**A CMOS Implementation of a Biological Neuron with Memristor as Synapse**

By

Zhao Li

A thesis submitted to the Graduate Faculty of
Auburn University
In partial fulfillment of the
Requirements for the Degree of
Master of Science

Auburn Alabama
May 7, 2017

Key words: Neuron Models, Memristor, CMOS Neuron



Approved by

Fa Foster Dai, Chair, Professor of Electrical & Computer Engineering
Bogdan M.Wilamowski, Professor of Electrical & Computer Engineering
Guofu Niu, Professor of Electrical & Computer Engineering

Abstract

A CMOS neuron with 130-nm SiGe technology is presented in this paper to simulate the biological activities of synapses and neurons in human brains. Biological neuron behaviors are introduced in the beginning then comes with its model. A device with name memristor is introduced to simulate synapse of the neuron. The circuit design can be separated into 3 stages: Dendrite input stage, LIF (leaky integrate and fire) stage, synapse stage. The first stage is achieved by a V-I converter and an Axon-Hillock circuit is implemented to simulate LIF stage, the synapse stage is modeled by resistor-memristor voltage divider based on STDP rule. The features of the whole neuron are based on the studies from neuroscience and mimicked by utilizing digital and analog circuits. The purpose of this paper is to give building blocks for future neural network applications.

Acknowledgements

It is an exciting and challenging period to pursue master's degree here at Auburn University. It is my honor that I can have this opportunity to learn and to research at the department of Electrical and Computer Engineering. I am very grateful for everyone who helped and enriched me during this time.

I would like to first express my sincere gratitude to Dr. Fa Foster Dai who has been my research advisor for master's degree. I would like to thank him for his support and guidance throughout my graduate years at Auburn University. His ELEC 6130 (RF Devices and Circuits) and ELEC 6190 (IC Design) are the two courses that give me insight on integrated circuit design. He is always giving feedbacks and suggestions that allow me to have progress on my research within weeks instead of months. I would also like to give my regards to Dr. Bogdan M. Wilamowski and Dr. Guofu Niu for their work to serve as a committee member of my committee. Their comments and advice are not only very helpful but very valuable as well.

I am also grateful for all my associates Dongyi Liao, Zhan Su, Ruixin Wang, Hechen Wang, Yucheng Tong, Haoyi Zhao, Xiao Liu and Ergen Tao. I would like to thank them for the help and discussions that inspired me when I encountered problems that I cannot solve by myself. I would also like to especially thank Dr. Kun Zhao who give me lots of useful suggestions during my research. It is a pleasure to work and discuss

with them all together.

Finally, I would like to thank my parents. Everything would not be possible without their love and support in my life.

Table of Contents

List of Figures

List of Abbreviations

| | |
|---|---|
| ANN | Artificial Neural Network |
| LIF | Leaky Integrate and Fire |
| LTP | Long-term potentiation |
| LTD | Long-term Depression |
| STDP | Spiking timing-dependent plasticity |
| OTA | Operational Transconductance Amplifier |

## Chapter 1 Introduction

Artificial Neural Networks (ANN) are a computational approach, which is based on a large number of artificial neurons, to simulate the way that a biological brain solves problems with a huge cluster of axons that connect neurons. Each individual neuron has a function that treats the weighted summation of all input signals as output. There will be a threshold on each connection such that the output signal of the pre-neuron must exceed the threshold before it is transmitted to other neurons. These kinds of systems are self-trained instead of programmed, so it has a better performance in the certain area like computer vision, speech recognition, such tasks problems are really hard to solve by ordinary rule-based programming [1]. The structure of an artificial neural network is shown below:

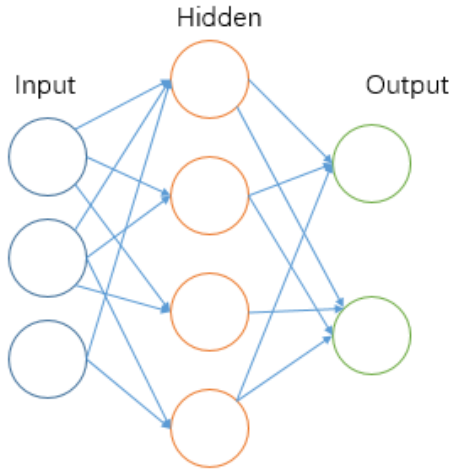



Figure 1.1 Structure of Neural Network.

With one more step of the thought of neural network, if we stack more neurons

with proper connections, can we have a chip that acts as human brains or even ‘smarter’ than human brains? The answer is yes. In the year 2011, IBM presented their prototype of brain-inspired CPU with 256 neurons, 256*256 synapses, and 256 axons. The function of this CPU is still simple for it can only play games like Pong. 3 years later, they gave us a CPU with 1 million neurons, 256 million synapses, and the power consumption is only 70mW, which is 100 times faster and 10,000 times lesser power consumption in the experiment of real-time recognition of human, bicycles, buses and carts that are videotaped 30 frames per second in Stanford University with 80% of accuracy than a laptop [2]. It is still having some disadvantages. For example, it is not a classical Von-Neumann architecture CPU, to make it work correctly, there should be a new type of software language based on neural network, simple logic like 1 and 0 will not work correctly in this new architecture. So, its applications are restricted for most of the software in the world are based on Von-Neumann architecture.

Now let’s focus on the building blocks of the ANN, which is a single neuron. To model the biological behavior of the neuron, we must know how the neurons work and what the electrical characteristics of the neuron is, then we may be able to model it.

This thesis is organized as follows: Chapter 2 will explain the characteristics of the neurons and explain the mechanism of a neuron, then we give 3 different types of the neuron models with simulations from Matlab. After a brief discussion, a certain model will be chosen to implement the neuron; Chapter 3 will cover the synapse of the neuron, a basic device called memristor will be introduced to mimic the synapse; Chapter 4 will present the CMOS implementation of the 3 parts of a single neuron,

simulation from Cadence tool of these parts will be discussed; The goal of Chapter 5 is to give a brief discussion of the applications of artificial-neural-network and give a conclusion.

# Chapter 2 Neuron Behavior and Models

## 2.1 Introduction

### 2.1.1 Basic function of Neurons and Membrane Potential

A neuron is a cell with special functions. Although a neuron has all structures including cytoplasm, nucleus, ribosome and mitochondria like normal cells, its specialty is to deal with the signal through electrical and chemical processes. The foundation of this function is the difference of the potential between the inside and the outside of the membrane of the neuron. This potential is because of the different consistency of sodium ion and potassium ion. For example, the amount of potassium ion inside the membrane is 20 times larger than outside. This is because there is a K-Na pump always transports potassium ions from outside to inside and sodium ions from inside to outside. Because of there is a difference concentration, ions are forced to move from high concentration part to low concentration part through ion tunnels. Finally, the potential caused by K-Na pump and the potential caused by concentration difference will cancel with each other to have a relatively steady state. The following figure shows the process:

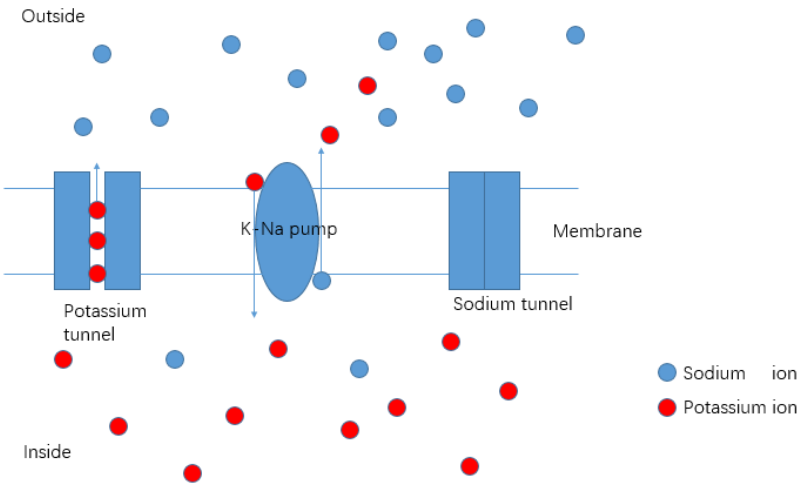



Figure 2.1 Membrane Potential Schematic Diagram.

### 2.1.2 Resting Potential, Active Potential, and Transportation Mechanism

As we know from 2.1.1, there is a relatively steady potential difference between the inside and outside of the membrane, this is called resting potential. When the neuron is not stimulated, potassium ion channel is open, sodium ion channel is closed. The different concentration of the ions gives rise to a positive potential outside the membrane and a negative potential inside the membrane. When the neuron is stimulated, Sodium ion channel will be open. Remember that a number of sodium ions outside the membrane is much larger than inside, so the sodium ions will flock in within a short period from outside to inside which cause a positive potential inside the membrane and a negative one outside. This is called active potential. It is also the main phenomenon indicates the neuron is stimulated. The mechanism for transportation the signal is this: When the neuron is stimulated, we know that the potential inside the membrane will be positive, another part of the synapse is still negative. The difference of the potential of the stimulated part and resting part will cause the current. When the current flows, the

positive potential inside the membrane will ‘travel’ with the current, in other words, the signal is transmitted. Full mechanism is shown in the picture below:

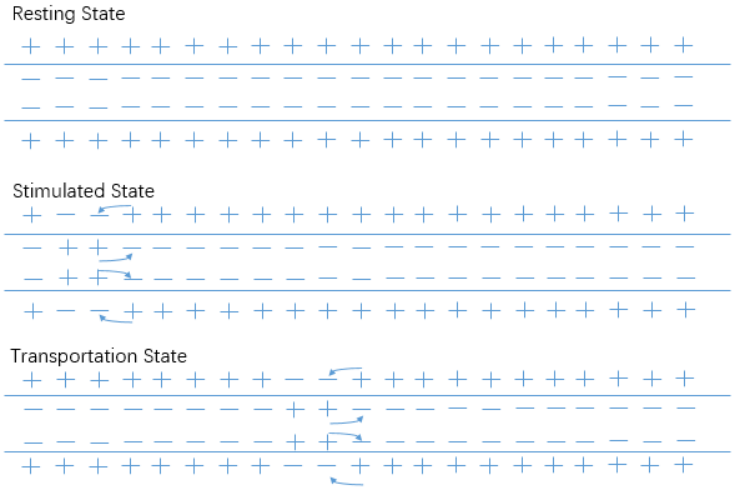



Figure 2.2 Resting, Activation and Transportation Mechanism.

### 2.1.3 Electrical Characteristics of Neuron

From above discussion, it is important to know what is the action potential looks like? From the study of a giant nerve fiber, Dr. Alan Lloyd Hodgkin and Dr. Andrew Fielding Huxley describe the ionic basis of nerve conduction in 1955 [3]. The details of this model will be discussed later, I only show the result of their experiment which explains the electrical part of the action potential of the neuron. From figure 2.3, V-t curve of the stimulated neuron, we can see that when the neuron is stimulated and if the stimulation is strong enough to surpass a threshold, the neuron will generate a ‘spike’, then the voltage falls down even below the resting level, after 2ms, it will be in resting state again.

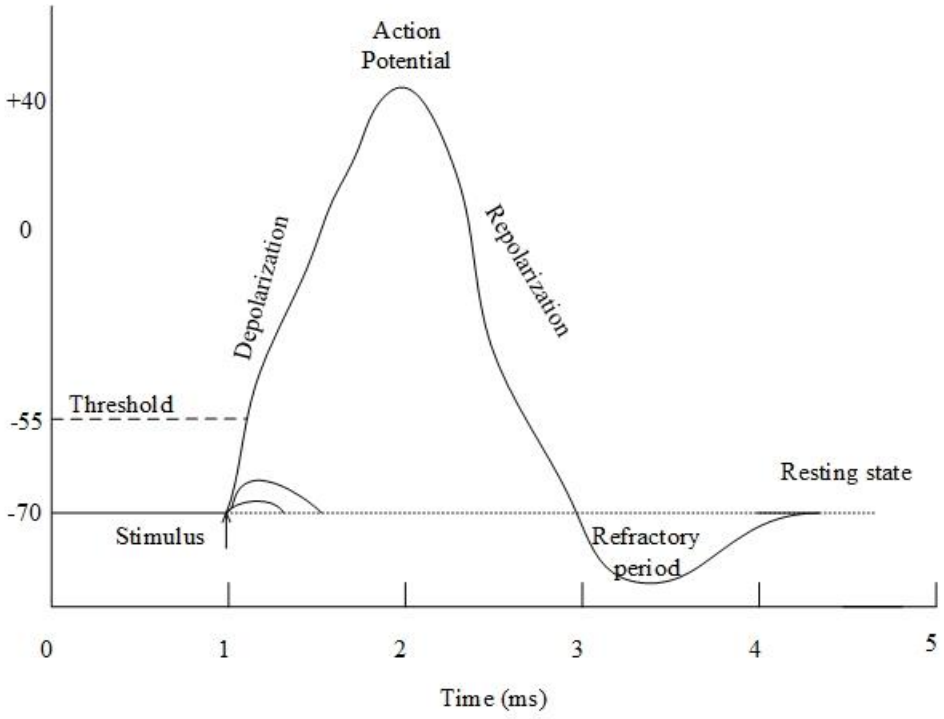



Figure 2.3 Approximate plot of the typical action potential.

## 2.2 Models of the Neuron

### 2.2.1 Hodgkin-Huxley Model

In the year 1952, people do not quite understand the mechanism of the ion channel, they simply believe that an ion channel is nothing but when ion go through, it is similar to capacitor discharge, the relationship of voltage and current is non-linear. Not like other people thought, Hodgkin and Huxley noticed that both dynamic state and steady state exist inside the ion channel. How to describe such complex progress? Their smartness is that they refer the thought from thermodynamics. First of all, they assume the conductance of the ion channel depends on the ion concentration difference between the inside part of membrane and outside part of the membrane; If the assumption is true, in other words, there is a relationship between conductance and concentration, the

distribution of the ions should follow Boltzmann principle in thermodynamics. According to Boltzmann principle, the probability of an ion inside or outside the membrane can be described as:

$$\frac{P1}{P2} = \exp(f) \tag{2.1}$$

Because that P1 and P2 are the probabilities for the same ion, then we know:

$$P1 + P2 = 1 \tag{2.2}$$

Then we have:

$$P1 = 1/(1 + \exp(f)) \tag{2.3}$$

Formula 2.3 is the steady state model of the ion channel.

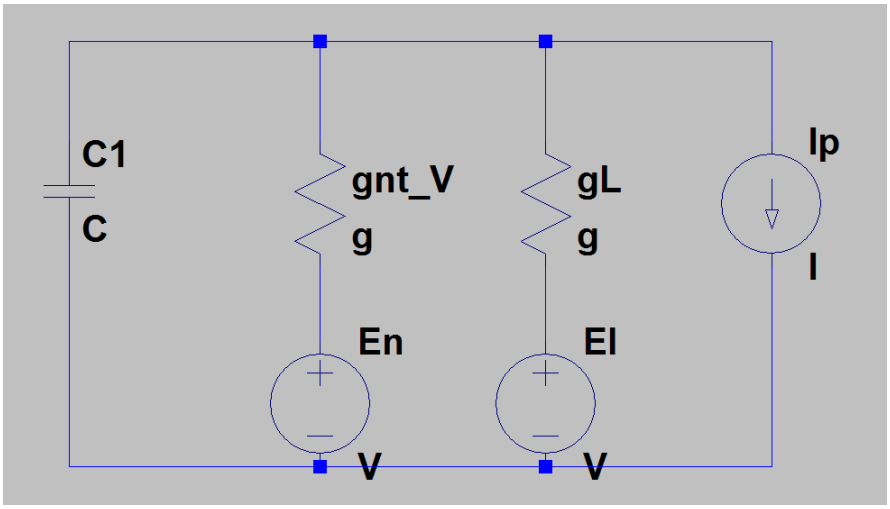



Figure 2.4 Electrical Model of the Membrane.

Figure 2.4 shows the electrical model which represents the biophysical characteristics of cell membranes. The lipid bilayer is represented as a capacitance(Cm). Voltage -gated and leak ion channels are represented by nonlinear(gn) and linear(gL) conductance respectively. The electrochemical gradients driving the flow of ions are represented by batteries(E), ion pumps and exchangers are represented by the current source(Ip).

From the electrical model, we have:

$$I = C_m \frac{dV_m}{dt} + \overline{g_k} n^4 (V_m - V_k) + \overline{g_{Na}} m^3 h (V_m - V_{Na}) + \overline{g_l} (V_m - V_l) \quad (2.4)$$

Where *I* is the current flows the membrane in a unit area. Other parameters are described below:

$$\frac{dn}{dt} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n \quad (2.5)$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1-n) - \beta_m(V_m)m \quad (2.6)$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1-n) - \beta_h(V_m)h \quad (2.7)$$

In formula 2.5,2.6 and 2.7, $\alpha_i\,(outside \rightarrow inside), \beta_i(inside \rightarrow outside)$ is the ith ion channel velocity, depend on voltage Vm and independent of time. $\overline{g_n}$ is the maximum value of the conductance, n, m and h are parameters that related to the activation of the potassium ion channel, the activation of the sodium channel, the deactivation of the sodium channel. The solution for these differential equations gives the expressions that describe the channel.

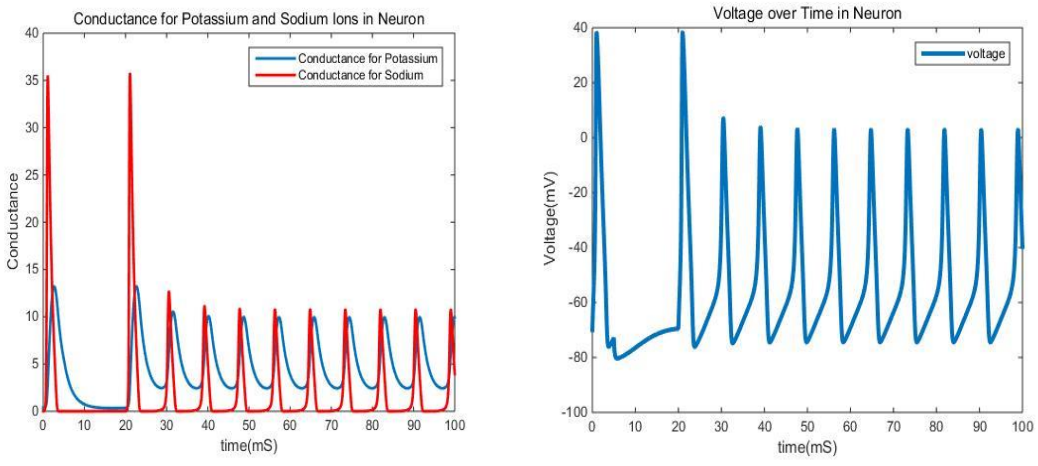



Figure 2.5 Simulation result from Matlab of HH model.

### 2.2.2 Izhikevinch Model

Hodgkin-Huxley model is the first neuron model that mimic the electrical behavior

of the neuron with accuracy, but it is too complex to implement with IC for its complexity of computation. Based on this model, Dr. Eugen M Izhikevich presented a new model to describe the behavior of neuron with more computational simplicity. The experimental result is quite good. His model can be described by the following formulas.

$$\dot{v} = 0.04v^2 + 5v + 140 - u + I \quad (2.8)$$

$$\dot{u} = a(bv - u) \quad (2.9)$$

If v >= 30mV, v = c, u = c+d (2.10)

v: membrane potential;

u: recover variable, take the place of the activation of potassium channel and deactivation of sodium channel;

a: recover time variable of membrane potential;

b: sensitivity that how parameter u was affected by v;

c: recover voltage value after the stimulation of the neuron;

d: reset value of parameter u after the stimulation of the neuron;

With different values of parameters a, b, c and d, this model can simulate all known behaviors of the neuron. Figure 2.6 gives the result of the simulation. These results are the known behaviors of the neuron.

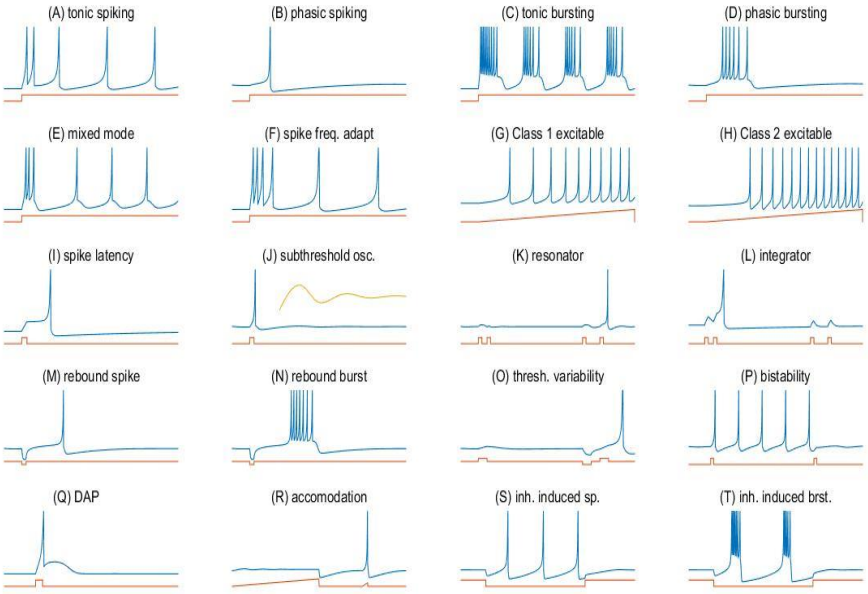



Figure 2.6 Different Behavior of Neurons.

### 2.2.3 LIF Model

LIF (Leaky Integrate and Fire) model is a basic simplification of Hodgkin-Huxley model. This model was presented by Louis Lapicque in 1907, a French neuroscientist. In the model, the neuron is treated as a node, the transmission inside the neuron is not in consideration. The formula describes the potential of the membrane is:

$$\tau_m \frac{du(t)}{dt} = -u(t) + RI(t) \tag{2.11}$$

Input current equals the summation of all currents at the same time, the voltage will leak to zero when there is no input current. There will be a spike generated by neuron when the membrane potential is large than the threshold voltage. We can know that if an action potential is generated at a point of the neuron, this potential will be transmitted to all other points, so we can calculate one point instead of all. The difference between Hodgkin-Huxley and LIF model is that LIF model treats the conductance of the

membrane as a whole constant, so the Hodgkin-Huxley model is simplified. Figure 2.7 is the electrical model of LIF.

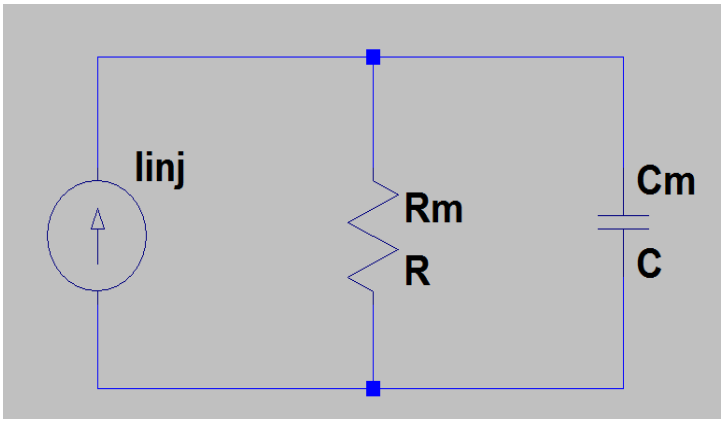



Figure 2.7 LIF Model of Neuron.

The circuit above can be described by:

$$C\frac{dV_m}{dt} = -\frac{V}{R} + I_{inj} \tag{2.12}$$

Iinj: Summation of all input currents;

The solution for Vm(t) when Iinj is constant:

$$V_m(t) = V_m(t_0)e^{-\frac{t-t_0}{\tau_m}} + R_m I_{inj}\left(1 - e^{-\frac{t-t_0}{\tau_m}}\right) \tag{2.13}$$

When Vm surpasses the threshold, it will be set to Vm(t0), so when Iinj is constant, the spikes that generated by the neuron is periodical. Assume t = t(1), a spike is generated, we can find that:

$$u(t) = RI_0\left[1 - \exp\left(-\frac{t-t^{(1)}}{\tau_m}\right)\right] \tag{2.14}$$

As $t \to \infty$ , $u(t) = RI_0$, there will be no spike when $RI_0 <$ Vth. Now assume the second spike is generated at time t = t(2), we have:

$$Vth = RI_0[1 - \exp(-\frac{t^{(2)}-t^{(1)}}{\tau_m})] \tag{2.15}$$

So the period of the spike when the input current is constant is:

$$\mathrm{T} = t^{(2)} - t^{(1)} = \tau_m In(\frac{RI_0}{RI_0 - V_{th}}) \tag{2.16}$$

Where $\tau_m = R_m C_m$ is the time constant.

Figure 2.8 shows the simulation results of LIF model in Matlab.

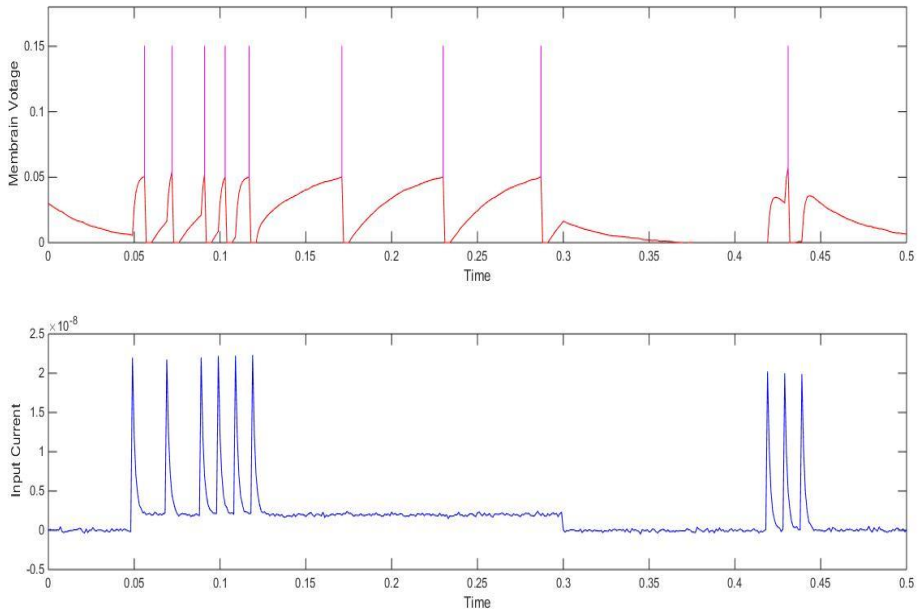



Figure 2.8 LIF simulation result.

From Figure 2.8 we notice that if there is no input, the neuron will be a leaky neuron with its voltage keep on decreasing. When we have some input pulses, the voltage will rise to the threshold and then reset to resting potential. Spikes drawn by hand are added when the spikes should be generated. When we have a constant input, the output will be periodical. When we have dense inputs, there is no output spike when the input is in the refractory period. These features are further discussed and simulated in Chapter 4.

### 2.3 Comparison of different models

LIF model treat membrane as a combination of resistor and capacitor, it treats multiple ion tunnels as a single resistor to simplify the Hodgkin-Huxley model. Izhikevich model is a dimensionality reduction of Hodgkin-Huxley model, there is not much difference between Izhikevich model and LIF model essentially. For LIF model,

when membrane voltage is larger than the threshold voltage, it immediately set to a constant. Izhikevich model treats the membrane voltage as:

$$C\dot{v} = k(v - v_{threshold})(v - v_{th}) + I(t) \tag{2.17}$$

This will cause the membrane potential rise very fast and approaches infinity. Figure 2.9 shows the action potential curve of these two models.

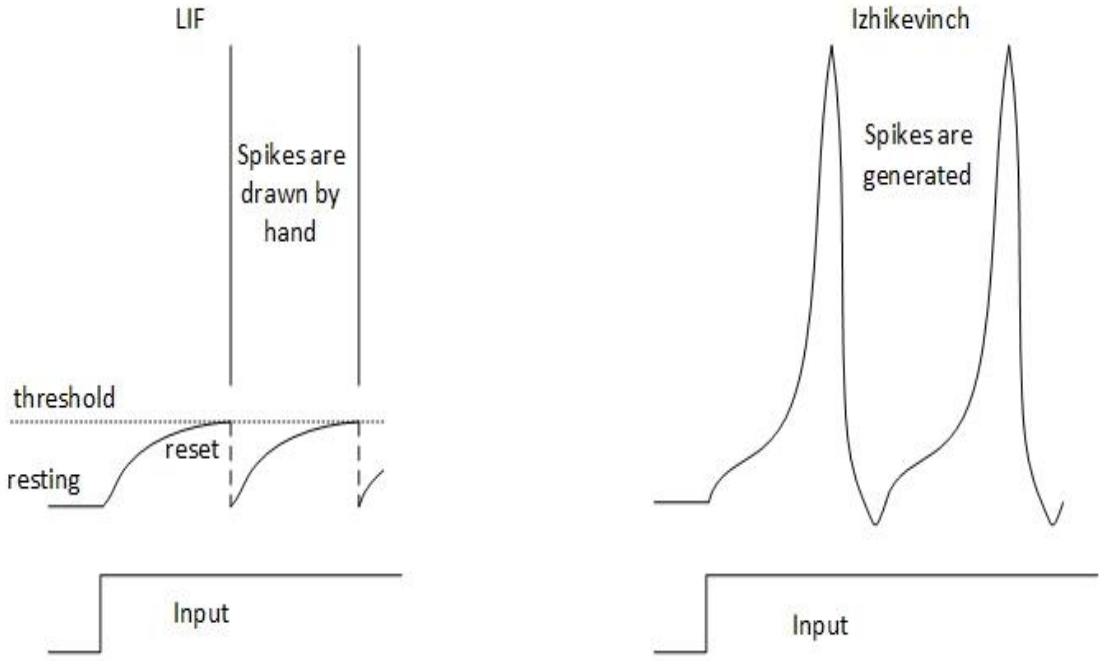



Figure 2.9 LIF spikes and Izhikevinch spikes.

In fact, there are many other models like GIF by Hutcheon and Yarom, QIF and EIF by Ermentrout and Kopell, LIF with adaptation by Treves, IFB by Rinzel, these models are the variations of the introduced models above and most of them were built for neuroscience study [5], [6]. The figure below gives the performance of biological plausibility versus implementation cost [7].

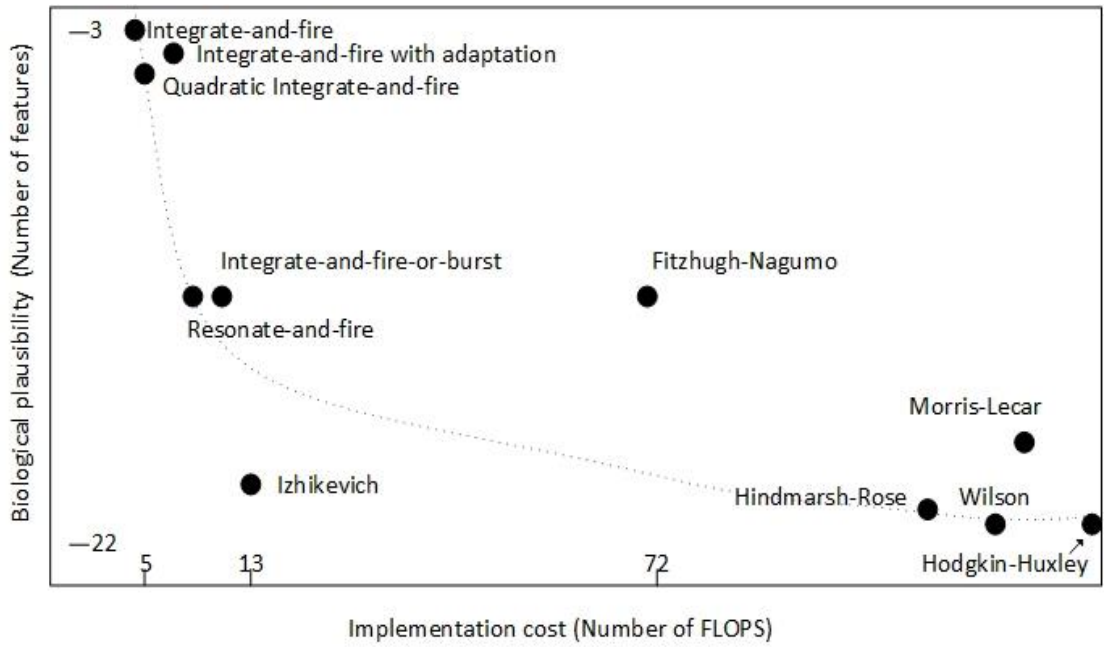



Figure 2.10 Comparison of the neuro properties of neuron models.

From the figure above, we can see that the LIF model is the simplest one to implement. I will choose LIF model as the implementation for the neuron for the consideration of power consumption and simplicity. A single neuron is not that smart until it is connected to each other to form a neural network, in other words, application for neural network may contain hundreds of thousands of neurons. If the neuron is too complicated, there will be much more difficult when we try to connect them together. What's more, when the neurons are stacked, we must take the stability of the circuit into account. The fewer components we introduced into the circuit, the more stability we will have. Though the LIF model does not have so many biological features of a real neuron, it is still got the basic need for a neuron. A lot of studies based on this model indicate that this is an efficient way to achieve some simple functions to simulate the behaviors of a biological neuron. For the applications of our design, we will need only simple functions of the neuron, like when input surpasses the threshold, a spike

should be generated and the membrane voltage will go back to resting potential, or when input signals are applied during the refractory period, there should be no output spikes. Based on these considerations, LIF neuron will meet our requirement.

## Chapter 3 Memristor- Artificial synapse

### 3.1 Introduction of memristor

The memristor is a fundamental device like a resistor, capacitor, and inductor in circuits. This device was first proposed by Dr. Leon O. Chua,1971 [8], and was discovered by D. Strukov, G. Snider, G. Stewart, and R. Williams in 2008 [9]. The memristor is regarded as the fourth fundamental elements in circuits along with inductor, capacitor, and resistor for its properties.

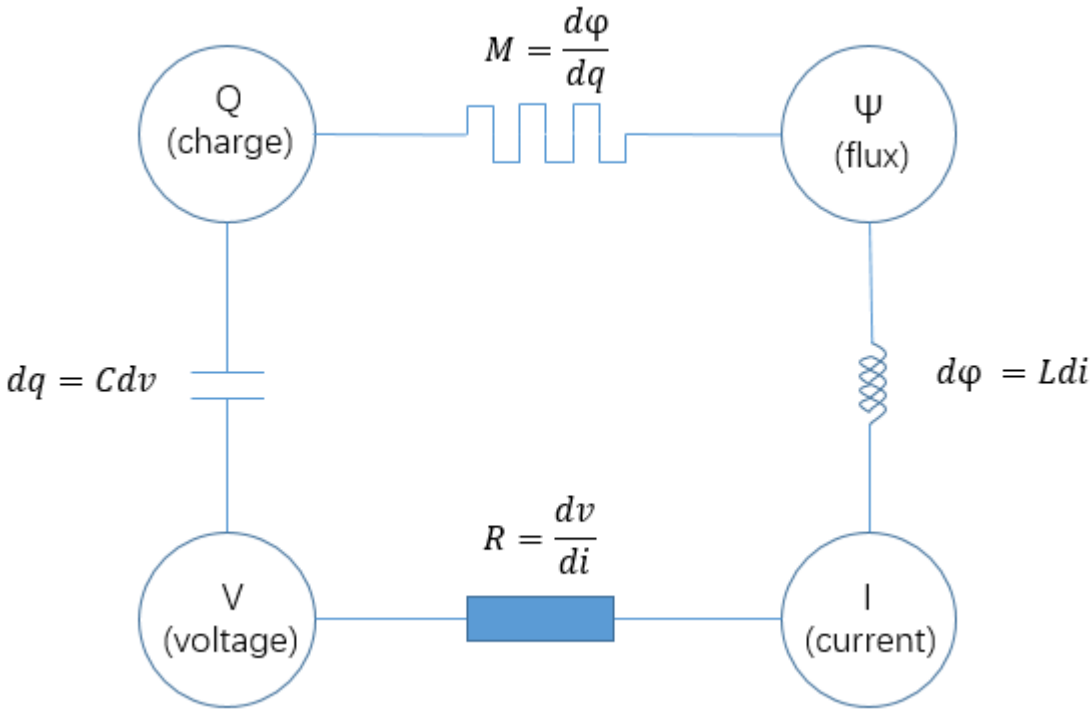



Figure 3.1 Circuit Elements and their relations

From the figure above we know that the memristor connects the charge Q and flux φ together:

$$M(q) = \frac{d\varphi}{dq} \tag{3.1}$$

If the charge Q is a function of time and flux φ is also a function of time, or $q = q(t)$ and $\varphi = \varphi(t)$, from the relations of charge and voltage, flux and current, we will have:

$$M\big(q(t)\big) = \frac{\mathrm{d}\varphi/dt}{dq/dt} = \frac{V(t)}{I(t)} \tag{3.2}$$

Equation 3.2 remind us that the memristor is a device that its resistance varies with the changes of its charge over time. This property can be used as synapse of the neuron in neural networks for different resistance can be treated as different weights of the neural network.

## 3.2 Models of Memristor

### 3.2.1 SPICE Model

The model of the memristor from [9], is fabricated by a two-layer thin film of Ti$O_2$, sandwiched between the platinum contacts. One layer acts like a semiconductor for it is doped with oxygen vacancies. The undoped region is an isolator. The resistance of the whole device can be described as:

$$M(w) = Ron\frac{w(t)}{D} + Roff(1 - \frac{w(t)}{D}) \tag{3.3}$$

W(t) is the width of the doped region, D is the total length of the Ti$O_2$, Ron and Roff are the device resistance when w(t) = 0 and w(t) = D.

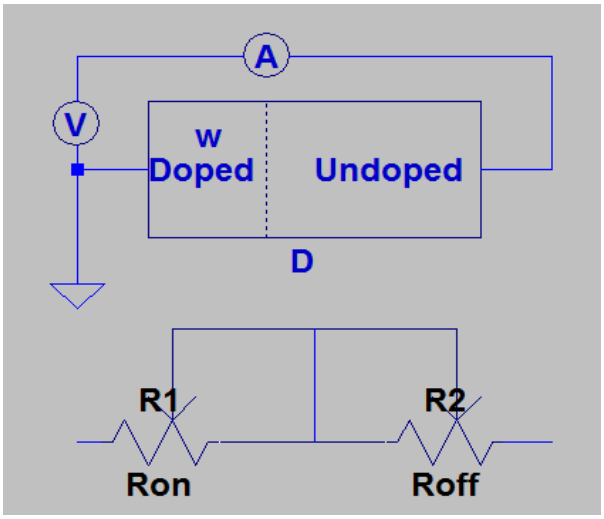



Figure 3.2 Memristor Model

The current and voltage relationship of the memristor can be defined by [10]:

$$v(t) = R(t)i(t) = \frac{d\varphi}{dq}i(t) \tag{3.4}$$

From formula 3.3 and 3.4, the relation between voltage and current of the memristor is :

$$v(t) = \left(R_{on}\frac{w(t)}{D} + R_{off}\left(1 - \frac{w(t)}{D}\right)\right)i(t) \tag{3.5}$$

For Ti$O_2$ memristor, $\frac{w(t)}{D}$ is the state variable, the rate of change of the state variable is defined as :

$$\frac{dw(t)}{dt} = \mu_V \frac{R_{on}}{D} i(t) \tag{3.6}$$

$\mu_V$ is the dopant mobility. Now we have:

$$\begin{aligned} M = \frac{d\varphi}{dq} &= R_{OFF}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)q(t)\right\} \\ &\approx R_{OFF}\{1 - \frac{\mu_v R_{ON}}{D^2}q(t)\} \end{aligned} \tag{3.7}$$

We can see from equation 3.7 that the memristance M is a linear function of the charge q(t). From equation 3.7, we can also see the relationship between memristance and the device length D. The dynamic part of the memristance is proportional to charge q(t), and its coefficient is inversely proportional to device length. In other words, if D is a

big number, the whole memristance of the memristor is almost a constant, q(t) part is too small to measure. That's the reason for we did not discover this device until 2008.

Now we can build a device that has the electrical properties above. From [11], a spice model of memristor was given. Figure 3.3 give the circuit implementation of this model using ideal elements:

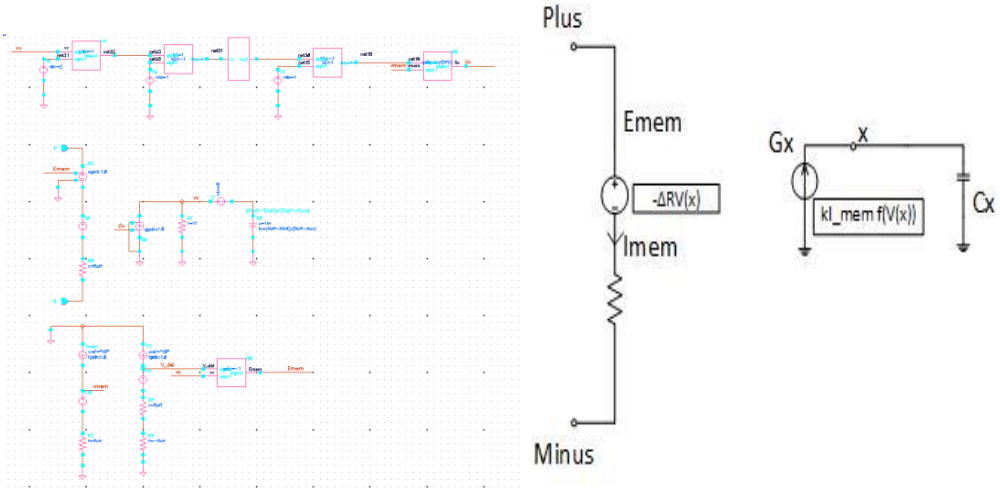



Figure 3.3 Block diagram of Memristor Spice model and Implementation.

The I-V curve of the above memristor is:

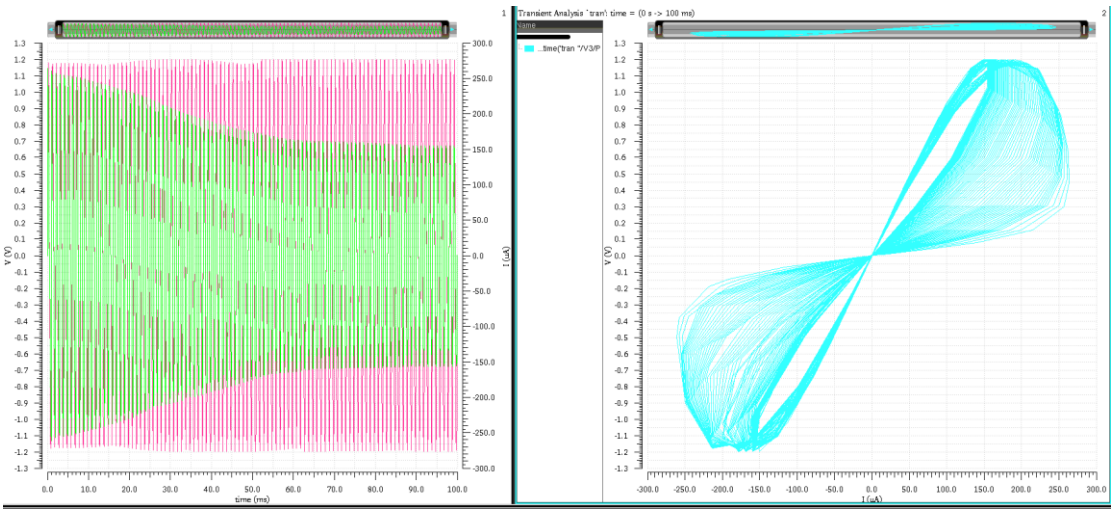


Figure 3.4 I-V curve of the Memristor.

Input: sinusoid wave with 1.2 Vpeak and frequency 1K

Runtime: 100ms

We can see that the memristor is a non-linear device with its resistance varies with input. In other words, the resistance of the memristor can increase or decrease with different input. The following simulation gives this property which is the same property we utilize as the function of synapse in the circuit.

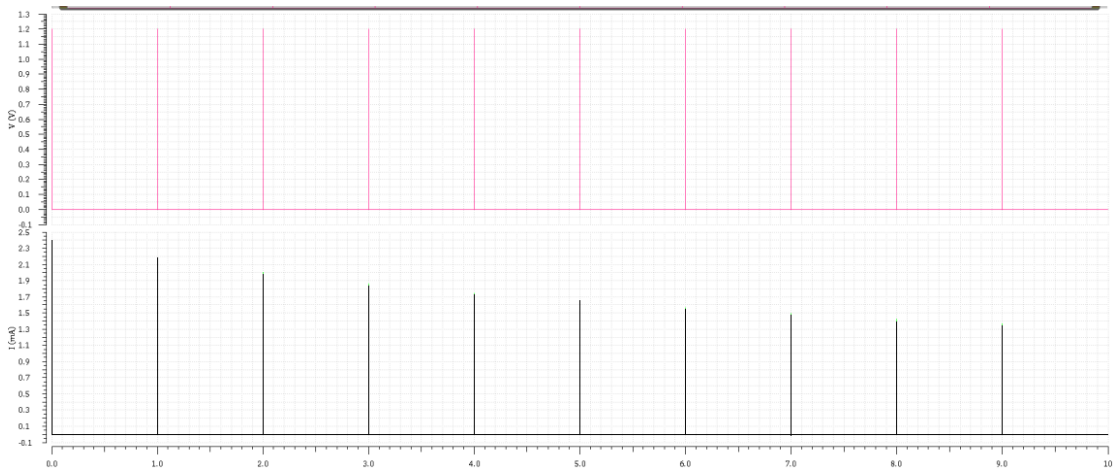


Figure 3.5 Variance of the resistance(Increase).

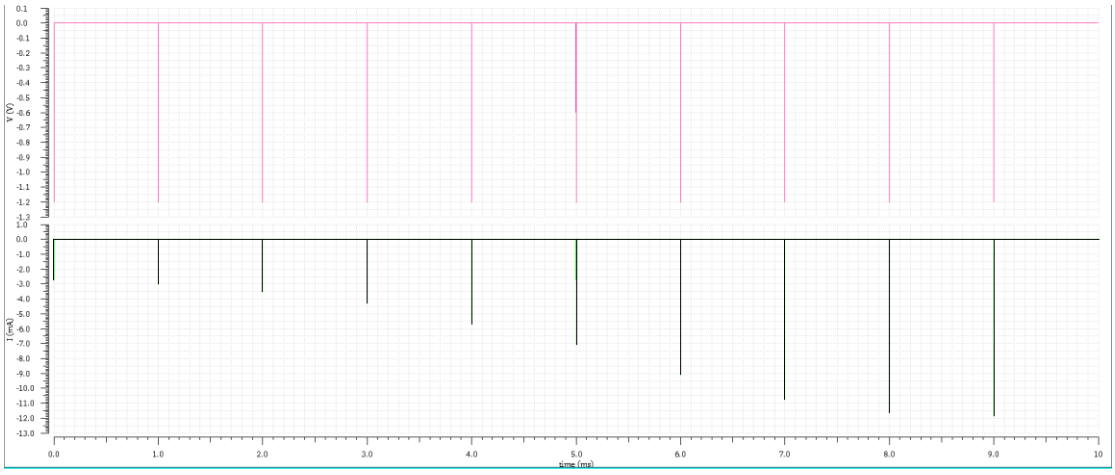


Figure 3.6 Variance of resistance(Decrease).

Table 3.1 Input information

| Figure 3.5 Input | Figure 3.6 Input |
|---|---|
| Square wave Va = 1.2V, Vb=0V, Freq=1K<br>Pulse width: 250n | Square wave Va = -1.2V, Vb=0V, Freq=1K<br>Pulse width: 250n |

Figure 3.5 and Figure 3.6 shows that if there is a positive pulse at the positive port of the memristor, the resistance of the memristor will increase; if there is a negative one, the resistance will decrease. These properties are also referred to as LTP and LTD, which will be discussed in Chapter 4. We can use this property to update the weight of the neural network. There is also a problem within this model. The input could only be applied to the positive port of the device. A real memristor should be a two-port device with positive and negative input ports.

### 3.2.2 TiO2 memristor emulator

Because the fabrication of the memristor is difficult and the cost is really high, people would like to have emulators of the memristor for different applications. In [10], an emulator was presented for people to study and discuss.

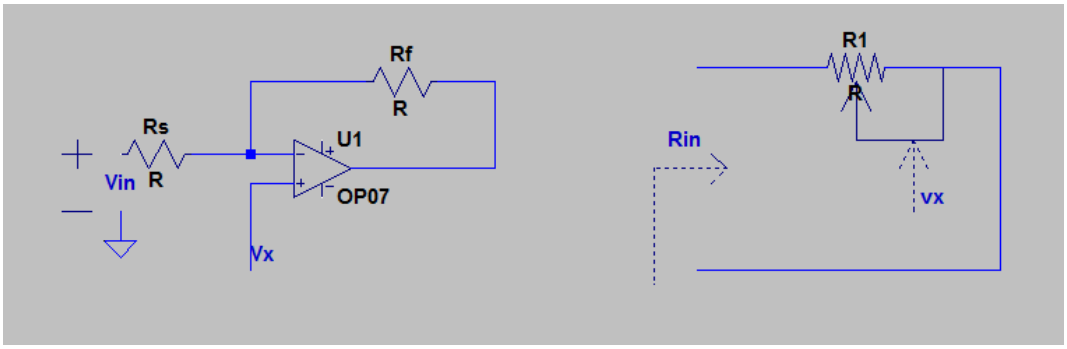



Figure 3.7 Concept of memristor and its equivalent circuit.

From Figure 3.7(a), we can see that the voltage at the input node of the circuit can be written as:

$$v_{in} = R_s i_{in} + v_x \tag{3.8}$$

$i_{in}$ is the input current, $v_x$ is the voltage at the positive node of the op-amp, $R_s$ is the resistance at the negative node. It is assumed that voltage $v_x$ is proportional to input current $i_{in}$, then we have:

$$v_{in} = R_s i_{in} + m i_{in} = (R_s + m) i_{in} \tag{3.9}$$

From equation 3.9, it shows that the resistance of the whole circuit is $R_s + m$ , if we can control m as the integral of the input current, we have the memristor.

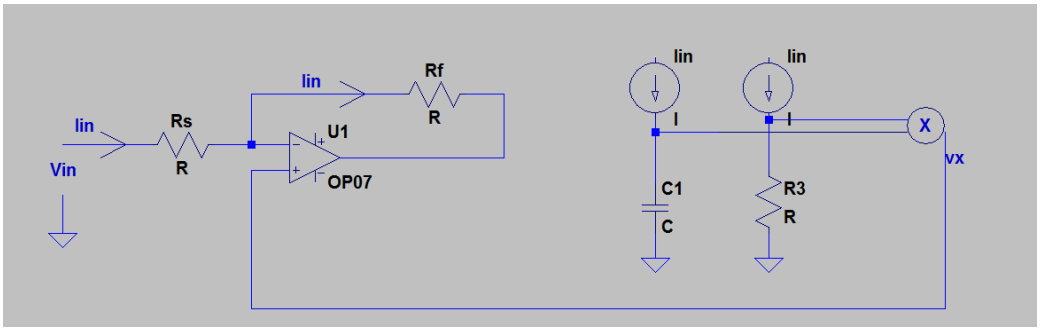



Figure 3.8 Basic configuration of a memristor.

From Figure 3.8, we have:

$$v_{in} = (R_s + \frac{q_C}{C} \times R_T) i_{in} \tag{3.10}$$

We can see that vx at the positive node of the op-amp is generated by the production of the integration of input current and voltage vt. This will give the result that the circuit will act like a memristor.

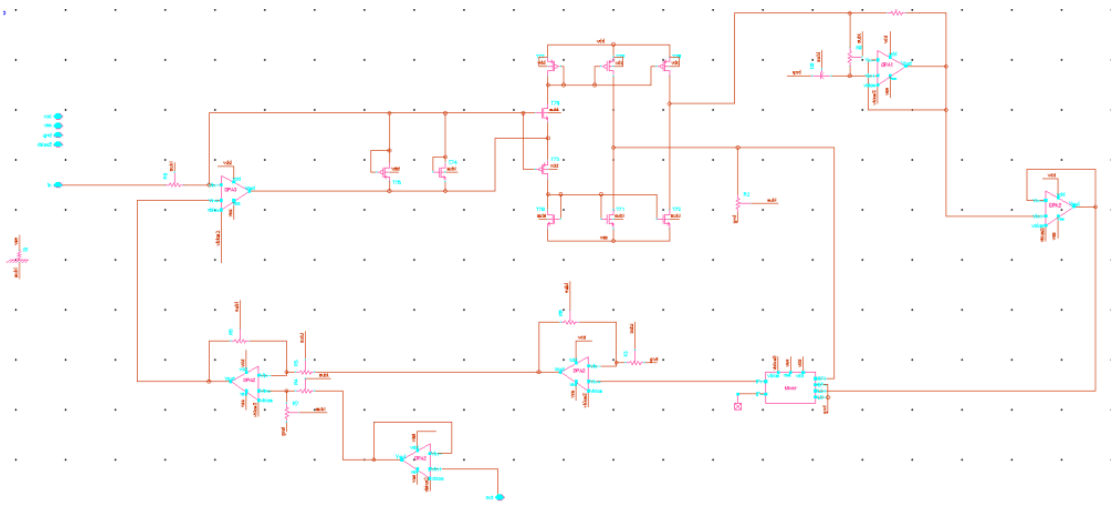


Figure 3.9 Implementation of the memristor.

Figure 3.9 is the implementation circuit based on figure 3.8. Two current mirrors were used to mirror the input current, then they were mixed together as the input of the positive node of the op-amp. From the circuit, we can see that this device is a one-port device. In our design, it should be a two-port device so that it can be connected to another device like a resistor. In paper [10], they claim that with certain connections, this memristor can be connected in series or in parallel, but we found that it doesn't work even in cadence simulation. Though we have adjusted the parameters and made some changes in the original circuit to meet our demand, it is still not working quite well for we would like to build a resistor-memristor voltage divider with proper control to update the weight of the neural network by changing the resistance of the memristor. We also layout the schematic diagram of the memristor based on figure 3.9 for the analytical purpose and hope to have a circuit that mimics the memristor good enough for us to utilize. Figure 3.10 is the layout.

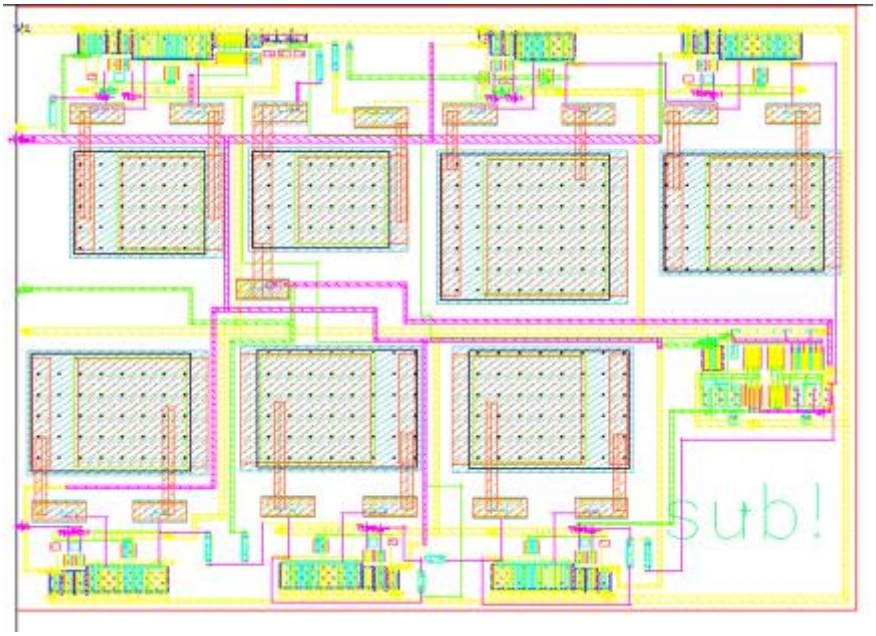



Figure 3.10 Layout of the memristor.

Simulation result of the above circuit shows in figure 3.11. The input signal is a sinusoid waveform with amplitude 700mV and the input frequency is 10KHz.

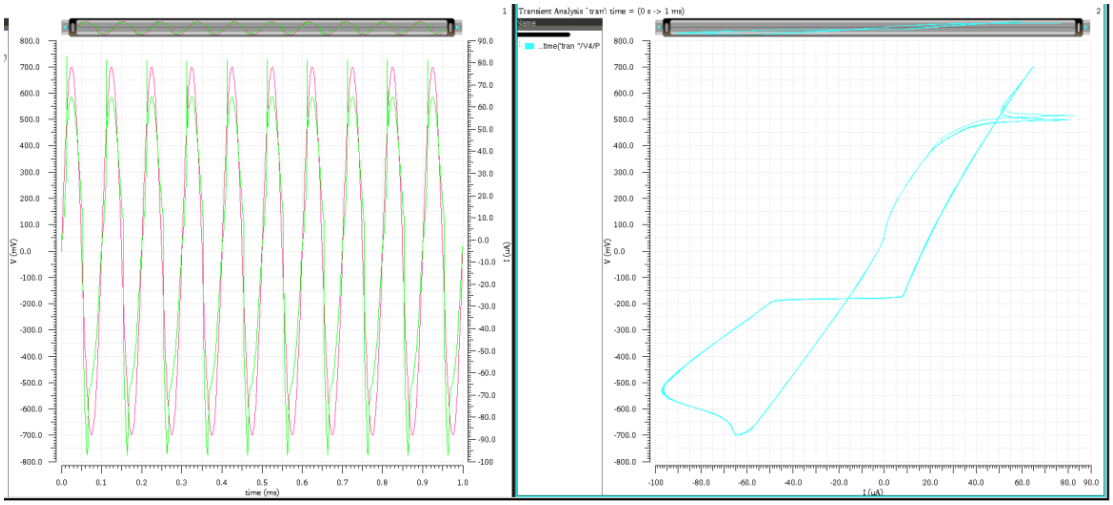


Figure 3.11 Simulation Result of the emulator.

From Figure 3.11 we can see it is not a perfect memristor for its I-V curve is not that smooth. According to Dr. Chua's statement, it is still a memristor.

### 3.2.3 Other models of memristor

In this section, I will introduce several different types of models of the memristor, these models are written in Verilog-A and suitable for EDA tools. [12]

a) Linear Ion Drift Model

This model treats the memristor as two resistors connect in series, one represents the isolated region or the oxide region with high resistance, the other resistor represents the dopants region with low resistance. It is also assumed that the ions inside the device have equal average ion mobility. HP model is the representative memristor of this one.

b) Nonlinear Ion Drift Model

In this model, a non-linear dependence between the voltage and the internal state derivative is assumed. That is to say, the ion mobility is not a constant compare to the model above.

c) Simmons Tunnel Barrier Model

This model assumes the asymmetric and nonlinear switching because of the exponential dependence of the movement of the ions, or changes in the state variable.

d) Threshold Adaptive(TEAM) Model

The TEAM model assumes that there is a threshold of current and polynomial dependence between the derivative of the state drift and the memristor current. The relationship of input current and voltage can be adjusted to have a linear or exponential expression.

Table 3.2 from[12] shows the characteristics of these models.

Table 3.2 The Characteristics of the memristor models

| Model | Linear ion drift | Nonlinear ion drift | Simmons tunneling barrier | TEAM |
|---|---|---|---|---|
| State variable | $0 \leqq w \leqq D$ Doped region physical width | $0 \leqq w \leqq 1$ Doped region normalized width | $a_{off} \leq x \leq a_{on}$ Undoped region width | $x_{on} \leq x \leq x_{off}$ Undoped region width |
| Control mechanism | Current controlled | Voltage controlled | Current controlled | Current controlled |
| Current-voltage relationship and memristance deduction | Explicit | I-V relationship-explicit Memristance deduction-ambiguous | Ambiguous | Explicit |
| Matching memristive system definition | Yes | No | No | Yes |
| Generic | No | No | No | Yes |
| Accuracy comparing practical memristors | Lowest accuracy | Low accuracy | Highest accuracy | Sufficient accuracy |
| Threshold exists | No | No | Practically exists | Yes |

After the simulation of these models, I found that none of these models would meet our frequency requirement. They can work well with the input frequency of a few hundred hertz, but if the frequency surpasses 1KHz, the I-V curve of these models looks like a line. In other words, these models work as a resistor in high frequency. We intended to use TiO2 emulator as our memristor, but the problem is it is a single port device, so it acts like a memristor when we see from the input port of the memristor. What's more, the incremental memristor and decremented memristor has different configurations. In our design, we would like to have the memristor has both characteristics. As a result, the spice model was applied to our design. There are plenty of work to do here. First of all, these memristors are single port devices, which means that input signal can only be applied from one port of it. It will bring some problems

like we have to provide both positive and negative power supplies for the circuit to simulate the increasing and decreasing resistance of the device in the circuit. What's more, when the devices are connected in series or in parallel, one port device need to be modified to get everything works correctly. These problems are discussed in Chapter 5 and treated as future work of the thesis.

## Chapter 4 CMOS Implementation of Single Neuron

A single biological neuron can be divided into 3 parts: dendrite, soma, and synapse. The function of the dendrite is that it can receive a signal from other neuron's synapse and then transmit the signal to soma. Here I would like to indicate that one neuron contains many dendrites that connect to different synapses. Soma of the neuron acts like a comparator when it receives the signal (current) from the dendrite and if the signal is strong enough, in other words, surpass a certain threshold, it will generate a spike. This spike will be transmitted by synapse to other neuron's dendrite. The trick here is the synapse is changed by this spike, either has a stronger connection or weaker connection to post-neuron. This is decided by the type of the spike. In other words, the weight of the neuron is changed. Before the implementation, a learning rule named STDP (Spiking timing-dependent plasticity) will be introduced.

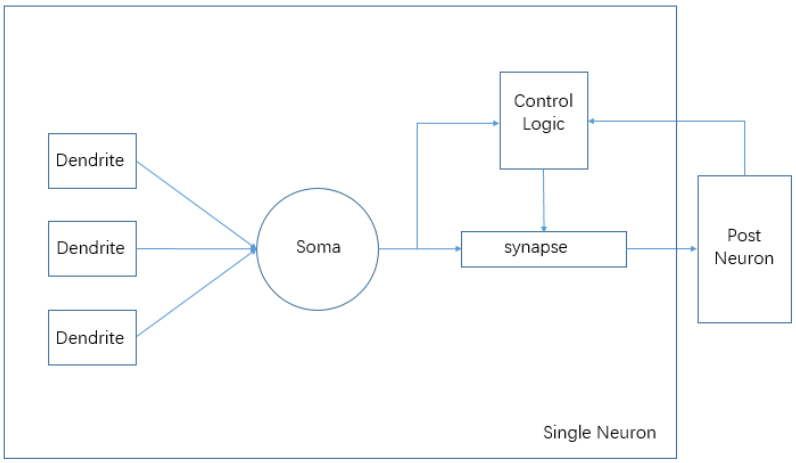



Figure 4.1 Block Diagram of Neuron.

### 4.1 STDP Learning Rule

STDP (Spiking timing-dependent plasticity) is widely used in models of circuits-level plasticity, development, and learning, it is a rule that determines the sign and magnitude of long-term potentiation (LTP) or depression (LTD) by different order and interval between pre-synaptic and post-synaptic spikes[13]. Here, LTP is known as the enhanced connection of the synapse, while LTD is the weaken connection. Guo-qiang Bi and Mu-ming Poo discover that when a pre-neuron spike is ahead of the post-neuron spike, LTP will occur and if the condition is the post before pre, LTD occurs. What's more, the pre-spikes and post-spikes that generated at a very close time point will have more influence than those generated far away in the strength of synapse[14]. These features of the neuron are now known as spike timing dependent plasticity (STDP). It is also learned from [13] that a lot of species (more than 20) including insects and mammals have this feature. For a neural network, it is a good choice to apply STDP rule and it is also a simple one to implement.

### 4.2 Dendrite Implementation

From the discussions above, the dendrite stage of the neuron should have an aggregation input for one neuron contains lots of dendrites. In other words, dendrite stage should add all inputs together, and then transmit the whole signal to next stage. Several source followers are connected to implement the dendrite input stage of the neuron. Figure 4.2 shows the dendrite input implementation. We can see that it is nothing but a V-I converter. It just transforms the input voltage signal into current and

adds them together. The transconductance of the OTA can be tuned to mimic the connection of the dendrites and pre-neurons. For example, smaller transconductance can be treated as a minor stimulus from pre-neuron. Figure 4.3 from [15] gives a basic implementation of input circuits.

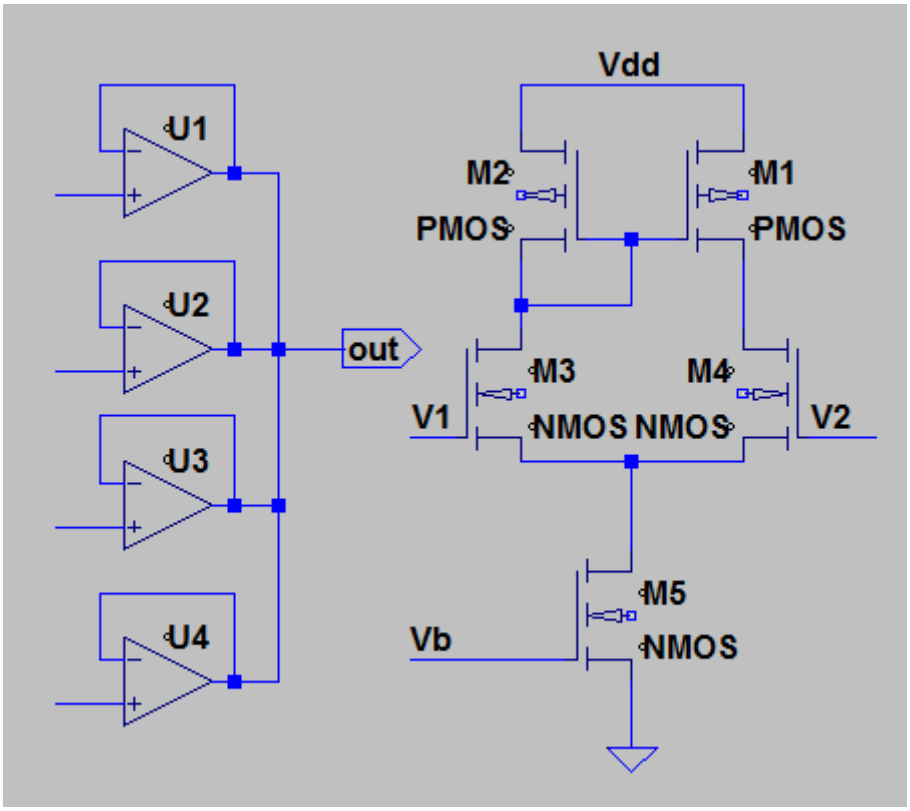


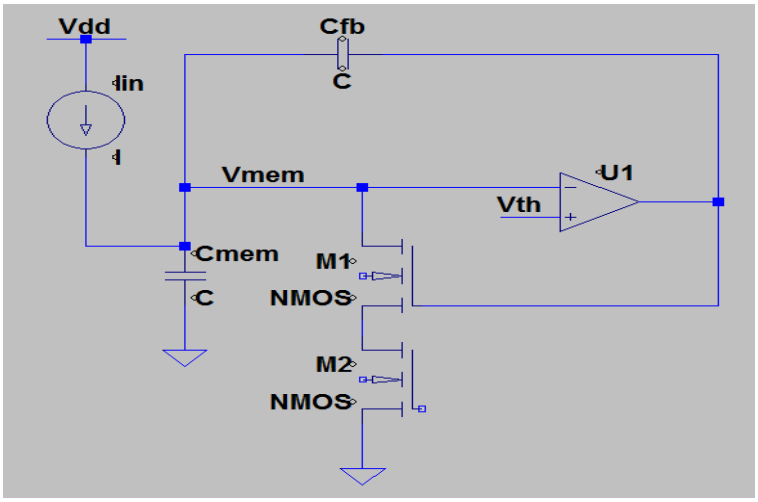

Figure 4.2 Aggregation Input and Single OTA.

Test bench and simulation result in Cadence tool of the above circuit with 130-nm technology are given in figure 4.3. There are 4 input signals with different delays to simulate the spikes that come to dendrite at different time points. We can see that as the number of input increases, the output of the dendrite increases that perform the summation property like a biological neuron performs.


Figure 4.3 Aggregation Input Simulation.

### 4.3 Soma Implementation

The model for the neuron is LIF. From the discussion of Chapter 2, [16] and [17], an Axon Hillock circuit was designed to meet the requirement of soma: If the input is strong enough, there will be a spike generated and then transmitted to next neuron through the synapse. The original schematic diagram from [16] and the cadence implementation is shown in Figure 4.4



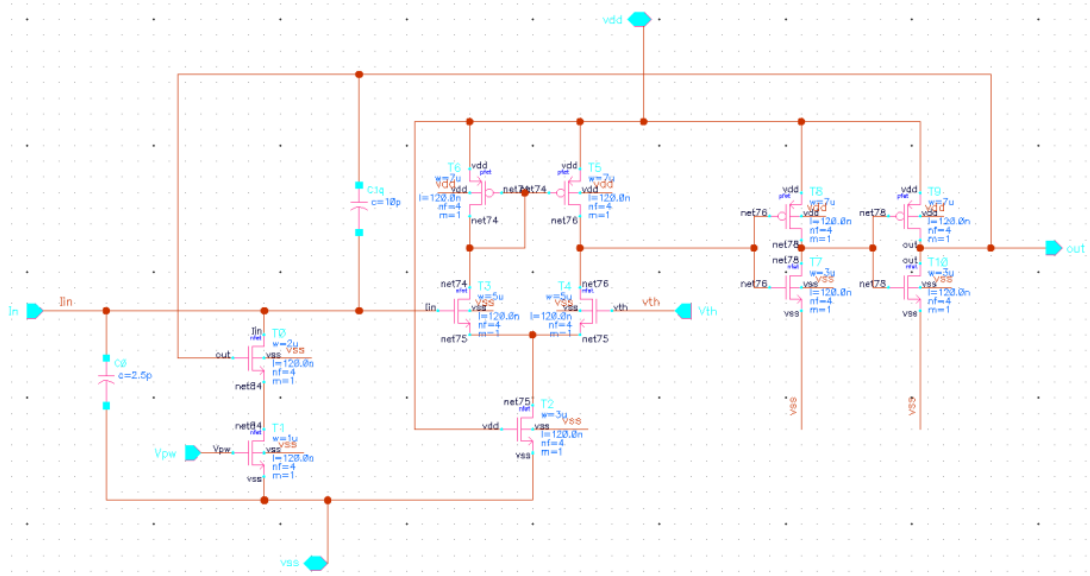


Figure 4.4 Axon-Hillock Circuit Implementation.

In the beginning, $C_{mem}$ which is the membrane capacitance, is charged by the input current $I_{in}$ , when the voltage of the membrane surpasses the threshold voltage ($V_{th}$), $V_{out}$ will go high and the reset transistor is turned on and a positive feedback path is formed through $C_{fb}$. Because the reset transistor is on, $C_{mem}$ will be discharged, then $V_{out}$ will be low again and turn off the reset transistor waiting for next input from aggregation stage. The width of the spike can be tuned by input voltage $V_{pw}$ and the firing rate of the circuit can be adjusted by different values of the ratio of $C_{mem}$ and $C_{fb}$. Figure 4.5 is the simulation result of the LIF circuit.

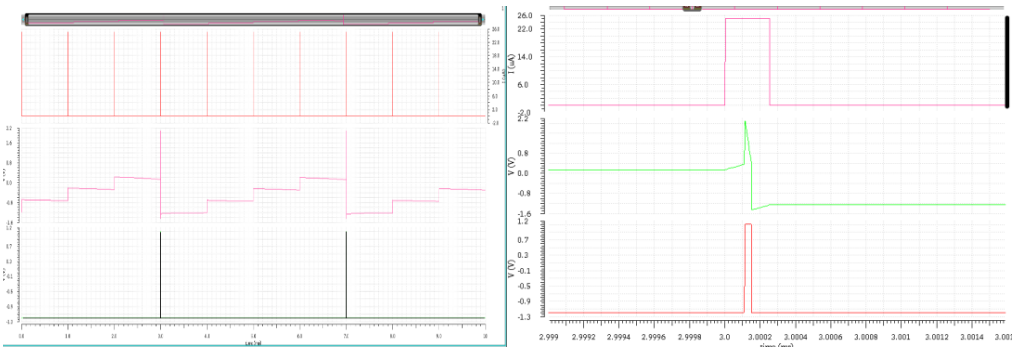


Figure 4.5 Simulation result of the Axon-Hillock Circuit.

The circuit is tuned that every four pulses of the input will trigger one spike. We can also adjust the width of the input which will give rise to a different behavior of the circuit. Figure 4.6 shows another kind of output by adjusting the input signal.

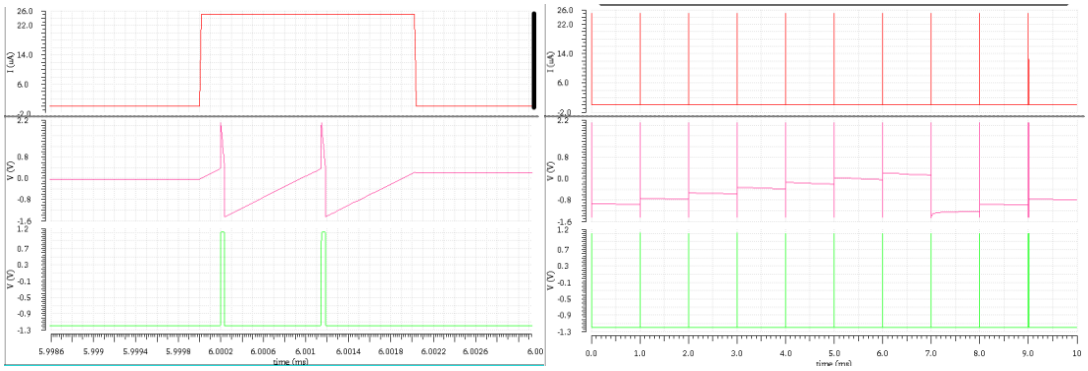


Figure 4.6 Axon-Hillock Simulation with Different Input.

Table 4.1 Input Parameter of Figure 4.5 and Figure 4.6

| Parameter | Figure 4.5 | Figure 4.6 |
|---|---|---|
| Iin | 1KHz with pulse width 250ns | 1KHz with pulse width 2us |
| Vth | 400mV | 400mV |
| Vpw | 150mV | 150mV |
| Vdd | 1.2V | 1.2V |
| Vss | -1.2V | -1.2V |

The difference between Figure 4.5 and Figure 4.6 indicates that if there is a strong input signal from dendrite, the firing rate of the Axon-Hillock circuit is faster than the weak input. This is also a property of biological neurons. It is clear that the magnitude of the output has nothing to do with the input strength (pulse width in circuits), so the strength of received signal of the post-neuron only depends on the synapse. This is really useful for we don’t have to compensate the output signal based on the different

strength of the input signal.

Figure 4.7 is another implementation and simulation of the soma from [24]. The advantage of this implementation is that it is simple and has certain functions like a biological neuron. As we see in the schematic diagram of the neuron, if the input current is strong enough, in other words, there is enough charge in capacitor C0 to turn on M0 transistor, a pulse is generated. When the voltage at the output node goes high, Vgs of M0 will decrease, thus, M0 turns off. M1 and M2 work together as a current mirror, serve as a positive feedback path when M0 turns on. I tried square wave as input signal and found out that no matter how wide the width of pulse is, I can only get one output pulse. To mimic the biological behavior of the neuron, Axon-Hillock circuit is chosen.

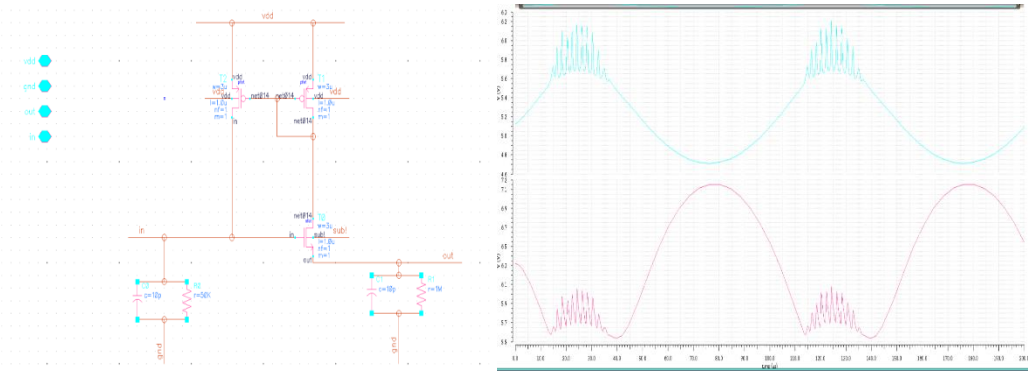


Figure 4.7 Pulse-coupled Neuron

### 4.4 Synapse Implementation

As we discussed in 4.1, to update the weight and obey the STDP rule, a resistor-memristor voltage divider with proper control is implemented. Figure 4.8 is the schematic diagram of update circuit. In Figure 4.8, we use 3 sets of transmission gates to update the weight of the synapse. According to STDP rule, if the pre-neuron spike is generated before the post-neuron, LTP should happen to strengthen the connection

between pre-neuron and post-neuron. If the condition reversed, in other words, pre-neuron spike generated after post-neuron, LTD should happen to weaken the connection.

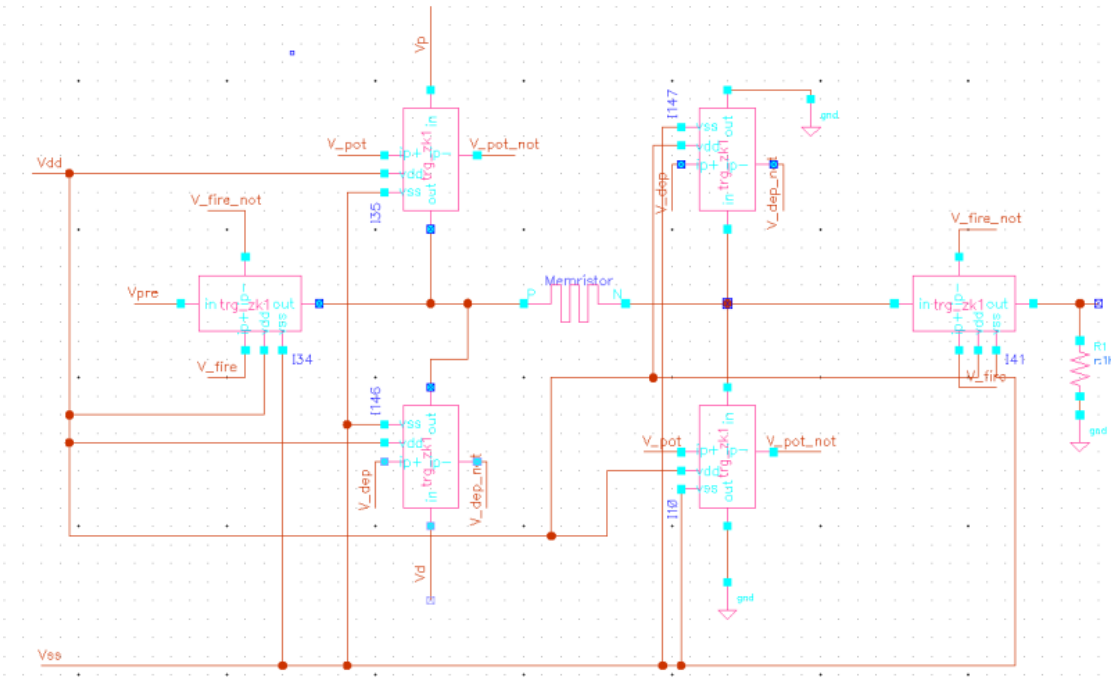



Figure 4.8 Synapse Update Circuit.

Here, $V_{pre}$ represents the spike from pre-neuron, $V_p$ and $V_d$ represent the LTP and LTD signal respectively. If there is a signal from pre-neuron, $V_{fire}$ sets to high that allows the pre-neuron signal pass transmission gates. We assume that $V_{pre}$ is modified that it will not exceed the threshold voltage of the next neuron. Now we assume a pre before post condition to figure out how the weight is changed. In this condition, $V_{pot}$ is high and two transmission gates with control signal $V_{pot}$ is open to let LTP signal $V_p$ pass. Remember the discussion in Chapter 3.2.1, the resistance of the memristor will change depend on the input signal. Here we would like to enhance the connection, so the resistance should be smaller than before. In our model, the resistance of the memristor varies from 100Ohms to 10,000Ohms, and it is connected with a resistor with resistance 1000Ohms. If the initial resistance of the memristor is 4000Ohms and

the input voltage is 1V, the output voltage is 0.2V; after the training, the resistance of the memristor falls to 500Ohms, with the same input voltage, the output is 0.67V. The process of LTD is almost the same, the difference between LTP and LTD is the polarity of $V_p$ and $V_d$.

From the discussion above, the essential part of the update weight circuit is the control logic. The schematic diagram of the control logic shows in Figure 4.9[18]. The upper one is the logic when the pre-neuron signal arrives before the post-neuron signal, the bottom one is post before pre. Delay units are introduced for the training should happen during the refractory period of the input signal according to STDP rule.

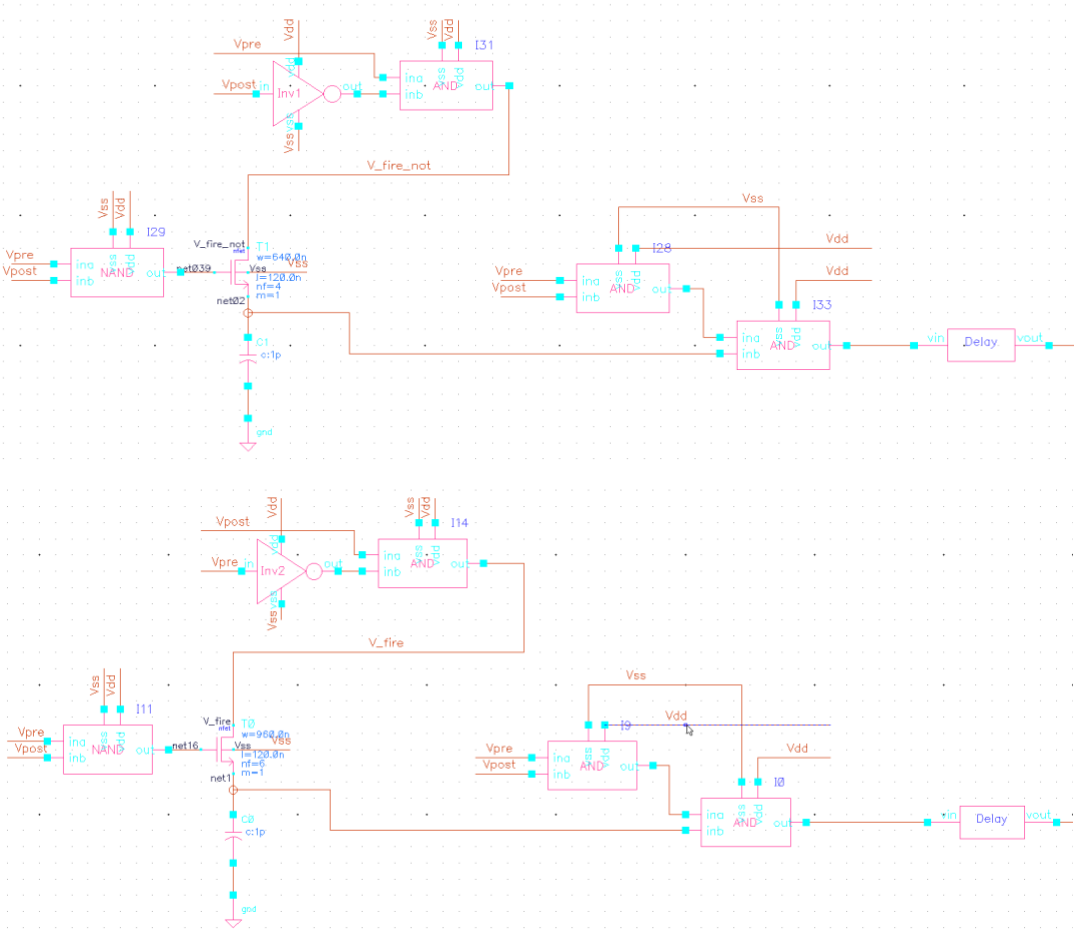
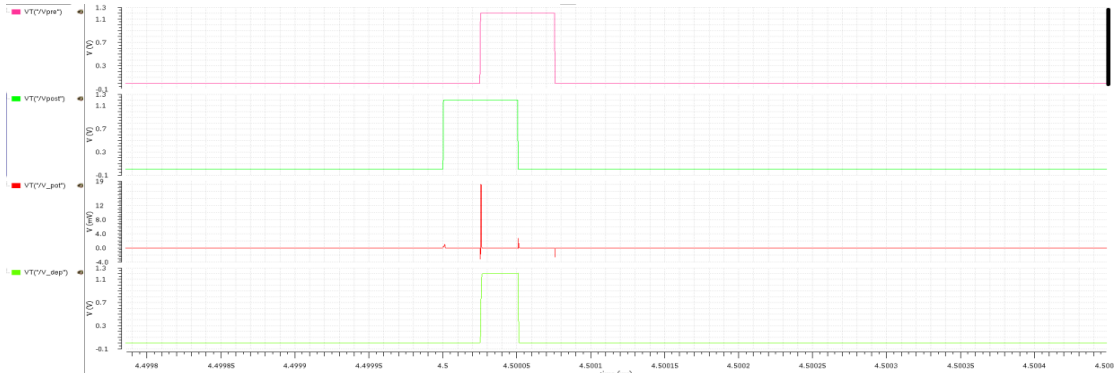



Figure 4.9 Control Logic of STDP.

The memristor should change its resistance only when pre-neuron pulse and post-neuron pulse happens at almost the same time point. From Figure 4.9, we can see that the capacitor in the circuit serves as a memory. When there is an only pre-neuron pulse or post-neuron pulse, the capacitor is charged, in other words, it is working in ‘write’ mode. When pre-neuron and post-neuron pulse happens at the same time, the transistor is off and the capacitor is working in ‘read’ mode and give a training pulse for next stage of the process. Simulation result of the circuit above shows that for the upper part of Figure 4.9, a certain pulse will be generated if the pre-neuron pulse happens before the post-neuron. This pulse is also used for the control signal of the transmission gates and training pulse for the memristor. During this period (refractory period), V_fire will be low and no pulse from pre-neuron is introduced to the memristor. As it is shown in Figure 3.5 and Figure 3.6, the resistance of the memristor can be changed by applying different pulses. Figure 4.10 is the simulation of decremental training pulses generated by logic circuits. What’s more, this is the pulse that is transmitted as the input signal of the next neuron.



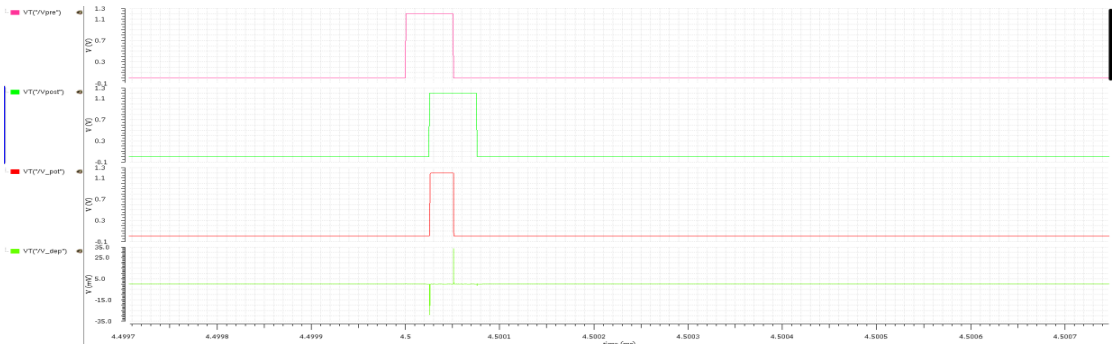


Figure 4.10 Training Signal Generation.

We can see the result of the logic circuit is right for if pre-pulses happen before post pulses, Vp will be generated and for the post before pre case, Vd will be generated. Here we should indicate that the training process should only happen during the refractory period of the input signal according to STDP learning rule. In other words, we must build a delay unit to make it right. Back to Chapter 1 and Chapter 2, from Figure 2.3 we can see what is the refractory period is. In this period, no matter what is the strength of the input signal, There won't be any spikes generated. The last part of Figure 2.8 also proves this property of the neuron. If we apply fast pulses during the refractory period, from Figure 2.8 we can see there won't be any response from the model. So, if we try to modify the weight of synapse, it should happen during the refractory period of the neuron to make sure the updating process will not affect the response of the neuron. The definition of the refractory period can be described the time with no pulses applied to the input port of the neuron. In other words, it is the resting state of the neuron. In the design, as we discussed Vd and Vp should only be applied to memristor-resistor divider with the related transmission gate open, and the control signal should also be delayed. We can use the same Vp and Vd as the control signal of transmission gates. That's why we need the delay unit. This unit is achieved by Verilog-

A for its delay time is long(500μs) compared with another kind of delays. We intend to use RC delay cell or a series of the not gates but none of them works for such a long delay. There is another way to figure it out which is a clock can be applied to this circuit. For the delay, accuracy is not an important stuff to deal with.

## Chapter 5 Conclusion

At present, ANN has a lot of advantages compare to traditional computers in the certain area like pattern recognition [19] [20], motion control [21], target classification [22]. It has lower power consumptions and higher efficiency over traditional computers.

In this thesis, a brief introduction of ANN was given and then neuron models were discussed in the beginning. Three different kinds of neuron models are given and the final choice of the model is LIF for its simple structure and it is the easiest one to stack. Then a basic device named memristor is discussed for its characteristics can be used as the synapse of the memristor. Several models of the memristor are introduced and the layout of one emulator is finished for the test. Then comes the implementation of different parts of the neuron. Dendrite input is achieved by a set of OTA, soma is implemented by Axon-Hillock circuit and synapses can be mimicked by memristor. Based on the study of neuroscience, for training neuron, STDP rule is introduced and a control logic is used to meet the requirement of STDP learning. Compare with a real neuron, the CMOS neuron acts like a biological neuron and it is working much faster than human neurons (less than 100Hz).

There is plenty work to do in future. First, a certain experiment can be done to check if the implementation of CMOS neuron has biological behaviors of the neuron. An associative learning experiment can do the work. Second, when the implementation is proved to have right behavior like biological neurons, a chip with stacked neurons

can be fabricated to have certain utilization like handwriting recognition or other simple functions. Then a processing unit can be built with massive neurons to finish more complicated tasks like pathfinding of smart cars, traffic administration like human beings. For the delay problem in Chapter 4, we can either increase the frequency of the input signal into gigahertz to apply a series of not gates as delay cell or a global clock can be introduced to modify the delay. The last and the most important work is that the memristor in this paper is an ideal one, though we have the emulator, it is still a one-port device. As we mentioned in Chapter 3, a two-port memristor is in need. We can achieve this by duplicate the circuit and a comparator can tell whether the input is applied in positive port or the negative port of the device. When this decision is made, the input signal can have the right path to the circuit, give rise to a two-port memristor. The benefit of a two ports device is that the power supply of the whole circuit can be reduced from -1.2V and 1.2V to 0V and 1.2V. Consider the difficulty of fabrication and high cost of the memristor, there is still a long way to go for practical.

References

[1] McCulloch, Warren, Walter Pitts (1943)， "A Logical Calculus of Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics. 5 (4): 115–133. doi:10.1007/BF02478259.*
[2] Dharmendra S. Modha, " Introducing a Brain-inspired Computer TrueNorth's Neurons to revolutionize system architecture ", *IBM THINKLab*
[3] A. L. Hodgkin and A. F. Huxley, " A quantitative description of membrane current and its application to conduction and excitation in nerve ", *J. Physiol.(1952), 500-544*
[4] Eugene M. Izhikevich, "Simple Model of Spiking Neurons ", *IEEE Transactions on Neural Networks, vol.14 No.6 November 2003*
[5] Lyle N. Long and Guoliang Fang, "A Review of Biologically Plausible Neuron Models for Spiking Neural Networks ", *AIAA Paper No.2010-3540*
[6] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input ", *Biol Cybem(2006)95: 1-19*
[7] Eugene M. Izhikevich, "Which Model to Use for Cortical Spiking Neurons? ", *IEEE Transactions on Neural Networks, VOL. 15, NO.5 SEPTEMBER 2004*
[8] Leon O. Chua, "Memristor-The Missing Circuit Element ", *IEEE Transactions on Circuit Theory, Vol.CT-18 No.5, September 1971*
[9] D. Strukov, G. Snider, G. Stewart, and R. Williams, "The Missing Memristor Found ", *Nature, vol. 453, pp.80-83, 2008.*
[10] Hyongsuk Kim, Pd. Sah, Changju Yang, Seongik Cho and Leon O. Chua, "Memristor Emulator for Memristor Circuit Applications ", *IEEE Transactions on Circuits and Systems,-I: Regular Papers, Vol.59, No.10, October 2012*
[11] Zdenek BIOLEK, Dalibor BIOLEK, Viera BIOLKOVA, "Spice Model of Memristor with Nonlinear Dopant Drift ", *Radioengineering, 18, 210-214.*
[12] Shahar Kvatinsky, Keren Talisveyberg, Dmitry Fliter, Eby G Friedman, Avinoam Kolodny and Uri C. Weiser, "Verilog-A for Memristor Models ", *CCIT Report #801, December 2011*
[13] Daniel E.Feldman, "The spike timing dependence of plasticity ", *Neuron. 2012 Aug.23*
[14] Guo-qiang Bi and Mu-ming Poo, "Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength and Postsynaptic Cell Type ", *Journal of Neuroscience, Dec.15,1998*
[15] Mary Ann C. Maher, Stephen P. Deweerth, Misha A. Mahowald and Carver A. Mead, "Implementing Neural Architectures Using Analog VLSI Circuits ", *IEEE Transactions on Circuits and Systems, Vol.36 No.5, May,1989*
[16] Giaocomo Indiveri, Bernabe Linares-Barranco, Tara Jullia Hamilton, "Neuromorphic silicon neuron circuits ", *Frontiers in Neuroscience, Review article, doi:10.3389/fnins.2011.00073*
[17] Bipin Rajendran, Yong Liu, Jae-sun Seo, "Specifications of Nanoscale Devices and Circuits for Neuromorphic Computational Systems ", *IEEE Transactions on*

*Electron Devices, Vol.60, January 2013*
[18] Jeff Sun, "CMOS and Memristor Technologies for Neuromorphic Computing Applications ", *Technical Report No.UCB/EECS-2015-219, Dec 1, 2015*
[19] Yang Zhang, Yi Li, Xiaoping Wang, Eby G Friedman, "Synaptic Characteristics of Ag/AgInSbTe/Ta-Based memristor for Pattern Recognition ",*IEEE Transactions on Electron Devices, Mar 2017*
[20] Anik Chowdhury, Mrinmoy Sarkar, Aqeeb Iqbal Arka, "Associative memory algorithm for visual pattern recognition with memristor array and CMOS neuron ", *ICECE,2016*
[21] Jing Lu, Jing Yang, Yong-Bin Kim and Kyung Ki Kim, "Implementation of CMOS neuron for robot motion control unit ", *ISOCC,2013*
[22] P. Molchanov, J. Astola, K. Egiazarian and A. Totsky, "Ground moving target classification by using DCT coefficients extracted from micro-Doppler radar signatures and artificial neuron network," *2011 Microwaves, Radar and Remote Sensing Symposium*", Kiev, 2011, pp. 173-176.
[23] Anderson, JA; Rosenfeld, E, eds. (1988). "Neurocomputing: Foundations of Research", *Cambridge, Mass.: The MIT Press. ISBN 978-0-262-01097-9. LCCN 87003022. OCLC 15860311*
[24] Bogdan M.Wilamowski, Mary Lou Padgett, Richard C.Jaeger, "CMOS implementation of a pulse-coupled neuron cell", *Neural Networks, 1996, IEEE International Conference, Volum:4*