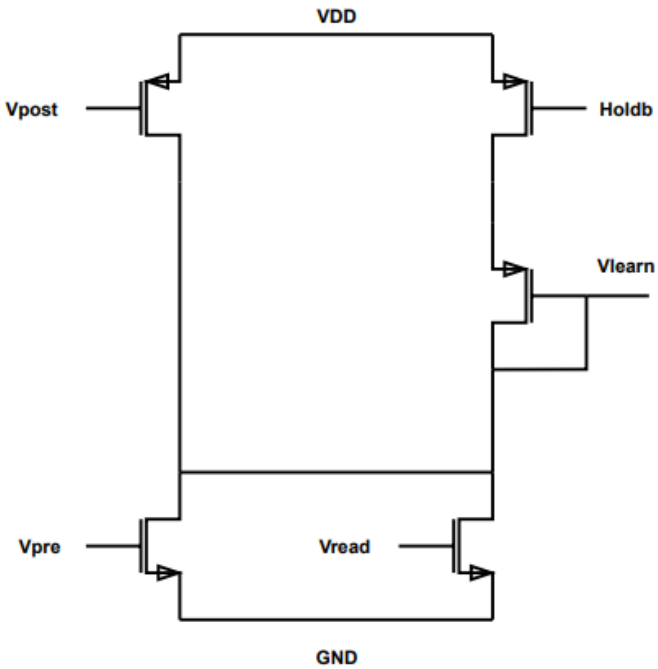

Figure 4.13: STDP summation circuit.

resistance state and storing a voltage equivalent to the memristor's current resistance value in the sample and hold circuit. This is combined with one of the two voltages stored on the two time decay circuits. This circuit consists of a capacitor that is charged and discharged. Given the post-neuron has not activated, and a pre-neuron activates, the time decay circuit related to the pre-neuron activates. This fully charges a capacitor and then allows it to discharge. When the post-neuron activates, the capacitor is block off from discharging. The final voltage on the capacitor gives a time relationship between the pre-neuron and post-neuron firing. Since the pre-neuron activated first the voltage has decreased from the fully charged capacitor. This voltage is sent to the summation circuit, and along with the voltage generated from the read operation, a new voltage for the current liming circuit is generated. The device is then set using a new gate voltage on the current limiting transistor. If the post-neuron activates before the pre-neuron, the corresponding capacitor is discharged and then allowed to charge up. This is the opposite process of a pre-neuron then post-neuron fire. In the summation circuit, instead of increasing the voltage, the summation decreases the voltage used at the gate of the current limiting transistor. The time decay circuit of whichever neuron activated second is not used in that learning operation. For a potentiation the post-neuron time decay circuit is held high, and for a depression the pre-neuron time decay circuit is held low. This guarantees the memristor is either programmed back into its previous state, or a state that is the correct increase or decrease in resistance. Chapter 6 shows the waveform operation of this circuit.

4.6 Synapse Conclusion

The synapse circuit provides an output current to a neuron circuit. It sets an output current by reading a memristor's resistance level. To accomplish this, the memristor needs to be initialized, or formed, and programmed. The programming uses only the process of creating the conductive filament in the memristor, and directly correlates a resistance

value to a DC input. For the different circuits presented in this section, the area estimates are shown in 4.2. The components DAC, Reference, and Feedback as well as parts of STDP can be distributed among many synapses. Table 4.3 the idle power of a circuit when its not in use, the energy per use of a synapse function, the current for that function, and the settling time of the function. The slowest settling time determines the fastest frequency the circuit can operate. For some operations, the max and minimum current is determined by the memristor's resistance. The reset operation produces a large amount of current for a brief moment before the device's resistance increases. Table 4.4 shows the expected current and total energy used for different steps in the STDP process. The idle current and maximum current during depression and potentiation are shown. The majority of current used in this circuit is setting up the summation circuit to drive the memristor.

Table 4.2: Area estimates for synapse components.

Circuit	Area (μm^2)
Memristor	0.09
Cell	96
Level shifter	144
Synapse Output	50
Current Control	96
DAC	128
Reference	344
Feedback	700
STDP	1446

Table 4.3: Energy usage for synapse circuits.

Structure	Idle Power (pW)	Energy (pJ)	Current (uA)	Settling Time (ns)
Memristor Read	18	1.56	1.3	120
Set		337	136	
Reset		542	1100 max 68 min	
Synapse	494	30.7	21	400
Program	426	27	22 max 5 min	18
Current Compliance	510	109	80 max 50 min	22

Table 4.4: Energy usage for STDP circuits.

Component	Idle	Current (dep)	Current (pot)
Pre	33pA	30pA	505nA
Post	26pA	140nA	400pA
Syn	16pA	20nA	20nA
Sum	20pA	57uA	63uA
Total Energy		151pJ	195pJ

Chapter 5

Neuron

The neuron is an integral part of neuromorphic computing. The purpose of the neuron circuit is to accumulate the current inputs from synapses and output a voltage when the accumulation crosses a threshold. These two functions are captured by an integrate and fire neuron. This is a simple mathematical model for a neuron that integrates the input current from synapses, stores the input as a voltage, compares the voltage against the threshold, fires an output spike upon crossing the threshold, and finally resets the accumulated voltage after activating. Other neuron models accomplish the same basic functions. There are adjustable factors in neurons, including dynamic thresholding and adjustable leakage. Leakage is loss of voltage in the accumulated voltage over time which yields the leaky integrate and fire neuron. The circuits that create the neuron need to implement the basic functions of a neuron.

5.1 Integrate and Fire Neuron circuit

The current output from the synapse is the input into the neuron. Creating a consistent accumulation of voltage is achieved using an integrator. The integrate and fire neuron uses an op-amp integrator to store the input current as a voltage. The op-amp helps create a consistent input current to accumulated voltage change. After the voltage storage, the

voltage is compared against a threshold. The integrate and fire circuit uses an op-amp as a comparator to provide a controllable threshold voltage. The output is a digital signal that is activated when the voltage accumulated crosses the voltage threshold. The capacitor used to accumulate the voltage is reset when the neuron fires an output spike. The process of resetting the neuron is called the refractory period.

The integrate and fire neuron shown in Figure 5.1 is my board level design used with the biomimetic synapse [71]. The input current from the synapse is integrated on the integrator consisting of Op Amp U1, an LT1793, and the capacitor C_{mem}. The LT1793 was chosen for its high input impedance. U2 is also a LT1793 since the part was already available for this circuit. The threshold voltage, V_{thr}, gives the voltage necessary for V_{mem} to cross before the output V_{spike} goes to a high voltage. When V_{spike} goes to a high voltage the p-type transistor at the input is turned off, blocking input current. The n-type transistor in the feedback of the integrator is turned on allowing current to flow and discharge the capacitor C_{mem}. The reset capacitor and resistor, C_{reset} and R_{reset}, help define the output spike width and reset voltage at V_{mem}. When V_{spike} initially goes high, the change in voltage is passed through C_{reset} onto V_{mem} causing V_{mem} to go farther below the threshold voltage. Since the path that R_{reset} is on is open, the voltage V_{mem} is discharge back across the threshold in the opposite direction. This causes the voltage V_{spike} to return to a low voltage. This voltage change is again passed through C_{reset} causing the voltage V_{mem} to return to a voltage above the threshold. This circuit does not have surrounding digital control logic and builds in the refractory period functionality with analog components. The on chip integrate and fire used for the MrDANNA and RAVENS wafer takes advantage of the clocked system to implement the refractory period timing with clocked flip flops.

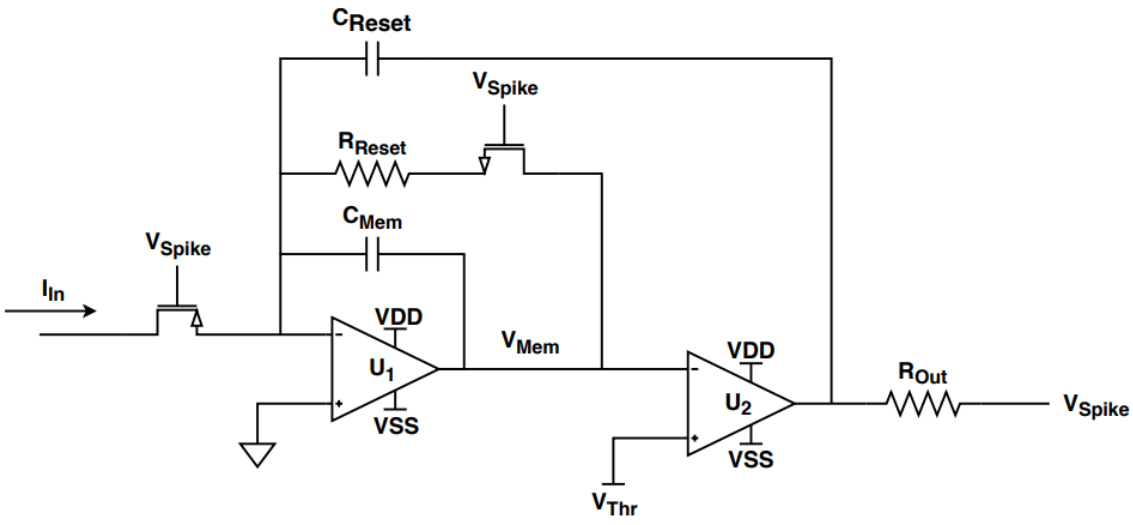

Figure 5.1: Integrate and Fire neuron [71].

5.2 Axon Hillock Neuron

The Axon Hillock circuit was designed by Carver Meade [40]. I implemented a version of this circuit shown in Figure 5.2 that takes advantage of the digital signals used on our on chip wafer [70]. The axon hillock in biology is the connection between the neuron cell body, or soma, and the axon. The axon hillock is the point where the accumulated charge in the body generates the output across the axon. This circuit recreates this feature of the neuron. Unlike the integrate and fire neuron, a floating capacitor is used. This creates difficulties in keeping the current input unaffected by the accumulated voltage. An ideal current source as the input into neuron would not cause any difference. The difference in accumulation does not pose problems in functionality, but a difference in modeling for high level simulation for application testing. An inverter circuit can be used like in the axon hillock neuron if the threshold can be constant and device defined.

The axon hillock neuron circuit shown in Figure 5.2 has similar functionality to the integrate and fire neuron. The voltage V_{mem} and capacitor C_{mem} share equivalent roles. The input current is integrated and stored on C_{mem} as V_{mem} and then compared against V_{thr} to activate the output. The output is generated with a dynamic inverter circuit and clocked with a D flip flop. The refractory period is clock cycle defined using p-type transistors at the input to block input currents, and an n-type to reset the capacitor.

5.3 Neuron Model

In addition to circuit solutions to the neuron, a Verilog A model can be used in place of the actual circuit design. The model I implemented has two basic purposes. It accumulates inputs and outputs voltages based on the accumulated input. The code snippet below shows the accumulation of an internal voltage as a function of input current into the model. The model has a minimum required current to begin accumulating voltage. The current into the model must be higher than the minimum for a short period of time to begin accumulating the current as an internal voltage. Once the accumulated voltage

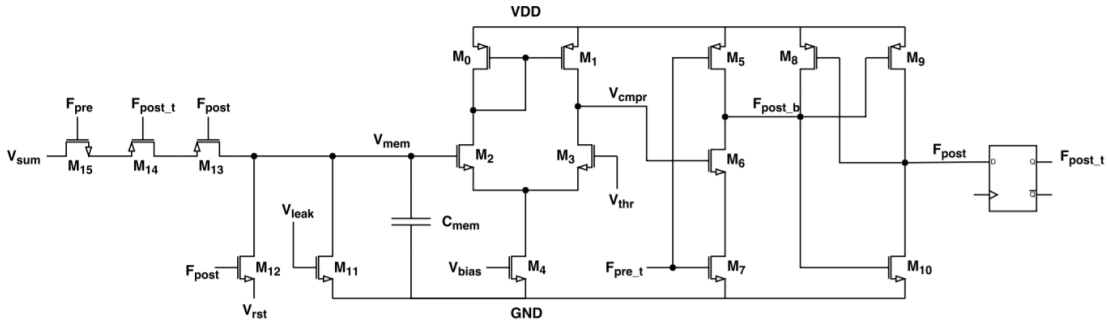

Figure 5.2: Synchronous axon hillock neuron [70].

crosses the specified threshold voltage, the output of the neuron is activated. While the neuron output is activated, the internal accumulated voltage is reset and inputs are blocked. This model is used in testing the output of the synapse without instantiating the circuit implementation of the neuron to speed up simulation time when debugging synapse neuron interaction. This model is used in Chapter 6 to confirm varying neuron charge accumulation determined by memristor resistance.

```
// Neuron accumulation upon input above
// minimum threshold for a set amount of time
@(cross ( $abstime -( nowtime + Tn), +1)) begin
        if (ttest == 1) begin
            startacc = 1;
        end
end
if ( startacc == 1) begin
        Vmemnew = Vmem - I(Vin)*1e5;
end
// Neuron output upon crossing accumulated voltage threshold
@ (cross(Vmem - Vthr, -1)) begin
        Vset = Vs;
        Vmemnew = Vr;
        tsamp = $abstime;
end
@ (cross($abstime - tsamp - Ts, +1)) begin
        Vset = 0;
end
Vmem = Vmemnew;
V(Vout) <+ transition(Vset);
V(Vmread) <+ (Vmem);
V(Vin) <+ (Vmid);
```

Chapter 6

Device Testing and Circuit Simulation

6.1 Hybrid CMOS memristor wafer

From the first hybrid CMOS memristor wafer, I tested multiple circuits including my axon hillock neuron circuit, the memristor forming circuit and the 1T1R memristor test structure. I found some positive results from my neuron circuit. I learned a great deal about probe station circuit testing. Unfortunately the intricacies of the testing were not well understood in the design. I was left with some understanding that the circuit worked, but without the ability to probe internal nodes I could not determine the exact response the synapses on this chip had on my neuron circuit. I took this information I learned about on chip circuit testing in the probe station and applied it to the next wafer. I implemented multiple structures that should build results to verify piece by piece the circuits I designed. Aside from testing my neuron circuit, I tested the memristor synapse forming circuit on the wafer. I found the circuit did not yield the results I expected. I had trouble debugging the forming circuit, and believe the difficulties I faced are a result of certain uncontrollable circuit parameters. The memristor forming circuit includes a current mirror intended to limit the current through the memristor during forming. This current limiting circuit is not adjustable, and possibly designed to a current that is too low to produce reliable forming in the memristor. Despite this, I tested the memristor test structure, the 1T1R,

and found profound results on the device characteristics including, forming, high and low resistance characteristics.

6.1.1 Hafnium Oxide testing and results

The hafnium oxide memristor testing followed the general steps outlined in Chapter 3. These tests used a 1T1R setup to limit the current directly on chip. The current limiting could be performed off chip, but would require an inline component like a transistor. The time needed to activate the current limiting features of equipment like a sourcemeter is not fast enough to properly limit the current through the device. For these tests I wrote a python code to control a sourcemeter to measure current and voltage while applying voltages. This provides sufficient analysis for DC measurements. I have written code for high speed analysis using the oscilloscope and wave form generator. These tested results come from sourcemeter measurements on the device under DC voltage or current supplies. This is intentional to match the usage in my memristive synapse.

6.1.2 Forming

For the forming process I had varying results. Figure 3.1 shows the current and voltage relationships recorded during the forming test. The gate voltage of the transistor is held at a DC 1.2 V while the voltage on the memristor is swept up to 4 V and down to ground. On the sweep up initially the current is very low due to the high resistance of the unformed memristor. When the device forms at about 2.5V in the test shown, the current jumps up to the current limited by the transistor. I tested forming at different temperatures. Temperature relationships seen in Figure 6.1 for a few test structures for one wafer showed the expected temperature relationship. The average forming voltage required increases as temperature decreases. What is interesting here is the lower forming voltage at room temperature and above. This shows promising results to utilize high temperature to lower the forming voltage to reduce the size of transistors needed near the memristor.

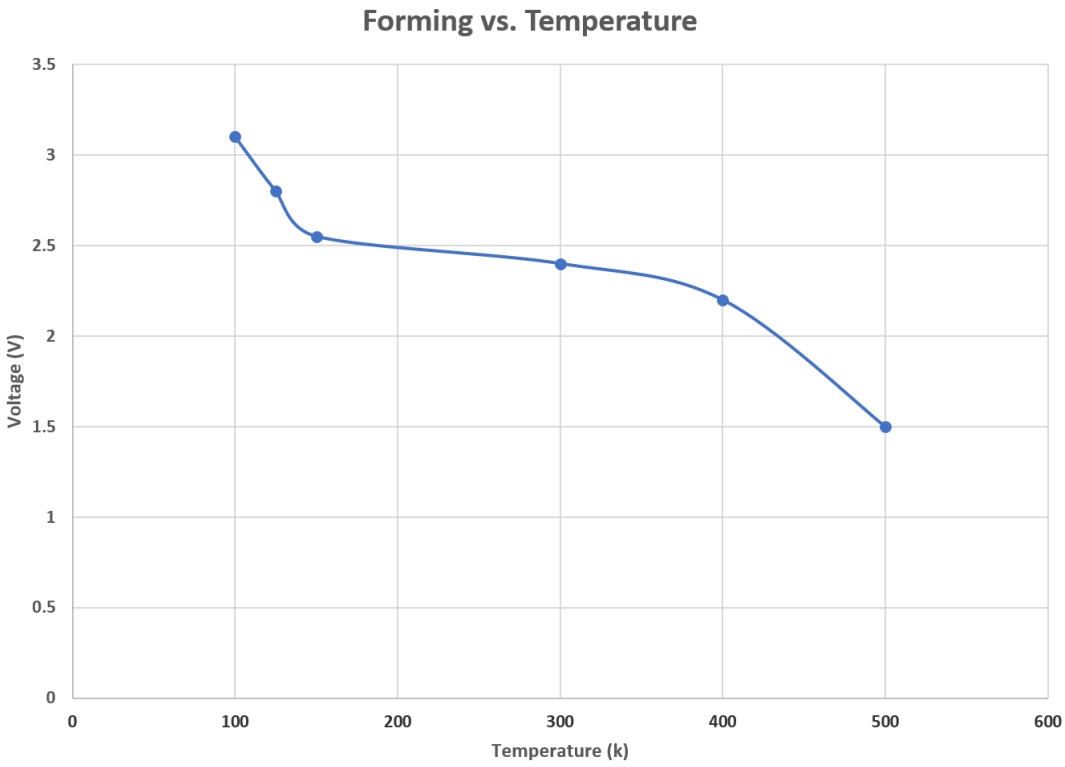

Figure 6.1: Averaged forming voltage at different temperatures.

6.1.3 Switching characteristics

After forming, the next test validates the switching characteristics of the device. DC Voltage sweeps shown in Figure 6.2 are applied to allow the device to set and reset. The voltage for set and reset are generally much lower than the forming voltage and thus the sweep is between $-1.5\,\mathrm{V}$ and $1.5\,\mathrm{V}$. The same jump in current in the positive voltage sweep aligns with the forming process, setting the device in to a low resistance state. For this device, the set voltage is around $0.5\,\mathrm{V}$. The negative voltage sweep shows the reset process. The resistance increases after the reset voltage is crossed, at around $-0.5\,\mathrm{V}$. Unlike the set process, the reset process does not have the pronounced jump in resistance. The high resistance is the flat, horizontal line passing through the origin and the low resistance is the sloped, vertical line. After the set process the current is limited by the transistor, which is why the current does not continue to increase and instead saturates at $150\,\mu\mathrm{A}$.

6.1.4 Cycle Variability

Once the switching characteristics of the device are verified via a voltage sweep, the next test applies singular voltages for set, reset, and read. In this test, the cycle to cycle variability of the device in the low resistance state is shown in Figure 6.3 and the high resistance state is shown in Figure 6.4. The device resistance is measured after every set and reset. The set is performed with $1.5\,\mathrm{V}$ applied to the memristor. The reset is performed with $-1.5\,\mathrm{V}$ applied to the memristor. The resistance is read using a $-200\,\mathrm{mV}$ pulse. The gate voltage remains constant throughout the test. Initial tests had the gate voltage static at $1.2\,\mathrm{V}$. For the device shown, the test shows a large consistent gap between the high and low resistances. the low resistance is around $5\,\mathrm{k}\Omega$ for the low resistance and $40\,\mathrm{k}\Omega$ for the high resistance. The process of gathering these results consisted of applying voltage pulses above and below the positive and negative switching thresholds and read pulses in between. While variation occur in both states, the variation in high resistance is

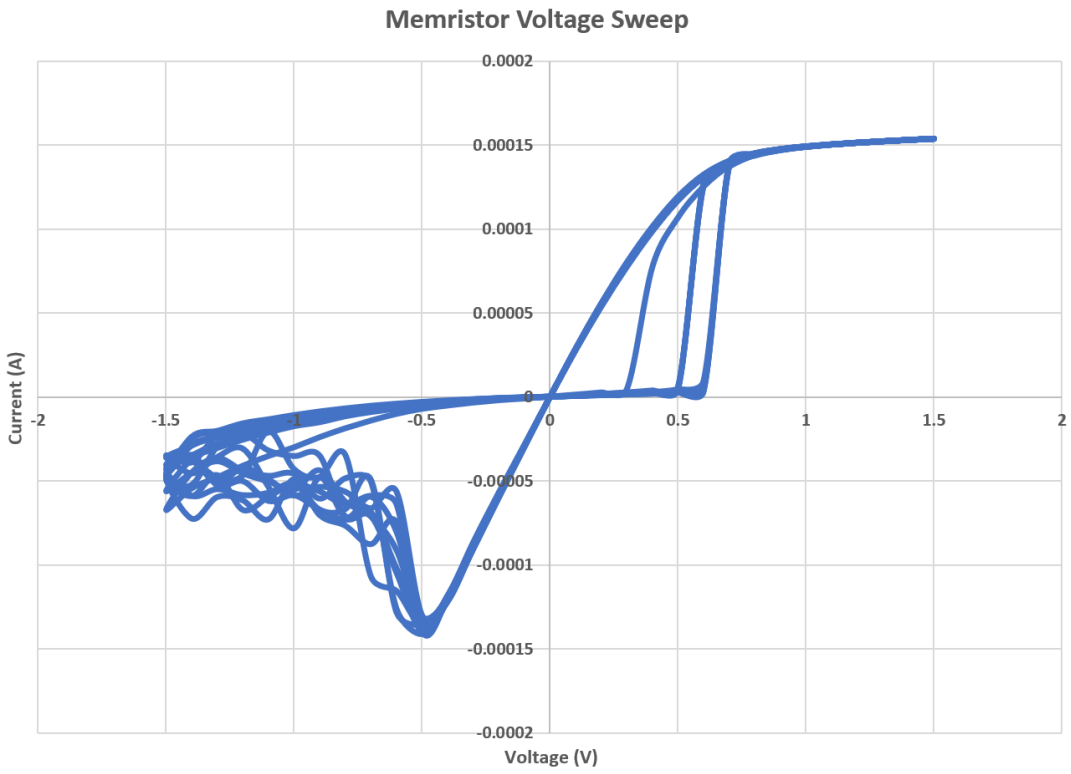

Figure 6.2: I-V sweep for set and reset operations.

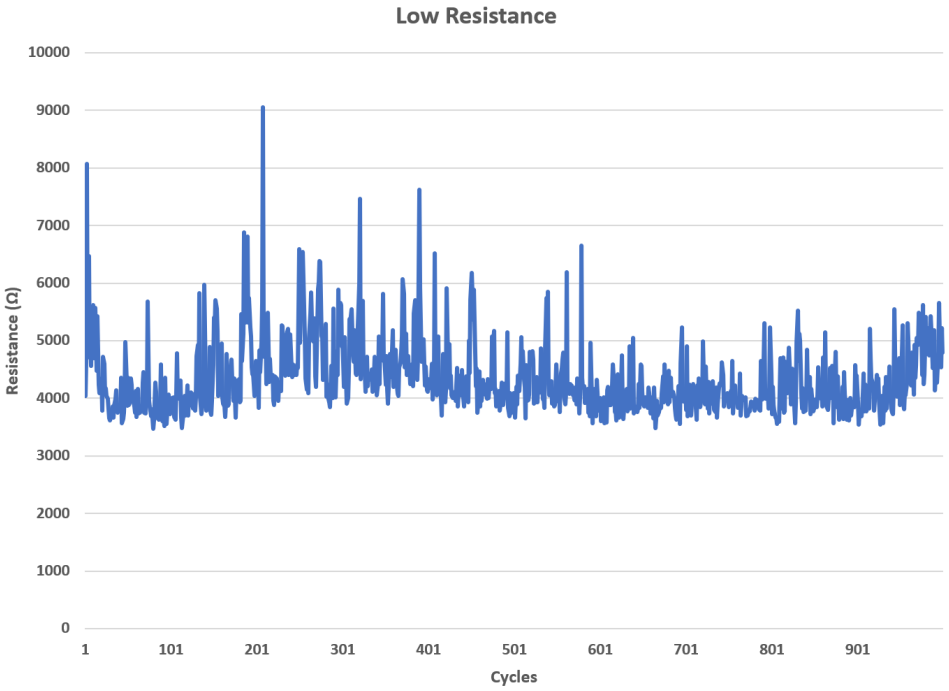

Figure 6.3: Cycles of low resistance states.

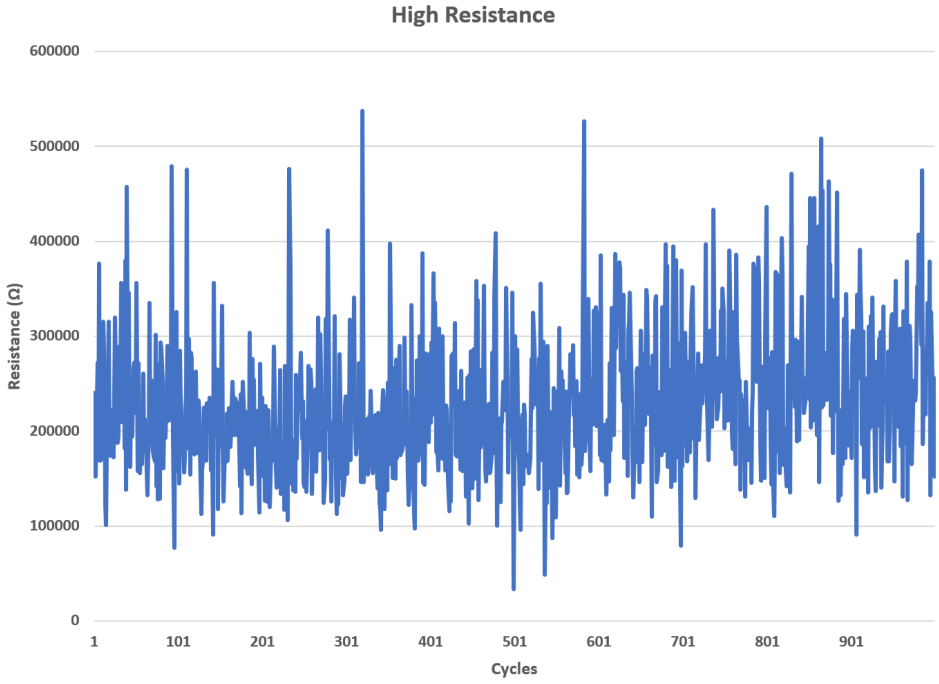

Figure 6.4: Cycles of high resistance states.

one factor that led to my synapse design. From this test the average resistance for the low resistance state

is 4.3 kΩ with a standard deviation of 620. The low resistance state had a maximum resistance of 9.1 kΩ and minimum resistance of 3.4 kΩ. The high resistance state had an average of 23 kΩ with a standard deviation of 73000. The high resistance state had a maximum resistance of 540 kΩ and minimum resistance of 34 kΩ.

The synapse design uses the low resistance state because consistent resistance levels can be achieved with identical voltage pulses from controlling the current limitation. The reset process can utilize both pulse width and magnitude shaping to achieve analog resistance states. For both the set and reset process, the device relies on different mechanics for adjusting the resistance level. Designing a circuit to implement consistent resistance changes using one method will reduce control circuitry needed. For this reason, I chose the low resistance state because controlling the current limitation to achieve specific resistance values within a certain range uses the same voltage pulses. Ignoring the high resistance state as an analog memory device reduces the range of resistances achievable. However, the inconsistency in switching given the voltage pulses, would reduce the reliable range of analog memory. As the resistance increases, the likelihood of a large resistance jump that takes the device to its highest resistance state increases. Consistent changes in resistance from the same voltage pulse only applies while the device is near the low resistance state. The usable range for controllable analog memory states using both the high and low resistance programming methods has a comparable resistance range. With a comparable range, the low resistance values and limited range of values is less of a detriment and the simplicity of current control with a singular consistent voltage pulse for resistance change should prove advantageous over the high resistance state which need pulse control.

A continuation of the cycle to cycle variability test using a static gate voltage is testing with incremental change in gate voltage. Figure 6.5 shows the low resistance state from a device when the gate voltage on the transistor increases from 1 V to 1.5 V in 50 mV increments. There is a clear relationship between the change in gate voltage and the

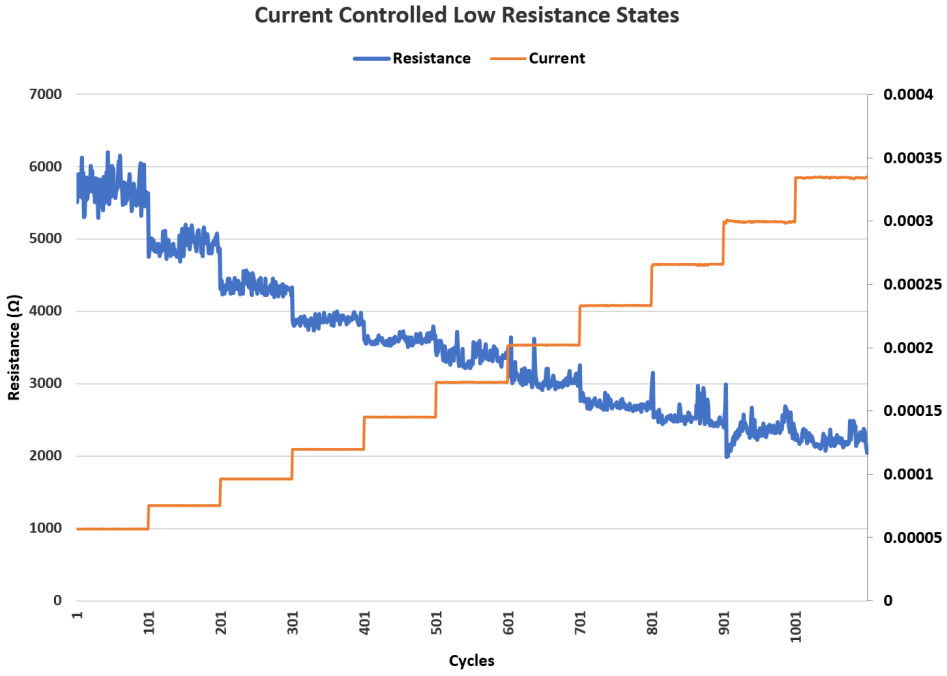

Figure 6.5: Increasing current limitation to achieve multiple low resistance states.

resistance value measured. As the gate voltage increases, the saturation current the transistor allows increases. The currents during the set process seen in orange increase from 50 µA to 350 µA. This allows the memristor to set into a different low resistance state. For this device the average resistance went from 5.7 kΩ to 2.2 kΩ from the minimum to maximum current with a noticeably different average for every saturation current. Figure 6.6 shows another change in current limitation but instead of sweeping from 1 V to 1.5 V in 50 mV increments, this test sweeps from 1.1 V to 1.5 V in 1 mV increments. For a similar test on a different device, the average resistance value vary when looking at the gate voltage, but the resistances achieved are closer when looking at the actual saturation current the current limiting transistor imposed. For example, at the saturation current of 96 µA the device from Figure 6.5 had an average resistance of 4.34 kΩ while at a similar current saturation, the device from Figure 6.6 had an average resistance of 4.51 kΩ. The currents ranged from less than 60 µA to over 300 µA on the device from Figure 6.5 while the device in Figure 6.6 range from 83 µA to over 129 µA. While the overall range of currents differs, in their overlap similar resistances are achieved. This helps validate the notion of a current limit based programming. However, this also indicates the need for highly uniform transistors for the current limitation in the design.

Another key aspect of the synapse design is the current read system. From the 1T1R test structure, I tested the feasibility of the current read. While the method is not exactly the same, I was able to read different low resistance states using a current read signal. For all tests on the device so far, a read operation was performed with a voltage signal. The test in Figure 6.7 gathered the resistances achieved using a current from a sourcemeter and measuring the resulting voltage. For a read operation the gate voltage of the transistor was reduced to 600 mV resulting in approximately 1 µA of saturation current. Then the memristor is supplied with 1 µA of current and the resulting voltage is read to determine the resistance state. Similar applying voltage and reading the current, this system for applying current and reading voltage shows varying low resistance states. In this test the low resistance state matches the changes seen in Figure 6.5.

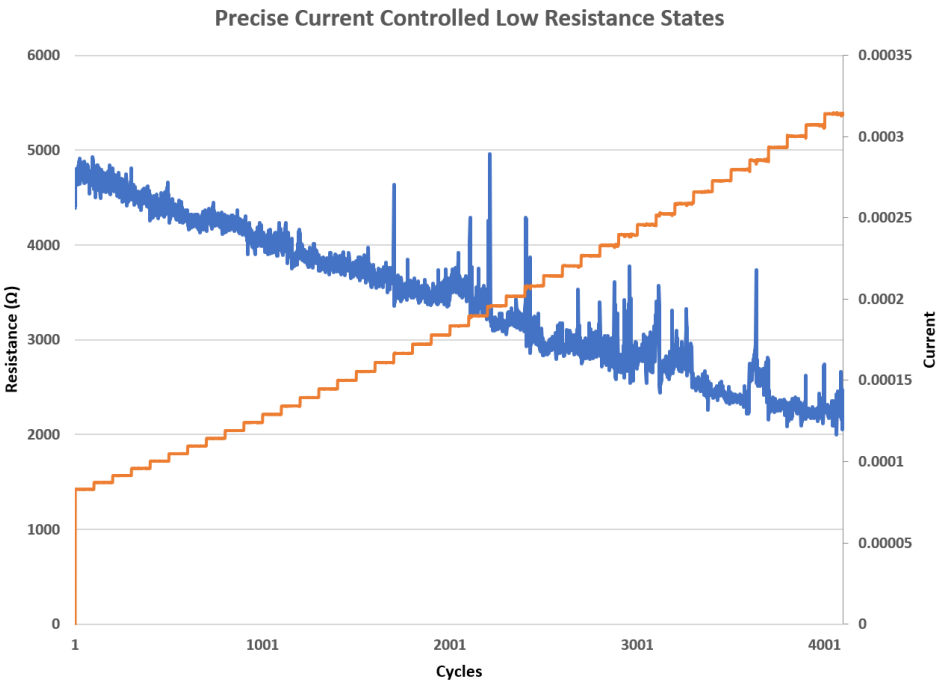

Figure 6.6: Higher precision increasing current limitation to achieve almost continuous low resistance states.

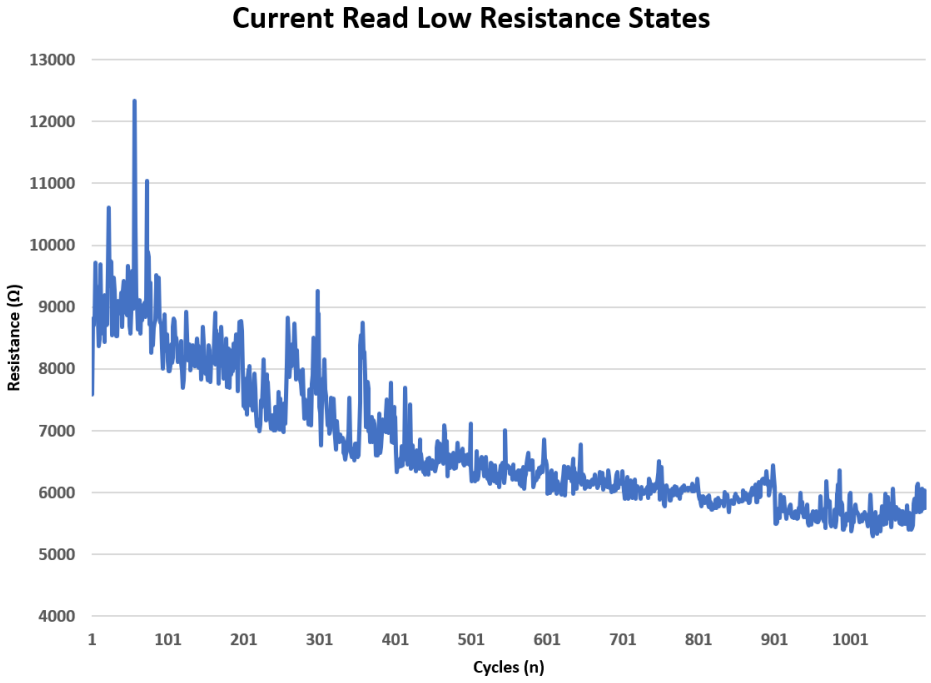

Figure 6.7: Current based read for low resistance states.

The resistances in Figure 6.8 are generated from increasing the high voltage applied to reset the device. Every ten cycles the voltage applied decreases by 5 mV from −1 V to −1.45 V. This was performed with set operations with increasing gate voltage seen in Figure 6.6. The high resistance states show a relationship to the voltage applied. As the voltage increases in magnitude the high resistance state achieved increases. The high resistance appears to vary greatly. However, this method could be used to program the device to high resistance states.

6.2 Circuit simulation

The second fabricated memristor run uses the same hafnium oxide memristive devices. On this wafer, my synapse circuit design and variations of the design will be tested. I implemented test structures that build up the design to validate each component sequentially. I will start with reevaluating the 1T1R structure that makes up the base of my synapse, and build up from there. Aside from building up the required components, different versions of the test structures are implemented. Specifically, the read reference generation is implemented with transistors, resistors, and separate memristors. I have a number of synaptic test structures to be tested. My synapse design is implemented in a core comprised of multiple synapses and one neuron. These tests will more heavily rely on digital communication in and out of the wafer. Testing all the circuits will utilize the same test equipment and have codes to automate tests.

6.3 Test structures

The circuits on the test structure are as listed:

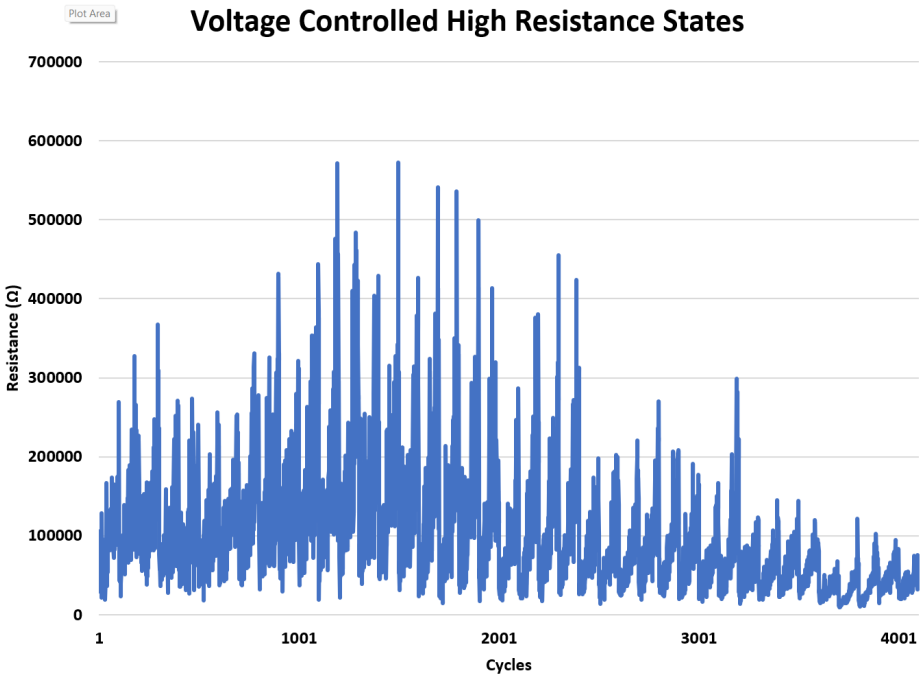

Figure 6.8: High resistance states from variable reset voltages.

1. One transistor one memristor (1T1R)
2. Memristor cell
3. Memristor cell with 1.2 V to 3.3 V converters and current generator
4. Full memristive synapse with 3 bit DAC
5. Synapse with feedback reference
6. Synapse with STDP

From this variation of a 1T1R circuit, there are three main tests to run. The first is to check the saturation currents during a set. The intended voltages at the gate of this transistor during form and set should allow usable currents for set programming. The gate voltages of 0.8 V to 1.2 V should result in a saturation current of 50 µA to 300 µA as seen in Figure 6.9. The DAC circuit and learning circuits should stay within 0.8 V to 1.2 V. Alongside checking the set saturation current limits, the lower current limit for read is checked. A gate voltage of 600 mV should result in a 1 µA current. The maximum expected output voltage generated from the memristor is dependent on the current limiting transistor. If the current limiting transistor is at too low or too high of a gate voltage the output will not be noticeably. Figure 6.10 shows the maximum expected voltage range for different sized pull up transistors at the optimal gate voltage for the current limiting transistor. In addition to testing the current limiting features, the last important test for the 1T1R circuit is the ability to reset the memristor. The reset operation on previous test structures relied on a negative voltage. In this test structure a positive voltage will be applied to the node between the memristor and the current limiting transistor and the top of the memristor will be held at ground. This should be equivalent to previous tests using a negative voltage to reset the device.

The next test structure continues on from the 1T1R by adding in the control transistors for each operation. The output current through the read transistor will be measured to determine the memristor's state. The high voltages for set and reset will pass through

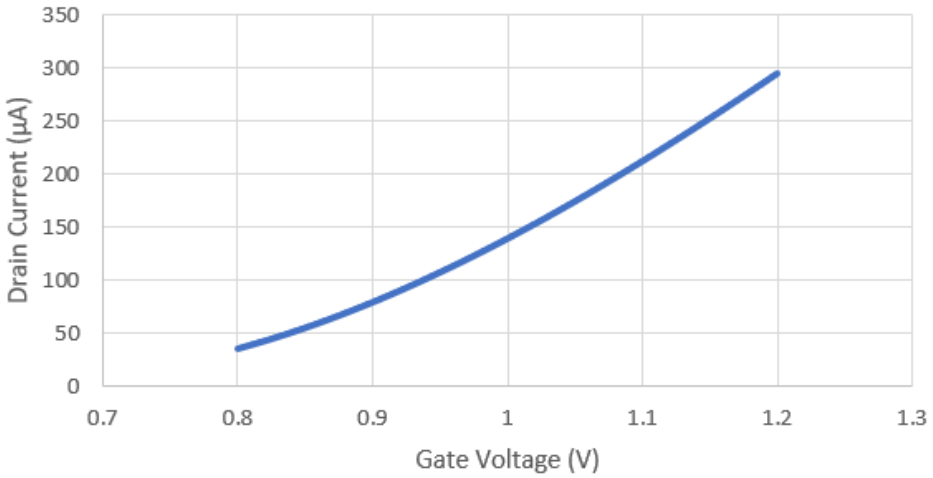

Figure 6.9: Expected current from gate voltage in 1T1R in Memristor Cell.

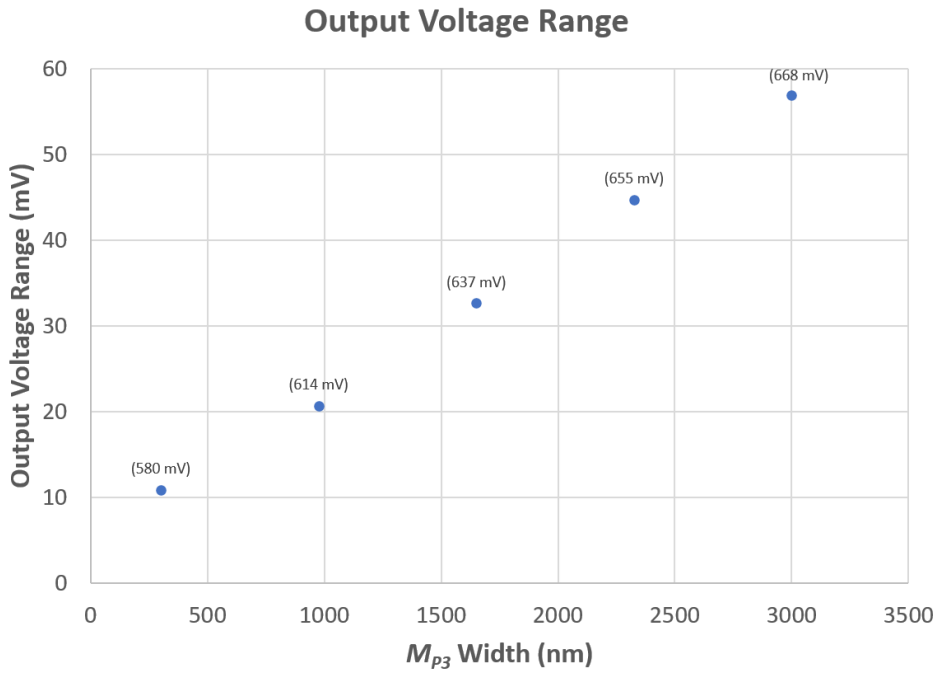

Figure 6.10: Maximum output voltage range at best case voltage for different widths.

a transistor to be applied to the memristor. The next structure further builds upon the memristor cell. This will take 1.2 V instead of 3.3 V. In addition the current generator circuit will create a current output. Using a sourcemeter to hold the output node at mid-rail, the measured output current should range from $-20\,\mu A$ to $20\,\mu A$ for different resistance states at the middle read reference voltage. These test structures should verify the operations of the memristor. The forming, set, reset, and read operations are directly measurable with the available pads.

The next test structure finalizes the basic synapse circuitry needed. This is a single memristor circuit, but includes all circuitry for a group of memristors. There is the DAC and current control circuitry to program and read the device, and the output control circuitry to guarantee the proper output. This circuit also has a second output. A voltage based output is generated using diode connected transistors to double check the read operation is working. The nodes for the DAC and the current control are able to be probed. This allows for the new critical components in the system to be verified. Figure 6.11 shows the expected voltages for the current control circuit. The output CC should be low when not in use. It will be high for either a set or a form, and at mid-rail for a read. Figure 6.12 shows the increasing current control gate voltage for increasing low bits.

After testing the DAC with the memristor, the next test structure uses the DAC to program both the synapse memristor and a feedback reference memristor. Figure 6.13 shows the voltages and currents expected when running the circuit. The output is a direct pad and does not feed into a neuron. The reference generation should create a zero output current point based on the memristor in the reference generator. The output current measured should be positive or negative depending on if it is above or below the reference memristor's resistance level. The internal nodes for the feed back loop are able to be probed.

The last test structure to be tested is the STDP online learning test structure. This test structure also does not have neurons attached on chip, but requires external signals to start the learning process. The signals, Pre and $Post$ activate the STDP circuitry. Depending

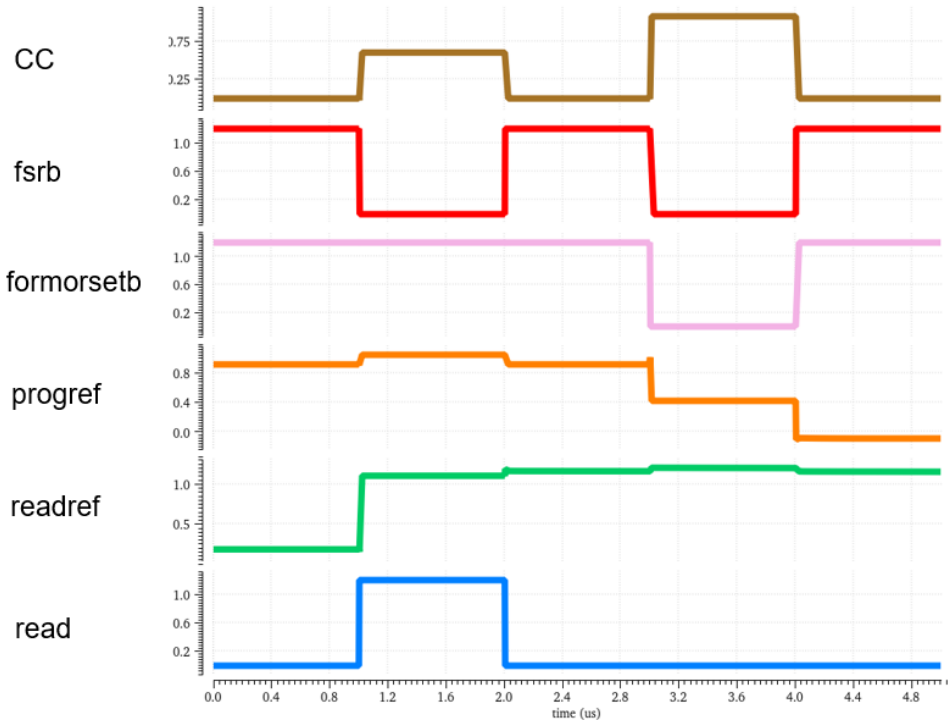

Figure 6.11: Simulation showing the different expected voltages for the current control circuit.

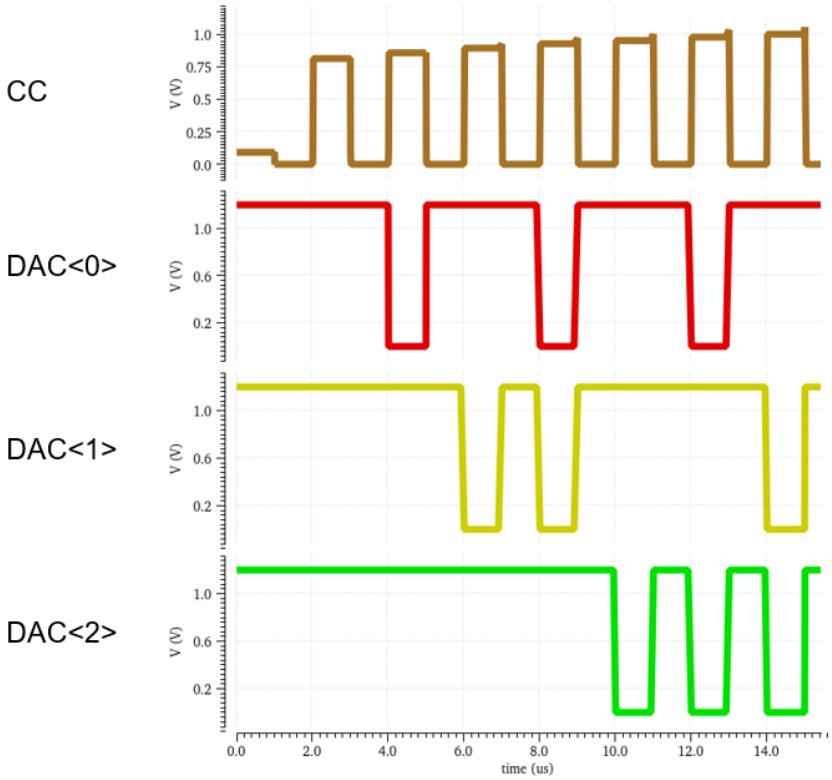

Figure 6.12: Simulation showing the different expected voltages for the 3 bit DAC.

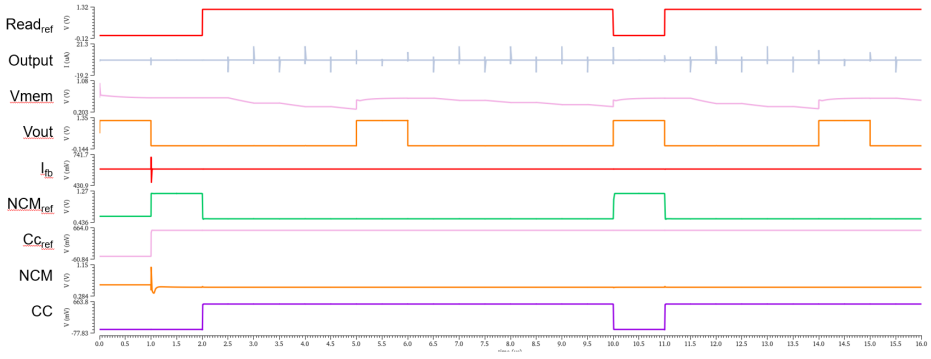

Figure 6.13: Simulation showing the voltage time relationships of the reference generator working with a synapse and neuron.

on the order, a potentiation or depression will occur. The potentiation function occurs when the signal *Pre* occurs before the signal *Post* shown in Figure 6.14. This results in a decrease in the memristors resistance which in turn increases the output current. The depression function occurs when the signal *Post* occurs before the signal *Pre* shown in Figure 6.15. This results in an increase in the memristors resistance which in turn decreases the output current. The polarity and the magnitude of change is a function of the time difference between *Pre* and *Post*. Figure 6.16 shows the decrease in change of memristor resistance as the signals *Pre* and *Post* occur further away.

6.4 Test Cores

The wafer also includes to analog synapse based test cores. Theses cores consist of sixteen synapses and one neuron each. There are two flavors of cores. Both use an integrate and fire neuron. There is one difference in the synapses for each core. One core uses an off chip reference for the current generator circuit, while the other uses an on chip memristor based reference. This is not the full feedback reference. Ideally, between the two cores, one will approximate the other. The off chip reference will be used to test the dynamic range of the current generating circuit. The best reference for the current generator should be achievable with the memristor reference circuit. The memristor reference version should show a direct relationship between the memristors on chip. From these cores, different neuron output rates should be achievable with different input patterns and weights at the synapse. Figure 6.17 shows the full wafer where this circuit is implemented. The bottom right corner blocks are the two synaptic cores. The bottom left corner contains the test structures for the synapse circuits.

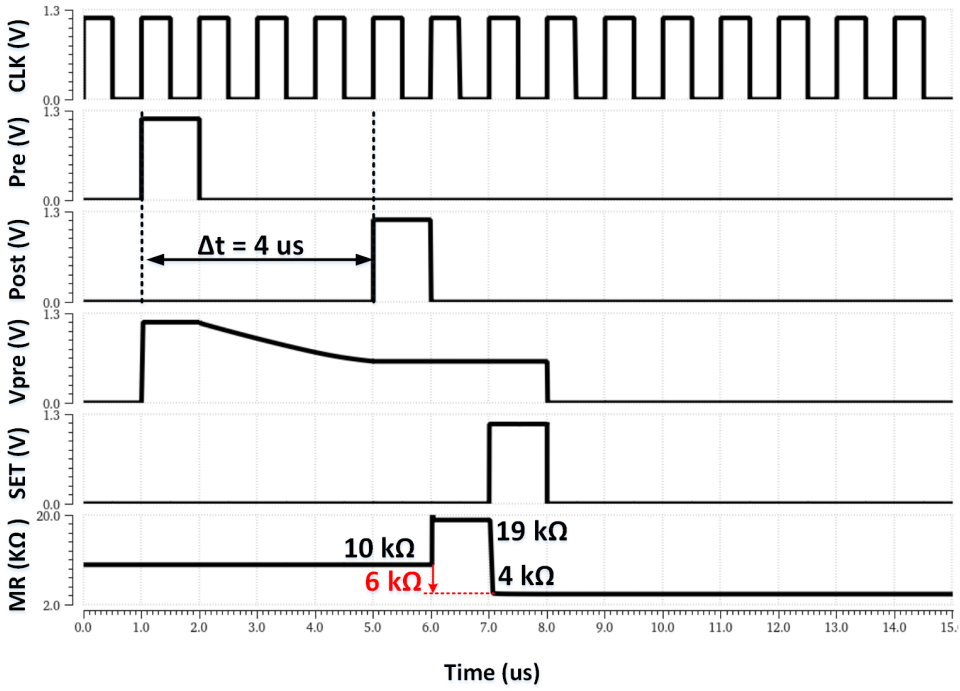

Figure 6.14: Simulation showing the voltage time relationships for a potentiation.

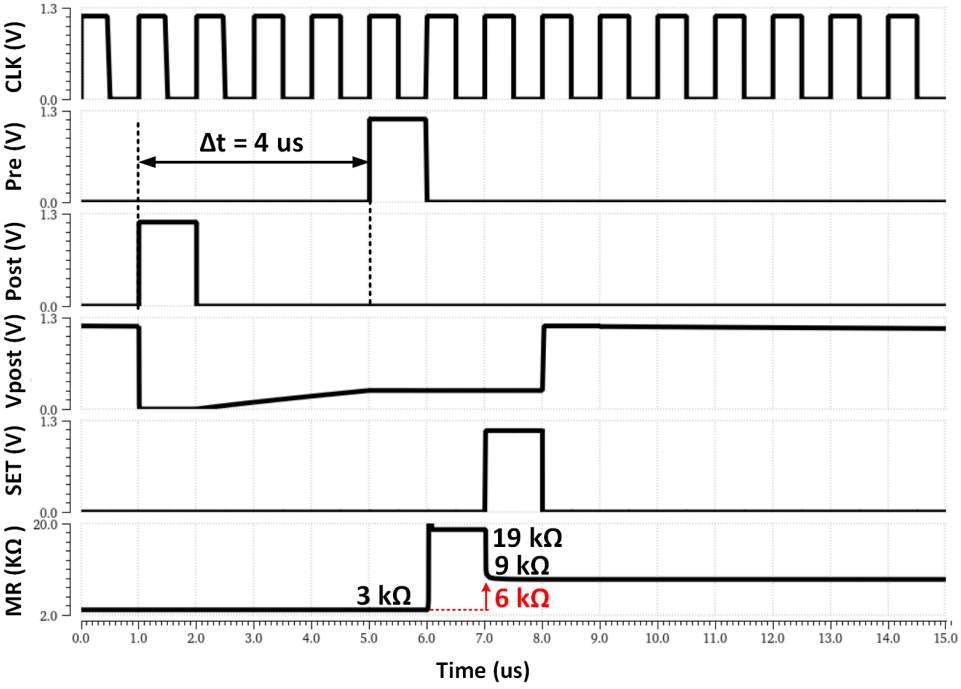

Figure 6.15: Simulation showing the voltage time relationships a depresion.

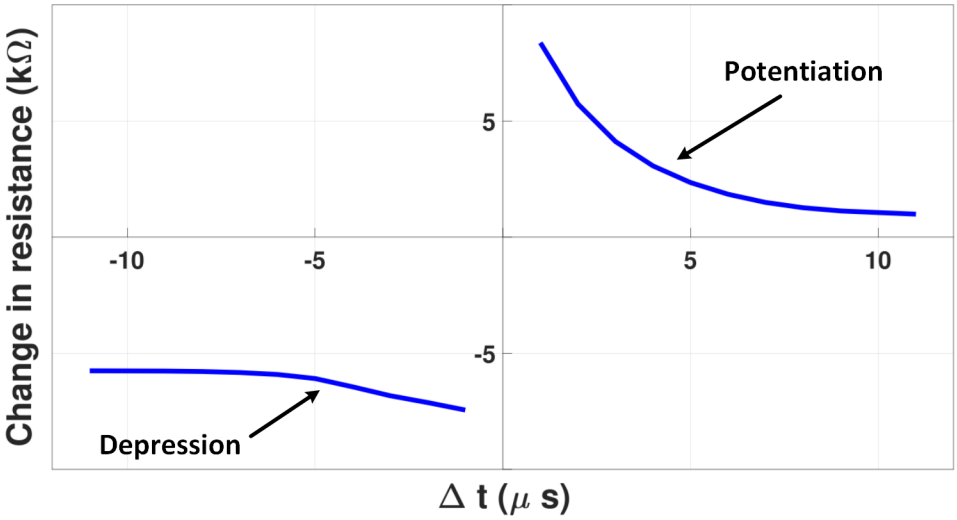

Figure 6.16: STDP curve showing decrease in magnitude change for neuron activity spacing.

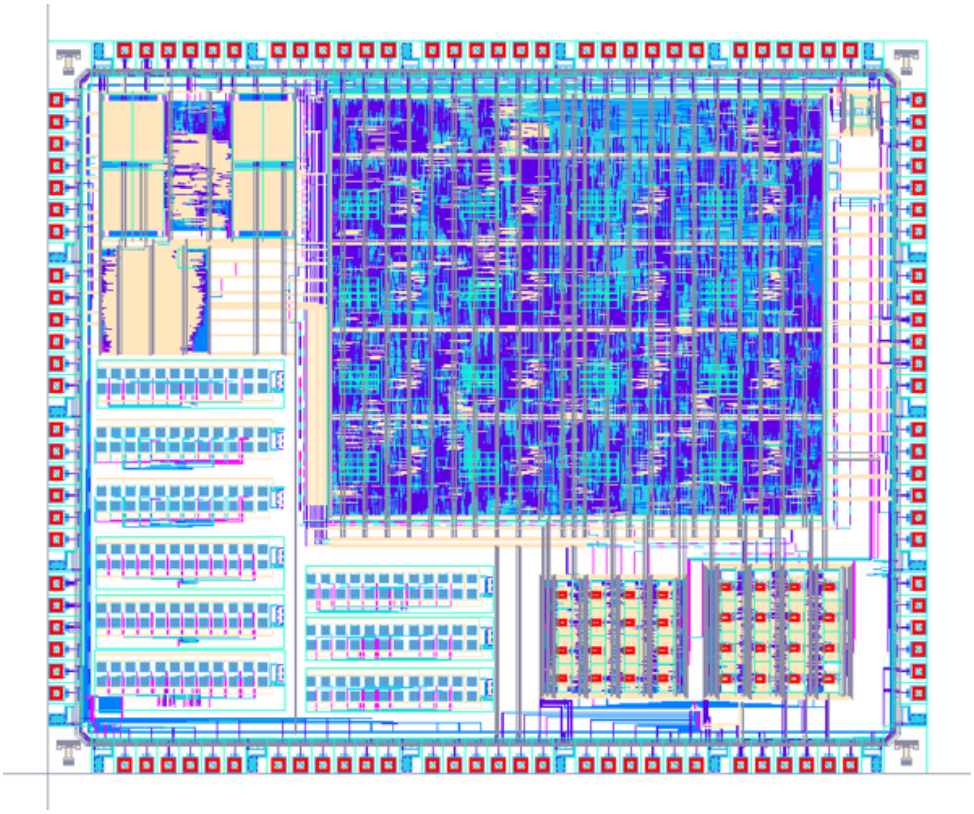

Figure 6.17: Layout image of full RAVENS wafer.

Chapter 7

Summary

Neuromorphic computing can increase computing capabilities. Analog circuits designed to directly implement the neuromorphic functions can increase the efficiency in resource constrained systems. Memristors can be used to implement the memory used in neuromorphic systems. Creating analog neuromorphic circuits using memristors poses unique challenges for different devices. The synapse designs take advantage of specific qualities of the fabricated hafnium oxide devices. The design trades off the potential full resistive range of the devices for a narrow range that has better analog memory capabilities. This paradigm creates challenges with power consumption and signal differentiability. The designs reuse transistors if possible to minimize the need for large transistors. The designs take advantage of the devices available while maintaining the benefits of analog computing. The presented work can be summarized as follows:

- A simple compact model with all operations for the memristor that highlights the relationship of DC current and DC voltage limitations on the switching characteristics of the device.
- A circuit consisting of a fabricated on-chip memristor that can consistently achieve programmable analog values to be used as an analog memory.

- A low-power, low-complexity read operation based on the same current limitation circuit used to update the device resistance.
- An implementation for current control for both programming and reading the memristor's resistance.
- A circuit that allows for the memristor to sink or source current into a neuron.
- A current programming of the memristor through a lightweight digital to analog converter.
- A memristor based reference generation circuit to match the expected output current of the synapse with other devices on chip.
- Unidirectional system for spike timing dependent plasticity using the current control update capabilities.

Table 7.1 shows a comparison of energy per spike for different memristive synapse circuits and different resistance values of the memory device used. The other works in the table utilize high resistance states and higher frequency. For this design, taking advantage of a DC programming routine using low resistance conductive filament growth, the power consumption is comparable to the state of the art in this field.

7.1 Future Work

Testing my circuits after fabrication will lead to potential improvements. The next wafer will have more device options that can be used as replacements for the current memristor where applicable. After studying the devices, if their functionality is equivalent, but show improvements in areas or usability or reliability, my designs can be fit for the use of other current controllable memristors. Other types of functional memristors will be included were their properties help in implementing neuromorphic computing. This will likely take place in the neuron and the plasticity of the synapse. Future fabrications can use

Table 7.1: Energy comparison for synapse circuits.

	Energy (pJ/Spike)	speed (MHz)	Resistance Range (kΩ)
This work	1.58	2.5	3-12
[77]	0.85	100	0.2-200
[30]	100	200	1-1000
[25]	14.6	2000	0.07-0.67
[1]	1.45	100	5-50

memristors were applicable to more efficiently create the learning rules used in this work. Aside from new memristors, the current hafnium oxide memristor and their control circuits can be further improved. While the high resistance state is variable, this may be desired for some applications. Using the same unilateral programming ideology and three step process for online learning. A synapse bypassing the low resistance state and only using the high resistance state is conceivable. This would require a DAC applying a voltage a the reset voltage. The DAC would need to operate at a high enough voltage to reliably reset the device into different states. In addition to a similar single sided use, the high resistance programming could be used in conjunction with the low resistance. This would have added difficulties in the continuous range were the highest low resistance and lowest high resistance meet. Ultimately this work provides a framework for unilaterally current controlled memristor devices. The method for programming and reading can be applied to other similar devices. The method for online learning can work with different types of DAC inputs. Future designs for the neuromorphic synapse will build upon the circuits designed and tested here.

Bibliography

[1] Md Musabbir Adnan et al. "A twin memristor synapse for spike timing dependent learning in neuromorphic systems". In: *2018 31st IEEE International System-on-Chip Conference (SOCC)*. IEEE. 2018, pp. 37–42 (cit. on pp. 5, 95).

[2] Md Musabbir Adnan et al. "Design of a Robust Memristive Spiking Neuromorphic System with Unsupervised Learning in Hardware". In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 17.4 (2021), pp. 1–26 (cit. on p. 50).

[3] James Bradley Aimone. *A Neuromorphic Future for Classic Computing Tasks.* Tech. rep. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020 (cit. on p. 19).

[4] Sherif Amer, Md Sakib Hasan, and Garrett S Rose. "Analysis and modeling of electroforming in transition metal oxide-based memristors and its impact on crossbar array density". In: *IEEE Electron Device Letters* 39.1 (2017), pp. 19–22 (cit. on p. 26).

[5] Sherif Amer et al. "A practical hafnium-oxide memristor model suitable for circuit design and simulation". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2017, pp. 1–4 (cit. on p. 26).

[6] Sherif Amer et al. “Design considerations for insulator metal transition based artificial neurons”. In: *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE. 2019, pp. 1131–1134 (cit. on p. 17).

[7] Sherif Amer et al. “Design techniques for in-field memristor forming circuits”. In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE. 2017, pp. 1224–1227 (cit. on p. 14).

[8] Arnon Amir et al. “Cognitive computing programming paradigm: a corelet language for composing networks of neurosynaptic cores”. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2013, pp. 1–10 (cit. on p. 19).

[9] Mostafa Rahimi Azghadi et al. “A hybrid CMOS-memristor neuromorphic synapse”. In: *IEEE transactions on biomedical circuits and systems* 11.2 (2016), pp. 434–445 (cit. on pp. 22, 24).

[10] Karsten Beckmann et al. “Towards synaptic behavior of nanoscale ReRAM devices for neuromorphic computing applications”. In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 16.2 (2020), pp. 1–18 (cit. on pp. 14, 15, 29).

[11] Mark Bohr. “A 30 year retrospective on Dennard’s MOSFET scaling paper”. In: *IEEE Solid-State Circuits Society Newsletter* 12.1 (2007), pp. 11–13 (cit. on p. 2).

[12] Irem Boybat et al. “Neuromorphic computing with multi-memristive synapses”. In: *Nature communications* 9.1 (2018), pp. 1–12 (cit. on p. 24).

[13] Geoffrey W Burr et al. “Neuromorphic computing using non-volatile memory”. In: *Advances in Physics: X* 2.1 (2017), pp. 89–124 (cit. on p. 3).

[14] Gangotree Chakma et al. “Memristive mixed-signal neuromorphic systems: Energy-efficient learning at the circuit-level”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8.1 (2017), pp. 125–136 (cit. on p. 21).

[15] Leon Chua. “Memristor-the missing circuit element”. In: *IEEE Transactions on circuit theory* 18.5 (1971), pp. 507–519 (cit. on pp. 3, 12).

[16] Antonio D Córcoles et al. "Challenges and opportunities of near-term quantum computing systems". In: *arXiv preprint arXiv:1910.02894* (2019) (cit. on p. 3).

[17] Loai Danial et al. "Two-terminal floating-gate transistors with a low-power memristive operation mode for analogue neuromorphic computing". In: *Nature Electronics* 2.12 (2019), pp. 596–605 (cit. on p. 9).

[18] M. Davies et al. "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning". In: *IEEE Micro* 38.1 (Jan. 2018), pp. 82–99. ISSN: 0272-1732. DOI: 10.1109/MM.2018.112130359 (cit. on p. 19).

[19] Javier Del Valle et al. "Challenges in materials and devices for resistive-switching-based neuromorphic computing". In: *Journal of Applied Physics* 124.21 (2018), p. 211101 (cit. on p. 17).

[20] Adam Z Foshie et al. "A Multi-Context Neural Core Design for Reconfigurable Neuromorphic Arrays". In: *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2021, pp. 67–72 (cit. on p. 21).

[21] Vittorio Fra et al. "Investigation on the Stabilizing Effect of Titanium in HfO2-Based Resistive Switching Devices With Tungsten Electrode". In: *Frontiers in Nanotechnology* 2 (2020), p. 592684 (cit. on p. 14).

[22] Steve Furber. "Large-scale neuromorphic computing systems". In: *Journal of neural engineering* 13.5 (2016), p. 051001 (cit. on p. 3).

[23] Jubin Hazra et al. "Improving the Memory Window/Resistance Variability Trade-Off for 65nm CMOS Integrated HfO 2 Based Nanoscale RRAM Devices". In: *2019 IEEE International Integrated Reliability Workshop (IIRW)*. IEEE. 2019, pp. 1–4 (cit. on p. 10).

[24] Jubin Hazra et al. "Optimization of Switching Metrics for CMOS Integrated HfO2 based RRAM Devices on 300 mm Wafer Platform". In: *2021 IEEE International Memory Workshop (IMW)*. IEEE. 2021, pp. 1–4 (cit. on p. 10).

[25] Miao Hu et al. “A compact memristor-based dynamic synapse for spiking neural networks”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36.8 (2016), pp. 1353–1366 (cit. on pp. 5, 50, 95).

[26] Giacomo Indiveri. “A low-power adaptive integrate-and-fire neuron circuit”. In: *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03.* Vol. 4. IEEE. 2003, pp. IV–IV (cit. on p. 22).

[27] Giacomo Indiveri et al. “Integration of nanoscale memristor synapses in neuromorphic computing architectures”. In: *Nanotechnology* 24.38 (2013), p. 384010 (cit. on p. 24).

[28] Giacomo Indiveri et al. “Neuromorphic Silicon Neuron Circuits”. In: *Frontiers in Neuroscience* 5 (2011). DOI: 10.3389/fnins.2011.00073 (cit. on pp. 22–24).

[29] Sung-Eun Kim et al. “Sodium-Doped Titania Self-Rectifying Memristors for Crossbar Array Neuromorphic Architectures”. In: *Advanced Materials* 34.6 (2022), p. 2106913 (cit. on p. 24).

[30] Duygu Kuzum, Rakesh GD Jeyasingh, and H-S Philip Wong. “Energy efficient programming of nanoelectronic synaptic devices for large-scale implementation of associative and temporal sequence learning”. In: *2011 International Electron Devices Meeting*. IEEE. 2011, pp. 30–3 (cit. on pp. 5, 95).

[31] Min-Woo Kwon et al. “Integrate-and-fire neuron circuit using positive feedback field effect transistor for low power operation”. In: *Journal of Applied Physics* 124.15 (2018), p. 152107 (cit. on p. 24).

[32] Steven Lequeux et al. “A magnetic synapse: multilevel spin-torque memristor with perpendicular anisotropy”. In: *Scientific reports* 6.1 (2016), pp. 1–7 (cit. on p. 9).

[33] Yesheng Li and Kah-Wee Ang. “Hardware Implementation of Neuromorphic Computing Using Large-Scale Memristor Crossbar Arrays”. In: *Advanced Intelligent Systems* 3.1 (2021), p. 2000137 (cit. on pp. 3, 24).

[34] Yibo Li et al. “Review of memristor devices in neuromorphic computing: materials sciences and device challenges”. In: *Journal of Physics D: Applied Physics* 51.50 (2018), p. 503002 (cit. on p. 3).

[35] Fu-Xiang Liang, I-Ting Wang, and Tuo-Hung Hou. “Progress and benchmark of spiking neuron devices and circuits”. In: *Advanced Intelligent Systems* 3.8 (2021), p. 2100007 (cit. on p. 24).

[36] Maximilian Liehr et al. “Fabrication and performance of hybrid RERAM-CMOS circuit elements for dynamic neural networks”. In: *Proceedings of the International Conference on Neuromorphic Systems*. 2019, pp. 1–4 (cit. on pp. 7, 14, 15, 17).

[37] Chris A Mack. “Fifty years of Moore’s law”. In: *IEEE Transactions on semiconductor manufacturing* 24.2 (2011), pp. 202–207 (cit. on p. 1).

[38] Joshua J Maraj et al. “Short-Term Facilitation-Then-Depression Enables Adaptive Processing of Sensory Inputs by Ion Channels in Biomolecular Synapses”. In: *ACS Applied Electronic Materials* 3.10 (2021), pp. 4448–4458 (cit. on p. 18).

[39] Danijela Marković et al. “Physics for neuromorphic computing”. In: *Nature Reviews Physics* 2.9 (2020), pp. 499–510 (cit. on p. 3).

[40] Carver Mead. *Analog VLSI and Neural Systems*. Addison Wesley Publishing Company, 1989. ISBN: 0201059924. URL: https://www.amazon.com/Analog-VLSI-Neural-Systems-Carver/dp/0201059924?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0201059924 (cit. on pp. 21–23, 69).

[41] Carver Mead. “Neuromorphic electronic systems”. In: *Proceedings of the IEEE* 78.10 (1990), pp. 1629–1636 (cit. on p. 21).

[42] Carver A Mead and Misha A Mahowald. “A silicon model of early visual processing”. In: *Neural networks* 1.1 (1988), pp. 91–97 (cit. on p. 21).

[43] Baker Mohammad et al. “State of the art of metal oxide memristor devices”. In: *Nanotechnology Reviews* 5.3 (2016), pp. 311–329 (cit. on p. 17).

[44] H Mulaosmanovic et al. "Evidence of single domain switching in hafnium oxide based FeFETs: Enabler for multi-level FeFET memory cells". In: *2015 IEEE International Electron Devices Meeting (IEDM)*. IEEE. 2015, pp. 26–8 (cit. on p. 10).

[45] H Mulaosmanovic et al. "Novel ferroelectric FET based synapse for neuromorphic systems". In: *2017 Symposium on VLSI Technology*. IEEE. 2017, T176–T177 (cit. on p. 10).

[46] Manu V Nair, Lorenz K Muller, and Giacomo Indiveri. "A differential memristive synapse circuit for on-line learning in neuromorphic computing systems". In: *Nano Futures* 1.3 (2017), p. 035003 (cit. on p. 24).

[47] Joseph S Najem et al. "Dynamical nonlinear memory capacitance in biomimetic membranes". In: *Nature communications* 10.1 (2019), pp. 1–11 (cit. on p. 18).

[48] SR Nandakumar et al. "A phase-change memory model for neuromorphic computing". In: *Journal of Applied Physics* 124.15 (2018), p. 152135 (cit. on p. 9).

[49] Antonio S Oblea et al. "Silver chalcogenide based memristor devices". In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2010, pp. 1–3 (cit. on p. 9).

[50] Aditya Kuber Parit et al. "Design and modeling of niobium oxide-tantalum oxide based self-selective memristor for large-scale crossbar memory". In: *Chaos, Solitons & Fractals* 145 (2021), p. 110818 (cit. on p. 18).

[51] Agostino Pirovano et al. "Electronic switching in phase-change memories". In: *IEEE Transactions on Electron Devices* 51.3 (2004), pp. 452–459 (cit. on p. 9).

[52] Agostino Pirovano et al. "Low-field amorphous state resistance and threshold voltage drift in chalcogenide materials". In: *IEEE Transactions on Electron Devices* 51.5 (2004), pp. 714–719 (cit. on p. 9).

[53] Manu Rathore et al. "A Compact Model for the Variable Switching Dynamics of HfO2 Memristors". In: *65th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE. 2022 (cit. on pp. 14, 16, 26).

[54] Garrett S Rose et al. “A system design perspective on neuromorphic computer processors”. In: *Neuromorphic Computing and Engineering* 1.2 (2021), p. 022001 (cit. on p. 21).

[55] Kauschick Roy, Saibal Mukhopadhyay, and Hamid Mahmoodi-Meimand. “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits”. In: *Proceedings of the IEEE* 91.2 (2003), pp. 305–327 (cit. on p. 2).

[56] Arianna Rubino et al. “Ultra-low-power FDSOI neural circuits for extreme-edge neuromorphic intelligence”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.1 (2020), pp. 45–56 (cit. on p. 24).

[57] Sagarvarma Sayyaparaju, Sherif Amer, and Garrett S Rose. “A bi-memristor synapse with spike-timing-dependent plasticity for on-chip learning in memristive neuromorphic systems”. In: *2018 19th International Symposium on Quality Electronic Design (ISQED)*. IEEE. 2018, pp. 69–74 (cit. on pp. 24, 47).

[58] Sagarvarma Sayyaparaju et al. “Circuit Techniques for Online Learning of Memristive Synapses in CMOS-Memristor Neuromorphic Systems”. In: *Proceedings of the on Great Lakes Symposium on VLSI 2017 - GLSVLSI 17*. ACM Press, 2017. DOI: 10.1145/3060403.3060418 (cit. on p. 24).

[59] Catherine D Schuman et al. “A software framework for comparing training approaches for spiking neuromorphic systems”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2021, pp. 1–10 (cit. on p. 24).

[60] Catherine D Schuman et al. “An evolutionary optimization framework for neural networks and neuromorphic architectures”. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2016, pp. 145–154 (cit. on p. 21).

[61] Catherine D Schuman et al. “Evolutionary optimization for neuromorphic systems”. In: *Proceedings of the Neuro-inspired Computational Elements Workshop*. 2020, pp. 1–9 (cit. on p. 21).

[62] John Shalf. "The future of computing beyond Moore's Law". In: *Philosophical Transactions of the Royal Society A* 378.2166 (2020), p. 20190061 (cit. on p. 1).

[63] John M Shalf and Robert Leland. "Computing beyond moore's law". In: *Computer* 48.12 (2015), pp. 14–23 (cit. on p. 1).

[64] Gagandeep Singh et al. "A review of near-memory computing architectures: Opportunities and challenges". In: *2018 21st Euromicro Conference on Digital System Design (DSD)*. IEEE. 2018, pp. 608–617 (cit. on p. 2).

[65] Dmitri B. Strukov et al. "The missing memristor found". In: *Nature* 453.7191 (May 2008), pp. 80–83. DOI: 10.1038/nature06932 (cit. on p. 14).

[66] M Suri et al. "Addition of HfO2 interface layer for improved synaptic performance of phase change memory (PCM) devices". In: *Solid-state electronics* 79 (2013), pp. 227–232 (cit. on p. 9).

[67] H Traff. "Novel approach to high speed CMOS current comparators". In: *Electronics Letters* 3.28 (1992), pp. 310–312 (cit. on p. 116).

[68] Navnidhi K Upadhyay et al. "Emerging memory devices for neuromorphic computing". In: *Advanced Materials Technologies* 4.4 (2019), p. 1800589 (cit. on p. 9).

[69] Adrien F Vincent et al. "Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems". In: *IEEE transactions on biomedical circuits and systems* 9.2 (2015), pp. 166–174 (cit. on p. 9).

[70] R. Weiss, G. Chakma, and G. S. Rose. "A Synchronized Axon Hillock Neuron for Memristive Neuromorphic Systems". In: *60th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. Boston, MA, Aug. 2017 (cit. on pp. 69, 70).

[71] Ryan Weiss et al. "A soft-matter biomolecular memristor synapse for neuromorphic systems". In: *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE. 2018, pp. 1–4 (cit. on pp. 18, 67, 68).

[72] Shiping Wen, Zhigang Zeng, and Tingwen Huang. "Exponential stability analysis of memristor-based recurrent neural networks with time-varying delays". In: *Neurocomputing* 97 (2012), pp. 233–240 (cit. on p. 13).

[73] Xinyu Wu et al. "A CMOS spiking neuron for brain-inspired neural networks with resistive synapses and in situ learning". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 62.11 (2015), pp. 1088–1092 (cit. on p. 50).

[74] Liming Xiu. "Time Moore: Exploiting Moore's Law from the perspective of time". In: *IEEE Solid-State Circuits Magazine* 11.1 (2019), pp. 39–55 (cit. on p. 1).

[75] Wei Yi et al. "Biological plausibility and stochasticity in scalable VO2 active memristor neurons". In: *Nature communications* 9.1 (2018), pp. 1–10 (cit. on p. 18).

[76] ZhenYu Yin et al. "What are memristor, memcapacitor, and meminductor?" In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 62.4 (2015), pp. 402–406 (cit. on p. 14).

[77] Shimeng Yu et al. "A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling". In: *2012 International Electron Devices Meeting*. IEEE. 2012, pp. 10–4 (cit. on pp. 5, 95).

[78] Andre Zeumault et al. "TCAD Modeling of Resistive-Switching Memristors: Efficient of Device-Circuit HfO2 Co-Design for Neuromorphic Systems". In: *Memristive Neuromorphics: Materials, Devices, Circuits, Architectures, Algorithms and their Co-Design* (2022) (cit. on p. 26).

[79] Yang Zhang et al. "Brain-inspired computing with memristors: Challenges in devices, circuits, and systems". In: *Applied Physics Reviews* 7.1 (2020), p. 011308 (cit. on p. 4).

[80] You Zhou and Shriram Ramanathan. "Mott memory and neuromorphic devices". In: *Proceedings of the IEEE* 103.8 (2015), pp. 1289–1310 (cit. on p. 18).

[81] Jiadi Zhu et al. "A comprehensive review on emerging artificial neuromorphic devices". In: *Applied Physics Reviews* 7.1 (2020), p. 011312 (cit. on p. 10).

[82] Xingqi Zou et al. "Breaking the von Neumann bottleneck: architecture-level processing-in-memory technology". In: *Science China Information Sciences* 64.6 (2021), pp. 1–10 (cit. on p. 2).

Appendix A

Abbreviations

CMOS	Complementary Metal Oxide Semiconductor
DANNA	Dynamic Adaptive Neural Network Arrays
DC	Direct Current
DIB	Dipole Induced Bilayer
DRAM	Dynamic Random Access Memory
EONS	Evolutionary Optimization of Neuromorphic Systems
FET	Field Effect Transistor
HRS	High Resistance State
LIF	Leaky Integrate-and-Fire
LRS	Low Resistance State
MSB	Most Significant Bit
MTJ	Magnetic Tunnel Junction
NIDA	Neuroscience-Inspired Dynamic Architecture
PCM	Phase Change Material
SOC	System on Chip
STDP	Spike Timing Dependent Plasticity
TMO	Transistion Metal Oxide
VLSI	Very-Large-Scale Integration

Appendix B

Test Structures

B.1 Test Structure Procedures

There are five test structure pads on the RAVENS wafer. Four of the five pad frames contain synapse circuits while the fifth contains a neuron circuit. The four synapse circuits will be described here for testing the components described in this work.

The pad frame for the test structures is twelve by two. The test structures use the twenty four pins to operate and probe the circuits. The layout of the pad frame is seen in Figure B.1. The pads are 60 µm square with 40 µm between pads and between rows. Either a twenty four pin probe card or two twelve pin probes are required to test all circuits. Due to a limited number of pins, pads are reused between different circuits in the first test structure.

B.1.1 First Test Structures

The first test structure seen in Figure B.2 consists of three memristors circuits. It is approximately 70 µm by 30 µm. There is a single memristor connected to a single n-type transistor, $memristor_3$, in the Table B.1. Building onto this is a single memristor cell, $memristor_2$, consisting of all high voltage transistor needed and including an output transistor for current, like Figure 4.2. The final addition of memristor circuitry, using

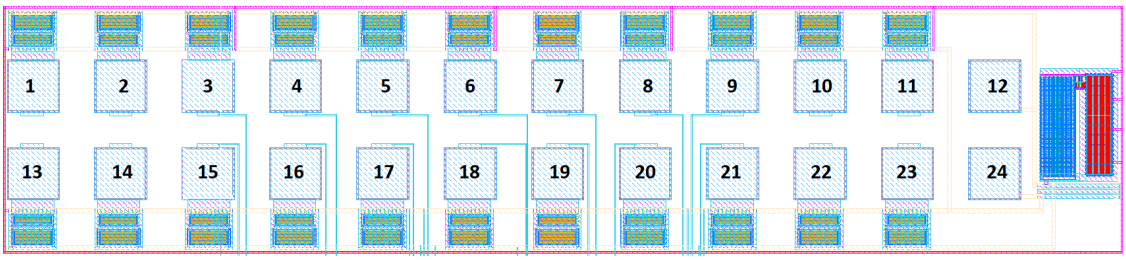

Figure B.1: Pad frame for test structure circuits.

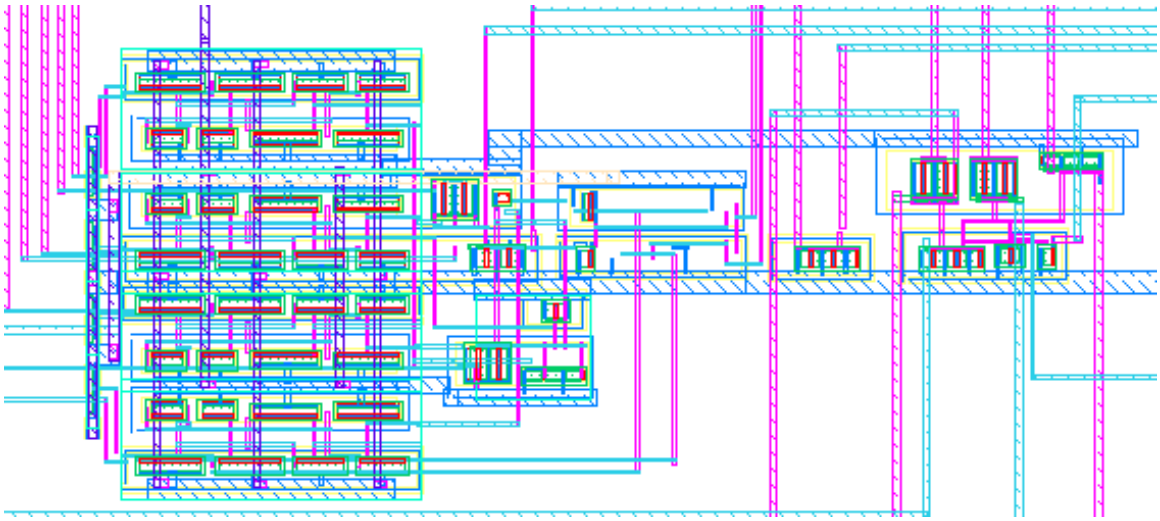

Figure B.2: Test structure 1 circuit layout image.

Table B.1: Pad frame for test structure 1 pad names.

Pin Number	Pin Name	Pin Type	Pin Description
1	Mtop	Analog Output	$Memristor_1$ top electrode
2	Vst	Analog Reference	Set voltage applied across memristors
3	Vrst	Analog Reference	Reset voltage applied across memristors
4	Vcc	Analog Input	Current control node for $memristor_1$
5	Synout	Analog Output	Current output from $memristor_1$
6	Read	Digital Input	Read $memristor_1$ output current enabled
7	Reset	Digital Input	Reset $memristor_1$ 0V-1.2V enable
8	Set	Digital Input	Set $memristor_1$ 0V-1.2V enable
9	Form	Digital Input	Form $memristor_1$ 0V-1.2V enable
10	Vncm	Analog Reference	Reference for synapse output current
11	VDD1v2	Power	1.2V power
12	VDD3v3	Power	3.3V power
13	Mbot	Analog Output	$Memristor_1$ bottom electrode
14	Mtop3	Analog Input	$Memristor_3$ top electrode
15	Vcc3	Analog Input	Current compliance for $memristor_3$
16	Out2	Analog Output	$Memristor_2$ output current
17	Reset2	Digital Input	Rest $memristor_2$ 0V-3.3V enable
18	SetB2	Digital Input	Set $memristor_2$ 3.3V-0V enable
19	ResetB2	Digital Input	Reset $memristor_2$ 3.3V-0V enable
20	Mbot2	Analog Output	$Memristor_2$ bottom electrode
21	Mtop2	Analog Output	$Memristor_2$ top electrode
22	FormB2	Digital Input	Forming $memristor_2$
23	Vcc2	Analog Input	Current compliance for $memristor_2$
24	GND	Power	Ground

$memristor_1$, on this test structure adds the current generation circuit in Figure 4.5. The circuit with $memristor_3$ also includes level shifters and the current generating circuit seen in Figure 4.5. The test circuit with $memristor_2$ uses the same reference voltages as the test structure with $memristor_3$ for set and reset. Pins 2 and 3 will be near 2 V to allow the memristor to set and reset and will be determined via testing the device. The digital signals for the first and second memristors apply the analog reference voltages to the devices. There is pad access to the top and bottom nodes of the memristor. Access to these nodes is critically important to evaluate the voltages achieved at the memristor. The testing procedure for this test structure flows from testing $memristor_3$ then $memristor_2$ and finally $memristor_1$. Testing device $memristor_3$ will be equivalent to testing previous memristor test structures. The device will be formed using a high voltage at pin 23. The voltage at pin 23 is directly applied to the memristor. While forming the device, pin 22 controls the current through the memristor. After forming, applying a negative voltage at pin 23 should reset the device, and applying a positive voltage should set the device. Expected voltage used for pin 22 is around 1.2 V. For pin 23 during forming voltages up to 3.3 V should form the device. For set and reset applying 1.5 V and -1.5 V should switch the device. The current through this pin is measured to determine the overall change in resistance. The device will be tested with different current compliance voltages on pin 22 during forming and set. Different magnitudes of voltage at pin 23 will be tested to determine the possible values for pin 2 and 3. Pins 2 and 3 are shared between $memristor_2$ and $memristor_1$. For both test structures, direct access to the memristor top and bottom electrode is available, but the intent is to measure the voltages produced during set and reset. Since the memristor switching is critically important, these nodes need to achieve the correct magnitudes to induce switching.

For $memristor_2$ the voltages for forming, set, and reset are applied through transistors that are digitally activated. The corresponding signal and transistor should allow the memristor to reach voltage levels at its electrodes to allow for switching. Applying the enable voltages on pins 15, 18, 20, and 19 for forming, reset, and set while controlling

current through pin 14 should operate the device equivalently to $memristor_1$. In addition to transistors applying switching voltages, $memristor_2$ has an output transistor and output current on pin 21. The output transistor should produce different output currents based on the resistance of $memristor_2$. There is not a dedicated read enabling transistor, so the read voltage is applied through the top electrode, pin 16. The current compliance voltage and voltage at the top electrode should produce a voltage at the bottom memristor electrode, pin 17, that is high enough to activate the output transistor and produce an output current. For $memristor_2$, pins 15, 18, 19, and 20 are digital control signals for form, reset, and set. Pins 16 is an analog input during read, the voltage will likely be between 0.6 V and 1.2 V, or an analog output during form, reset, and set. Pin 14 is used to control current during forming, set, and read. The voltage at pin 14 during read should be 0.6 V and during forming and set it should be near 1.2 V. Pin 17 will be an analog output during all $memristor_2$ operations. Pin 21 is an analog output during read operations, equivalent to Iout in Figure 4.2. Pin 21 should be a positive voltage, with a measured current. For testing at DC, apply 0.6 V to 1.2 V and measure current. This internal node in the full synapse should be about 0.7 V. Additionally, the output current should be measured with an off chip resistor with a high voltage applied. An oscilloscope measuring the voltage drop across the resistor will give timing information on the current during the read operation. The voltage at the output node should be equivalent to the tests at DC, likely 0.6 V to 1.2 V is applied to the resistor. The size of the resistor should give a voltage drop at the expected current levels that is visible on the measuring oscilloscope. Other test structures will also use an off chip resistor to measure current.

The final device on this test structure uses 1.2 V logic and uses level shifters to activate the high voltage transistors. Pins 6, 7, 8, and 9 control the read, reset, set, and forming operations. Current compliance is handled through pin 4 with similar voltage to the previous test circuits. This circuit has a reference voltage for generating the output synapse current. Unlike the circuit with $memristor_2$, $memristor_3$ has a push-pull synapse output which allows for by directional current output. The reference voltage on pin 10

sets the pull down current for the output synout on pin 5. The voltage at pin 10 should be 0.6 V or slightly lower. The output current should be measure similarly to the previous test structure. The difference here is the output voltage should be held at a constant 0.6 V. Measuring current for different memristor resistances should provide correlating currents. This test circuit is used to find the reference voltage for the cores with off chip references. Ultimately the goal of this test structure is to determine the memristor characteristics have not changes. Switching occurs with positive and negative voltages applied after the initial forming. The output current can be adjusted by changing the device resistance. The output current can be bidirectional with the right reference voltage.

B.1.2 Second Test Structures

The second test structure seen in Figure B.3 consists of one memristor circuit which is an entire synapse circuit from Figure 4.1 with some addition output circuitry. Table B.2 shows the pin location and type for this test structure. It is approximately 220 µm by 10 µm. This test structure has one test circuit with four main points to test. The first critically testing point, like the previous test structure is the ability for the memristor device to switch. Pins 7, 8, and 9 are used to reset, set and form by applying 1.2 V digital signals. During these operations current compliance is no longer determined by an external pin. It is internally generated and the value is determined by the 1.2 V digital signals on pins 21, 22, and 23 which controls a circuit as seen in Figure 4.6. Changing the bits on these pins should result in different resistance values when running the set process after a reset on a formed device. The MSB is pin 21 and a higher value results in more current and thus a lower resistance when setting the device. The values can be read out using the normal operation of the synapse, but pins 1 and 4 give direct access to the memristor. This setup consists of all circuitry used for the base synapse circuit seen in the cores. The references on pins 2 and 3 come from the values used in the first test structure that give the best results for switching. The reference on pin 10 also comes from the first test structure. Pin 16 in this test structure is an internal node that is also present in the first test structure.

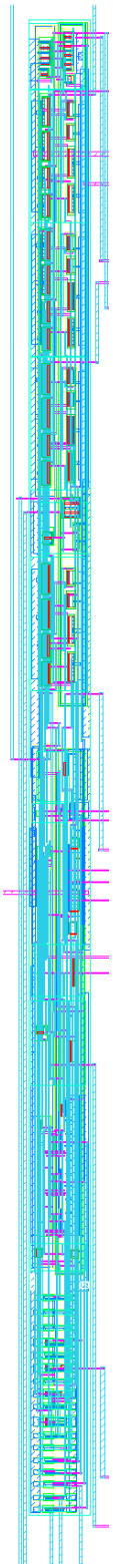

Figure B.3: Test structure 2 circuit layout image.

Table B.2: Pad frame for test structure 2 pad names.

Pin Number	Pin Name	Pin Type	Pin Description
1	Mtop	Analog Output	Memristor top electrode
2	Vst	Analog Reference	Set voltage applied across memristors
3	Vrst	Analog Reference	Reset voltage applied across memristors
4	Mbot	Analog Output	Memristor bottom electrode
5	synout	Analog Output	Current output from memristor
6	Read	Digital Input	Read memristor output current enabled
7	Reset	Digital Input	Reset memristor 0V-1.2V enable
8	Set	Digital Input	Set memristor 0V-1.2V enable
9	Form	Digital Input	Form memristor 0V-1.2V enable
10	Vncm	Analog Reference	Reference for synapse output current
11	VDD1v2	Power	1.2V power
12	VDD3v3	Power	3.3V power
13	OUT<2>	Digital Output	Output current level digitized
14	OUT<1>	Digital Output	Output current level digitized
15	OUT<0>	Digital Output	Output current level digitized
16	PGR	Analog Output	Voltage node after memristor cell
17	Vcc	Digital Input	Current control node
18	CLK	Digital Input	Clock for synapse control
19	SYNOUT	Digital Output	Clocked synapse output
20	readref	Analog Reference	Reference voltage determining read current compliance
21	DACV<2>	Digital Input	Bit for set/form current compliance
22	DACV<1>	Digital Input	Bit for set/form current compliance
23	DACV<0>	Digital Input	Bit for set/form current compliance
24	GND	Power	Ground

The voltage here is equivalent to Out2 in the first test structure. The read out uses a reference from pin 20 that generates the current compliance. The voltage used on this pin should be close to 0.6 V. When running a read operation, the memristor top and bottom nodes should be equivalent to the first test structure. The output on pin 5 should be equivalent to the output of $memristor_1$ on the first test structure. The current can be read similarly by applying a 0.6 V source and measuring current or attaching a resistor to a high voltage source. Pin 5 activates with the read signal, while pin 19 is only activated if clock is low. The clock, pin 18, is used only to gate the output in this circuit. In the cores, the read signal will be connected to the clock, but here these signals can be controlled independently. The circuit should be tested at a frequency of 1 MHz. The maximum speed can be determined by the time it takes for pin 5 to reach steady state after the read signal is activated. The time this takes multiplied by two is the minimum clock period. At this speed the output on pin 19 should be valid during the entire time clock is low. In addition to the normal synapse output currents, this test structure also has some current comparators to digitize the output current. Pins 13, 14, and 15 use a current comparator similar to [67] to digitize the current generated from different memristor resistances. The three outputs use different comparison thresholds and should activate sequentially upon higher output current. Ultimately this test structure should provide the basis that the synapse circuits used in the cores will function properly. They are programmable with pins 21, 22, and 23 and the current is constant and consistent on pin 19 for different programmed values.

B.1.3 Third Test Structures

The third test structure seen in Figure B.4 consists of two memristors to create one system. Table B.3 shows the pin location and type for this test structure. It is approximately 315 µm by 25 µm. This test structure has two memristors, one used as a reference as seen in Figures 4.8 and 4.9. The other memristor is in a synapse circuit with a programming

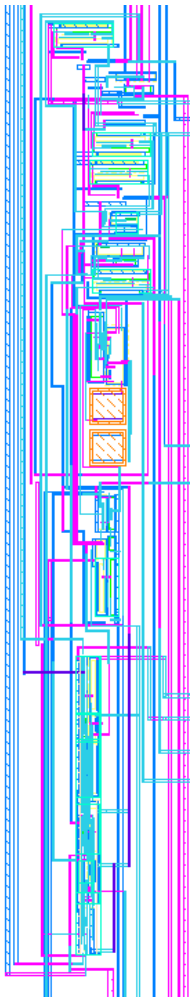

Figure B.4: Test structure 3 circuit layout image.

Table B.3: Pad frame for test structure 3 pad names.

Pin Number	Pin Name	Pin Type	Pin Description
1	Vmid	Analog Reference	Midrail voltage reference 0.6V
2	Vst	Analog Reference	Set voltage applied across memristors
3	Vrst	Analog Reference	Reset voltage applied across memristors
4	Prog	Analog Output	Voltage generated from DACV values
5	setref	Digital Input	Set reference memristor 0V-1.2V enable
6	formref	Digital Input	Form reference memristor 0V-1.2V enable
7	read	Digital Input	Read memristor 0V-1.2V enable
8	reset	Digital Input	Reset memristor 0V-1.2V enable
9	set	Digital Input	Set memristor 0V-1.2V enable
10	form	Digital Input	Form memristor 0V-1.2V enable
11	VDD1v2	Power	1.2V power
12	VDD3v3	Power	3.3V power
13	SYNOUT	Analog Output	Clocked synapse output
14	refread	Analog Reference	Reference voltage determining read current compliance
15	Iref	Analog Output	Internal feedback voltage from reference
16	Vcc	Analog Output	Current control node
17	readref	Digital Input	Read reference memristor 0V-1.2V enable
18	NCMref	Analog Output	Reference for synapse output current
19	resetref	Digital Input	Reset reference memristor 0V-1.2V enable
20	CLK	Digital Input	Clock for synapse control
21	DACV<2>	Digital Input	Bit for set/form current compliance
22	DACV<1>	Digital Input	Bit for set/form current compliance
23	DACV<0>	Digital Input	Bit for set/form current compliance
24	GND	Power	Ground

circuit like in the second test structure. This test structure has one additional test points to the synapse and programming circuits, pin 4. Pin 4 is a voltage based on the combination of values on pin 21, 22, and 23. This circuit has additional memristor control signals for the memristor reference circuit. Pins 5, 6, 17, and 19 are for the memristor reference and are used equivalently to pins 7, 8, 9, and 10 which is for the synapse memristor. To activate the reference generation circuit, pin 17 must be 1.2 V. When pin 17 is enabled, pins 14, 15 and 18 are valid. The voltages on these pins are determined by the reference memristor's resistance. They setup the references for the synapse such that when the memristors have an equal resistance the synapse will give no output current. There should be no output current when both memristor's are programmed to the same value and pin 13 is at 0.6 V. The voltages at pins 14, 15, and 18 should all be relatively close to 0.6 V when the reference memristor is in a low resistance state and pin 17 is enabled. Beyond testing operating point, this circuit should enable and run at a 1 MHz clock frequency. The clock signal on chip only controls the output enable and correlating signals must be handled off chip. Since the reference uses feedback signals, the output needs to be checked for ringing and oscillation upon enabling and disabling pin 17. Determine when a valid current output on pin 13 occurs after enabling pin 19 and followed by pin 7. Varying both the synapse memristor resistance and the reference memristor resistance to determine all possible output currents.

B.1.4 Fourth Test Structures

The fourth and final test structure seen in Figure B.5 consists of two memristors to create one system. Table B.4 shows the pin location and type for this test structure.It is approximately 130 µm by 55 µm. This test structure has one memristor used in a synapse with STDP control, Figures 4.11. The control circuits for the synapse are used equivalently to the signals on the previous test structures. Pins 8, 9, and 10 are for the memristor manual switching operations, set, reset, and form. Pin 7 is the equivalent to read. To enable an STDP event, pin 1 needs to be enabled. When this pin is enabled and both pins

Table B.4: Pad frame for test structure 4 pad names.

Pin Number	Pin Name	Pin Type	Pin Description
1	Learn	Digital Input	STDP learning control 0V-1.2V enable
2	Vst	Analog Reference	Set voltage applied across memristors
3	Vrst	Analog Reference	Reset voltage applied across memristors
4	SYNOUT	Analog Output	Clocked synapse output
5	Vcc	Analog Output	Current control node
6	Vncm	Analog Output	Reference for synapse output current
7	Pre	Digital Input	Read memristor 0V-1.2V enable
8	reset	Digital Input	Reset memristor 0V-1.2V enable
9	set	Digital Input	Set memristor 0V-1.2V enable
10	form	Digital Input	Form memristor 0V-1.2V enable
11	VDD1v2	Power	1.2V power
12	VDD3v3	Power	3.3V power
13	Mbot	Analog Output	Memristor bottom electrode
14	Prog	Analog Output	Voltage generated from DACV values
15	readref	Analog Reference	Reference voltage determining read current compliance
16	Vlearn	Analog Output	Summed voltage for update programming
17	PGR	Analog Output	Voltage node after memristor cell
18	Mtop	Analog Output	Memristor top electrode
19	Post	Digital Input	Reset reference memristor 0V-1.2V enable
20	CLK	Digital Input	Clock for synapse control
21	DACV<2>	Digital Input	Bit for set/form current compliance
22	DACV<1>	Digital Input	Bit for set/form current compliance
23	DACV<0>	Digital Input	Bit for set/form current compliance
24	GND	Power	Ground

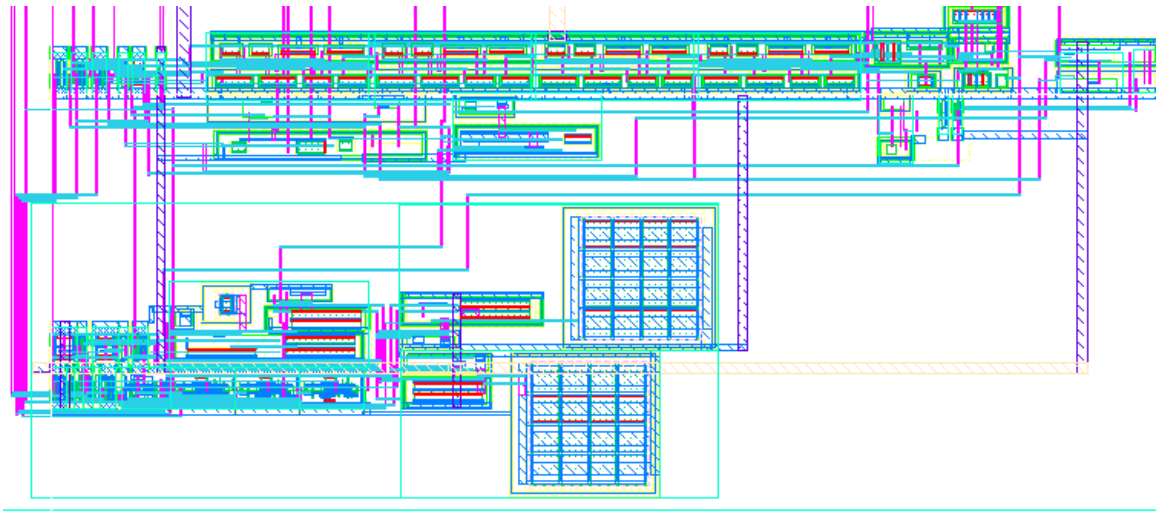

Figure B.5: Test structure 4 circuit layout image.

7 and 19 have been enabled an STDP learning event will occur. During this event, each operation on the memristor needs a clock cycle to activate. During the last of the three clock cycles pin 16 is valid. The voltage at this pin should correspond to the resulting memristor's resistance. To test this circuit enable pin 1 and vary the time between enabling pin 7 and pin 19. This should be timed with a corresponding clock running. Validate the change in pin 16 with different time differences and the resulting memristor resistance.

Appendix C

Verilog-A Code for Memristor and Neuron Model

The complete verilog-a codes described in Chapters 3 and 5 for the memristor and neuron are presented.

C.1 Memristor Model

```
// VerilogA for memristor model

`include "constants.vams"
`include "disciplines.vams"

module Memr_model(p,n);

  inout          p;         //positive pin
  inout          n;         //negative pin
  electrical     p, n;

//memristor parameters
```

```
13
14    parameter real Vtp           =  0.75;
15    // positive threshold voltage
16    parameter real Vtn           = -0.75;
17    // negative threshold voltage
18    parameter real tsw_p         =  1e-8;
19    // time to switch under +V bias
20    parameter real tsw_n         =  1e-6;
21    // time to switch under -V bias
22    parameter real delR          =  40e3;
23    // resistance change rate
24
25  // max/min LRS/HRS values
26    parameter real HRS_max = 1e6; //maximum HRS
27    parameter real LRS_min = 100; //minimum LRS
28
29    //forming parameters
30    parameter real form  = 0; //Memristor model include forming = 1
31    parameter real Rf    = 1M; //Pre forming resistance
32    parameter real Rinit = 5k; //Initial formed resistance
33    parameter real Vf    = 2.1;
34    parameter real Vfl   = 1.2;
35    parameter real Tf    = 2m;
36
37
38    // local variables
39    real td;             // simulation time step
40    real Rm;               // memristance
41    real Rm_tmp;           // temp memristance variable
42    real time_last;        // previous simulation time reading
```

```
43    real Vwr;              // input voltage
44    real Iwr;              // input current
45    real Vtpl;             // negative threshold voltage
46    real Itnl;             // end reset process
47    real dRm;              // resistance change rate
48
49  // forming variables
50    real f;
51    real tmp = 0;
52    real c = 0;
53    real formend = 0;
54
55  // Set and Reset variables
56
57          real Setstart = 0;
58          real Resetstart = 0;
59
60  analog begin
61
62    @ ( initial_step or initial_step("dc") ) begin
63      td              = 0;
64      time_last         = 0;
65          f                         = form;
66          tmp                       = 0;
67          formend           = 0;
68          if (f == 0)
69                  Rm   = Rinit;
70          else
71                  Rm   = Rf;
72          Rm_tmp = Rm;
```

```
73    end
74
75    td        = $abstime - time_last;
76    time_last   = $abstime;
77    Vwr         = V(p,n);
78    Iwr             = Vwr/Rm;
79    dRmp        = delR/tsw_p;
80    dRmn        = delR/tsw_n;
81
82  // Vwr, voltage across device
83  // Vf, forming voltage to initiate the forming process
84  // Vfl, forming voltage threshold to end forming process
85  // Tf, forming delay, this indicates how long a high
86  //     voltage has to be applied to the device before
87  //     beginning the forming process
88  // c, holds timing data for Tf check
89  // tmp, flag to begin forming process
90  // formend, flag to end forming process
91  // Rm, current device resistance
92  // Rm_tmp, fufture device resistance
93
94
95
96    if (f == 1) begin
97    // The variable f determines if the device has
98    // undergone the forming process
99          if (Vwr < Vf && formend == 0) begin
100         // Check if the device has received enough voltage
101         // to initiate the forming process
102                 c = 0;
```

```
                tmp = 0;
        end
        else if (Vwr >= Vf && formend == 0)begin
        // Check if the high voltage has been applied
        // for long enough to initiate forming process
                c = c+ td ;
                if (c >= Tf)begin
                    tmp = 1;
                end
        end
        if (Vwr >= Vfl  && tmp == 1) begin
        // Checking for the conditions to start and continue
        // lowering resistance
                Rm = Rm -  td * dRmp;
                // linear decrease in resistance
                if (Rm_tmp < LRS_min) begin
                        Rm_tmp = LRS_min;
                        f = 0;
                end
                formend = 1;
        end else begin
                Rm_tmp = Rm;
        end
        if (formend == 1 && Vwr < Vfl) begin
        // Exit condition for forming process
                        f = 0;
                        Rm_tmp = Rm;
        end
end

```

```
// Functions to define set and reset process end
Vtpl = -0.0001*Rm + 1;
Itnl = 8e-11*Rm-2.19e-5;

// Switching processes
  if (f == 0) begin
                if (Vwr >= Vtp && Rm > LRS_min) begin
                        Setstart = 1;
                end
                if (Vwr > Vtpll && Setstart == 1) begin
            Rm = Rm -  td * dRmp;
            if (Rm_tmp <= LRS_min) begin
                                Rm_tmp = LRS_min;
                                Setstart = 0;
                        end
                end
                else
                        Setstart = 0;

                if (Vwr <= Vtn && Rm < HRS_max ) begin
                        Resetstart = 1;
                end

                if (Iwr <= Itnll && Resetstart == 1) begin
                        Rm = Rm +  td * dRmn;
                        if (Rm_tmp >= HRS_max) begin
                                Rm_tmp = HRS_max;
                                Resetstart = 0;
                        end
                end
```

```
                else
                        Resetstart = 0;

        end


  Rm = Rm_tmp;

  I(p,n) <+ Vwr / Rm;
end          // end analog


endmodule
```

C.2 Neuron Model

```
// VerilogA for neuron

`include "constants.vams"
`include "disciplines.vams"

module Neuron(Vin, Vout, Vmread);
        inout Vin;
        output Vout, Vmread;

electrical Vin, Vout, Vmread;

parameter real Vs = 1.2;
```

```
13 parameter real Vr = 600e-3;
14 parameter real Tn = 1e-8;
15 parameter real Ts = 1e-6;
16 parameter real Cmem = 1e-9;
17 parameter real Vthr = 550e-3;
18 parameter real Imin = 1e-12;
19 parameter real Vmid = 600e-3;
20
21 real Vmem;
22 real Vmemnew;
23 real tsamp;
24 real Vset;
25 real nowtime;
26 real startacc;
27 real ttest;
28
29 //branch (Vin) insamp;
30
31 analog begin
32
33         @ ( initial_step or initial_step("dc") ) begin
34
35                 Vmem = Vr;
36                 tsamp = 0;
37                 Vset = 0;
38                 nowtime = 0;
39                 startacc = 0;
40                 ttest = 0;
41
42         end
```

```
43
44 // accumulate input current upon crossing input current threshold
45
46         if (startacc == 0) begin
47                 Vmemnew = Vmem;
48         end
49
50         @(cross ( $abstime -( nowtime + Tn), +1)) begin
51                 if (ttest == 1) begin
52                  startacc = 1;
53                 end
54         end
55
56         if ( startacc == 1) begin
57                 Vmemnew = Vmem - I(Vin)*1e5;
58         end
59
60         if ( Vset == Vs ) begin
61                 Vmemnew = Vmem;
62         end
63
64         @(cross(I(Vin)-Imin, +1)) begin
65                 nowtime = $abstime;
66                 ttest = 1;
67         end
68
69
70         @(cross(I(Vin)-Imin, -1)) begin
71                 startacc = 0;
72                 ttest = 0;
```

```
73          end
74  //Output voltage spike upon crossing threshold
75
76          @ (cross(Vmem - Vthr, -1)) begin
77                  Vset = Vs;
78                  Vmemnew = Vr;
79                  tsamp = $abstime;
80
81           end
82
83          @ (cross($abstime - tsamp - Ts, +1)) begin
84                  Vset = 0;
85          end
86
87                  Vmem = Vmemnew;
88
89                  V(Vout) <+ transition(Vset);
90                  V(Vmread) <+ (Vmem);
91                  V(Vin) <+ (Vmid);
92  end
93
94  endmodule
```

Vita

Ryan Weiss is originally from Huntington Beach, California. He graduated from Father Ryan High School in Nashville Tennessee in 2012 and began his studies at the University of Tennessee, Knoxville in the pursuit of a Bachelor of Science degree in Electrical Engineering. He graduated cum laude, with his Bachelor of Science degree in Electrical Engineering in May of 2016.