Spring 4-5-2017

A LEAKY INTEGRATE-AND-FIRE NEURON WITH ADJUSTABLE REFRACTORY PERIOD AND SPIKE FREQUENCY ADAPTATION

Jacob N. Healy
University of New Mexico

Jacob N. Healy

Candidate

Electrical and Computer Engineering

Department

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

Approved by the Thesis Committee:

Payman Zarkesh-Ha , Chairperson

Thomas Caudell

Jim Plusquellic

A LEAKY INTEGRATE-AND-FIRE NEURON WITH ADJUSTABLE REFRACTORY PERIOD AND SPIKE FREQUENCY ADAPTATION

by

JACOB N. HEALY

B.A. IN ENGLISH LANGUAGE AND LITERATURE, FORT LEWIS COLLEGE, DURANGO, CO, 2008

M.A. IN ENGLISH LANGUAGE AND LITERATURE, UNIVERSITY OF NEW MEXICO, ALBUQUERQUE, NM, 2011

M.S. IN COMPUTER ENGINEERING, UNIVERSITY OF NEW MEXICO, ALBUQUERQUE, NM, MAY 2017

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Computer Engineering

The University of New Mexico
Albuquerque, New Mexico

May, 2017

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Payman Zarkesh-Ha for is guidance and support throughout the research and writing of this thesis, my committee members, Dr. Thomas Caudell and Dr. Jim Plusquellic for their service and critique of this thesis, and Javad Ghasemi for his assistance and advice regarding the circuit layout.

A LEAKY INTEGRATE-AND-FIRE NEURON WITH ADJUSTABLE REFRACTORY PERIOD AND SPIKE FREQUENCY ADAPTATION

by
JACOB N. HEALY
B.A. IN ENGLISH LANGUAGE AND LITERATURE, FORT LEWIS COLLEGE, DURANGO, CO, 2008
M.A. IN ENGLISH LANGUAGE AND LITERATURE, UNIVERSITY OF NEW MEXICO, ALBUQUERQUE, NM, 2011
M.S. IN COMPUTER ENGINEERING, UNIVERSITY OF NEW MEXICO, ALBUQUERQUE, NM, MAY 2017

ABSTRACT

As standard CMOS technology approaches its physical limitations there is increased motivation to explore new computing paradigms. One possible path forward is to develop an array of computational architectures which specialize in distinct tasks. Neural computing architectures excel at pattern recognition and processing low-fidelity sensory input, but one of the biggest challenges in the field has been implementing architectures which strike an appropriate balance between biologically-plausible performance and the simplicity needed to make large neural systems practical. This work proposes a new VLSI neural architecture which seeks to provide such a balance.

The design described here builds on an implementation first proposed by van Schaik. Van Schaik's circuit has the advantage of simplicity. It uses a Leaky-Integrate-and-Fire model while offering some biologically analogous behavior and maintaining a very compact layout profile. However, the circuit lacks the ability to emulate certain desirable biologically inspired features, most notably spike frequency adaptation (SFA).

The circuit depicted receives a current stimulus as its input. If the current is greater than the neuron's leakage current, then it charges a capacitor which drives a comparator circuit. When the voltage on the capacitor exceeds the threshold voltage a

spike is generated. The design makes use of four parametric inputs to tune its behavior and also contains circuitry for a tunable refractory period and SFA.

Rather than operate in biological time, the circuit operates in accelerated time with a spike frequency in the nano-second region. This allows smaller capacitors to be used and reduces the overall layout area. The circuit layout was created using Tanner EDA's L-Edit software and designed for fabrication with a 180nm technology node. It occupies 386.497μm^2. The circuit was extracted and simulated using Tanner Tools T-Spice. Simulations show an average power consumption in the micro-Watt range.

Table of Contents

List of Figures

List of Tables

Chapter I: Introduction

Spiking Artificial Neural Networks (ANN) are networks designed in either hardware or software and attempt to emulate the structure and functionality of networks of biological neurons. Neural cells operate by converting input stimulus into a spike pattern (a pulse train) in which the information they convey about the input is encoded into the frequency of pulses generated at the output. Circuits modeled on neural behavior are built around neural models which recreate the behavior of biological neurons with varying degrees of accuracy. Figure 1 presents a basic block diagram of this principle at a high level of abstraction. Hardware circuits generally process input currents and convert the stimulus into a spike train at the output.

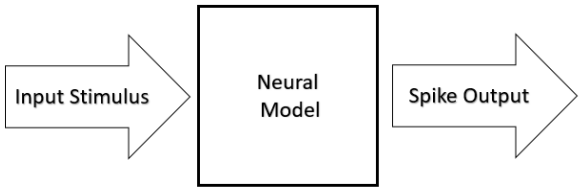

Figure 1: Basic Neural Block Diagram. Neural circuits are built around neural models which accept an input stimulus, typically in the form of a current, and output a voltage spike.

Figure 2 depicts the waveform containing the basic characteristics of a typical neural circuit. Such circuits are characterized by an integration period, during which the potential on the neuron builds. This potential reflects the behavior of a neural cell's membrane which can be characterized as having a resting ionic potential characterized by the cell's biochemical processes. When an action occurs that causes the neuron to start receiving stimulation this potential will continue to increase until it reaches a threshold, at which point, biochemical processes cause the potential to increase more rapidly (spike).

This spiking behavior is followed by a reset in which the potential is returned to its resting state. The time it takes to trigger the reset action defines the spike width. Finally, there is a period of time after spiking, the refractory period, during which the neuron is unable to spike and remains at its resting potential. Information is encoded into the spike pattern generated by the neuron. Circuits which attempt to emulate this behavior can be designed to implement a range of the biological complexes involved in the spiking process. Some models attempt to account for as many of the biochemical processes

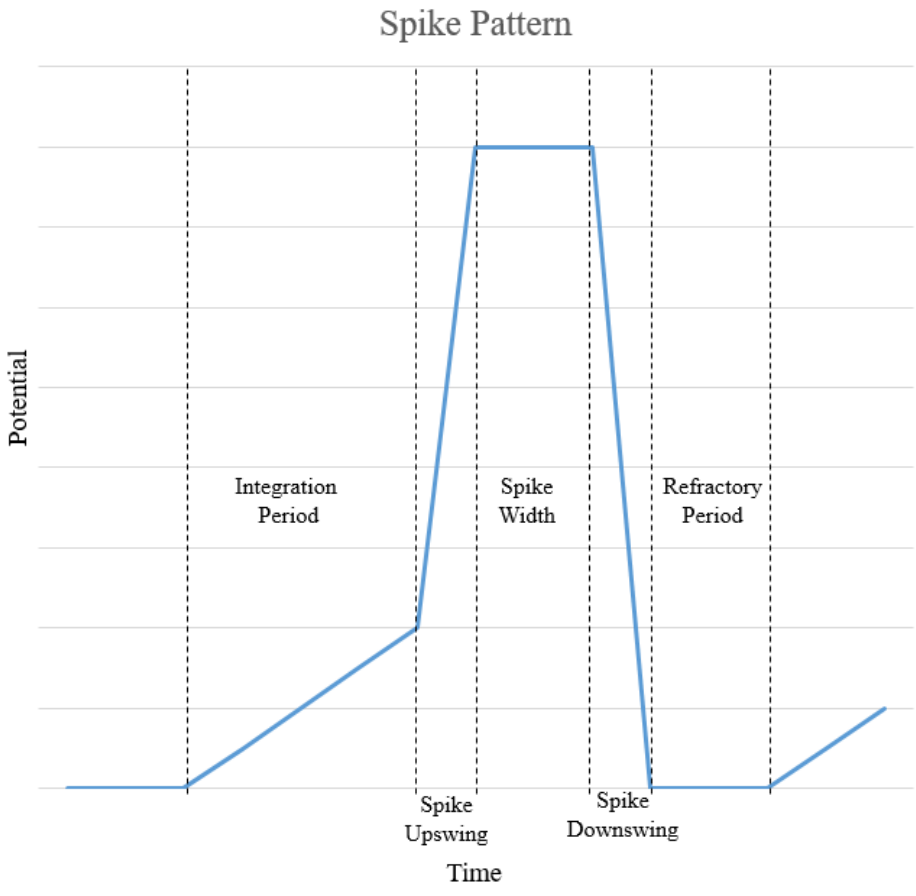

Figure 2: Basic Neural Spiking Pattern. The spiking characteristics of a neural circuit are defined by the integration period, spike width, and refractory period as shown above. The spike width is typically very small and is exaggerated here for illustration purposes.

involved in actual neural cells as are currently understood, while other models abstract the inner-working of the neural cell and attempt only to recreate the spiking behavior.

Developing computational paradigms based on spiking neural structures has the potential to offer tremendous benefits over the traditional von Neumann computing model. The human brain, for example, contains on the order of 100 billion neurons and has a total processing rate that is on the order of 10^{16} bits per second [1], a rate far surpassing that found in modern computers. Furthermore, neural networks have the potential to excel at tasks that have traditionally been considered difficult for computers to process and therefore are ideal for purpose built computing applications [2]. Such applications include rapid pattern recognition, low-fidelity real-time processing of sensory input, and artificial intelligence applications. While the potential benefits to be derived from ANNs are great, their implementation faces a number of challenges. Simulating neural structures in software comes the cost of speed, and while hardware implementations tend to be faster, chip area and power consumption become concerns.

The field of neuromorphic engineering aims to tackle the problems associated with implementing neural networks in hardware. It is a field that is becoming increasingly important as traditional von Neumann architectures begin to reach their theoretical performance limits [3]. Hardware implementations of neural networks offer the potential to approach computing in new ways; thus enabling problems which were not traditionally considered easily computable to be approached [4]. Neural networks seek to use biologically inspired models to create architectures for solving problems utilizing imprecise logic.

However, the field brings with it a wide array of challenges not present in conventional computing paradigms. The complexity of the circuitry necessary to simulate complex neural networks make cost effective implementations challenging [5]. Hardware must strike a balance between accurately modeling neural functions and being simple enough to implement in a compact layout. Additionally, the advantages and disadvantages of using digital or analog implementations of these circuits must be weighed. While analog implementations offer higher speed and lower power performance they are more susceptible to noise [5]. While these challenges must be considered when designing the hardware for implementing a neural network, they can be overcome or at least minimized through careful circuit optimization.

There have been a wide range of approaches to neuromorphic designs ranging from analog to digital implementations. For example, Liu et al. and Cruz-Albrecht et al. both propose the use of analog memristive synaptic systems that make use of a digital processing core in [6] and [7] respectively and Cassidy et all discuss the successes of IBM's all digital design, named TrueNorth in [8].

In addition to the mode of implementation used, architectures which enable these circuits to be used in meaningful ways are also necessary. To this end, Roy et al. describe a method for configuring a reservoir of Leaky Integrate and Fire neurons into a soft-Winner-Take-All (sWTA) network which can be used for pattern recognition in [9]. When coupled with Spike-Time-Dependent-Plasticity (STDP) mechanisms such Spike-Frequency-Adaptation (SFA), a feature found in many biological neural systems which is believed to play a role in cognitive processing, sWTA's could potentially be used to emulate the kind of cognitive processing found in biological brains [2]. SFA allows the

neural circuit to reduce the firing rate of a neuron experiencing constant stimulation and has the potential to be useful in encoding neural information and producing additional behaviors. At the center of all of these designs is the neural model that is used to emulate the functionality of a biological neuron.

1.1 Neural Models

Biological neurons transmit signals using complex chemical processes in which the release of neurotransmitters modulates the electrical potential of individual neurons [5]. When looking at the spiking neuron as a core building block of ANNs, at its most basic it can be modeled by a comparator circuit that compares an input voltage to a pre-defined threshold and if the input is over the threshold, it generates a voltage spike as output (i.e. a voltage pulse with a fixed pulse width is generated). As long as the input voltage remains above the threshold, the circuit will continue spiking. In biological systems these spikes typically have a frequency on the order of milliseconds. Many designs maintain this firing rate in order to mimic biological neurons as closely as possible, though some proposed circuits operate in accelerated time. There are a number of approaches to modeling neurons that attempt to replicate this spiking behavior with varying degrees of biological accuracy. The most common are the Hodgkin-Huxley model, the Izhikevich Model, and the Leaky Integrate and Fire model. Below is a brief summary of each of these models and their potential uses.

Hodgkin-Huxley Model

The Hodgkin-Huxley neuron model is one of the most biologically accurate models. It attempts to emulate as many of the biochemical processes that take place in real

neurons as possible and typically has on the order of 30 tunable parameters to help realize this complexity. As such, it allows researchers to construct very detailed models of brain structures in hardware, thus providing the means to simulate real brain function with a plausible degree of accuracy. Such detailed simulations provide the potential to develop insight into how brains work and could provide breakthroughs in artificial intelligence applications.

Because of the biological accuracy of this model it is most commonly used to conduct medical research. For example, Hsin Chen et al. have used this model to test neural variability with the intent of developing a model for exploring the causes of diseases such as Parkinson's [10], and Chuanxin M. Niu et al. have proposed using this model to explore an array of diseases which impact motor function [11]. Other projects have sought to add to the understanding of how the brain works through simulation [1]. However, the accuracy of this model comes at the cost of high power consumption and large chip area [12].

Izhikevich Model

The Izhikevich model represents a compromise between maintaining a reasonable degree of biologically inspired functionality, while being more compact and power efficient than the Hodgkin-Huxley model. Izhikevich model neurons typically have roughly half the number of tunable parameters as the Hodgkin-Huxley model does. This simplifies the circuitry while still allowing enough biological realism to do biological research with these circuits. Nazari et al. have implemented an Izhikevich model based neural network for the purpose of using the circuit to understanding neuron-astrocyte

interactions, for example [13]. Their implementation uses digital logic to construct the neural model, which can be configured to accurately model the effects of sodium and potassium receptors in actual biochemical processes. For a complete description of their circuit design see [13].

Others have implemented Izhikevich model circuits for computational purposes. Wijekoon and Dudek have proposed such a circuit for the purpose of controlling robotic locomotion. Their circuit focuses on emulating dopamine receptors in silicon. They have variations of the circuit which operate in biological time and in accelerated time, lowering the power consumption, as described in [14] and [15] respectively.

Leaky Integrate and Fire Model

The leaky integrate and fire (LIF) model had been the most widely adopted for computational applications. This is largely due to its simplicity: Mead originally proposed the Axon-Hillock circuit which uses a pair of inverters and an amplifier to implement this model [16]. A basic Axon-Hillock circuit, which uses capacitive feedback to modulate the output is depicted in Figure 3.

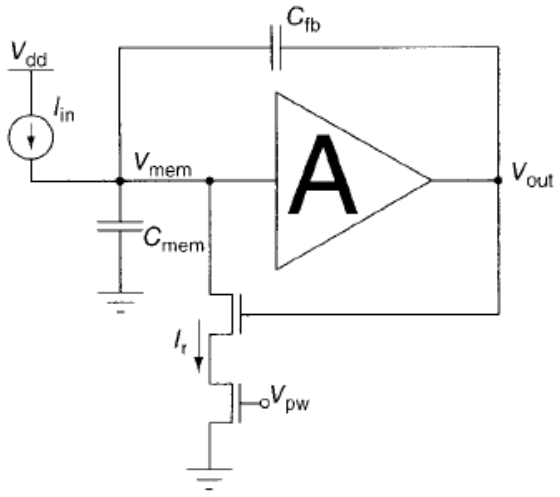

Figure 3: Axon-Hillock Neural Circuit. This simple implementation of the LIF model uses an amplifier and capacitive feedback to generate spikes. [24]

The LIF model emulates the basic functionality of biological neurons (the spiking behavior), often without accounting for the biological principles that cause spiking to

occur in biological systems. As a result, it represents a good abstraction of neural behavior but does not provide the precise accuracy necessary to model real biological systems.

A number of variations on this model have been proposed for use in a wide array of applications. Shapero et al. described an architecture for using LIF neural circuits to do sparse approximation in [17] and Cassidy et al. have proposed using abstracted digital arithmetic logic units to model LIF neurons for the purposes of implementing learning circuits [18]. Because of the ease of implementing LIF neural circuits, a large amount of recent innovation has occurred using this model. Versions of LIF circuits which attempt to reintroduce more biological features can be found in [19], [20], [21], and [22]. These designs all incorporate circuitry that mimics more complex behaviors than are typically found in LIF neural models such as bursting behavior in which the neuron produces high frequency bursts of spikes in response to high levels of excitation and SFA, in which the neuron adapts to periods of prolonged excitation by reducing its spike frequency. These modifications to the LIF model allow for the design of neuromorphic circuits that leverage the learning capability of more complex models while still being a compact and relatively simple to implement.

1.2 Neuron Implementation

Silicon neurons have been proposed using both analog and digital implementations. Ease of implementation is a primary concern when comparing these two approaches. FPGAs provide a rapid means of prototyping and testing neural networks. As such they offer an advantage in that complex networks can be rapidly constructed and tested. This

approach makes sense for research applications such as those presented in [13]. However, FPGA implementations come at a high cost in terms of space and power. This is true for digital implementations in general as well.

Digital Implementations

The synchronous nature of digital circuitry allows these implementations to be more precise. This makes them scale to large networks more readily and some digital implementations have even overcome the issue of space efficiency by designing the neuron to function in accelerated time. By making the neuron function on a smaller timescale, a single circuit can be reused many times over to simulate multiple neurons functioning on a biological timescale as is done in [18].

Analog Implementations

Analog implementations, on the other hand, offer the potential to be much more power efficient and tend to consume less area on chip [2]. Analog circuits can respond to changes in input parameters in real-time thereby allowing biological neural behaviors to be modeled more realistically and they tend to be more power efficient. The difficulties in accurately synchronizing the signals from a large array of neurons can be overcome by using asynchronous communications protocols such as Address Event Representation (AER) [23].

1.3 Summary

While the principle behind the neural circuit is quite simple, designs that adequately recreate the behavior of biological neurons can quickly grow complex. While it is tempting to design a silicon neuron that will emulate every aspect of a biological

neuron's electro-chemical processes as accurately as possible, such designs are not feasible for neural networks intended to perform computational tasks because the complexity required allocating large amounts of area on the chip to a single neuron, such designs require too much power to operate, and because not enough is known about the functionality of biological neurons to perfectly recreate their function in silicon.

The design proposed here works from the opposite end of the spectrum: taking the simplest possible LIF circuit model and building functionality on top of it to improve power efficiency and to introduce a degree of plasticity that makes self-learning networks possible with the design. This work proposes an analog neuron with SFA and a tunable refractory period for its spiking behavior. While digital designs offer more precision and are easier to synchronize, the analog design allows for rapid responses to input stimulus, simpler, more compact design, and a more biologically realistic behavior.

Rather than operate in biological time, the circuit operates in accelerated time with a spike frequency in the nano-second region. This allows smaller capacitors to be used and reduces the overall layout area. The layout was created using Tanner Tools L-Edit software and designed for fabrication with a 180nm technology node. It occupies 386.497^2. The circuit was extracted and simulated using Tanner Tools T-Spice. Simulations show an average power consumption that ranges from 5-200μW when properly configured. Actual average power varies depending on the configuration of the neuron's spiking behavior.

1.4 Organization

Chapter 2 of this paper will provide a detailed review of the most relevant literature to this work, examining the designs and approaches that influenced the proposed design. Chapter 3 will then describe the design and functionality of the proposed circuit and provide a detailed description of the circuit's inputs and outputs as well as its layout. Chapter 4 provides the simulation results for the designed circuit and finally, Chapter 5 offers conclusions about the design's functionality and offers suggestions for future work related to this circuit.

Chapter 2: Overview of the Literature

As is noted in [24], the fundamental difference between spike-event generation in a biological neuron and a silicon neuron is that biological spiking is a smooth-curved continuous event whereas most silicon implementations model this behavior as a more discrete process. Mead's original Axon-Hillock circuit [16], reflects this distinction. In the Axon-Hillock circuit, an amplifier is used to generate spike events. An input current is used to charge a capacitor, which represents the neural circuits membrane capacitance, until the switching threshold is reached and the output moves to VDD. Once a spike is generated, a feedback circuit is used to discharge the membrane capacitor and cause the amplifier to switch back to ground. In its most straight forward implementation this circuit uses a basic two-inverter amplifier and the neuron's threshold voltage is entirely dependent on the switching characteristics of the transistors being used to implement it. The circuit implemented in this work builds off of the basic principles presented in the Axon-Hillock circuit and the following works discussed in this section represent modifications to it implementation.

2.1 The Octopus Retina

In 2003, Culurciello et al. proposed a neural circuit that builds off the Axon-Hillock design and behaves similarly to an octopus's retinal neuron. The circuit was designed to be used to process sensor data. In "A biomorphic digital image sensor," Culurciello et al. note that biologically inspired retinal circuits offer the potential to parallelize the processing of pixels in digital image sensors and to process the information much more rapidly than conventional circuitry allows [25].

The circuit leverages an AER protocol to request access to the bus when a spike event generator circuit has reached its spiking threshold [25]. In this case the spike event generator is simply a pair of coupled inverters used to amplify the input signal. Capacitive feedback from the output of the amplifier is used to accelerate the transition from low to high, reducing the switching time and therefore conserving power as depicted in Figure 4. The spike event generator proposed Culurciello et al. is dependent on the CMOS device design, having a threshold voltage for spiking that is equal to that of the threshold voltage of the inverter used in the circuit.

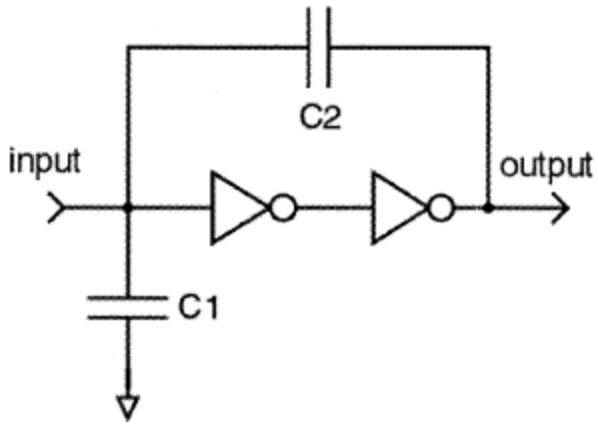

Figure 4: Culurciello et al.'s Spike Event Generator. Note that generation of a spike is dependent on the device characteristics of the inverters [25].

2.2 Electronic Spiking Neurons

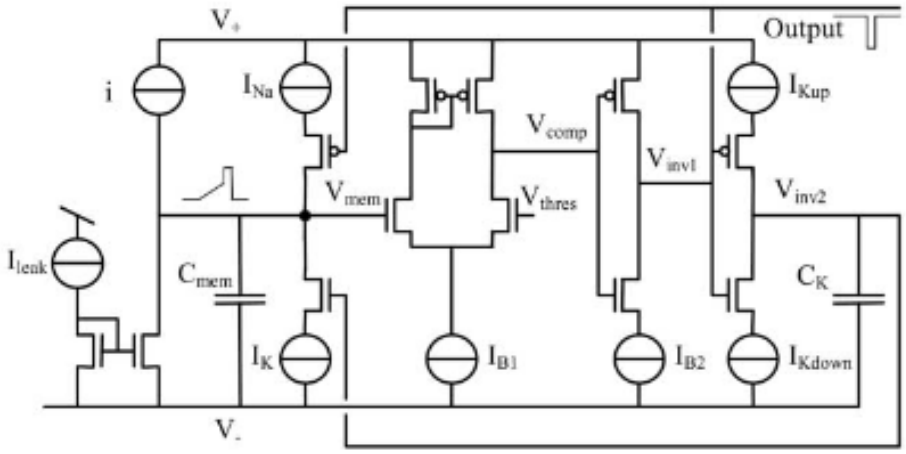

Figure 5: Neural Circuit Proposed by van Scaik et al. Spiking behavior in the circuit is largely governed by competing current sources which bias its switching behavior [26].

In 2001, van Scaik et al. proposed a neural circuit which uses a similar principle to Culurciello et al.'s octopus retina circuit but introduces a transconductance amplifier, allowing the spiking threshold to be defined dynamically. The circuit was designed to model neural behavior for both inter-neural communication and for communication between external events and neurons [26]. The primary goal of the research presented in [26] was to propose circuits which provide building blocks for exploring the capabilities of neural computing. The central focus of this exploration rests in examining how spike events convey computationally significant meaning. Van Scaik et al. conclude that such meaning primarily rests in the timing of the events generated rather than in the spikes themselves and their circuit includes features which attempt to emulate how biological neurons govern the timing of spike events [26].

The circuit models a neuron's membrane capacitance and leakage current using the capacitor, C_{mem} and current I_{mem} shown in Figure 5. The neuron is largely modeled after biological neurons: the current sources I_k and I_{Na} are used to model the effects of sodium

and potassium channels in biological neurons and impact the rate at which the spike is generated and reset [26].

The design described above was implemented by van Scaik et al in 1µm technology with 10pf capacitors and it operates with a spike frequency on the order of milliseconds and a spike width on the order of microseconds. Van Scaik et al. note that one of the circuit's key advantages is that its all-analog implementation allows changes to the parameters which govern the circuit's behavior to affect the neuron in real time.

2.3 Neural Circuits for Cognitive Processing

Chicca et al. propose a set of circuits which can be used to model neural interactions for the purpose of exploring their cognitive processing capabilities in [27]. They begin by noting that one of the fundamental challenges in neuromorphic engineering is the need to develop autonomous, cognitive systems. Their goal is to explore circuitry which can be used to implement autonomous learning structures.

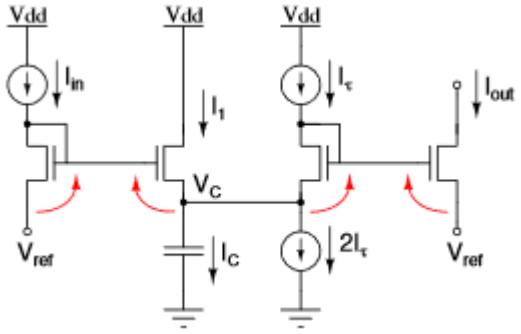

Figure 6: Tau-Cell circuit. The circuit allows the neural dynamics of membrane conductance to be effeciently modeled using biolgically plausable time constants. [27]

Chicca et al. use a Tau-Cell circuit, first proposed by van Scaik and Jin in [28] and depicted in Figure 6 to mimic the behavior of membrane conductance. This circuit allows for highly realistic modeling of neural behavior but comes at the cost of size efficiency and complexity. In order to accurately model the differential equations that govern the circuit's behavior the transistors in the circuit must be sized, and additionally, the two

current sources must be matched with a high degree of precision for the circuit to behave correctly [27].

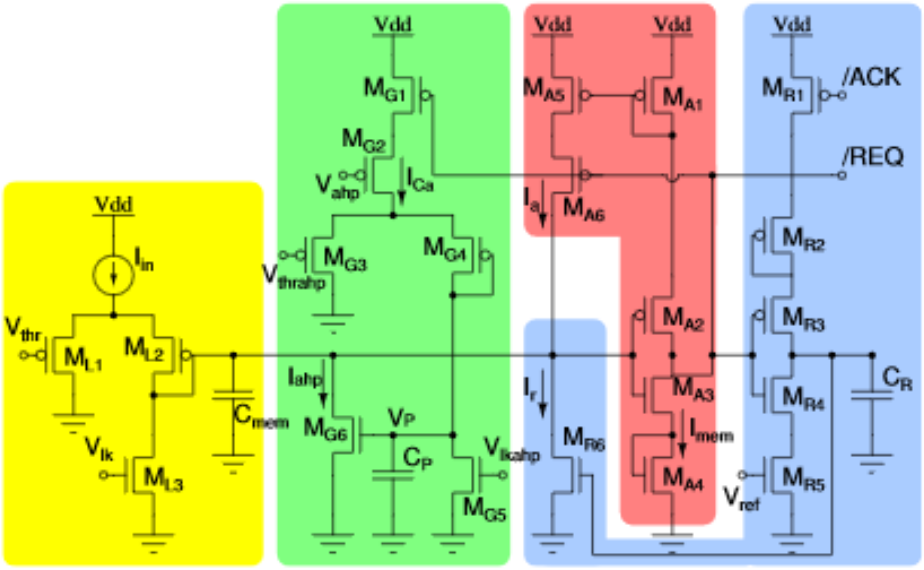

Figure 7: Adaptive Exponential Integrate and Fire Circuit. From left to right: a Tau-Cell circuit models the membrane capacitance, feedback from the output provides SFA functionality, the integrated input current triggers an inverter which is responsible for spike event generation, feedback from the spike event is used to trigger the refractory period. [27].

Chicca et al.'s proposed circuit is depicted in Figure 7. They note that the circuit was designed as a compromise between circuit complexity and computational power. Importantly, this circuit accurately reflects key biological features such as realistic time constants and it makes use of a refractory period and spike-frequency adaptation [27]. Membrane capacitance, Sodium channel activation, and Calcium and Potassium conductance are also modeled by this circuit. The spiking threshold is dictated by the switching threshold of the inverter that governs spike event generation, but the rate at which C_{mem} charges can be modulated by the tau-cell circuit.

The paper also describes experiments designed to test spike-frequency adaptation by stimulating a silicon neuron with constant input current and measuring the membrane potential. They found they were able to tune adaptation circuits to produce bursting behavior. Chicca et al. note that the circuit was able to demonstrate this capability without having to integrate additional circuits into the design as many other models have done and attribute this to the fact that their tau-cell circuit allows for more flexibility via the added control parameters it incorporates.

An array of these neurons was implemented by Chicca et al in a sWTA network and their experimental results demonstrate its ability for selective amplification and state dependent computation [27]. The network consisted of 128 Integrate and Fire neurons all configured to excite its first nearest neighbors, second nearest neighbors, and for global inhibitory neurons.

The first sWTA experiment stimulated the network with two distinct regions of activation. In each case the neurons receiving the strongest input won and enhanced their activity by suppressing the other neurons using the global inhibitory neurons. The second experiment used a sWTA network to construct Finite State Machines. To do this two populations of neurons were stimulated subsequently. The second stimulus triggers a state transition by suppressing the first population of neurons and activating the second population. When the stimulus is removed the state remains in a self-sustained state.

Chicca et al conclude by addressing the largest problem facing their design approach: device mismatch and knowledge of the brain. Chicca et al. attempt to mitigate this problem by utilizing circuit designs which do not require precisely matched transistors. They also suggest that the AER protocol could be used to help reduce this

problem on multi-chip designs. Finally, they suggest that the largest problem facing the field of neuromorphic engineering is accurate knowledge of how biological brains actually work. Because of this they stress the need for an interdisciplinary approach to the field.

Chapter 3: Circuit Design

The circuit proposed here builds primarily on the design van Schaik proposed in [26]. It first minimizes the circuit to its most direct implementation by removing the current sources I_{Kup}, I_{Kdown}, I_{Na}, and I_k as shown in Figure 5. In van Schaik's circuit, these current sources govern the spiking behavior of the neuron and by manipulating them, the neuron's biologically inspired behaviors can be manipulated in order to produce a range of distinct spiking patterns in response to an identical input stimulus. While this behavior is interesting and useful for exploring the range of behaviors the neuron can provide, it does not impact the basic functionality of the circuit. While removing these sources from the circuit sacrifices some of the biological accuracy, it allows for a design that is more compact and reduces the number of inputs required to drive the circuit while still maintaining the circuits most important features.

3.1 Circuit Functionality

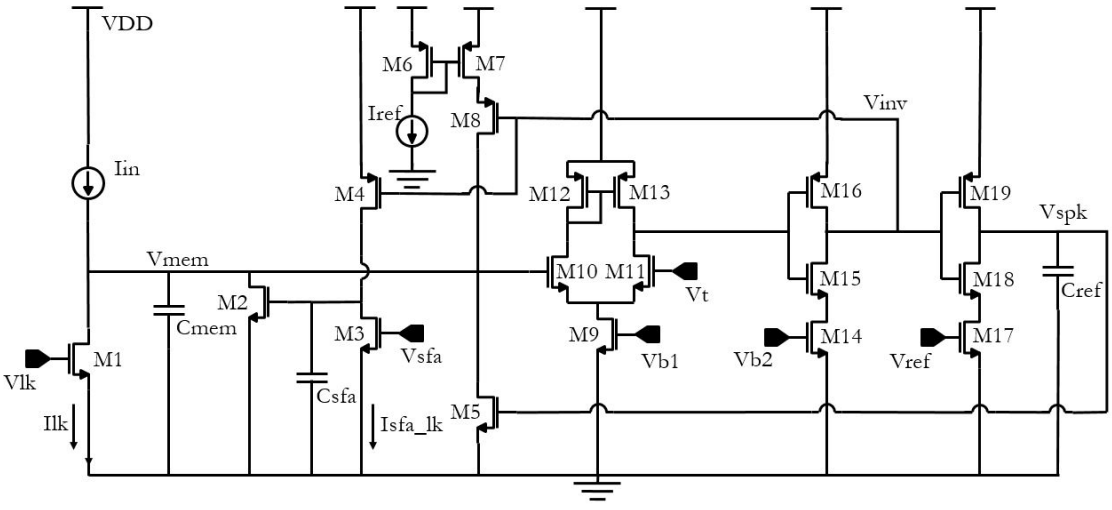

Figure 8: Neural Circuit: The proposed circuit has inputs Iin, Vlk, Vsfa, Vt, Vref, Vb1, and Vb2 which are used to control the circuit's spiking behavior and Iref which governs the current feedback to Vmem. The output is the voltage, Vspk.

The proposed circuit, as depicted in Figure 8, accepts a current source as its stimulus input. The current, I_{in} can either charge or discharge the capacitor, C_{mem}, depending on its direction. As I_{in} charges C_{mem}, it must compete with the neural circuits leakage current, represented by I_{lk}. Therefore, the current charging the capacitor, I_{mem} can be represented by:

$$I_{mem}(t) = I_{in}(t) - I_{leak}(t) \tag{1}$$

and the membrane potential, V_{mem}, can be modeled by the equation:

$$V_{mem}(t) = \frac{1}{C_{mem}} \int_{-\infty}^{\tau} I_{mem}(t)\, dt \tag{2}$$

Requiring I_{in} to compete with a leakage current reflects a degree of biological plausibility and allows the charging rate of the membrane potential despite C_{mem} being a fixed value. The entire neural circuit is composed of nineteen transistors and three capacitors. Additionally, the current source I_{in} would normally be implemented using synaptic circuits. However, for the purpose of testing the neuron's functionality, an array of current mirrors was used to provide current to the circuit.

The circuit's inputs are all described in Table 1. It has four voltage-source inputs and a current source input which govern the spiking behavior of the neuron: V_{lk}, V_{sfa}, V_t, V_{ref}, and I_{ref}. Each input drives a different portion of the circuit which can be divided into four blocks: the leakage current block, the SFA block, the refractory block, and the spike generator. Each of these blocks is described in greater detail below. The inputs, V_{b1} and V_{b2}, are used to conserve power and when configured properly do not significantly affect the behavior of the circuit.

Table 1: Circuit Inputs. This table shows all 7 of the inputs to the circuit described in this section. The operating range of each input represents the suggested values for ideal functionality.

Input	Ideal Range	Function
I_{in}	µA	Provides stimulus to membrane capacitor
I_{ref}	nA	Reference Current for refractory block
V_{lk}	V_{thresh} - VDD/2	Sets the magnitude of the leakage current discharging the membrane capacitor
V_t	V_{thresh} - VDD/2	Defines the potential at which the neuron begins firing
V_{sfa}	V_{thresh} - VDD/2	Defines the sensitivity of the SFA circuit (Higher values are less sensitive)
V_{ref}	≈VDD/2	Adjusts the spike width and the refractory period of the output
V_{b1}	V_{thresh} - VDD/2	Used to increase power efficiency
V_{b2}	V_{thresh} - VDD/2	Used to increase power efficiency

Leakage Current Block

The leakage current is implemented using a single NMOS transistor, M1 in Figure 9. A voltage source, V_{lk}, drives M1 and sets the rate at which the leakage current dissipates the charge on V_{mem}. When V_{lk} is greater than the threshold voltage for M1, I_{lk} can be described by M1's characteristic equation:

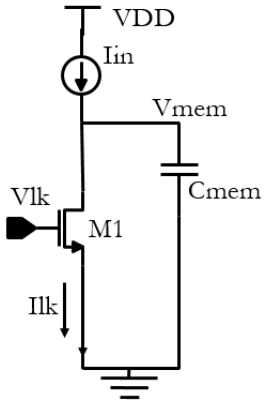

Figure 9: Leakage Current Block consisting of the Input current, and a leakage current governed by Vlk

$$I_{lk} = k_n' \frac{W}{L} \left[(V_{lk} - V_{thresh}) V_{min} - \frac{V_{min}^2}{2} \right] (1 + \lambda V_{mem}) \quad (3)$$

where V_{thresh} is the threshold voltage of the transistor M1 and V_{min} represents the minimum of V_{mem} and V_{lk}-V_{thresh}, and when V_{lk} is less than M1's threshold voltage, the current is equal to the transistor's subthreshold conduction rate:

$$I_{lk} = I_s e^{\frac{V_{lk}}{nkT/q}} \left(1 - e^{-\frac{V_{mem}}{kT/q}} \right) \quad (4)$$

where I_s and n are characteristics of the device and kT/q is the device's thermal voltage. Importantly, the leakage current through M1 will always gradually return V_{mem} to its grounded resting potential, just as a biological neuron would. Thus, any input current must compete with this leakage current to charge V_{mem}, as described in the above section and if the neuron has not received an input stimulus recently, whatever charge has built up on the membrane will gradually dissipate. As long as V_{mem} remains below the spiking threshold, V_t, I_{lk} will act as the primary source of competition for I_{in}.

Spike Frequency Adaptation Block

The SFA block is inspired by the SFA circuit proposed in [27] and consists of transistors M2-M4 and the capacitor, C_{sfa}, as depicted in Figure 10. The SFA block provides the neuron with an adaptive means of modifying its behavior according to the input it is receiving in real time. When the neuron receives input stimuli in sparse intervals, the SFA block has little to no effect on the pattern of spike generation. However, if the neuron is the recipient of a high degree of excitation, the SFA block is engaged and reduces the frequency of output spikes. This allows the neuron to adapt to its input, preventing a strong input signal from overly biasing the output.

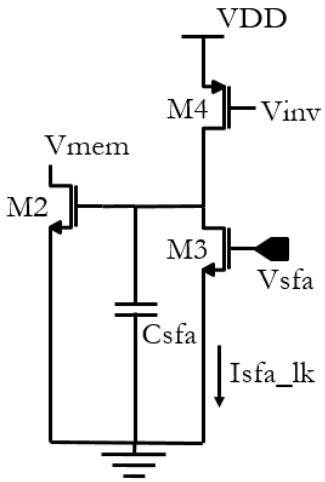

Figure 10: SFA Block. The input Vsfa governs the sensitivity of the spike frequency modulation provided by this circuit

Each time the transconductance amplifier, (M12-M13) produces a spike (generates an output of VDD), the spike travels through a pair of inverters before reaching the output

stage. The output of the signal after traveling through the first inverter, V_{inv}, is used to drive the PMOS transistor in the SFA circuit, M4. M4 governs whether or not current is provided to the SFA block. Because of the delays introduced by the inverters, this means that the SFA circuit is only powered for brief intervals of time: just after a spike has been generated, but before the spike's output signal has been produced. When V_{inv} is close to ground, M4 will be in the saturation region and the current through M4 can be described by:

$$I_{M4} = k_p' \frac{W}{L}(VDD - V_{inv} - |V_{tresh}|)^2[1 + \lambda(VDD - V_{DM3})] \quad (5)$$

which will be roughly equivalent to:

$$I_{M4} = k_p' \frac{W}{L}(VDD - |V_{tresh}|)^2[1 + \lambda(VDD - V_{DM3})] \quad (6)$$

where V_{thresh} is the threshold voltage of M4, and when V_{inv} is close to VDD (the state in which it will spend most of its time), M4 is in the cut-off region; therefore, no current will flow through M4, and the potential at M3's drain, V_{DM3}, will equilibrate to ground, provided there is enough potential driving the gate of M3 to allow this to happen.

The transistor, M3, is then used to govern the amount of current which will charge the capacitor, C_{sfa}. This means that as the potential of V_{sfa} on M3's gate increases, the amount of current charging C_{sfa} decreases. Provided transistors M3 and M4 are designed for symmetry, when V_{sfa} is equal to VDD, the current through M3 will be roughly equivalent to the current through M4 and, therefore, almost no current will be provided to C_{sfa}. When V_{sfa} is between M3's threshold voltage and VDD the current through M3 behaves similarly to the current I_{lk} in Equation 3 and can be described as:

$$I_{sfa\_lk} = k_n' \frac{W}{L} \left[(V_{sfa} - V_{thresh}) V_{min} - \frac{V_{min}^2}{2} \right] (1 + \lambda V_{DM3}) \quad (7)$$

where V_{thresh} is M3's threshold voltage and V_{min} is the minimum of $V_{sfa} - V_{thresh}$ and V_{DM3}. In this range of operation, the current that charges C_{sfa} can be described as:

$$I_{sfa} = I_{M4} - I_{sfa\_lk} \quad (8)$$

and thus, potential on C_{sfa} will increase at a rate of:

$$V_{DM3}(t) = \frac{1}{C_{sfa}} \int_{-\infty}^{\infty} I_{sfa}(t)\, dt \quad (9)$$

Finally, when M3 is in the cut-off or subthreshold region, the current I_{sfa_lk} will be very small and therefore, I_{sfa} will be approximately equal to I_{M4}.

The potential on C_{sfa}, V_{DM3} is then used to drive the gate of M2. When V_{DM3} is greater than M2's threshold voltage, a second channel is created between V_{mem} and ground. This means that the current discharging the capacitor C_{mem} becomes equal to:

$$I_{discharge} = I_{lk} + I_{M2}$$

where I_{M2} is the current through M2. When V_{DM3} is close to VDD, M2 will be in saturation and I_{M2} becomes large enough to discharge C_{mem} completely and thereby prevent the neuron from continuing to produce spikes at its output. When no spike is being produced, I_{M4} will be close to 0 and therefore, the current I_{sfa} as described in Equation 8 will become $-I_{sfa\_lk}$ and will discharge C_{sfa} at the rate described by Equation 9. Once V_{DM3} has fallen below M2's threshold voltage, the current, I_{M2} will become very small and the current charging C_{mem} will return to being roughly equal to that described in Equation 1.

Given the above description of the SFA block, this means that when the neuron is exposed to a high degree of excitation (i.e. is receiving a constant stream of input stimuli), C_{sfa} will initially take a longer period of time to charge from ground to VDD. Once this happens, the SFA block will prevent the neuron from continuing to fire until the current I_{M2} is sufficiently small to allow C_{MEM} to build a charge again and the neuron will once again fire. This time however, the potential at C_{sfa} will be just under the threshold voltage of M2 and therefore it will take less time to build a sufficient charge on C_{sfa} to once again cause I_{M2} to prevent the neuron from being able to fire.

The input voltage, V_{sfa} governs the sensitivity of the SFA block. As a result, if V_{sfa} is very small, C_{sfa} will both charge rapidly and discharge slowly and the neuron will only spike initially and then will be prevented from continuing to produce spiking output. If, on the other hand, V_{sfa} is very large, C_{sfa} will be unable to develop a sufficient charge to engage I_{M2} and the circuit will behave as it would if the SFA block had not been implemented.

Refractory Block

The refractory block consists of transistors M5-M8 and the capacitor, C_{ref} as depicted in Figure 11. The input V_{ref}, which drives M17 (seen in Figure 8) and controls the discharge rate of C_{ref} also impacts the functionality of this block. M6 and M7 are implemented as a simple current mirror that governs the current provided to V_{mem} through the refractory block and makes the block more power

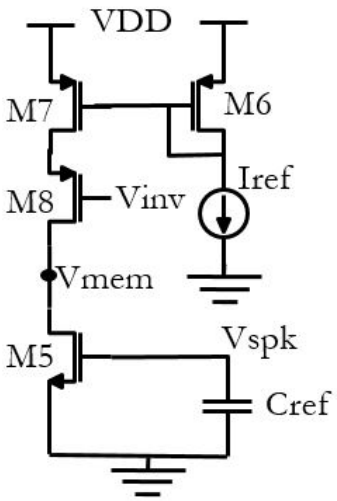

Figure 11: Refractory Block. Vinv is low when a spike has been generated. Thus both the upper and lower portions of the circuit are active during spiking. However, there is a propagation delay between the time Vinv goes low and Vspk goes high.

efficient. M8 governs when current is supplied to C_{mem} through the refractory block and is driven by V_{inv}. This means that whenever a spike is produced, The M8 enters saturation mode and the current through M8 charges C_{mem}. Therefore, if the potential at C_{mem} that caused the spike is just above the threshold voltage, the refractory block will drive V_{mem} to VDD. This insures a stable enough potential to drive the spiking functionality of the circuit.

After a short delay, the other portion of the refractory block is activated. The output spike charges the capacitor, C_{ref} until the potential at V_{ref} is equal to VDD. V_{ref} is then used to drive M6 which causes current to follow through M6 which will discharge C_{mem} and prevent the neuron from continuing to be in a spiked state. When V_{mem} falls below V_t, current stops flowing through the upper portion of the refractory block, allowing C_{mem} to discharge faster. Additionally, C_{ref}, also begins to discharge at a rate governed by V_{ref}, which controls the current through M17. This allows the pulse-width of the output, $V_{spk,}$ to be modulated and allows the spike frequency to be adjusted. The longer it takes V_{spk} to return to ground, the longer it will be before the neuron can fire again: until V_{spk} falls below the threshold voltage of M6, a leakage current will be working to inhibit the neurons spiking behavior.

Spike Generation Block

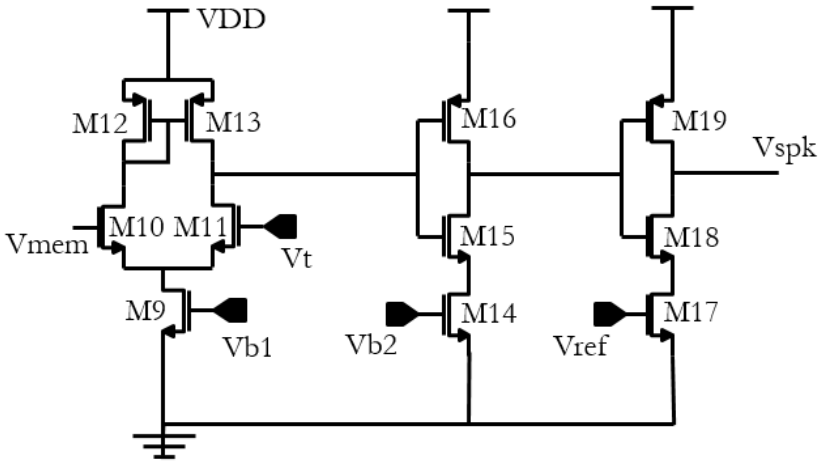

Figure 12: Spike Generation Block consisting of a transconductance amplifier and two inverters. The inverters introduce a propagation delay between the time spike generation and spike output which enables the functionality of the Refractory and SFA blocks.

The spike generation is accomplished using transistors M9-M19 (depicted in Figure 12) and consists of a five transistor transconductance operational amplifier (M9-M13) and two inverters (M14-M19). The transconductance amplifier compares the potential V_{mem} to an arbitrary threshold voltage defined by the input V_t and its behavior can be described as follows:

$$V_o = \begin{cases} VDD, & if\ V_{mem} \geq V_t \\ 0, & if\ V_{mem} < V_t \end{cases} \quad (10)$$

where V_o is the output of the transconductance amplifier. The M9, NMOS transistor connected to the source nodes of M10 and M11 does not contribute to the spiking functionality. It is placed there to help the circuit be more power efficient and does not have a significant impact on the neuron's functionality as long as V_{b1} is greater than or equal to 0.6V.

The pair of inverters functions as a buffer, introducing a set of delay times between when the spike is generated at V_o and when it is output at V_{spk}. These delays are used to

activate the SFA and refractory circuits before the actual spike is output. M14, attached to the source node of M15 in the first inverter, functions similarly to M9 by helping reduce power consumption. Like M9, the behavior of the circuit as a whole is impacted minimally when V_{b2} is greater than or equal to 0.6V.

3.2 Circuit Layout

Table 2: Layout Area. This table shows the layout area of each block implemented in the circuit. In the layout, the leakage current block and the refractory block were implemented together and the refractory block area reflects the layout area for both components

Block	Layout Area
Capacitor Bank	144.816μm^2
Input Array	91.872 μm^2
SFA Block	23.292μm^2
Refractory Block	29.075μm^2
Spike Generation Block	60.386μm^2
Entire Neuron	294.625μm^2

The circuit layout was done for 0.18μm technology node in Tanner EDA's L-Edit v2016.2 layout software by Mentor Graphics. All transistors in the layout have a channel length of 0.18μm. The NMOS transistors all have a width of 0.99μm, giving them a W/L ratio of 5.5. The PMOS transistors have a width 2.25μm, giving them a W/L ratio of 12.5 and making them roughly 2.3 times bigger than the NMOS transistors. This sizing difference allows the NMOS and PMOS transistors to perform with a high degree of

symmetry. Table 2 displays a breakdown of each block in the circuit and the area it consumes. The three capacitors represent the largest single portion of the circuit.

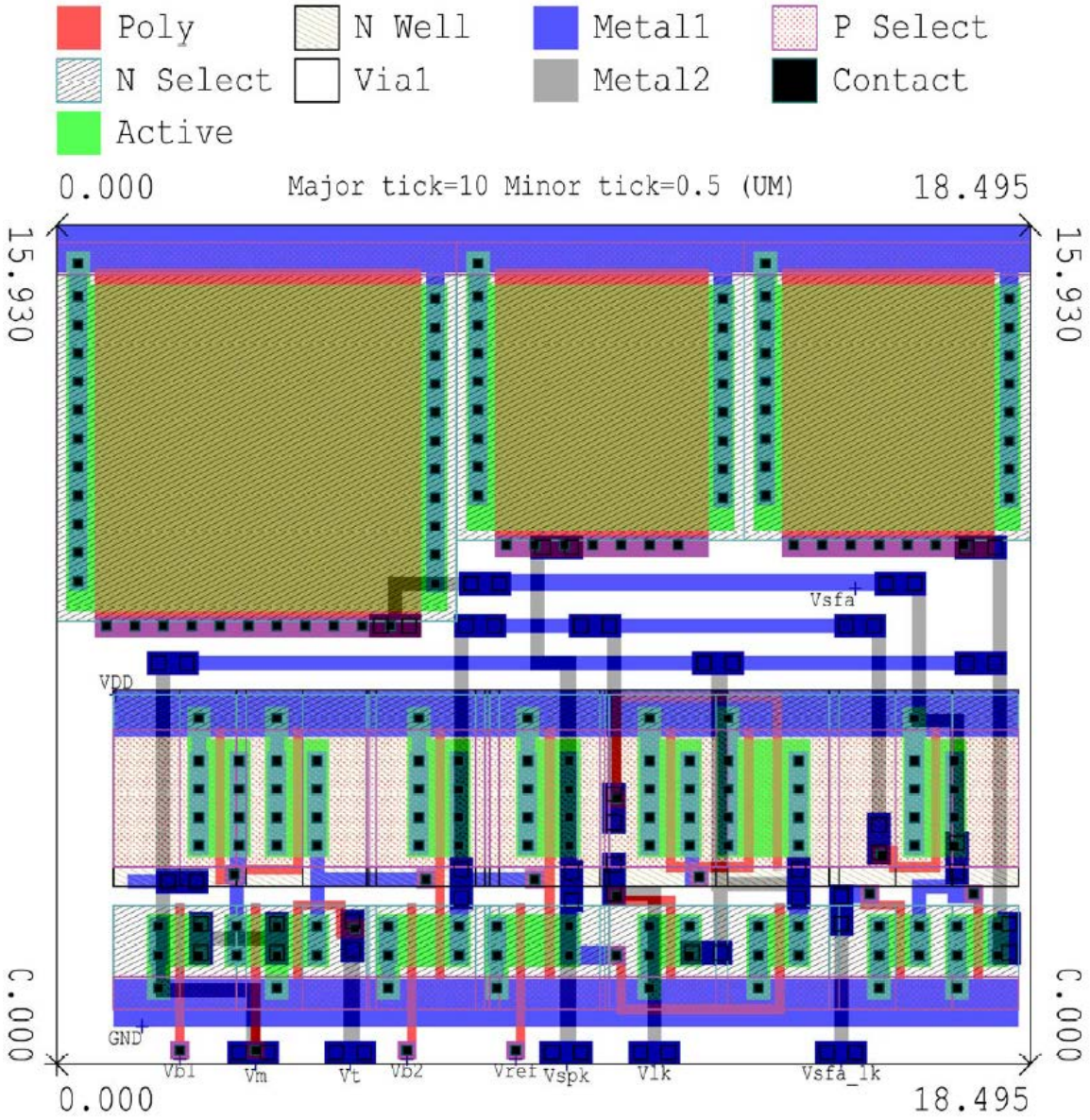

Figure 13: Circuit Layout. The three capacitors represent the upper portion of the circuit, while the lower portion is the implementation of M1-M19

The layout of transistors M1-M19 and the three capacitors C_{mem}, C_{ref}, and C_{sfa} consumes an area of 294.625µm^2 as seen in Figure 13. Additionally, it is implemented

using only two metal layers. It forms a tight square, making it easy to efficiently form an array of neurons on chip.

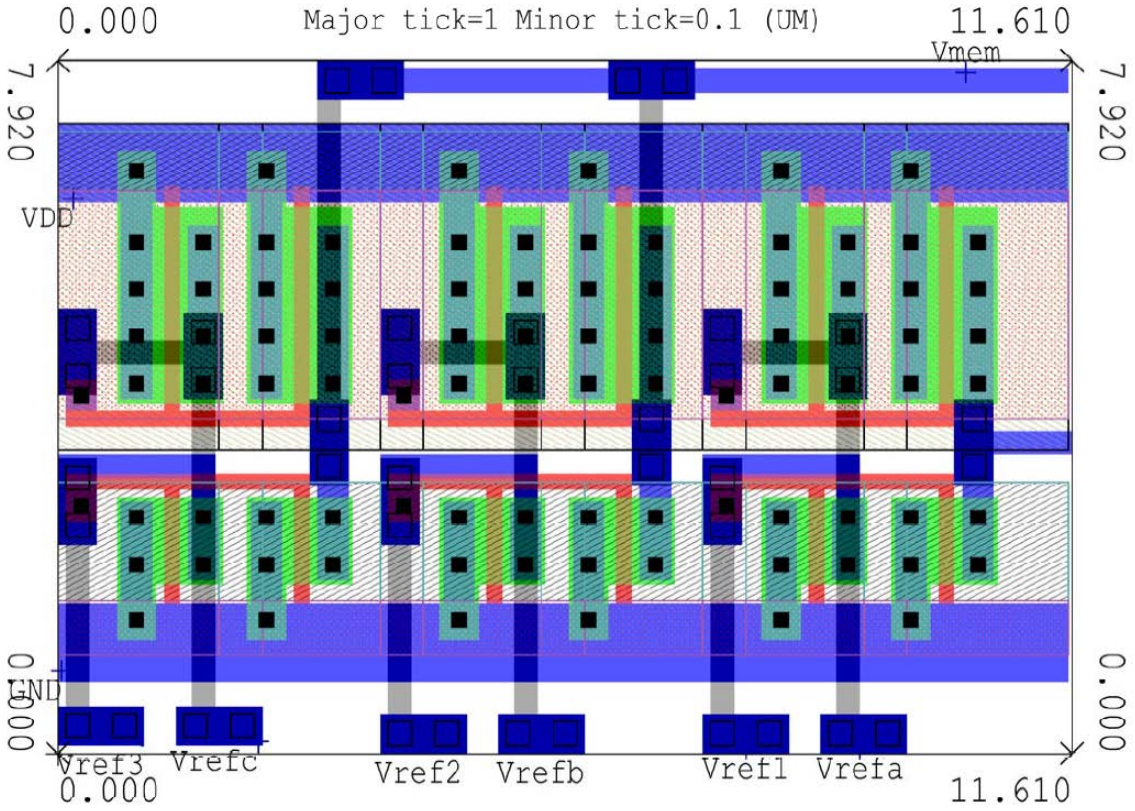

Figure 14: Current Mirrors. The numerically labeled inputs are inhibitory while the alphabetically labeled inputs are excitatory

Because a synapse was not implemented along with this circuit, there is no direct way to connect to it. To address this, a bank of current mirrors (Figure 14) was added to the layout. The inputs V_{refa}, V_{refb}, and V_{refc} are excitatory inputs which provide stimulus to charge the capacitor C_{mem} and cause the neuron to begin firing. Conversely, the inputs, V_{ref1}, V_{ref2}, and V_{ref3} are inhibitory and will accelerate the discharging of C_{mem}, thereby making it more difficult for the neuron to fire. The area of this bank is 91.872 μm^2, bringing the total layout area to 386.497μm^2 as shown in Figure 15. The layout can be divided into five components: the capacitor bank, the input array, the SFA block, the

refractory block, and the spike generator. The input array is discussed above. Each of the remaining components is discussed below.

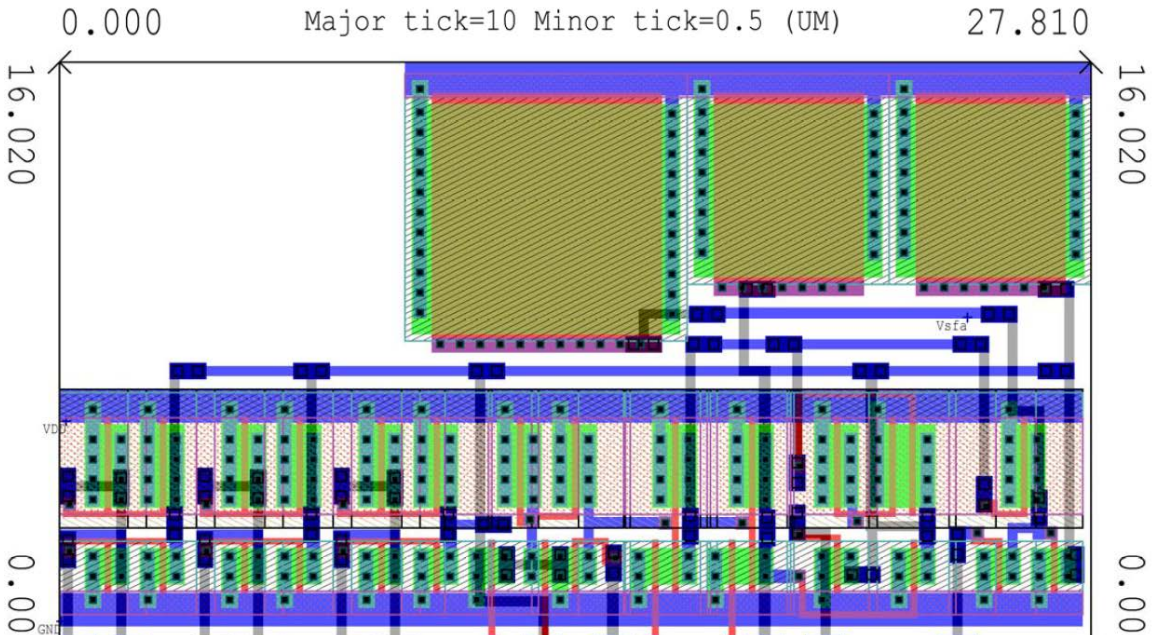

Figure 15: Neural Circuit with Current Mirrors. The entire circuit including six current mirrors to allow the circuit to be tested

Capacitors

The three capacitors in the circuit are implemented as MOS capacitors and consume a total area of 144.816μm^2 as shown in Figure 16. The sizing of the capacitors was done to

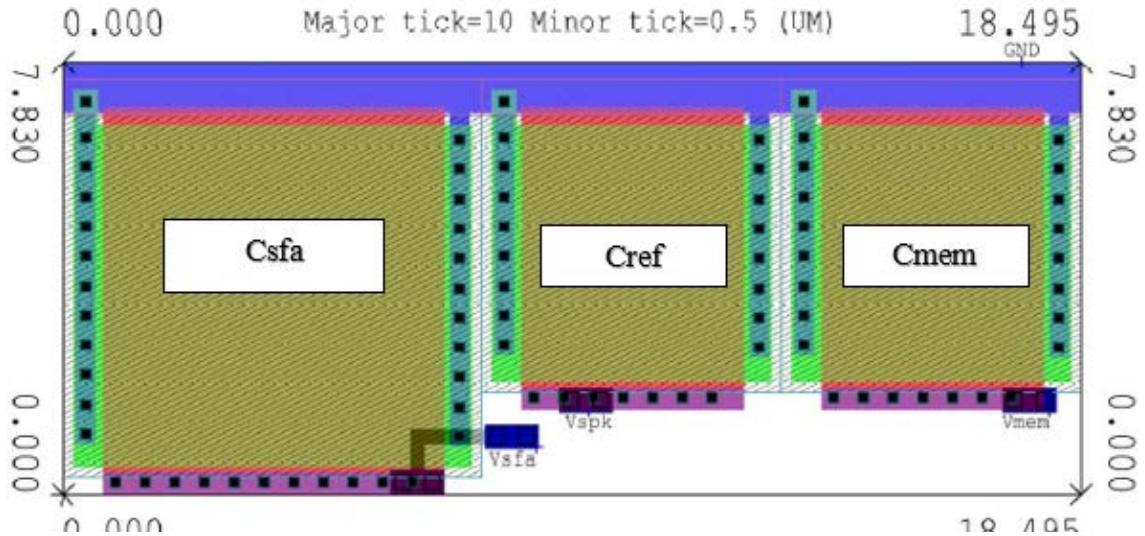

Figure 16: Bank of three capacitors. Csfa needs to charge more gradually than Cref and Cmem and is thus 2x the size of the other two capacitors

produce the desired capacitance values for each capacitor. Simulations done in Tanner EDA T-Spice v2016.2 by Mentor Graphics showed that the substrate has a capacitance of approximately 26fF/μm^2. Table 3 provide a breakdown of the area consumption and implemented capacitances for each of the capacitors.

C_{mem} and C_{ref} were both implemented as 0.5pF capacitors. Their layouts each consume a total area of 33.335μm^2 and have an area of poly-over-active of 18.772μm^2 as shown in Figure 17. The capacitance value chosen for C_{mem} and C_{ref} represents a compromise between area consumption and biological plausibility of the neuron. Instead of using a larger capacitor which would take longer to charge and discharge, therefore resulting in more biologically plausible time scales for the neuron's behavior, a smaller capacitor with a more could be implemented with a more compact profile was chosen.

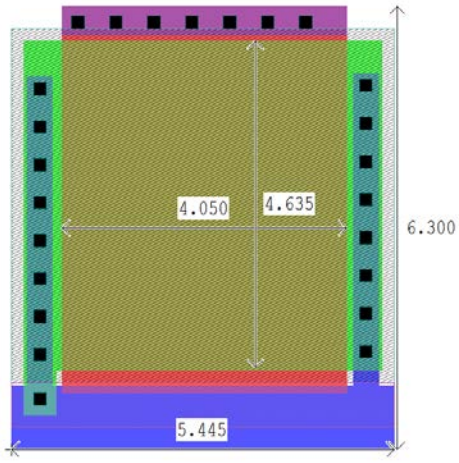

Figure 17: 0.5pF MOS capacitor. The capacitor has a layout area of 33.335 square microns

C_{sfa} was implemented as a 1pF capacitor and its layout consumes an area of 59.547μm^2 and has an area of poly-over-active of 38.564μm^2 as shown in Figure 18. C_{sfa} is larger than the other two capacitors because, in order to adapt to the neuron's firing behavior, it must have some ability to

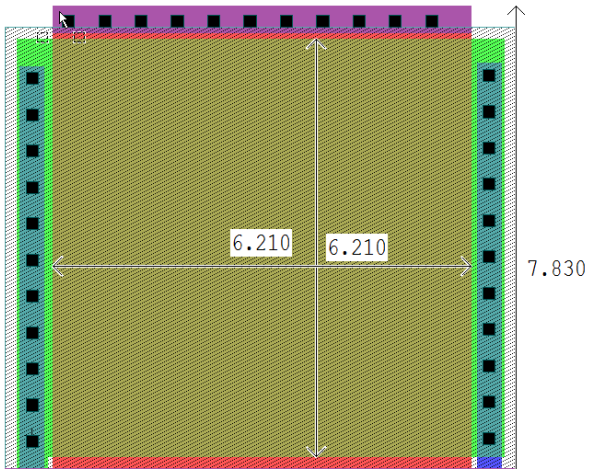

Figure 18: 1pF MOS capacitor. The capacitor has a layout area of 59.547 square microns.

remember what the previous behavior was. The larger capacitance means that it charges and discharges at a slightly slower rate and will not respond to a single spike in the same way that it would respond to a sequence of spikes. Likewise, it will hold its charge long enough to prevent the neuron from firing for a period of time.

Table 3: Capacitors. This table shows the implemented capacitance values for each of the circuit's three capacitors as well as the area they consume in the layout.

Name	Capacitance	Layout Area	Poly-Over-Active Area
C_{mem}	0.5pF	33.335µm^2	18.772µm^2
C_{ref}	1.0pF	33.335µm^2	18.772µm^2
C_{sfa}	1.0pF	59.547µm^2	38.564µm^2

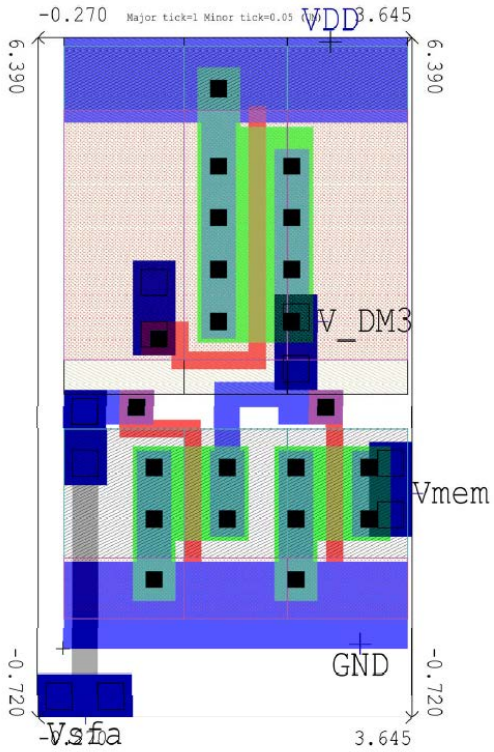

Figure 19: SFA Block Layout. The input Vsfa governs the sensitivity of the blocks behavior. Higher voltages result in less sensitive spike frequency adaptation and when Vsfa=VDD the block is disabled.

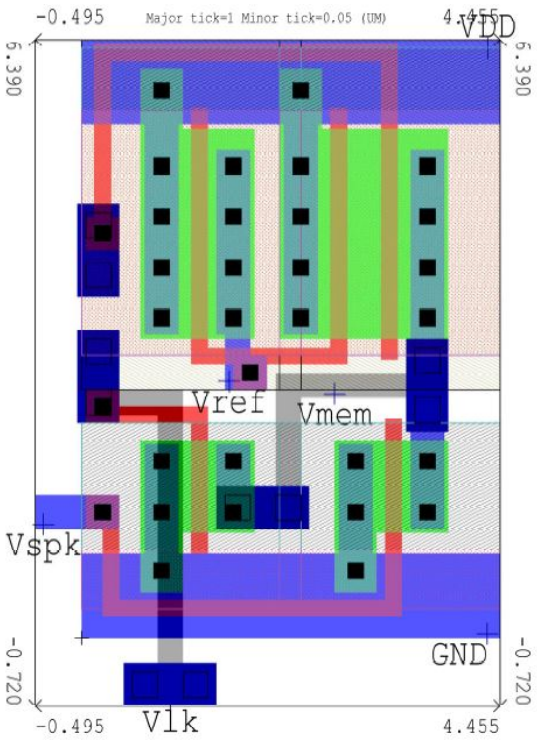

Figure 20: Refractory Block Layout. Vlk implements the neuron's membrane leakage current. The refractory circuit consists of the three PMOS transistors and the NMOS transistor on the lower right.

Spike Frequency Adaptation Block

The layout of the SFA block, as seen in Figure 19, consumes 23.292μm^2 and consists of two NMOS and one PMOS transistors. The output V_DM3 connects to the capacitor C_{sfa} and the output V_{mem} connects to the capacitor C_{mem}. The PMOS's gate (unlabeled in Figure 19) connects to the output of the first inverter in the spike generation block.

Refractory Block

The refractory block, depicted in Figure 20, combines the refractory and leakage current blocks described in section 3.1. It consumes a total area of 29.075μm^2 and implements three PMOS and two NMOS transistors. The three PMOS transistors and the NMOS transistor on the bottom right implement the refractory functionality. The PMOS on the far right hand side of the figure is driven by the output of the first inverter in the spike generation block. The other two PMOS transistors comprise the current mirror driven by input, I_{ref}. The port, V_{mem}, is connected to the capacitor C_{mem} and the port V_{spk} is connected to the capacitor, C_{ref}. The membrane leakage current is implemented with the NMOS transistor on the left hand side of the figure. Its drain is connected to the capacitor C_{mem} and the input voltage, V_{lk}, governs the magnitude of the leakage current which discharges C_{mem}.

Spike Generation Block

The spike generation block is implemented using four PMOS transistors and seven NMOS transistors. It consumes an area of 60.386μm^2 as shown in Figure 20. The output V_{mem} is connected to the capacitor, C_{mem}, and the output, V_{spk} is connected to the

capacitor, C_{ref}. V_{b1}, V_t, V_{ref}, V_{b1}, and V_{b2} are all external inputs. V_{inv1} is the output which is used to drive the refractory and SFA blocks.

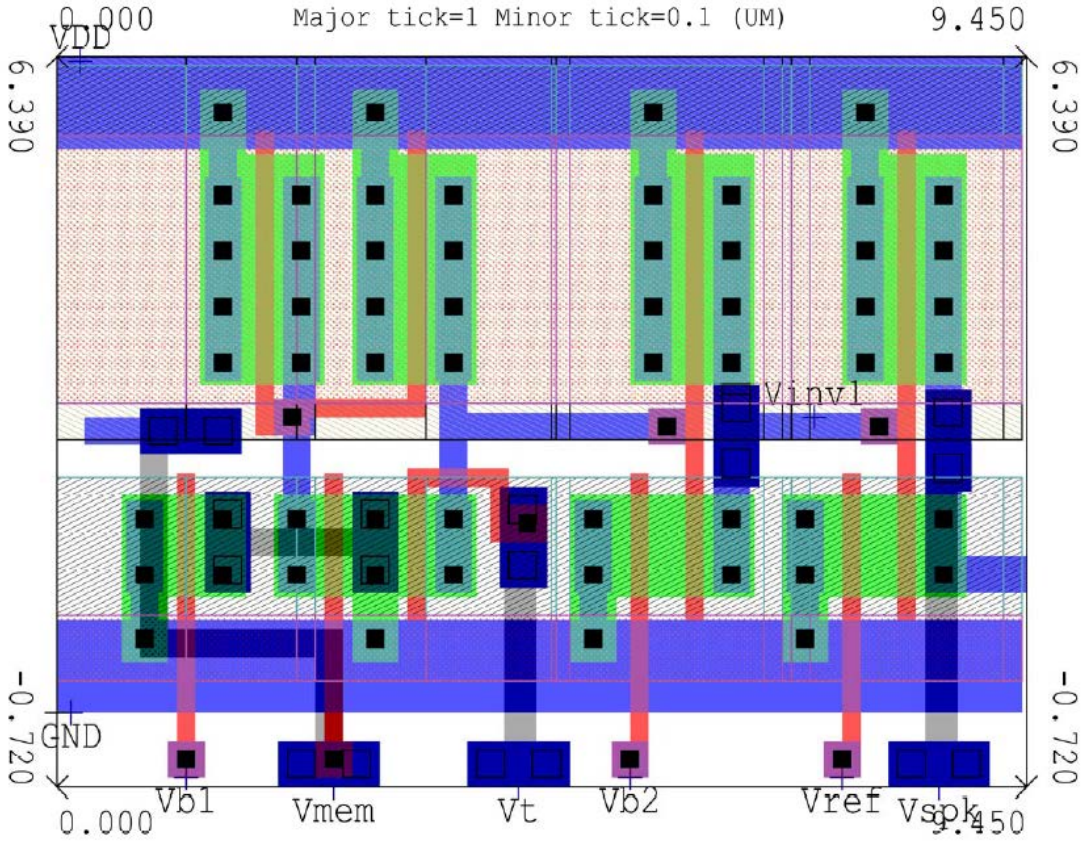

Figure 21: Spike Generation Block Layout. Vinv1, is the output port from the first inverter which is used to drive the refractory and SFA blocks.

Chapter 4: Simulation Results

This section presents the simulation results for the neural circuit. The circuit layout (described in Chapter 3) was extracted from Tanner EDA's L-Edit v2016.2 layout software by Mentor Graphics using an IBM 7RF transistor model. Simulations were conducted in Tanner EDA's T-Spice v2016.2. All input voltages discussed in this section were attached to transistors with a threshold voltage of 0.4V and the circuit was operating with a VDD potential of 1.8V. Voltage sources that served as inputs to tune the circuit's behavior were tested in 100mV increments to provide a range for the neuron's operating behaviors. All simulations, unless otherwise specified, were conducted for a 1µs period during which a constant 10µA input current was used to stimulate the neuron. Additionally, in all simulations a small reference current, I_{ref}, of 1nA was supplied to the current mirror in the refractory block of the circuit.

4.1 Bias Inputs

This section discusses the effect the bias inputs, V_{b1} and V_{b2}, have on the circuit's average power consumption. Table 4 shows the simulation results for a range of input values for V_{b1} and V_{b2}.

Simulation 1 in Table 4 depicts the average power when V_{b1} and V_{b2} are configured to have minimal impact on the circuit. Simulations 2-4 indicate that as V_{b1} is reduced the average power decreases. A potential of 0.5V achieves the maximum power reduction. Because the transistor governed by V_{b1} affects the rate at which the amplifier swings from high to low, 0.5V represents the minimum value for V_{b1} at which the circuit will still properly function. Moving below this value interferes with the spiking pattern produced

by the neuron. Critically, adjusting the value of V_{b1} has little input on the circuit's spiking pattern if V_{b1} does not drop below its minimum operating value. As shown in Table 4, while the average power is impacted substantially by adjusting V_{b1}, the Inter-Spike-Interval (time between spikes) only changes by 11ps and the spike width is only affected by 2ps.

Table 4:Bias inputs. This table depicts the relationship between the bias inputs Vb1 and Vb2 and the power consumption and spiking behavior of the neuron. The input parameters of relevance to this section are highlighted in bold.

Sim #	I_{in} (µA)	V_{lk} (V)	V_t (V)	V_{sfa} (V)	V_{ref} (V)	V_{b1} (V)	V_{b2} (V)	Average Inter-Spike-Interval (ps)	Average Spike Width (ps)	Spike Freq (Spikes per µs)	Average Power (µW)	Joules per Spike (pJ)
1	10	0.4	0.4	1.8	1.8	**1.8**	**1.8**	45	109	65	644.8	9.92
2	10	0.4	0.4	1.8	1.8	**0.9**	**1.8**	42	107	67	637.2	9.51
3	10	0.4	0.4	1.8	1.8	**0.6**	**1.8**	41	109	67	628.7	9.38
4	10	0.4	0.4	1.8	1.8	**0.5**	**1.8**	53	109	62	566.6	9.14
5	10	0.4	0.4	1.8	1.8	**1.8**	**0.9**	48	113	63	620.0	9.84
6	10	0.4	0.4	1.8	1.8	**1.8**	**0.7**	52	120	58	584.0	10.07
7	10	0.4	0.4	1.8	1.8	**1.8**	**0.6**	57	156	47	605.7	12.89
8	10	0.4	0.4	1.8	1.8	**0.5**	**0.7**	57	110	60	491.1	8.19

Simulations 5-7 depict the impact of reducing the potential at V_{b2}. This input affects the rate at which the first inverter in the spike generation block can swing from high to low, so just like with V_{b1} the limit to how much the average power can be reduced is determined by the point at which the circuit will no longer produce a spike pattern in the output. As can be seen by the results of simulation 6, a maximum power reduction is achieved with an input value for V_{b2} of 0.7V. Moving below this value affects the switching rate of the first inverter in the spike generation block and actually causes the circuit to be less efficient, increasing average power consumption. Modulating V_{b2} has a more dramatic effect on the Inter-Spike-Interval and spike width than does modulating

V_{b1}. However, these values are still only minimally effected, increasing by 12ps and 11ps respectively when V_{b2} is at its minimum operating value of 0.7V.

When both V_{b1} and V_{b2} are set at their minimum values, simulation 8 shows only a slight increase in spike width and Inter-Spike-Interval, but a substantial decrease in average power consumption. Figure 23 shows the spike pattern produced by the circuit when the inputs are configured as shown in simulation 1 in Table 4 and Figure 22 depicts the spike pattern when the inputs are configured as depicted in simulation 8. While there is a slight reduction in the number of spikes generated over a 1µs period the overall

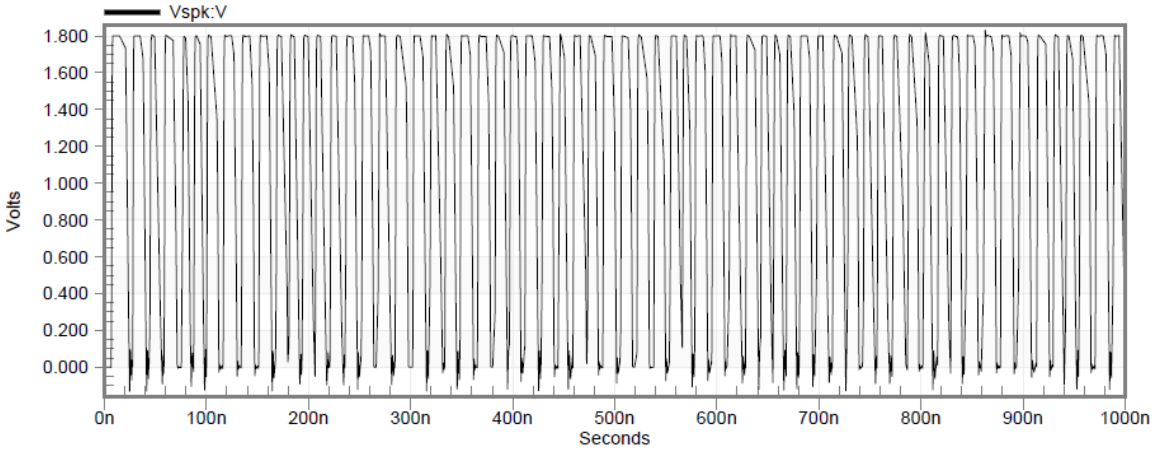

Figure 22: Spiking Behavior of Neuron in Simulation 1. This figure shows the spike pattern produced when Vb1 and Vb2 are configured to have minimal effect on the circuit.

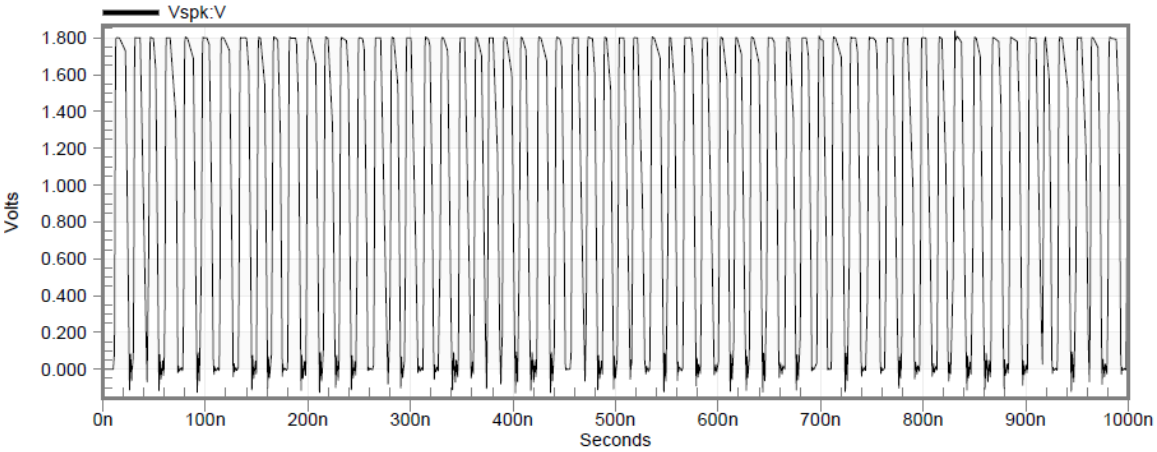

Figure 23: Spiking Behavior of Neuron in Simulation 8. This figure shows the spike pattern produced when Vb1 and Vb2 are configured to have maximal effect on the circuit.

behavior remains unaffected by adjusting the values of V_{b1} and V_{b2}. For all other simulations the inputs V_{b1} and V_{b2} will have values of 0.5V and 0.7V respectively.

4.2 Leakage Current

The leakage current discharging the capacitor, C_{mem}, can be adjusted using the input V_{lk}. As the value of V_{lk} increases, so does the current through transistor M1 (Figure 9). The circuit operates most efficiently when the firing threshold is set to 0.6V as will be explained in Section 4.3. Figure 24 shows the magnitude of the current through M1 when the membrane potential is at this maximal value of 0.6V for varying values of V_{lk} and represents the maximal inhibitory effect vales for V_{lk} ranging between 0.3V and 0.6V will have on the membrane potential.

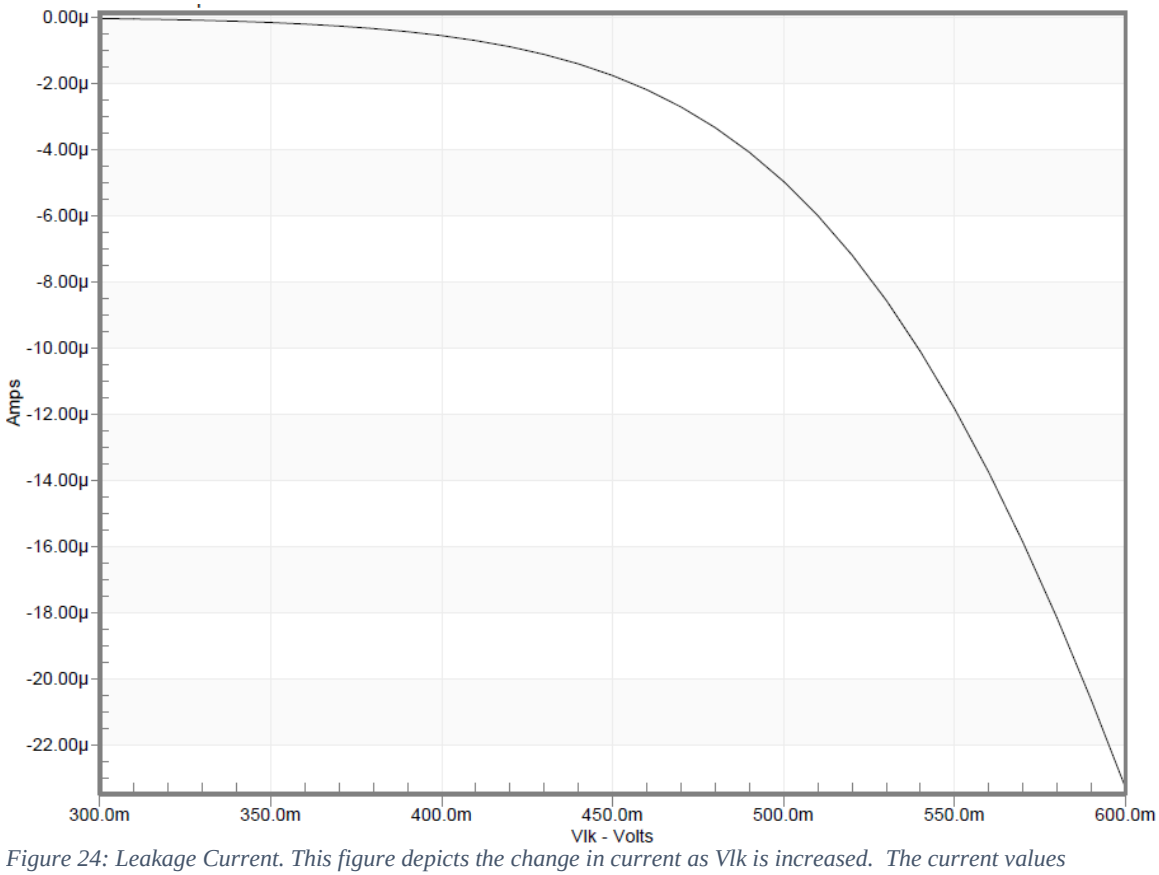

Figure 24: Leakage Current. This figure depicts the change in current as Vlk is increased. The current values simulated are representative of the case where the potential at Cmem is 0.6V.

Figure 25 show the change in leakage current as the membrane potential changes and V_{lk} is fixed at 0.4V (the minimal value V_{lk} can take and still have M1 be on). In this case the inhibitory leakage current peaks at roughly half a micro-amp and is negligible when the charge on C_{mem} is small. This behavior means that the leakage current will have a significant impact on the circuit without dominating its behavior and therefore represents a good configuration with which to operate the neuron.

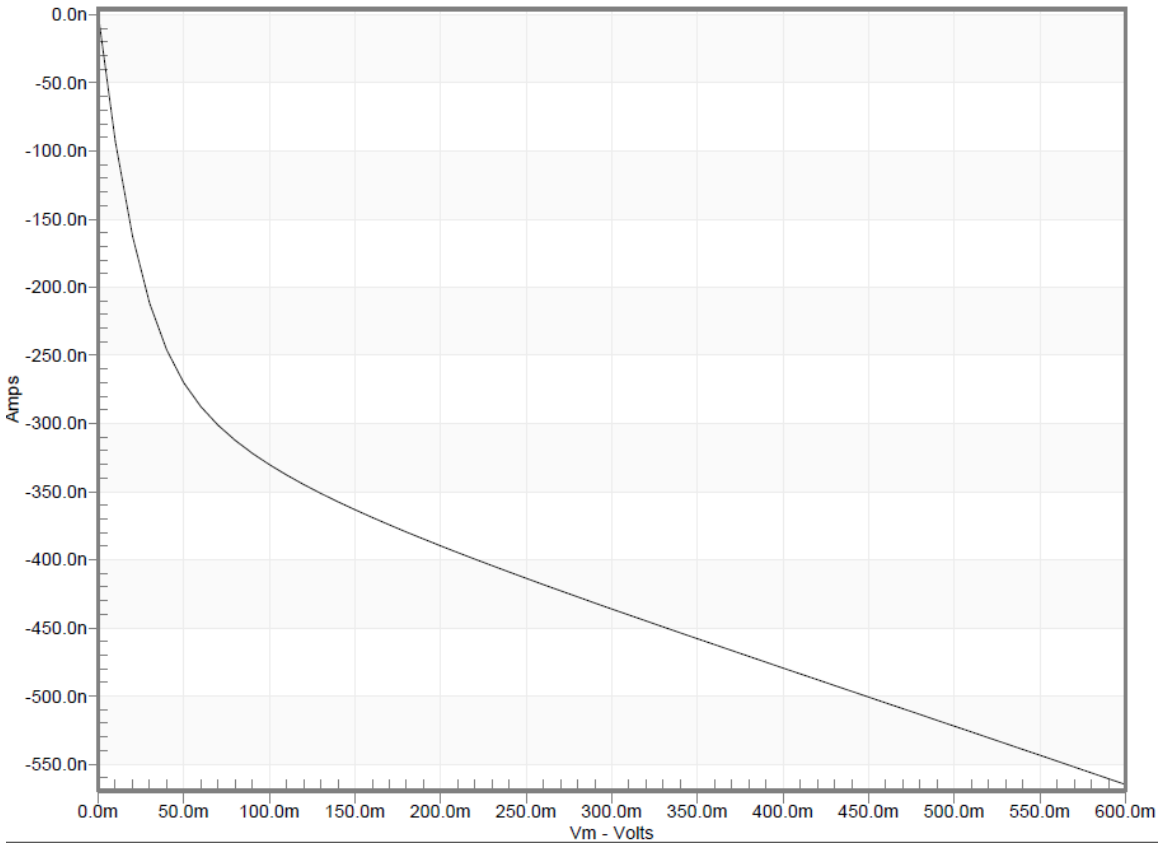

Figure 25: Leakage current for varying membrane potentials when Vlk is 0.4V

Table 5: Leakage Current. This table depicts the effect modulating the leakage current has on the behavior of then neuron. Relevant input values are bolded in the table.

Sim #	I_{in} (µA)	V_{lk} (V)	V_t (V)	V_{sfa} (V)	V_{ref} (V)	V_{b1} (V)	V_{b2} (V)	Average Inter-Spike-Interval (ps)	Average Spike Width (ps)	Spike Freq (Spikes per µs)	Average Power (µW)	Joules per Spike (pJ)
24	10	**0**	0.4	1.8	1.8	0.5	0.7	55	114	60	501.3	8.50
8	10	**0.4**	0.4	1.8	1.8	0.5	0.7	57	110	60	491.1	8.19
10	10	**0.5**	0.4	1.8	1.8	0.5	0.7	79	103	55	408.4	7.43
9	10	**0.6**	0.4	1.8	1.8	0.5	0.7	N/A	N/A	0	0.00197	N/A

Table 5 shows how the circuit's behavior changes as a result of increasing the leakage current. Increasing the leakage current decreases the circuit's average power consumption because the firing rate is reduced. Simulation 8 demonstrates that 0.4V is the largest value for V_{lk} does not impact the spike frequency. Furthermore, increasing the value of V_{lk} to 0.6V allows for enough of an inhibitory current to prevent the neuron from firing altogether.

4.3 Firing Threshold

The threshold at which the neuron fires is controlled by the input, V_t, which defines the potential the membrane capacitance must reach in order to cause the neuron to spike. Table 6 shows the effect that adjusting this parameter has on the circuit's behavior. Notably, as the threshold increases, the inhibitory characteristics of the neuron, such as the leakage current have a more pronounced effect on the circuit. This means that at higher threshold potentials the neuron produces a more sharply pronounced spike (the spike width narrows substantially).

When raising the threshold from 0.4V the spike frequency increases at the same time as the spike width decreases and the Inter-Spike-Interval increases until the threshold reaches 0.6V. This means that the neuron is firing faster and, at the same time, the spikes

are farther apart and narrower. Beyond a threshold of 0.6V, the frequency falls off, but the spike width and Inter-Spike-Interval continue their respective trends.

Table 6: Firing Threshold. This table depicts the effect of adjusting the threshold for the membrane potential will cause the neuron to begin firing. The relevant input parameters are highlighted in bold.

Sim #	I_{in} (µA)	V_{lk} (V)	V_t (V)	V_{sfa} (V)	V_{ref} (V)	V_{b1} (V)	V_{b2} (V)	Average Inter-Spike-Interval (ps)	Average Spike Width (ps)	Spike Freq (Spikes per µs)	Average Power (µW)	Joules per Spike (pJ)
8	10	0.4	**0.4**	1.8	1.8	0.5	0.7	57	110	60	491.1	8.19
11	10	0.4	**0.6**	1.8	1.8	0.5	0.7	114	35	67	193.4	2.89
12	10	0.4	**0.7**	1.8	1.8	0.5	0.7	129	32	62	178.3	2.88
13	10	0.4	**0.8**	1.8	1.8	0.5	0.7	151	34	54	170.0	3.15
14	10	0.4	**1.5**	1.8	1.8	0.5	0.7	274	20	34	84.5	2.89
15	10	0.4	**1.6**	1.8	1.8	0.5	0.7	N/A	N/A	N/A	87.7	N/A

Figure 26 and Figure 27 depict the neurons spike patterns for a 0.4V and 0.6V threshold respectively. The spikes generated by the 0.6V thresholded circuit are much

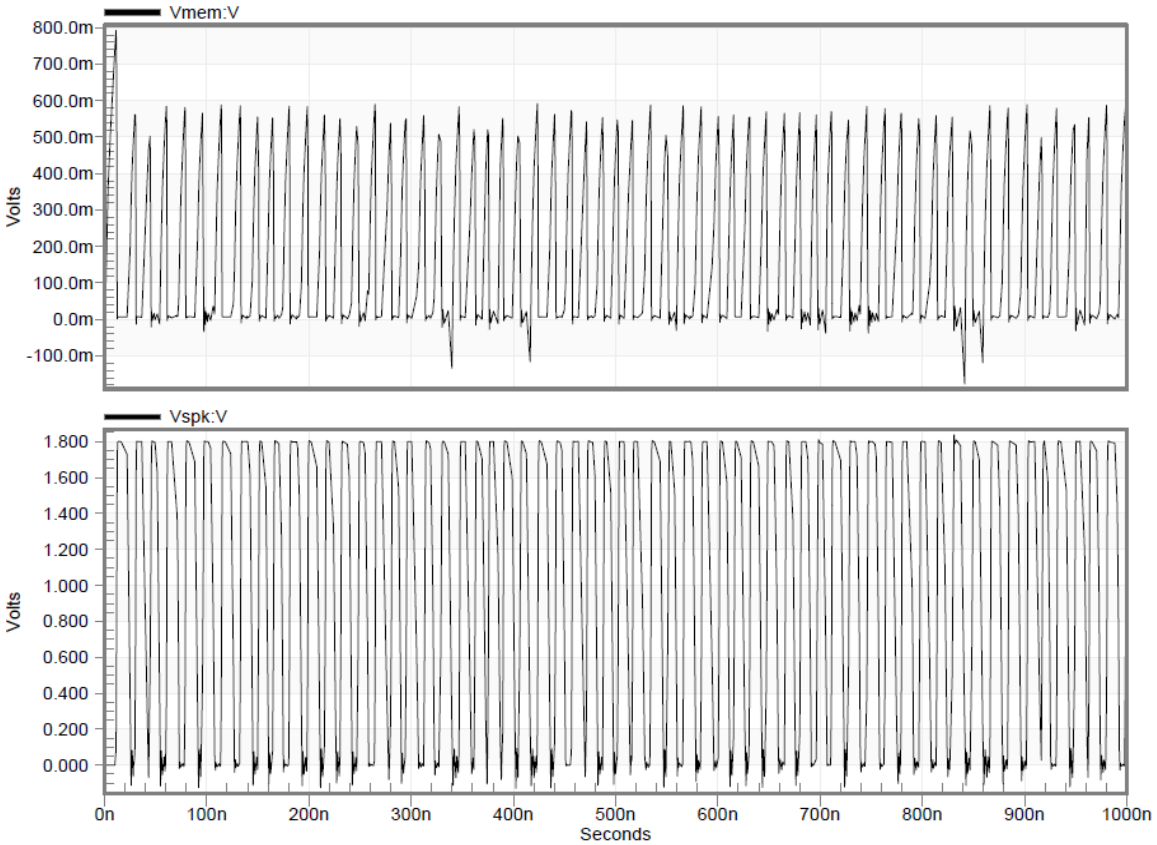

Figure 26: Spiking Pattern with 0.4V Threshold. This figure depicts the spiking output pattern in the lower image and the membrane potential in the upper image. The membrane potential reflects a relatively smooth charge/discharge rate and the spike width is relatively large.

narrower. This is representative of a more ideal behavior than those generated by the 0.4V thresholded circuit. Additionally, the steeper slope at the peak of the waveform for Vmem reflects the increased charge/discharge rate that results at higher membrane potentials.

As the spike width decreases, the average power is reduced substantially. Operating the circuit with a threshold of at least 0.6V means that the circuit has an average power usage of under 200µW resulting in a minimum energy efficiency of 2.9pJ per spike.

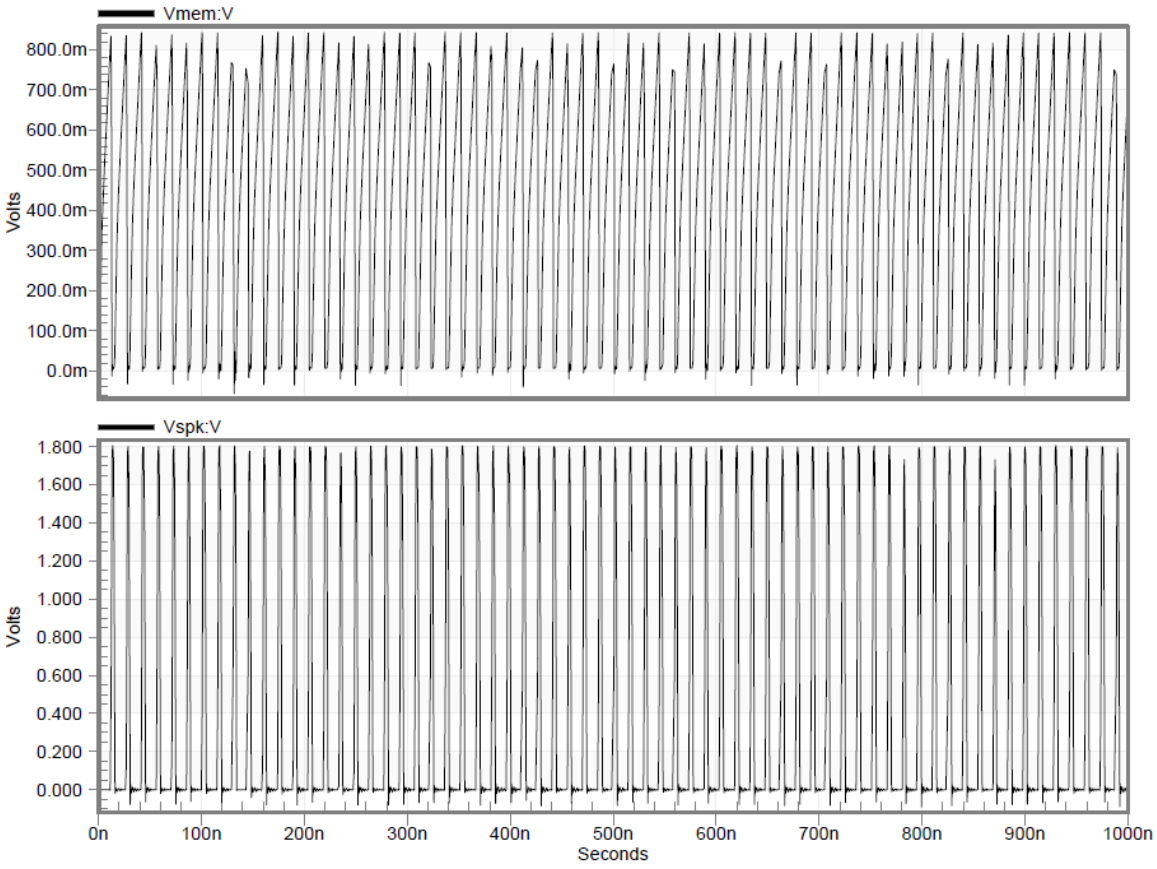

Figure 27: Spiking Pattern with 0.6V Threshold. The upper image represents the membrane potential. It has a much steeper charge/discharge cycle that that of the 0.4V thresholded simulation. The lower image represents the spiking output and has a much more pronounced spike pattern than in the 0.4V thresholded circuit.

4.4 Spike Width and Refractory Period

The refractory period (the duration of the Inter-Spike-Interval) can be tuned to produce slightly different firing patterns through the use of the input, V_{ref}. Decreasing V_{ref} from a maximal value of VDD causes the capacitor C_{ref} to discharge at a slower rate which simultaneously prolongs the duration for which the membrane capacitor sees an additional leakage current and slows the rate at which the spike can switch from high to low.

Table 7: Refractory Period. This table shows the effect of tuning the refractory period on the circuits firing pattern. Relevant inputs are highlighted in bold.

Sim #	I_{in} (µA)	V_{lk} (V)	V_t (V)	V_{sfa} (V)	V_{ref} (V)	V_{b1} (V)	V_{b2} (V)	Average Inter-Spike-Interval (ps)	Average Spike Width (ps)	Spike Freq (Spikes per µs)	Average Power (µW)	Joules per Spike (pJ)
11	10	0.4	0.6	1.8	**1.8**	0.5	0.7	114	35	67	193.4	2.89
16	10	0.4	0.6	1.8	**0.9**	0.5	0.7	116	47	64	175.3	2.74
17	10	0.4	0.6	1.8	**0.8**	0.5	0.7	113	49	62	168.1	2.71
18	10	0.4	0.6	1.8	**0.7**	0.5	0.7	113	66	56	157.6	2.80
19	10	0.4	0.6	1.8	**0.6**	0.5	0.7	109	119	44	114.4	2.60
20	10	0.4	0.6	1.8	**0.4**	0.5	0.7	N/A	N/A	N/A	7.98	N/A

Table 7 shows the effect of decreasing V_{ref} on the firing patterns. Between 1.8V and 0.9V, the Inter-Spike-Interval increases by 2ps and the spike width increases 12ps. This allows the firing patterns to be very finely tuned within this range. Additionally, as

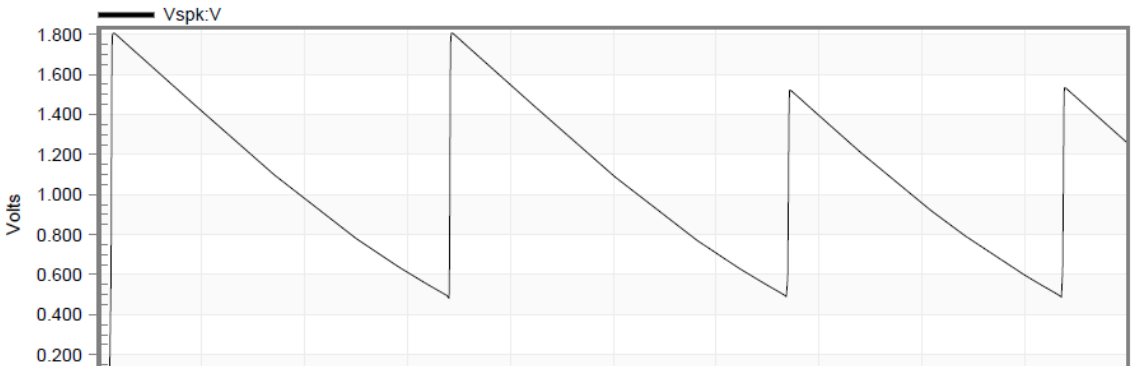

Figure 28: Spike Width. At low values for Vref, the neuron stops functioning and its output is a waveform which no longer resembles a spike pattern.

V_{ref} is further decreased the Inter-Spike-Interval begins to decrease more substantially, while the spike width continues to increase. While distinct spikes are still produced with V_{ref} values as low as 0.6V, going below this potential will cause the neuron to stop spiking altogether. Figure 28 shows the output of the neuron when V_{ref} is 0.4V. At this point, the output potential does not completely return to ground and the output becomes more of a saw tooth than a spike.

4.5 Spike Frequency Adaptation

SFA provides the neuron with a limited memory about previous events by modulating the refractory period based on past input stimulus. Transistor M3 in Figure 10 governs the sensitivity of the SFA circuit. Reducing the amount of current flowing through M3 increases the charge on capacitor C_{sfa} which, in turn, activates M2 and creates an additional path from V_{mem} to ground. This makes it harder to build a charge on the membrane capacitor and prevents the neuron from firing.

Table 8 demonstrates the impact adjusting V_{sfa} has on the neuron's output. It is important to note that, while the Inter-Spike-Interval is affected, the spike width is not. Adjusting V_{sfa} with values between 1.8V and 0.7V. In this same range, the Inter-Spike-Interval increases by 146ps. Simulation 26 shows that with a V_{sfa} of 0.6V, the spike width is only reduced by 1ps. Values of V_{sfa} below 0.6V cause the spike width to drop significantly. This minimum value for V_{sfa} at which the neuron can still produce spikes is 0.4V. At this setting, the output produces only 2 spikes per microsecond and the Inter-Spike-Interval increases from picoseconds to nanoseconds. However, this also changes the spike behavior: the spike width is nearly halved.

Table 8: SFA. This table demonstrates the effect adjusting Vsfa has on the refractory period and n power consumption. Relevant inputs are highlighted in bold.

Sim #	I_{in} (µA)	V_{lk} (V)	V_t (V)	V_{sfa} (V)	V_{ref} (V)	V_{b1} (V)	V_{b2} (V)	Average Inter-Spike-Interval (ps)	Average Spike Width (ps)	Spike Freq (Spikes per µs)	Average Power (µW)	Joules per Spike (pJ)
11	10	0.4	0.6	**1.8**	1.8	0.5	0.7	114	35	67	193.4	2.89
21	10	0.4	0.6	**0.9**	1.8	0.5	0.7	126	35	62	174.3	2.81
25	10	0.4	0.6	**0.7**	1.8	0.5	0.7	165	35	50	126.8	2.54
26	10	0.4	0.6	**0.6**	1.8	0.5	0.7	260	34	34	82.1	2.41
22	10	0.4	0.6	**0.5**	1.8	0.5	0.7	740	30	13	31.0	2.38
23	10	0.4	0.6	**0.4**	1.8	0.5	0.7	4982	18	2	5.61	2.81

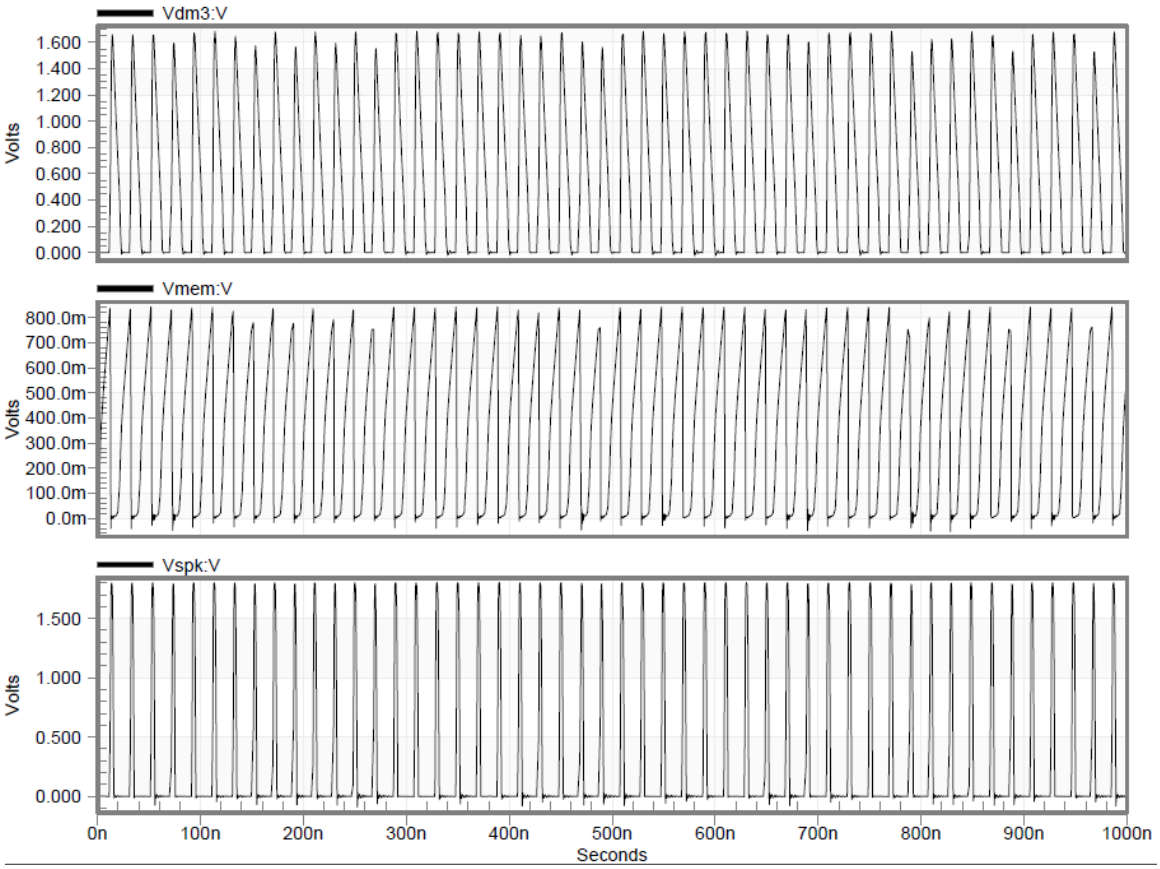

Figure 29: SFA with Vsfa=0.7V. This figure shows the spike pattern that results when SFA is enabled. The top graph shows the potential stored on Csfa. The middle graph is the membrane potential. Finally, the bottom graph is the neuron's output.

An added benefit of SFA, is the effect it has on power consumption. When SFA is enabled, the neuron spends less time switching. This results in less power consumption and can cause the circuits average power to drop from hundreds of micro-Watts to tens of micro-Watts.

Figure 29 displays the results of simulation 25 in Table 8: representing the maximum impact SFA can have on the output without affecting the spike width. Comparing the output, V_{spk}, here with the same output in Figure 27 shows the increase in Inter-Spike-Interval caused by SFA. The only difference in the settings of the input parameters that resulted in these two figures is the value of the input, V_{sfa}. Figure 27 corresponds to simulation 11 in Table 8.

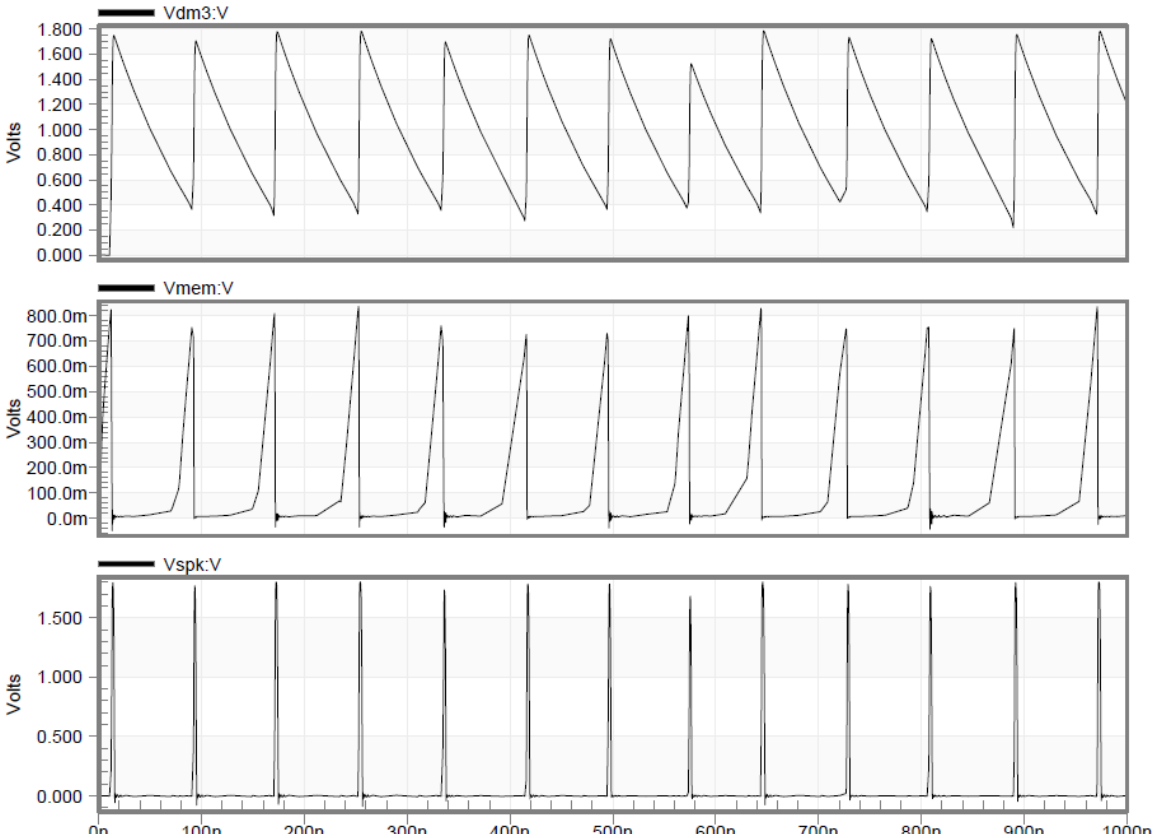

Figure 30: SFA with Vsfa=0.5V. This figure shows the spike pattern that results when SFA is configured aggressively. The top graph shows the potential stored on Csfa. The middle graph is the membrane potential. Finally, the bottom graph is the neuron's output.

Comparing the potentials V_{dm3} and V_{mem} in Figure 29 shows that the two are essentially compliments of each other. This means that the SFA block has a memory that goes one spike back in time. Figure 30, which corresponds to simulation 22 in Table 8, demonstrates how adjusting V_{sfa} impacts the duration of this memory. V_{sfa} modulates the discharge rate on C_{sfa}. Smaller values of V_{sfa} decrease the magnitude of the current through M3, thereby maintaing the potential on C_{sfa} for a longer duration of time.

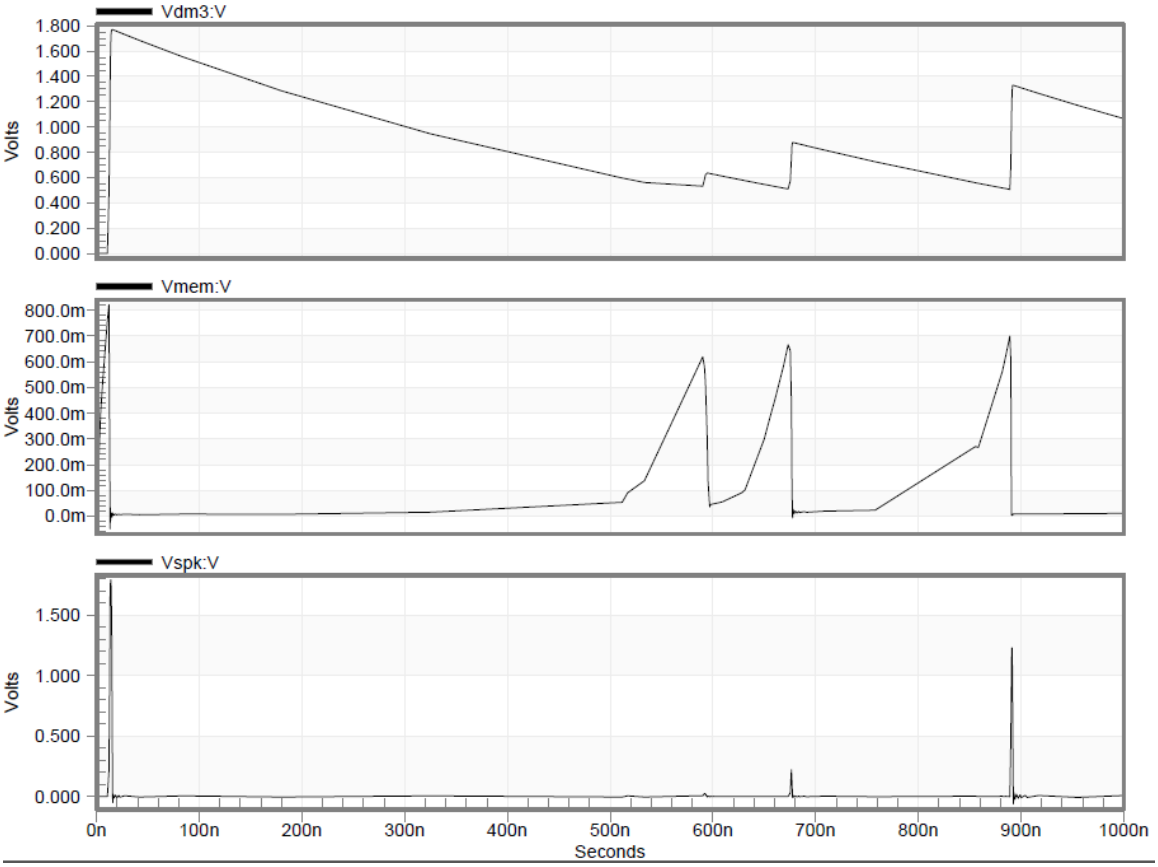

Figure 31: SFA with Vsfa=0.4V. This figure shows the spike pattern that results when SFA is configured to have a maximum impact on the firing pattern. The top graph shows the potential stored on Csfa. The middle graph is the membrane potential. Finally, the bottom graph is the neuron's output. Note, that at times, even when Vmem > Vt, the neuron is unable to fire.

Figure 31 shows the impact SFA has on the circuit when it is confiuged to be at its most sensitive and corresponds to simulation 23 in Table 8. In this case, the memory of C_{sfa} is so sensitive that it prevents future spikes from occuring altogether. At both

580ns and 680ns, the potential at V_{mem} is greater than V_t and thus a spike should be produced in the output. However, there is still enough charge on C_{sfa} to pul V_{mem} back below V_t before a spike can be produced.

4.6 Spike Generation Delay

This section addresses the propagation delay from the time V_{mem} surpasses the firing threshold potential to the time a spike is produced in the output. It first examines the delay through each of the two inverters used in the spike generation block as depicted in Figure 12. It then covers the delay through the transconductance amplifier, and finally, discusses the total delay. 0.5V and 0.7V were the potentials set for V_{b1} and V_{b2} respectively, a spike threshold voltage of 0.6V was used, V_{sfa} was configured to have

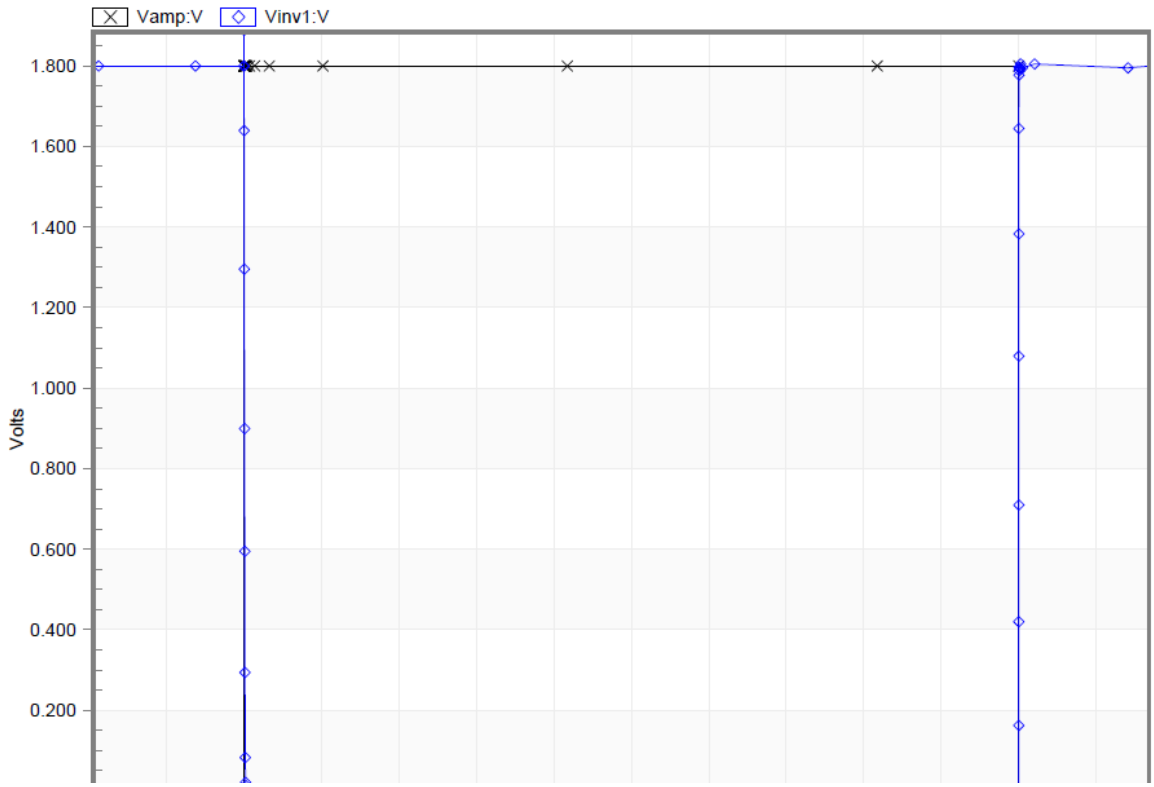

Figure 32: Delay through Inverter 1. The blue-diamond trace represents the output pulse and the black-x trace represents the input pulse.

minimal impact on the circuit and V_{ref} was configured to demonstrate its maximum and minimum impacts on the circuit.

Inverter Delay

To measure the delay through each of the inverters, a pulse was sent from the input of each inverter to the output and the rise and fall times were measured. Figure 32 shows the input and output pulse waveform for the first inverter in the spike generation block and simulation A in Table 9 provides its delay times. The first inverter has a significantly higher high to low delay than its low to high delay. This is caused by the bias transistor, M14 in Figure 12.

Table 9: Delay through Inverters. This table shows the propagation delay through the two inverters. Simulation A corresponds to the first inverter, while simulations B and C correspond to the second inverter.

Sim #	V_t (V)	V_{sfa} (V)	V_{ref} (V)	V_{b1} (V)	V_{b2} (V)	Low to High Delay (ns)	High to Low Delay (ns)
A	0.6	1.8	1.8	0.5	0.7	0.047	0.415
B	0.6	1.8	1.8	0.5	0.7	0.214	0.413
C	0.6	1.8	0.5	0.5	0.7	0.214	26.346

The delay through the second inverter is affected by both the output capacitor, C_{ref} and the transistor M17 which is governed by V_{ref}, and is substantially larger than the delay through the first inverter. Simulations B and C in Table 9 show the delay times for the second inverter. This delay represents the time difference between when feedback from the output of the first inverter reaches the Refractory and SFA blocks and feedback reaches those same blocks from the second inverter. Figure 33 and Figure 34 show the waveforms used to calculate the delays in simulations B and C respectively.

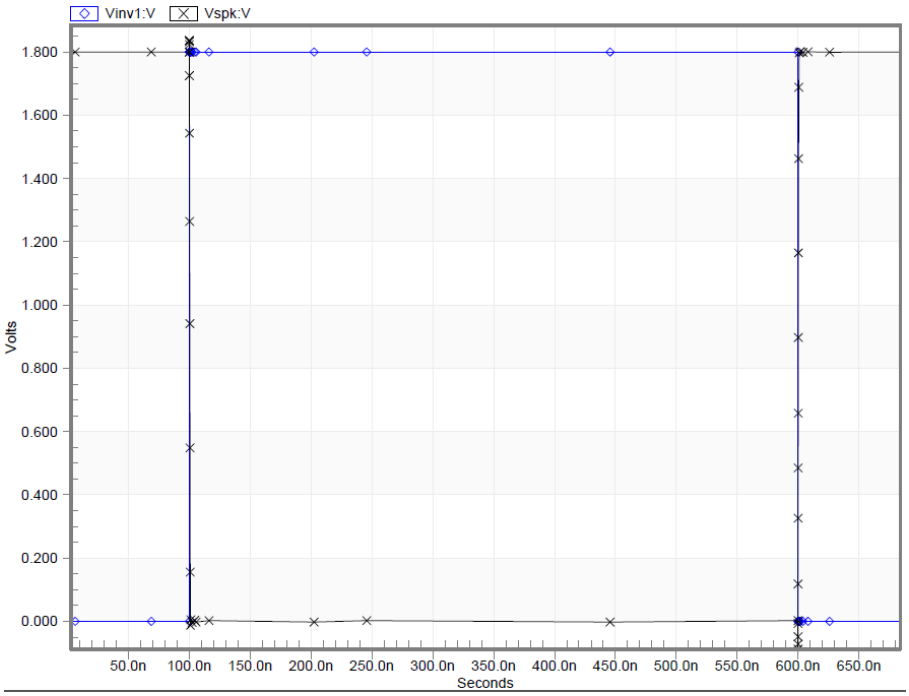

Figure 33: Minimum Delay through Inverter 2. The blue-diamond trace represents the output pules and the black-x trace represents the input pulse.

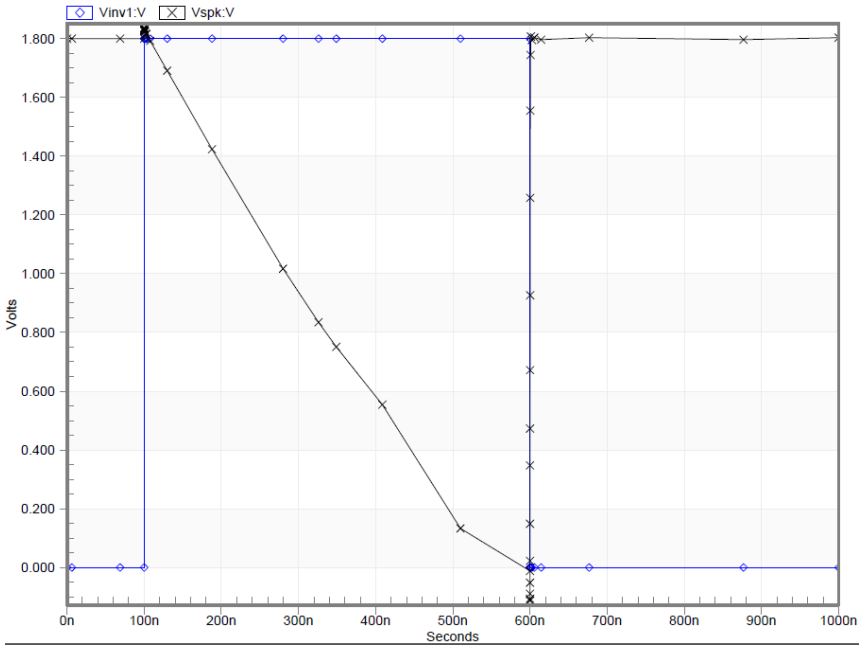

Figure 34: Maximum Delay through Inverter 2. The blue-diamond trace represents the output pulse and the black-x trace represents the input pulse.

Transconductance Amplifier and Spike Delay

Table 10: Delay through Amplifier and of Spike Generation. Simulation D represents the delay through the transconductance amplifier and simulations E and F represent the delay from Vmem to Vspk.

Sim #	V_t (V)	V_{sfa} (V)	V_{ref} (V)	V_{b1} (V)	V_{b2} (V)	Rising Delay (ns)	Falling Delay (ns)
D	0.6	1.8	1.8	0.5	0.7	1.960	6.510
E	0.6	1.8	1.8	0.5	0.7	3.390	5.893
F	0.6	1.8	0.5	0.5	0.7	3.294	31.729

The delays through the transconductance amplifier and from start to finish in the spike generation process were measured by placing an input pulse at the node, V_{mem} and measuring its propagation through the remainder of the circuit. As with the inverters' delays, the fall time delays are longer than the rise time delays due to the biasing of the spike generation circuit. Because the spike is measured based in the arrival of a low-high pulse, it is more important to have a minimal rising delay. In Table 10, simulation D shows the delay through the transconductance amplifier. The waveform used to measure the delay is depicted in Figure 35. Simulation E reports the delay from V_{mem} to V_{spk} when V_{ref} is configured to minimize the high to low delay (V_{ref} = 1.8V) and the waveform used to calculate the results is depicted in Figure 36. Finally, simulation F reports the delay from V_{mem} to V_{spk} when V_{ref} is configured to maximize the high to low delay (V_{ref} = 0.4V) and the waveform used to calculate the results is depicted in Figure 37.

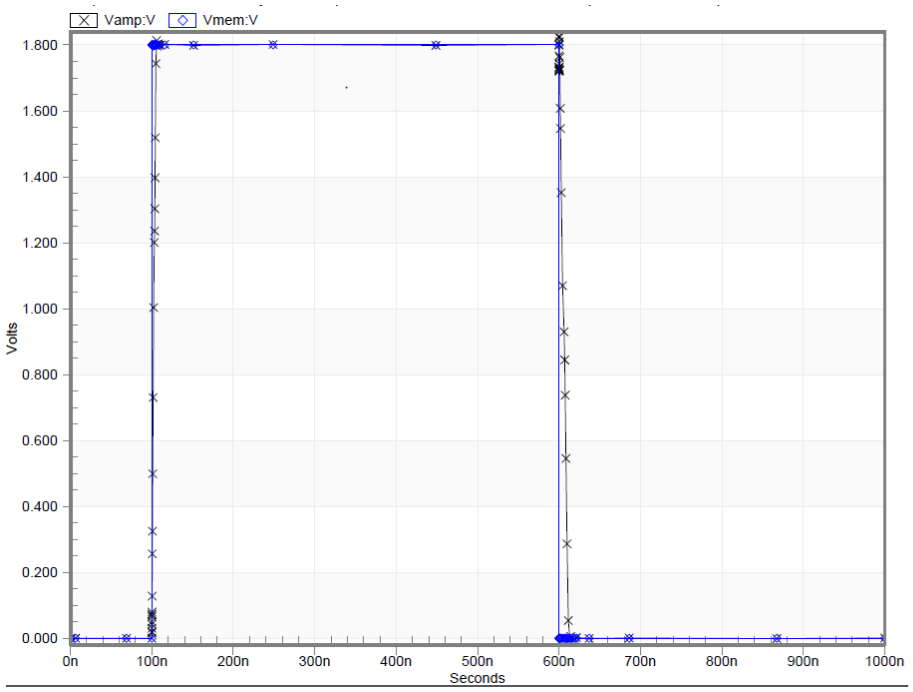

Figure 35: Delay through Transconductance Amplifier. The blue-diamond trace represents the output pules and the black-x trace represents the input pulse.

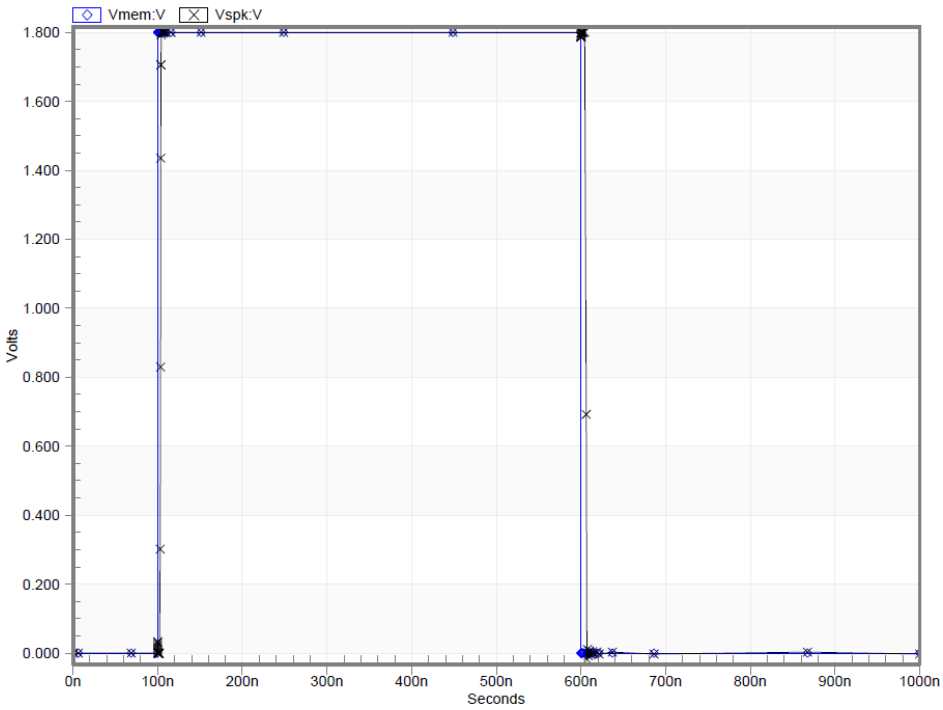

Figure 36: Minimum Delay from Vmem to Vspk. The blue-diamond trace represents the output pules and the black-x trace represents the input pulse.

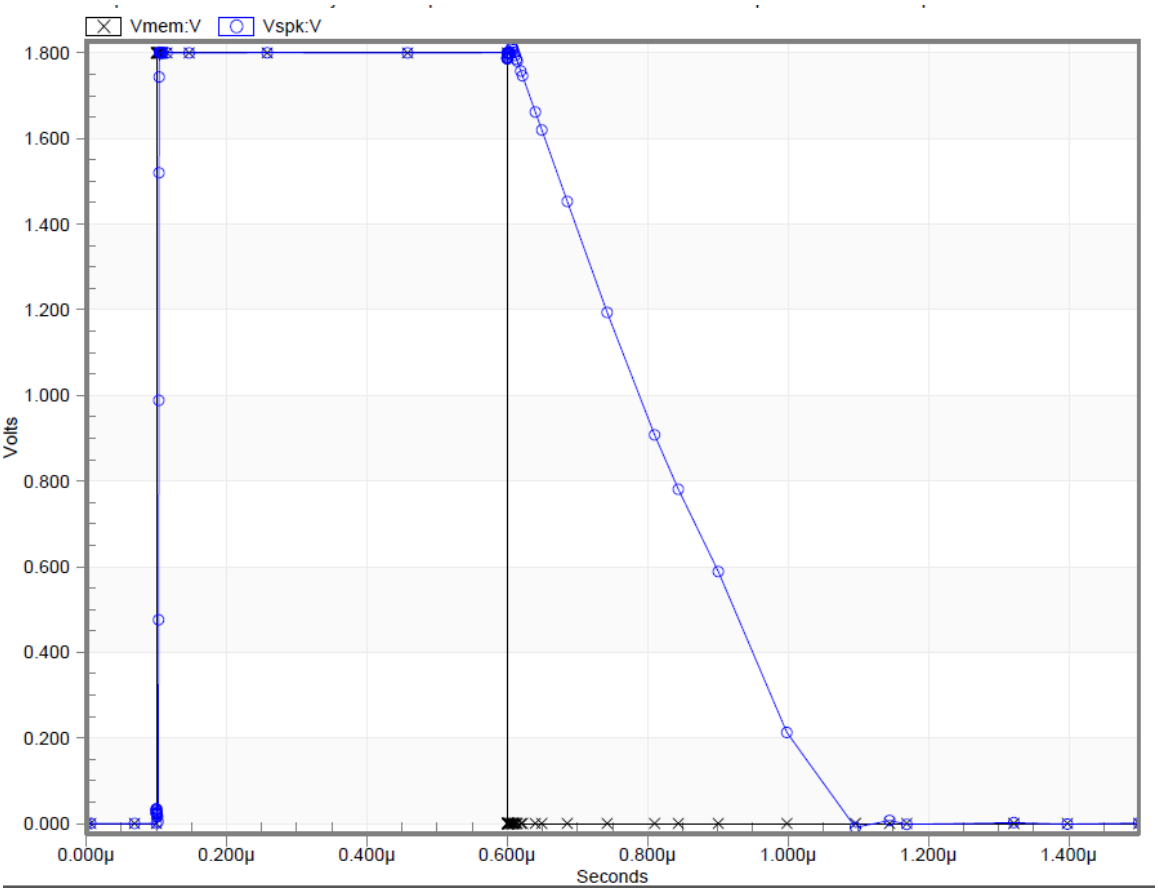

Figure 37: Maximum Delay from Vmem to Vspk. The blue-diamond trace represents the output pules and the black-x trace represents the input pulse.

Chapter 5: Conclusion

5.1 Synopsis

This work presents an analog Leaky-Integrate-and-Fire neural circuit. The circuit has tunable parameters that allow the firing behavior of the neuron to be altered by adjusting its refractory period and using Spike Frequency Adaptation to adjust to input stimulus in real time. The circuit operates in accelerated time, with a spike frequency in the nanosecond range. It consumes on the order of 100-200μW when properly tuned and has a spike efficiency on the order of 2-3pJ per spike. The layout of the circuit for a 0.18μm technology node occupies 386.497μm^2. This area includes the area used for one 1pF capacitor and two 0.5pF capacitors which were implemented as MOS-capacitors.

The circuit has a minimal number of adjustable parameters. This makes tuning its behavior relatively simple. Additionally, its ability to adapt to input stimulus using SFA, provides the circuit with a degree of Spike Time Plasticity which emulates the behavior of biological neurons. These features make the circuit ideal for use in computational networks which seek to emulate cognitive behaviors as well as in networks designed to process sensory input.

5.2 Future Work

The layout of the circuit can be further refined. Implementing the capacitors using a metal-insulator-metal model would allow for a more compact layout and perhaps, for the implementation of slightly larger capacitors. The current layout of the capacitors represents a compromise between layout area and circuit functionality. A slightly larger

capacitor in the SFA block would provide the neuron with a longer memory of past excitation and make its spiking behavior more dynamic.

A synaptic circuit also needs to be designed in order to allow this circuit to be configured into larger networks. Currently, the circuit can be tested using voltage source inputs to stimulate it directly, but it cannot be configured to respond to other neural inputs in its current state. Adding a synaptic circuit would allow for such configurations.

References

[1] H. de Garis, C. Shuo, B. Geortzel and L. Ruiting, "A world survey of artificial brain projects. Part I: Large-scale brain simulations," *Neurocomputing,* vol. 74, pp. 3-29, 2010.

[2] F. M. Dias, A. Antunes and A. M. Mota, "Artificial neural networks: a review of comercial hardware," *Engineering Applications of Artificial Intelligence,* vol. 17, pp. 945-952, 2004.

[3] D. Monroe, "Neuromorphic Computing Gets Ready for the (Really) Big Time," *Communications of the ACM,* vol. 57, no. 6, pp. 13-15, 2014.

[4] H. Mahdiani, A. Ahmadi, S. Fakhraie and C. Lucas, "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 57, no. 4, pp. 850-862, 2010.

[5] J. Misra and I. Saha, "Artificial Neural Networks in Hardware: A Survey of Two Decades of Progress," *Neurocomputing,* vol. 74, no. 3, pp. 239-255, 2010.

[6] B. Liu, Y. Chen, B. Wysocki and T. Huang, "Reconfigurable Neuromorphic Computing System with Memristor-Based Synapse Design," *Neural Processing Letters,* vol. 41, no. 2, pp. 159-167, 2015.

[7] J. Cruz-Albrecht, D. T and S. N, "A Scalable Neural Chip with Synaptic Electronics using CMOS Integrated Memristors," *Nanotechnology,* vol. 24, no. 38, 2013.

[8] A. Cassidy, P. Merolla, J. Arthur, S. Esser, D. Rubin, F. Akopyan, E. McQuinn, W. Risk, D. Modha and J. Sawada, "Cognitive Computing Building Block: A versatile and Efficient Digital Neuron Model for Neurosynaptic Cores," *International Joint Conference on Neural Networks,* 2013.

[9] S. Roy, A. Banerjee and A. Basu, "Liquid State Machine with Dendritically Enhanced Readout for Low-Power, Neuromorphic VLSI Implementations," *IEEE Transactions On Biomedical Circuits And Systems,* vol. 8, no. 5, pp. 681-695, 2014.

[10] H. Chen, S. Saighi, L. Buhry and S. Renaud, "Real-Time Simulation of Biologically Realistic Stochastic Neurons in VLSI," *IEEE Transactions on Neural Networks,* vol. 21, no. 9, pp. 1511-1517, 2010.

[11] C. M. Niu, S. K. Nandyala and T. D. Sanger, "Emulated muscle spindle and spiking afferents validates VLSI neuromorphic hardware as a testbed for sensorimotor function and disease," *Frontiers in Computational Neuroscience,* vol. 8, 2014.

[12] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomputing,* vol. 74, pp. 239-255, 2010.

[13] S. Nazari, K. Faez, M. Amiri and E. Karami, "A novel digital implementation of neuron-astrocyte interactions," *Journal of Computational Electronics,* vol. 14, pp. 227-239, 2015.

[14] J. H. B. Wijekoon and P. Dudek, "A CMOS circuit implementation of a spiking neuron with bursting and adaptation on a biological timescale," *2009 IEEE Biomedical Circuits and Systems Conference,* pp. 193-196, 2009.

[15] J. H. B. Wijekoon and P. Dudek, "VLSI circuits implementing computational models of neocortical circuits," *Journal of Neuroscience Methods,* vol. 210, no. 1, pp. 93-109, 2012.

[16] C. Mead, Analog VLSI and Neural Systems, Reading: Addison-Wesley, 1989.

[17] S. Shapero, C. Rozell and P. Hasler, "Configurable hardware integrate and fire neurons for sparse approximation," *Neural Networks,* vol. 45, pp. 134-143, 2013.

[18] A. S. Cassidy, G. J. and A. G. Andreou, "Desing of silicon brains in hte nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization," *Neural Networks,* vol. 45, pp. 4-26, 2013.

[19] E. Neftci, E. Chicca, G. Indiveri and R. Douglas, "A systematic Method for Configuring VLSI Networks of Spiking Neurons," *Neural Computation,* vol. 23, pp. 2457-2497, 2011.

[20] G. Indiveri, E. Chicca and R. Douglas, "A VLSI Array of Low-Power Spiking Neurons and Bistable Synapses with Spike-Timing Dependent Plasticity," *IEEE Transactions on Neural Networks,* vol. 17, no. 1, pp. 211-221, 2006.

[21] P. Livi and G. Indiveri, "A current-mode conductance-based silicon neuron for Address-Event neuromorphic systems," *International Symposium on Circuits and Systems,* pp. 2898-2901, 2009.

[22] E. Radhika, S. Kumar and A. Kumari, "Low power Analog VLSI Implementation of Cortical Neuron with Treshold Modulation," *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on,* pp. 561-566, 2015.

[23] D. Badoni, M. Giulioni, V. Dante and P. Del Giudice, "An aVLSI Recurrent Network of Spiking Neurons with Reconfigurable and Plastic Synapses," *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on,* pp. 1227-1230, 2006.

[24] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley and R. Douglas, Event-based Neuromorphic Systems, Wiley, 2015.

[25] E. Culurciello, Etienne-Cummings and K. A. Boahen, "A Biomorphic Digital Image Sensor," *IEEE Journal of Solid-State Circuits,* vol. 38, no. 2, pp. 281-295, 2003.

[26] A. Van Schaik, "Building blocks for electronic spiking neural networks," *Neural Networks,* vol. 14, pp. 617-628, 2001.

[27] E. Chicca, F. Stefanini, C. Bartolozzi and G. Indiveri, "Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems," *PROCEEDINGS- IEEE,* vol. 102, no. 9, 2014.

[28] A. Van Scaik and C. Jin, "The Tau-Cell: A New Method for the Implementation of Arbitrary Differential Equations," *Proceedings of the 2003 International Symposium on Circuits and Systems,* vol. 1, pp. I-569-I-572, 2003.

[29] P. Livi and G. Indiveri, "A current-mode conductance-based silicon neuron for Address-Event neuromorphic systems," *2009 IEEE International Symposium on Circuits and Systems,* pp. 2898-2901, 2009.

[30] J. H. B. Wijekoon and P. Dudek, "VLSI circuits implementing computational models of neocortical circuits," *Journal of Neuroscience Methods,* vol. 210, no. 1, pp. 93-109, 2012.

[31] A. Van Schaik, "Building Blocks for Electronic Spiking Neural Networks," *Neural Networks,* vol. 14, pp. 617-628, 2001.