ADVANCED ENCODING SCHEMES AND THEIR HARDWARE IMPLEMENTATION FOR BRAIN INSPIRED COMPUTING

BY

LAKSHMI SRAVANTHI KOUTHA

B.Tech., CVR College of Engineering, 2015

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the requirements for the degree of Master of Science.

_____

Chair Dr. Yang Yi

_____

Dr. Christopher Allen

_____

Dr. Glenn Prescott

Date Defended: 05/05/2017

The thesis committee for Lakshmi Sravanthi Koutha certifies that this is the approved version of the following thesis:

ADVANCED ENCODING SCHEMES AND THEIR HARDWARE IMPLEMENTATION FOR BRAIN INSPIRED COMPUTING

_____

Chair Dr. Yang Yi

Date approved:05/05/2017

ABSTRACT

According to Moore's law the number of transistors per square inch double every two years. Scaling down technology reduces size and cost however, also increases the number of problems. Our current computers using Von-Neumann architectures are seeing progressive difficulties not only due to scaling down the technology but also due to grid-lock situation in its architecture. As a solution to this, scientists came up architectures whose function resembles that of the brain. They called these brains inspired architectures, neuromorphic computers. The building block of the brain is the neuron which encodes, decodes and processes the data. The neuron is known to accept sensory information and converts this information into a spike train. This spike train is encoded by the neuron using different ways depending on the situation. Rate encoding, temporal encoding, population encoding, sparse encoding and rate-order encoding are a few encoding schemes said to be used by the neuron. These different neural encoding schemes are discussed as the primary focus of the thesis. A comparison between these different schemes is also provided for better understanding, thus helping in the design of an efficient neuromorphic computer. This thesis also focusses on hardware implementation of a neuron. Leaky Fire and Integrate neuron model has been used in this work which uses spike-time dependent encoding. Different neuron models are discussed with a comparison as to which model is effective under which circumstances. The electronic neuron model was implemented using 180nm CMOS Technology using Global Foundries PDK libraries. Simulation results for the neuron are presented for different inputs and different excitation currents. These results show the successful encoding of sensory information into a spike train.

ACKNOWLEDGEMENTS

I sincerely thank Dr. Yang Yi for her dedicated support throughout my Masters. I am indebted to her constant guidance in helping me understand many analog circuit design concepts that helped me accomplish this thesis. I thank her for her wonderful courses that reinforced VLSI concepts. She is a great inspiration to me in many aspects. Thank you very much Dr. Yi.

I sincerely express my gratitude to Chenyuan Zhao for his immense help in completion of this project. He has been a great guide and has been very patient in answering my questions when I needed technical assistance the most.

I also thank Dr. Christopher Allen for High-Speed Digital Design course that improved my skills applicable to this thesis. I thank Dr. Glenn Prescott and Dr. James Stiles for their courses that improved my knowledge horizons. I extend my gratitude to Dr. John Gibbons for being very understanding throughout my time working with him. I feel grateful for his words of motivation that helped me keep up with deadlines. I would also like to thank Pam, Victoria and others at the EECS department for helping me from time-to-time for updating me about deadlines.

I thank my parents Sri. Satyanarayana Murthy and Smt. Sundari for their unconditional love, motivation, trust and encouragement. I also thank my sister Mahathi and my brother-in-law Vijay for their support here in the States. I thank my friend Rick for all his help both personally and professionally. I also thank Bijal, Kaushik, Hara Madhav, Elaina, Shweta, Jyoti and Sayali for being a great pillar of support during my Masters.

In conclusion, I thank all my well-wishers, classmates and friends who have taught me many things making me the person I am today.

Thank you all.

CONTENTS

ABSTRACT iii
ACKNOWLEDGEMENTS iv
LIST OF FIGURES vii
LIST OF TABLES ix
CHAPTER 1 INTRODUCTION 1
1.1 BACKGROUND AND MOTIVATION 1
1.2 VON NEUMANN VS NEUROMORPHIC: 2
1.3 TIME-LINE OF NEUROMORPHIC ARCHITECTURES: 5
1.4 THESIS OUTLINE: 7
CHAPTER 2: NEURON MODELS 8
2.1 WORKING OF NEURON 9
2.2 ELECTRICAL NEURON MODELS 12
2.2.1 INTEGRATE-AND-FIRE MODEL 12
2.2.2 HODGKIN AND HUXLEY MODEL 14
2.2.3 LEAKY INTEGRATE-AND-FIRE MODEL 17
2.2.4 COMPARISON BETWEEN DIFFERENT NEURON MODELS 18
CHAPTER 3: NEURAL ENCODING SCHEMES 19
3.1 BINARY ENCODING 21
3.2 RATE ENCODING 23
3.3 TEMPORAL ENCODING 26
3.4 POPULATION ENCODING 28
3.5 SPARSE CODING 30
3.6 RANK-ORDER ENCODING 32
3.7 COMPARISON OF DIFFERENT ENCODING SCHEMES 33
CHAPTER 4: NEURON IMPLEMENTATION 34
4.1 IDS EQUATION AND OPERATING REGIONS OF A TRANSISTOR 36
4.2 NEURON IMPLEMENTATION 42
4.2.1 BAND GAP REFERENCE 42
4.2.2 CLOCK GENERATOR 54
4.2.3 CURRENT GENERATOR 56
4.2.4 NEURON DESIGN 59

CHAPTER 5: PRE-LAYOUT SIMULATION RESULTS 62
5.1 BANDGAP REFERENCE SIMULATION 62
5.1.1 OP-AMP SIMULATION 62
5.1.2 BANDGAP REFERENCE SIMULATION 63
5.2 CLOCK GENERATOR 64
5.2.1 COMPARATOR SIMULATION 64
5.2.2 CLOCK GENERATOR SIMULATION 66
5.3 CURRENT GENERATOR 67
5.4 NEURON SIMULATION 68
CHAPTER 6: CONCLUSION 69
REFERENCES 70

LIST OF FIGURES

Figure 1.1: Moore's Law being shown graphically ……1
Figure 1.2: Von Neumann Architecture ……...3
Figure 1.3: Architecture of a Neuromorphic Computing Chip ……...4
Figure 2.1: Neuron ……...8
Figure 2.2: Working of the neuron to generate action potentials…….11
Figure 2.3: Lapicque's Integrate-and-Fire Neuron Model……..13
Figure 2.4: Circuit representing the membrane……...14
Figure 3.1: McCulloch and Pitts neuron model ……..22
Figure 3.2: Step function of the McCulloch-Pitt's neuron with θ……..23
Figure 3.3: Spike firing rate ……..……25
Figure 4.1: Neuron Schematic…….……..35
Figure 4.2: Cross-section of NMOS and PMOS transistors …….37
Figure 4.3: NMOS Transistor ……37
Figure 4.4: Generation of Positive To Absolute Temperature voltage ……..44
Figure 4.5: Typical BGR circuit ……46
Figure 4.6: Schematic of Band Gap Reference……..47
Figure 4.7: CMOS realization of a two-stage amplifier…….47
Figure 4.8: Common-source amplifier with a current mirror active load ……..48
Figure 4.9: Small signal equivalent circuit for the common-source amplifier……...48
Figure 4.10: Small signal model for a MOS transistor in active region …….49
Figure 4.11: Input and gain stages of the two-stage op-amp ……..52
Figure 4.12: Schematic of Operation amplifier……...54
Figure 4.13: Schematic of Clock Generator……54
Figure 4.14: Schematic of Comparator……...55
Figure 4.15: Current-mirror biasing using ideal current source …….56
Figure 4.16: Current-mirror biasing using a resistor …….....56
Figure 4.17: Self-biased current mirror ……..57
Figure 4.18: Additional resistor added to the current mirror ……..58

Figure 4.19: Schematic of the current generator…….58
Figure 4.20: Neuron Schematic……..59
Figure 5.1: Transient simulation of an operational amplifier…….62
Figure 5.2: AC Analysis for the operational amplifier……63
Figure 5.3: Simulation result for Band Gap Reference……..63
Figure 5.4: Simulation results of comparator with one waveform over the other……..64
Figure 5.5: Simulation results of the comparator……65
Figure 5.6: Simulation results of the clock generator…….66
Figure 5.7: Simulation results of the current generator……..67
Figure 5.8: Simulation results of the neuron with excitation current =120 nA……68
Figure 5.9: Individual Spike……...68

LIST OF TABLES

Table 1.1: Difference between Von-Neumann and Neuromorphic Computers……5
Table 2.1: Comparison between various discussed neuron models……18
Table 3.1: Comparison between different neuron encoding schemes……33
Table 4.1: Different VLSI Design Tools……34
Table 4.2: Parameters of each component used in the schematic……36
Table 4.3: Different operating regions and each region's current equations ……42

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

According to Moore's law, the number of transistors per square inch double approximately every two years [1]. This observation was made by Gordon Moore, co-founder of Intel, was described in a paper in 1965 where he predicted that the number of components double every year for approximately ten years. After a decade, 1975 he revised his prediction to be doubling every two years [1].

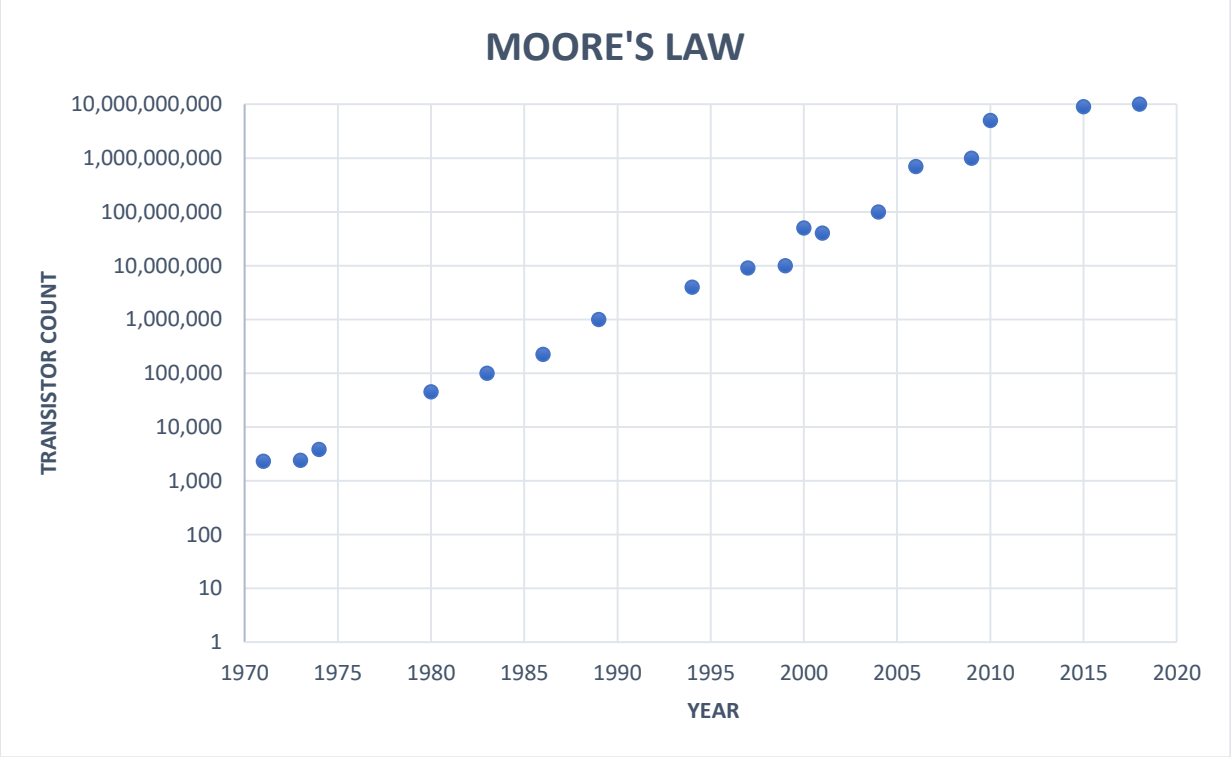

Figure 1.1: Moore's Law being shown graphically (*Source*: [1])

This prediction turned out to be right and the number of transistors per inch has been increasing ever since. With increase in the number of transistors also increased the number of problems. Increasing the number of transistors not only reduces size and cost but also reduces power requirements. One of the problems occurring to semiconductor technologies in the recent

past to increasing the number of transistors is leakage current, in generic terms the power getting wasted is higher than the power being used. There are many such problems being faced due to scaling down of technology and our traditional computer architectures are not meeting the needs of energy efficient and compact systems and the law has slowed down its pace.

Our current computers use Von Neumann architectures and have been so far successful keeping up with the Moore's law. However, they are seeing progressive difficulties not only due to scaling down of technology but also due to a bottle-neck situation in the architecture [2]. To approach these issues and keep up with the emerging technology researchers have decided that, to make advancements and to keep up with reducing the size and power, the architecture of these computers must be changed. Since the traditional architecture has reached its limits and the best computer known to man is the human brain, researchers thought why not model the brain electronically. In 1980 Carver Mead came up with this concept which was defined as VLSI systems having electronic circuits whose function resembles that of the brain [3]. They called these brains inspired architectures, neuromorphic computers. Neuromorphic computing has been trending in the field of computing. But are Neuromorphic Computers better than our traditional computers?

1.2 VON NEUMANN VS NEUROMORPHIC:

Von-Neumann architecture is the most traditional architectures have four functional units; the memory unit, the control processing unit, the arithmetic and logic unit and data paths. But the issue with these architectures is that the data is constantly juggled to and fro between the functional units of the system and memory creating a grid-lock situation. This problem persists even when parallel processors are used [2]. Though our current computers are faster than the brain it gets extremely hard to implement smart features. In an attempt to include these smart features in our regular

processors, Google Corporation made an effort which required 16000 processors for just identifying a cat [4] [5]. This is a very large number for performing one task. The number of processors to implement many such tasks would be very high, which is why we need neuromorphic computers.

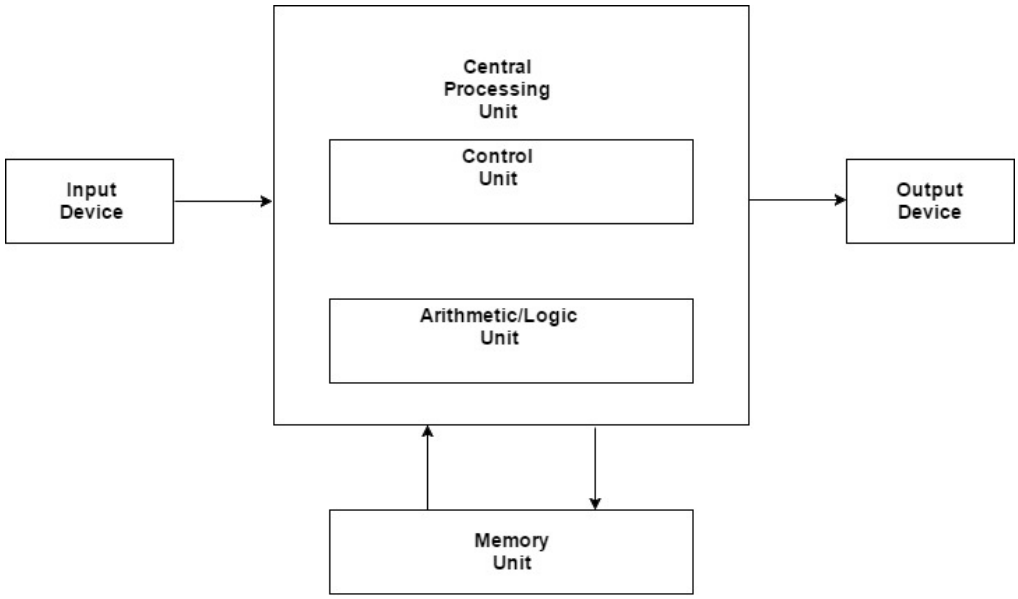

Figure 1.2: Von Neumann Architecture (*Source*: [6])

These issues are however not seen in neuromorphic computers, the main reason being the brain has the capability to change itself per the given requirements. The other important aspect to pay attention to is that brain doesn't need any additional programming like our traditional computer which have huge codes written for their operations. The neuromorphic computers go through something called "learning", which is basically training the brain to learn some things like identifying numbers, alphabets etc. Adapting itself to different conditions is brain's chief quality [2]. Brain being the most mysterious human organ, debates on some of its functions are still active. The brain is far more advanced and much efficient than many of our current computers. Supercomputers require about a few megawatts of power while the human brain only uses 20 watts. The power consumption used by the brain is very less and this meets our requirements of

energy efficient systems. One chief quality of the biological nervous system is that it has the ability to operate even after failure of a few cells, it doesn't fail to operate unlike our traditional processors which fail after a single transistor malfunction [5].

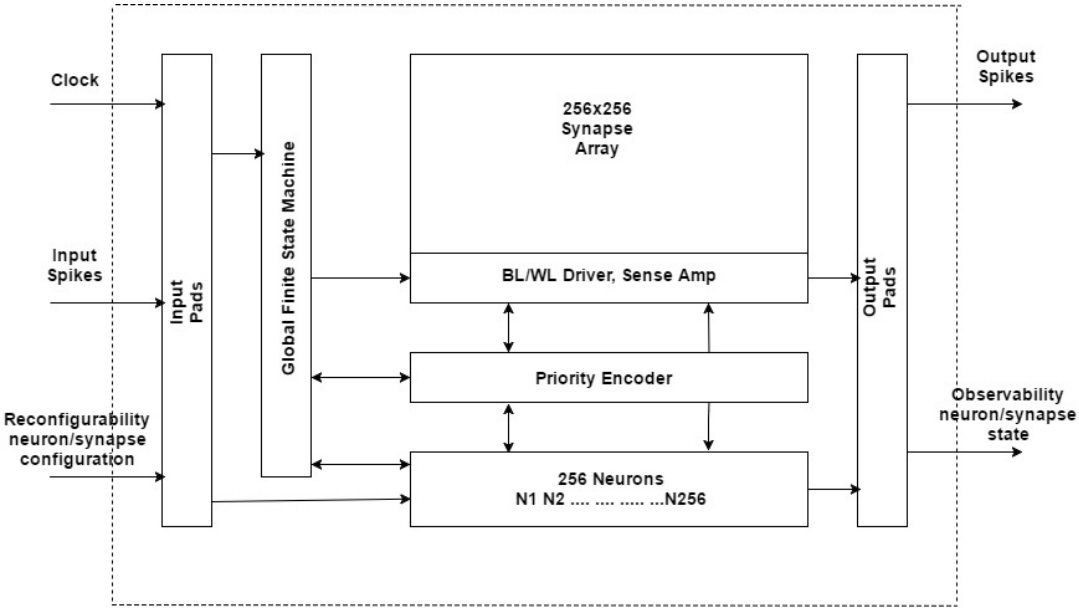

Figure 1.3: Architecture of a Neuromorphic Computing Chip (*Source:* [7])

Looking at the neuromorphic architecture we can certainly say that there is no much juggling of data to and fro between functional units and memory. The bottle-neck or gridlock situations in the Von-Neumann architecture are not seen in the neuromorphic system architectures in addition to providing great balance [2]. The learning process is what makes things easier by making the system ready for eclectic inputs [2].

The mystery content of the brain is what keeps creating complexions for researchers across the world. Man-kind has achieved great heights in every field let it space-exploration, under-water exploration, mineral-exploration etc., but the human brain in many aspects is still a mystery which is why it's modelling is hard.

	VON-NEUMANN ARCHITECTURE	NEURMORPHIC ARCHITECTURE
COMPLEXITY	Yes (for implementing smart features)	Yes (Brain is a mystery)
SPEED FOR BASIC COMPUTATION	Very fast	Slow compared to Von-Neumann
SPEED FOR SMART FEATURES	Slow	Very Fast
POWER CONSUMPTION	Very high	Low
BOTTLE-NECK	Yes	No
CAPABILITY DURING CELL MALFUNCTION	Won't operate as required	Operates almost efficiently
ADAPTING TO SITUATIONS	No	Yes
LENGTHY PROGRAMMING	Yes	No
SIZE	Huge	Small
COST	Cheap (In comparison to Neuromorphic chips)	Expensive (This is a new territory being explored, prices will get lowered if everything is Figureured out)

Table 1.1: Difference between Von-Neumann and Neuromorphic Computers

1.3 TIME-LINE OF NEUROMORPHIC ARCHITECTURES:

With these concepts, many people came forward to implement this idea into practice. They have tried to bridge the gaps between the human brain and computers. Field Programmable Neural Array was the first ever attempt in this field and this dated back to 2006. This system was analogous to Field Programmable Gate Array. This was the first-time neural networks were implemented on silicon [3]. Following this achievement, in 2011 researchers at Massachusetts Institute of Technology implemented an analog neural network on a chip using 400 transistors [3] [8]. Following this in 2012 at Purdue, the Spintronic researches designed a neuron which used

memristors and claimed that their design required even low power than the designs present during the time.

Researchers at HP labs developed a new device called neuristor in which they claimed to have used memristors and this device behavior is that of the biological neuron [3]. The innovation in creating neuromorphic systems picked up great pace. Researchers at Stanford University then made an attempt to club analog computation and digital communication techniques for the hardware implementation of the biological brain [9]. The board they built had 16 neurocores and each of this core had about 65 thousand neurons and this board less than 2 watts in power [9]. In October 2013, the European Union started a project called the Human Brain Project based in Switzerland. Among their wide objectives one of their key objective was neuromorphic computing. Karlheinz Meier is the executive member of the Human Brain Project along with overseeing the neuromorphic computing team. In the ten-year tenure of this project they anticipate to simulate the brain and use these simulations in building neuromorphic systems using the VLSI techniques [9] [10]. BrainScaleS and SpiNNaker are products of this project.

In similarities to the Human Brain Project, the White House in April, 2013 issued orders to release funds to study the human brain called the BRAIN Initiative (Brain Research through Advancing Innovative Neurotechnologies Initiative) [11]. The key idea of this project was to understand the brain better and find the reason for neural disorders [11]. A similar project was started in March 2016 in China called the China Brain Project which also targets to study of the brain in its fifteen-year tenure [12].

In addition to all these projects, the most powerful product in this field coming from the States, is the TrueNorth chip by IBM. Dharmendra Modha is the manager and a scientist in Cognitive Computing Group at IBM [13]. It was said that this chip was the only chip that was more close to

the human brain than any other design. This design had a total of 256 programmable silicon neurons. The TrueNorth chip successfully removed the grid-lock or bottle-neck situations seen in the Von-Neumann architectures. This chip consumed only 70 milliwatts of power [14].

1.4 THESIS OUTLINE:

The above description was a detailed background of the neuromorphic architectures and their need in the world. This project focuses on the different encoding schemes followed by the biological neuron and its implementation. Chapter 2 discusses the working of the neuron and different electrical neuron models. Chapter 3 describes the different encoding schemes of the biological neuron. In Chapter 4 the detailed hardware implementation of the neuron is presented. Chapter 5 has results of the neuron. The thesis will end in a conclusion which gives the scope of future work.

CHAPTER 2: NEURON MODELS

From Chapter 1 we learn that the key part of the brain is the neuron and to successfully implement neuromorphic architectures we must understand the function of the neuron. Neurons are the cells that make up the nervous system [15]. Neurons process information and transmit them to rest of the body through other neurons. The neuron picks up signals and communicates with the other neurons for respective action. This transmission happens with the help electrical and chemical signals. Neurons have voltage levels which are used as reference and in situations when the voltage changes drastically an all-or-none electrochemical pulse is released. This all-or-none pulse is called the action potential [16]. In order to construct the neuromorphic chip, it is very important to understand how each element of the biological system works to replicate it close enough.

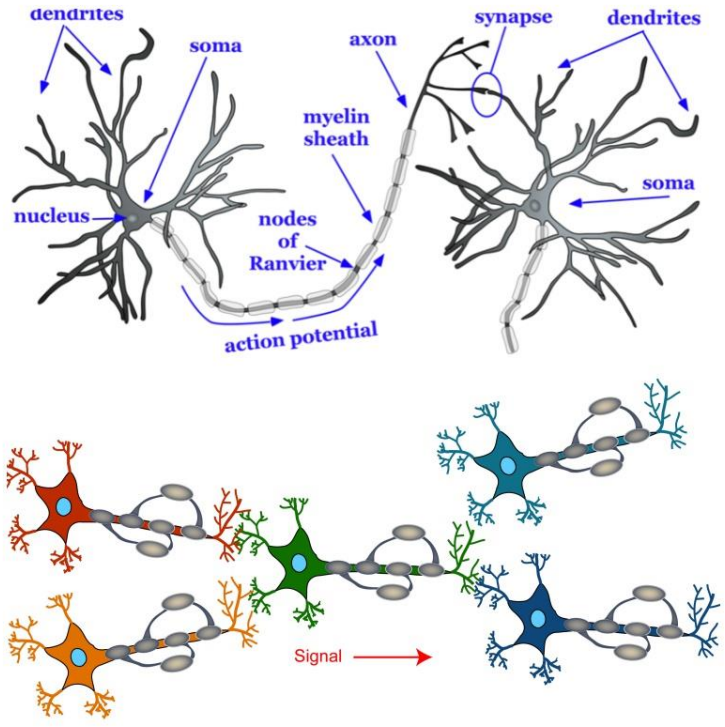

Figure 2.1: Neuron (*Source:* [17] [18])

Figure 2.1 shows the connection of neurons through points called synapses. There are approximately 100 billion neurons in the human body making this system a very complicated one to build [17]. The different parts of the neuron include soma, dendrites and axon. Neurons are connected to each other through their dendrites and this connection is called synapse. The information transfer takes place with the help of axon. It is believed that axons can be as long as a meter [19].

2.1 WORKING OF NEURON [15]:

Neurons receive signals from other neurons through neurotransmitters. When these neurotransmitters bind to receptors on the dendrites, this acts as a chemical signal. This binding opens ion channels, allowing charged ions to flow in and out of the cell converting the chemical signal into an electrical signal. Since a single neuron can have many dendrites receiving input, if the combined effect of the multiple dendrites changes the overall charge of the cell enough, the action potential is triggered. This action potential is transmitted through the axon to the dendrites at speeds of 100m/sec triggering the neurotransmitter at the other end [16].

Neurotransmitters are used to communicate from neuron to neuron while the axon is used for the neuron's internal communication. The cell has electrical charge because of different ion concentration on the inside and outside. The cell initially has negative charge of about -65mV relative to outside. This is called the resting membrane potential. When the binding of neurotransmitter to receptor occurs a ligand gated ion channel opens to allow ions on the outside of the cell to flow in. When positive ions from the outside flow in, the cell becomes less negative causing depolarization. In general, there are many ligand gated channels for different ions thus resulting to the flow of ions in and out. The cell has positive potassium ions and negatively charged

anions on the inside, while it has positive sodium ions, positive calcium ions and negative chlorine ions on the outside. In the end, if there is net influx of positive charge it is called Excitatory Post-Synaptic Potential. If there is net influx of negative charge it is called Inhibitory Post-Synaptic Potential and this makes the cell more negative causing repolarization [15].

A single EPSP/IPSP has very little effect on the membrane potential. If there are enough EPSP's on multiple sites on the dendrites it pushes the membrane potential to a specific threshold value, -55mV. When this happens the voltage gated sodium channels are triggered and these channels open with the changing voltage. When these channels open, positive sodium ions rush into the cell. This results in nearby voltage gated sodium channels to open as well continuing the entire length of the axon and this is our action potential and when this happens we say the neuron is fired [15].

Once many sodium ions rush across the neuronal membrane the cell becomes positively charged relative to the outside up to +40mV the sodium channels stop allowing sodium to flow through them. This is called the inactivation stage. This stage is different from closed state. This happens after depolarization until the cell repolarizes and goes into closed state. It is only during the open state of the channel, sodium is allowed into the cell [15].

We also have the potassium channels that are slow in operation and are not opened until the sodium channels go to inactivation stage. Due to this, potassium ions flow outside the cell in contrast to depolarization. The potassium channels do not have inactivation stage and they end up staying open for a while during which more positive ions leave the cell causing the membrane potential to become more negative or repolarize. During this repolarization stage the cell also depends on a sodium/potassium pump which is an active transporter that moves three sodium ions out and lets two potassium ions in. It is during this repolarization stage the cell is in its Absolute

Refractory Period because the sodium channels are inactivated and won't respond to any kind of stimuli. This Absolute Refractory Period keeps the action potentials from happening too close together in time and moves the action potential in one direction. Due to the moving of ions using the pump the cell become hyperpolarized for a small amount of time to -75mV during which the sodium channels close and the potassium channels stay open. This is called the relative refractory period. After this, the potassium channels close and the cell returns to its resting membrane potential, -65mV [15]. The duration of these spikes is usually about 1-2ms [20].

From this description, we can say that the soma behaves as the CPU (Central Processing Unit), the dendrites act as the input devices and the axon behaves as the output device. Another conclusion that we can make is that neuron is a non-linear device and if we are building a computing processor based on this logic, it will be non-linear as well unlike the Von-Neumann architecture.

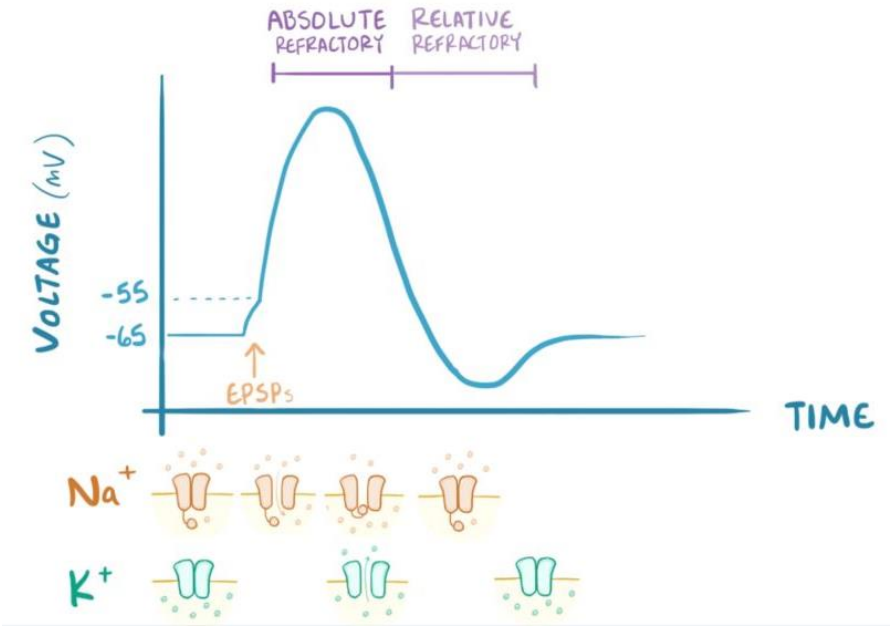

Figure 2.2: Working of the neuron to generate action potentials (*Source:* [15])

2.2 ELECTRICAL NEURON MODELS [21]:

The following are the Electrical models of the neurons currently in existence.

- Integrate and Fire model
- Hodgkin and Huxley model
- Leaky Integrate and Fire model
- Fractional Order Leaky Integrate and Fire model
- Galves Locherbach model
- Exponential Integrate and Fire model
- FitzHugh Nagumo model
- Morris Lecar model
- Hindmarsh Rose model
- Wilson model [22]
- Izhikevich model [22]

2.2.1 INTEGRATE-AND-FIRE MODEL:

The integrate and fire neuron model was one of the neuron models [21]. This was developed by Louis Lapicque in 1907. According to this model the neuron's electrical model is a capacitor parallel to a resistor [23].

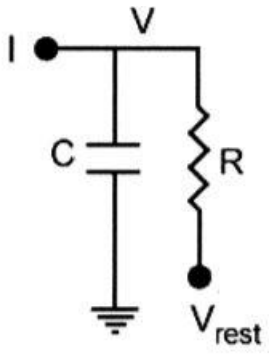

Figure 2.3: Lapicque's Integrate-and-Fire Neuron Model [23].

According to Lapicque, this circuit behaves like "the capacitance and leakage resistance of the cell membrane" [23]. However, there were arguments about this circuit that it does not have the capability to fire spikes. Defending this Lapicque stated that by the nature of capacitor, the capacitor would charge to its threshold causing action potentials to be triggered. After the charge of the capacitor, it would discharge making it come back to resting membrane potential.

From the definition of capacitance, we can say that

$$Q = C\,V$$

Where, Q = Charge

C=Capacitance

V=Voltage

Applying derivation with respect to time to the above equation we get,

$$I(t) = C\,\frac{dV(t)}{dt}$$

Here, I(t) = excitation current

$C=C_m$=Capacitance in the membrane

$V(t)=V_m(t)$=Membrane Potential

In simple words, when there is an excitation current on the neuron, the membrane potentials reach a threshold after which the spiking occurs. This model at this stage doesn't include the refractory period of the cell going below resting membrane potential and it coming back to its original state. But this can be implemented using this design.

2.2.2 HODGKIN AND HUXLEY MODEL:

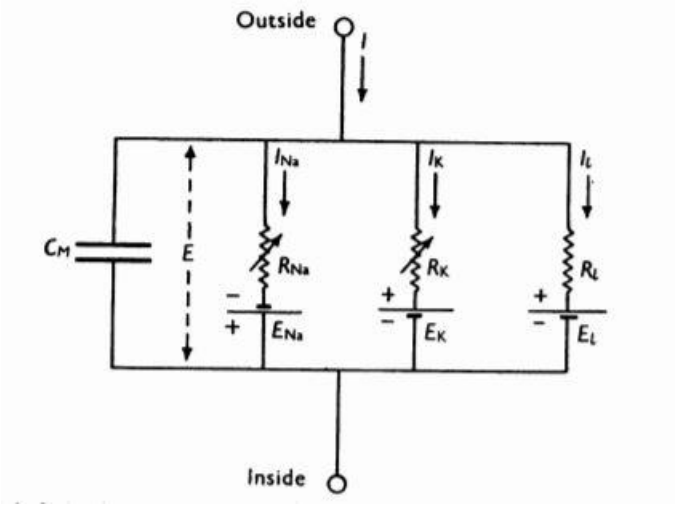

Figure 2.4: Circuit representing the membrane [24].

If we observe the circuit it represents the membrane very closely. Like discussed in section 2.1 of this thesis, we know that the main ions that cause potential difference between outside of the cell and the inside are positive Sodium and Potassium ions. These ions cause ionic current in the cell. There is another current often termed as leakage current caused by the negative chloride ions and the other ions. These individual currents can be determined by the product of potential

difference and permeability coefficient. This permeability coefficient has units as that of conductance. For example, if we are finding the ionic current of Potassium ions, it is the product of Potassium conductance and the difference between the Potassium ion's equilibrium potential and membrane potential. We know that, during depolarization stage that the sodium and potassium conductance increases. The potassium conductance increases at a slower pace than the sodium conductance. This increase is reversed during the repolarization state. Hodgkin and Huxley thought that this needed a mathematical expression [24].They gave an equation for total membrane current which is described as follows:

$$I = C_M \frac{dV}{dt} + I_i \qquad [24]$$

Where, I = Total membrane current

C_M = Membrane capacitance

V = Displacement of membrane potential from resting membrane potential

t = time

I_i = Ionic current

Note that the magnitude of V does not affect the membrane capacity

The ionic current I_i can be described as follows:

$$I_i = I_{Na} + I_K + I_l \qquad [24]$$

Where, I_i = Ionic current

I_{Na} = Current due to sodium ions

I_K = Current due to Potassium ions

I_l = Current due to other ions

Now from the theory that Hodgkin and Huxley proposed we can write the each individual ionic current in terms of potential difference and conductance. We know from Ohm's law,

$$I = \frac{V}{R}$$

Where I = Current

V = Voltage

R = Resistance

And we know,

$$R = \frac{1}{g}$$

Where g = Conductance

Therefore, from above two equations using it in our situation we get,

$$I_{Na} = g_{Na}(V - V_{Na})$$

$$I_K = g_K(V - V_K)$$

$$I_l = g_l(V - V_l)$$

Hodgkin and Huxley won Nobel Prize in 1963 in Physiology for this model. Their work is more extensive than just the above equations. They also gave terms for ionic conductance and reconstructed the nerve behavior which can be found extensively in [24].

2.2.3 LEAKY INTEGRATE-AND-FIRE MODEL:

In the different neuron models over the time there have been many issues with large power consumption, spike frequency adaptation and power dissipation. These issues have been addressed in some models previously however, they were not optimized well enough. So, this leaky integrate-and-fire model uses the same design principles of some of the few older models however, targets to get rid of most of the issues discussed above. This model is explicitly explained in Section 4.

The key components of this model are:

- Membrane Capacitor
- Leaky Current Source
- Synaptic Input Current

Like the integrate-and-fire model, this model also has the Synaptic current source. But, to solve the memory issue, membrane term is added to leakage current as follows.

$$I(t) - \frac{V_m(t)}{R_m} = C_m \frac{dV_m(t)}{dt} \text{ [21]}$$

In the previous models discussed here, it was assumed that the membrane was a perfect insulator however, this is not true. Good part about this model is that it is not assumed that it is a perfect insulator [21]. I_{th} is the threshold current and this model forces the Synaptic input current to exceed

I_{th} to release a spike. In cases this doesn't happen, there is leakage current released without change in voltages [21].

The firing frequency is given as follows [21]:

$$f(I) = \begin{cases} 0, & when\ I \leq I_{th} \\ \left[t_{ref} - R_m C_m \log(1 - \frac{V_{th}}{I\ R_m})\right]^{-1}, & when\ I > I_{th} \end{cases}$$

This neuron model is used widely due to simplicity in the design and very good accuracy. Some of the previous models discussed here use capacitors and resistors in more number. In electronic circuits, these two components occupy large space and having more than one in a design will only make the design large in area. While implementing such models is counterproductive since the main objective to design neurons is reduce cost and power. In the simple Integrate-and-Fire model, we are not resetting the neuron back to its resting membrane potential. The Hodgkin and Huxley model is very complicated for a practical implementation. However, the Leaky Integrate-and-Fire model is comparatively having better performance, smaller size, good resemblance to the biological neuron and accurate [25].

2.2.4 COMPARISON BETWEEN DIFFERENT NEURON MODELS:

NEURON MODEL	COMPLEXITY	RELIABILITY	SIZE	POWER REQ	COST
INTEGRATE AND FIRE	Low	Low	Small	Low	Low
HODGKINS AND HUXLEY	High	High	Large	High	High
LEAKY INTEGRATE AND FIRE	Medium	High	Medium	Low	Medium

Table 2.1: Comparison between various discussed neuron models

CHAPTER 3: NEURAL ENCODING SCHEMES

It is the brain that helps us perceive the world and its fundamental elements are the neurons. A neuron fires a spike in response to the stimulus. This means that the encoding of information is done by the neurons [26]. The neuron is known to generate a spike train and the spike train is usually defined as "a sequence of stereotyped events occurring at regular or irregular intervals" [20].

It is not only the spike that transmits the data it is also its rate and timing that contains information. These spikes cannot happen immediately because there is delay in the neuron to come back to its original state. This is one flaw of the neuromorphic systems because they do not operate at high speeds and also need time to get back to their original state causing delays [20].The entire information for an action to take place is encoded by the neuron and sent to the other neurons accordingly. Many of the neurons communicate only with their peers through signals and only a few of them receive straight forward inputs [26]. To make this disguised communication make sense, they must be decoded however, it is still a mystery as to how this happens [26].

From the working of the neuron it is evident that the main computing happens inside the neuron and as a consequence of this spikes are fired. To make this make sense and understand what the brain is processing we should inspect the spikes at each of brain's stages. This inspection from stage to stage helps us understand as to how the brain is processing [27]. Most of the neuron's response is not symmetric, this means that it is hard to find neurons that respond similar to a stimulus. This characteristic of the neuron enables it to be able to represent different stimulus. From this it can be said that the encoding process is affected depending on the reaction of the neuron to the wide range of stimulus it receives. To understand the brain's function, it is necessary to go down to individual neuron level rather than stage level [27].

To make a conclusion on how the process of communication occurs internally in between the neurons it is important to study an individual neuron's activity and response. It is also required to determine the rate at which messages are transferred between the cells. To explicitly determine what data is being transferred it is very important to study how much data is being carried for the same situation in the same time frame. This is the same technique applied to realize to what extent the data is unnecessary or what parts of the data is useful. The rate of data transfer is done by measuring rate of data transfer from each neuron response. These measurements ensure answers to basic questions about brain's processing. Information theory is what provides solutions to all these discussed information retrieval. And this is done by studying the spike fire timing in the individual neuron and relative time of firing between neurons [27].

The spike train is not periodic nor has any correlation with the nearby neural spikes. It is difficult to differentiate noise from the code because we do not know how the code looks. It could be an efficient way to implement the code or pure noise or mixture of data and noise. This is something we are unsure about [20]. Differentiating data from noise from the codes obtained is one of the most crucial issues to be solved. The cause of this noise varies, it could be a synaptic failure or it could be due to some internal changes or it could be due to external changes or it could be due to failure in the transmission, the reason is however unknown. In some hardware designs, noise is added externally to imitate the brain entirely.

It has been believed for a long time that rate codes are used by the neuron and researchers used this for neural codes. But in contrast, the neuron has different encoding schemes used and this is often a topic of argument. Neuron encoding is learning how neurons react to different stimulus [28]. Depending on the action electrical signals are fired and these spikes vary from each activity. For a while now there have been studies on how these spikes keep changing depending

on the activity and the intensity of that activity. Researchers argue about which encoding scheme the neurons use and this is still a debate. The following are the different encoding schemes [28] [26].

3.1 BINARY ENCODING [26] [29]:

This was one of the early encoding schemes that was thought to have been used by the neurons. When this technique was hypothesized, it was assumed that encoding process in the neurons involves either true or false i.e., Boolean functions. This analysis was done by McCulloch and Pitt and their neuron model was called McCulloch-Pitts Neuron in which they assumed that the neuron either spikes or doesn't spike at all [29]. When we talk in general terms true or false in numeric systems accounts to binary either '0' or '1'. And the neuronal response was assumed to provide such response in a few situations and they are:

- A spike is generated by the neuron even though there is no on-going activity but the cell receives an apt input i.e., an apt stimulus [26].
- A spike is generated by the neuron during bursting behavior of the neurons in spite of having an on-going activity [26].
- A spike is generated in reactance to an input [26].

These conditions look like if-else statements in a loop returning a Boolean function in a typical C program. The roots of today's neural code understanding come from McCulloch-Pitts inspection about them which dates back to 1943 [29]. This code is overlooked due its simplicity yet it is still consequential.

The neuron model given by McCulloch and Pitt is very simple and also, not realistic. Because this design used basic digital logic circuits like AND, OR and XOR gates. And all the networks are built using just the three gates in this mode. This is too good to be true.

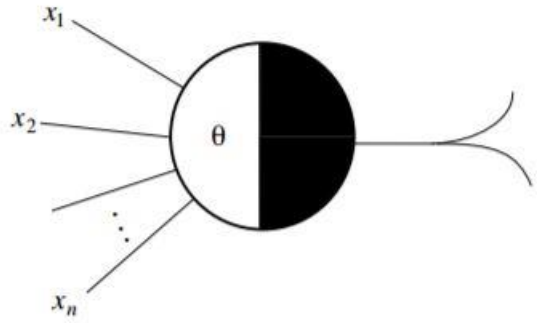

Figure 3.1: McCulloch and Pitts neuron model (*Source:* [30])

According to McCulloch-Pitt's neuron model, a neuron is something that which can output 'high' or 'low' states having multiple inputs however, there is only one inhibitory input. Every neuron also has a threshold value [29]. The neuron has a step function according to McCulloch and Pitt. Inputs are denoted by x, y for inhibitory inputs, w for the weights and θ for threshold value. The two scientists explained that the neural networks have either excitatory or inhibitory edges. The inhibitory edges denoted by circles (see figure 3.1).

$$Sum = \sum_{i=0}^{N} x_i w_i$$

$$y = f(Sum)$$

The activation function of this neuron is a step function. At θ the output changes from low to high. When θ is greater than the excitatory inputs this neuron will fire. Even if there is one inhibitory

input the neuron will not fire. The x axis is the sum and the y-axis is the function ‘y’ as denoted above [31].

$$y = \begin{cases} 0, & Sum < \theta \\ 1, & Sum \geq \theta \end{cases}$$

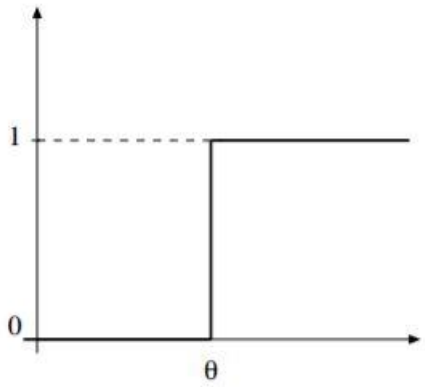

Figure 3.2: Step function of the McCulloch-Pitt’s neuron with θ (*Source:* [30])

The research is continuing using these codes. Carina Curto, Anda Degeratu and Vladmir Itskov from the University of Nebraska Lincoln are working on the binary codes and have published a paper in the same focus [32]. Georgina Cruz and Graeme Lowe from the Monell Chemical Senses Center, Philadelphia are working in the same aspect studying these characteristics on mice [33].

3.2 RATE ENCODING [26] [28]:

Scientists have suggested that the rate of the neural code also gives information. And in fact, it is believed that this rate of the spike train carries information. Researchers figured that understanding how information is being transmitted through rate according to the problem will help them with studying the encoding process used by the neurons. Scientists have studied the

characteristics of an individual neuron and its reactions to a certain stimulus. But, as research advanced in this aspect, scientists started studying a group of neurons' response and from the data received they studied individual neural behavior. This group response however, does not give as much information as an individual neuron response [26] [28]. From different experiments performed, scientists observed the neural behavior by straining certain parts of the body and the conclusion of these experiments was that it is not the amplitude that carries information but the frequency of the train that carries information.

Rate coding was first introduced by E D Adrian and Y Zotterman in 1926 [28]. When they first came up with this concept, they used an instrument which they had designed and built that has the capability to record action currents. They used this instrument on a frog by hanging weights to its muscle. In their experiment, they observed that time between two spikes was lesser than the absolute refractory period. This is an indication that these spikes are not generated by just one neuron. They repeated this experiment for different weights. And they observed that as the weight increased, the frequency rate of the spike train increases as well.

The temporal average [28] is nothing but the spike-count rate [28] which is defined as follows.

$$v_k = \frac{n_k^{sp}}{T} \qquad [20]$$

Where, k is trial

n_k^{sp} is spike count

T is the duration

v_k is the spike firing rate

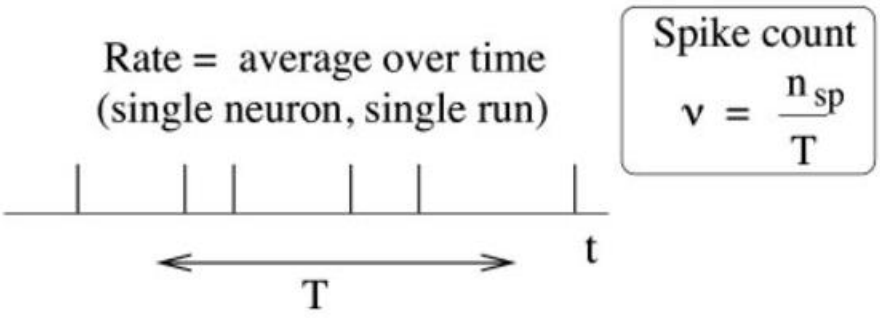

Figure 3.3: Spike firing rate (*Source:* [20])

Figure 3.3 only says that the temporal average is nothing but number of spikes over a given time [28]. This idea of the neuron containing information in the frequency was widely accepted and implemented even today. However, the spike rate is never constant for the same situation under same conditions for the same time frame. And because of this time-dependent firing rate [28] is used because the information the same situation, same time frame under the same conditions is repeated and then average for all these examinations is taken [28]. This style of measurement also helps when we want to measure the spikes over a plethora number of neurons. It is always easier to acquire from an individual neuron a couple of times than get information from a plethora of neurons. In fact, when we acquire information from so many neurons it gets difficult to know which neuron fired when, it becomes very clustered and so the time-dependent firing rate [28] is very helpful.

It is assumed in rate coding that any variations in the intervals between spike are considered to be noise though they cause fluctuations. This is the biggest draw-back of rate coding. Because the inter-spike interval might carry valuable information. It is a known fact that humans evolve. If the variations in the spike intervals is noise why haven't they evolved over the generations?

Scientists do have an explanation for this, they say that in certain sensory systems neurons fire with a very specific timing indicating rate encoding is not the only encoding scheme used by the biological neuron. The other explanation that was provided was that this variation might be observed because the neurons are working in their extreme limits. Contradicting to these, some claim that these variations that are noise can be used as an added advantage. It is said that this variation makes the system more sensitive to weak signals.

In 1960's Richard B. Stein provided substantial evidence on rate encoding of the neuron in [34]. The Brain Initiative, The Human Brain Project and other associations working on one sole purpose to get to know the brain better than what we know now. In [35] Anil has a whole new argument that the rate and the time codes work together.

3.3 TEMPORAL ENCODING [26] [28]:

From the previous sections, we learned about binary and rate codes, in this section we will study the temporal code. If the inter spacing or timing between spikes has data, it is considered as temporal code [28]. The reason temporal code was brought up by scientists is because neurons have a unique gift of very accurate spike firing. Many arguments are still active on how the precise spike fire happens. This spike firing accuracy also raises many questions if the neural response which appears to be noisy is actually noise. To the list of different arguments about the nervous systems adds another argument, which encoding scheme do the neurons use? Still unknown. Some researchers suggest that if neurons are firing the spikes at such precise timing, the timing definitely means something.

"The high-frequency fluctuations the neurons display" [28] could carry information or could be noise. Ignoring these variations by claiming it could be noise which results in the loss of valuable information. Many experiments were performed on the nervous systems of animals and birds and from the results, it was concluded that spike timing plays a significant role [26]. The neurons deliver a spike precisely and several times the spike trains have to be delayed for the action to be performed at the exact time. This delay is induced by the long axons [26]. The basic reason temporal code came into existence is to avoid the loss of information due to ignoring part of the code considering noise [28]. Learning more about these codes helps know the root causes for many neural disorders and finding a solution for them.

In [36] David and Nelson claim that sometimes for a different input the firing frequency doesn't change but the time between spikes gives information. They worked on neurons that aid vision. They provided three different inputs to see how the code looks like after passing through the neuron. In their experiment, they found out that the average firing rate from all the three-different stimulus was the same which shouldn't be true if the neurons are using the rate encoding scheme. In this paper, they claimed that the interval between the spikes gives specific information about the stimulus and that rate does give information about the stimulus but only the inter-spike interval gives information about the duration of the input provided [36]. They also said that the spike by itself is not a reference in such situations it is the phase of the spikes that is the reference.

According to [37], the authors say that building a temporal encoder from a single neuron it is not possible to obtain a temporal code. They explain that a cluster of neurons is what creates a temporal code.

There are three types of temporal codes and they are as follows.

- Time to First Spike Encoder:

  In this encoding scheme, all the information is encoded to a single spike, the time from the beginning to the time to first spike contains data relative to the stimulus. This is also called latency coding and according to this, only one spike is generated per neuron.

- Inter-Spike Encoder:

  In this type, the information is gathered from the inter-spike interval in the spike train. This is proven to be the case in real neurons however, not many designs have come out yet.

- Phase of Firing Encoder:

  This type is more like a combination of rate and temporal encoder where the firing of the neuron depends on the oscillations.

To the best of my knowledge there are very few implementations of the temporal encoding scheme. Efforts have been made in the past to implement the temporal encoding scheme in software but there have been very few efforts in building it in hardware. In [37], the authors claim that it was the first time ever a temporal encoder was successfully implemented in hardware.

3.4 POPULATION ENCODING [28]:

This type is used by the sensory and motor neurons. This type of coding represents input with respect to group activities done by neurons. It was studied that this collective information is very useful was easier to decode than others however, this encoding process is not followed by all the neurons. In this coding, the response of the neuron is scattered among the stimulus coming in [28]. In [38] the authors have given equations that would help analyze the population code.

The i^{th} neuron's neural activity can be given as:

$$r_i = f_i(x) + \sigma\varepsilon_i$$

Where, i = 1,2,…..N

r_i = neuron's activity

$f_i(x)$ = tuning function of the i^{th} neuron representing the mean value of the response

σ = Noise Intensity

ε_i = Gaussian Random variable with mean = 0 and variance = 1

And,

$$f_i(x) = \frac{1}{\sqrt{2\,\pi\,a}}\, e^{\frac{-(c_i - x)^2}{2\,a^2}}$$

Where, a= tuning width

c_i = preferred stimulus position for i^{th} neuron

The encoding process can be given as:

$$Q(r|x) = \frac{1}{\sqrt{(2\,\pi\,\sigma^2)^2\,\det(A)}}\, \exp\left[\frac{-1}{2\,\sigma^2}\sum_{ij} A_{ij}^{-1}\big(r_i - f_i(x)\big)\Big(r_j - f_j(x)\Big)\right]$$

Where A_{ij} = convariance matrix

det (A) = determinant of A

The population code has three different types and they are:

- Correlation coding [28]:

  Like the name suggests, this model assumes that information is also conveyed by the relationship between different spikes of the same spike train.

- Independent-spike coding [28]:

  Contrary to the previous type this model suggests that no two spikes of the train depend on each other.

- Position coding [28]:

  "This code is for continuous variables like color, eye position, sound frequency".

3.5 SPARSE CODING [28]:

It is believed that this type of coding is used when the being is adapting itself to new environments. This kind of code occurs when a small group of neurons are activated. When there is a dense population of neurons that are activated it is easy to encode. But, if the number of neurons activated is sparse none of the encoding schemes work. It is believed that this encoding scheme is used by neurons many sensory operations. Sometimes the sparseness is in a few selected neurons. It is unaware how the timing works in sparse codes however, [39] claims that the timing in the sparse codes does carry information. While keeping the advantages of the sparseness, in this kind of coding, the variations in spike-timing according to the stimulus to provide a useful channel increases the coding capacity of the neurons [39]. This coding type uses the concepts of rate and temporal encoding applied to sparse neuron fires. In [40] the authors gave an expression for sparseness in neurons and it is given as follows:

$$k = \frac{1}{n}\sum_{i=1}^{N}\frac{(r_i - r_m)}{\sigma^4} - 3$$

Where, k = sparseness

r_i = response of the neuron

r_m = mean response

σ = standard deviation of the neuron

The units of sparseness is Kurtosis.

Most sparse models use something called a linear generative model which is used to approximate the input-stimulus [28]. The approximation function looks as follows:

$$\xi \approx \sum_{j=1}^{n} s_j \, b_j$$

Here, ξ and b_j are vectors.

Where, ξ = set of real-numbered vectors

b_j = set of basis vectors (real)

s_j = sparse vector of weights (real)

The goal of sparse coding is to find the basis vector b. In software-neural networks, depending on the basis vector coefficients of typical inputs, the sparseness is categorized into two types:

- Soft Sparseness:

This type of code's distribution is more Gaussian and has many zero values, a few small absolute values, very few large absolute values and very very few very large absolute values. The result of this is most part of the basis vector is active.

- Hard Sparseness:

  This type of code's distribution is more Gaussian and has many zero values, no small absolute values, very few large absolute values and very very few very large absolute values. The result of this is only a part of the basis vector is active.

This area of neural coding is not as touched as the previous coding schemes discussed because of less generality. However, there is active research going on at the University of Oxford's Gero Miensenbock's lab. They are studying mushroom cells to have a more detailed take on sparse coding.

3.6 RANK-ORDER ENCODING [26]:

This code is very different from what was discussed so far. Previously, the codes discussed about the rate or spacing. This code depends on the order in which the action potentials arrive from different neurons. Research is ongoing to understand how this is done by the brain [26]. This was first proposed by Simon J. Thorpe and Jacques Gautrais in 1997. In their work presented in [41] they said that most neurons have very low frequency rates which is a strong indication that there is only one spike per neuron causing a result. And they said that the neurons which are strongly activated are the first to generate a spike. They applied this coding scheme only to visual processing and had encouraging results. They concluded that for image processing purposes spike

frequency encoder cannot be used because a human can process a whole new image in 150 ms which makes it hard for a neuron to generate more than one spike. Not a whole lot of research is going on this field but it is a side of neurophysics still to be explored.

3.7 COMPARISON OF DIFFERENT ENCODING SCHEMES:

	BINARY	**RATE**	**TEMPORAL**	**POPULATION**	**SPARSE**	**RATE-ORDER**
ACCURACY	Low	Medium	High	Medium	Medium	Medium
COMPLEXITY	Low	Low	High	High	Medium	High
COST	Low	Low	High	High	High	High
POWER REQ.	Low	Low	Medium	Medium	Medium	Medium
SIZE	Small	Medium	Large	Large	Large	Large
RELIABILILTY	Low	Medium	High	Somecases high	Somecases high	Somecases high
NEURONS USING THIS SCHEME	More general	Most neurons in peripheral nervous system, motor neurons	Mainly by cochlea neurons, used generally as well	Motor neurons, sensory neurons	Used by neurons activated which are only a few in the whole cluster	Image processing neurons

Table 3.1: Comparison between different neuron encoding schemes

CHAPTER 4: NEURON IMPLEMENTATION

In this thesis, spike-time dependent encoder is implemented using 180nm CMOS technology. Global Foundries PDK libraries have been used for the implementation. Cadence Virtuoso tool was used for the implementation of the circuit. The following table gives information about the available software existing in the market.

SL. NO	CAD TOOL	OPEN SOURCED/ LICENSED	TYPE	FUNCTION
1	Cadence	Licensed	Analog and mixed signals	Complete CAD flow
2	Mentor graphics	Licensed	Analog and mixed signals	Complete CAD flow
3	Synopsys	Licensed	Analog and mixed signals	Complete CAD flow
4	Tanner	Licensed	Analog and mixed signals	Complete CAD flow
5	Alliance	Open source	Mixed signals	Logic to layout
6	Electric CAD	Open source	Mixed signals	Logic to layout
7	Magic	Open source	Mixed signal	Circuit layout
8	SystemC	Open source	Electronic system level	Library for digital design
9	myHDL	Open source	Electronic system level	Hardware description language

Table 4.1: Different VLSI Design Tools [42]

Figure 4.1 shows the entire circuit of the encoder including its peripherals.

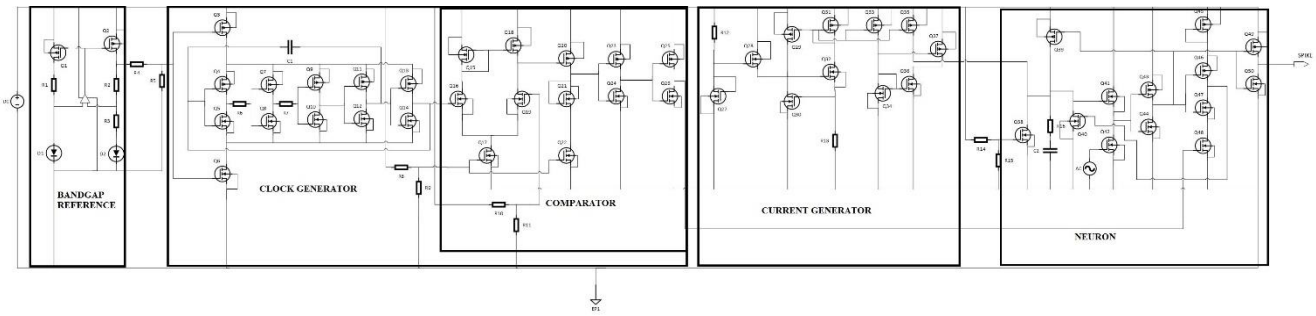

Figure 4.1: Neuron Schematic

Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q1	10/4.5	Q2	10/4.5	Q3	1/2	Q4	3.6/3.5
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q5	1.6/3.8	Q6	0.5/2	Q7	1.6/3.8	Q8	0.5/2
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q9	1.6/3.8	Q10	0.5/2	Q11	1.6/3.8	Q12	0.5/2
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q13	1.6/3.8	Q14	0.5/2	Q15	30/0.18	Q16	50/0.18
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q17	15/0.18	Q18	30/0.18	Q19	50/0.18	Q20	4/4
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q21	2/0.18	Q22	15/0.18	Q23	1.8/0.18	Q24	0.6/0.18
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q25	1.8/0.18	Q26	0.6/0.18	Q27	4/4	Q28	5/4
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q29	10/4	Q30	10/2	Q31	10/4	Q32	10/4
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q33	5/4	Q34	2.5/4	Q35	1.1/4	Q36	2/4
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q37	4/4	Q38	0.8/4	Q39	1/2	Q40	0.7/0.4

Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q41	2/2	Q42	1/4	Q43	1.8/0.18	Q44	0.6/0.18
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**Q no.**	**W/L µm/µm**
Q45	1.8/0.38	Q46	1.8/0.18	Q47	0.6/0.18	Q48	1.1/0.18
Q no.	**W/L µm/µm**	**Q no.**	**W/L µm/µm**	**R1 kΩ**	**R2 kΩ**	**R3 kΩ**	**R4 kΩ**
Q49	1.8/0.18	Q50	0.6/0.18	51.91	52.42	4.25	9.72
R5 kΩ	**R6 kΩ**	**R7 kΩ**	**R8 kΩ**	**R9 kΩ**	**R10 kΩ**	**R11 kΩ**	**R12 kΩ**
11.46	0.968	0.968	19.34	5.73	17.49	5.73	9.32
R13 kΩ	**R14 kΩ**	**R15 kΩ**	**R16 kΩ**	**C1 pF**	**C2 fF**	**D1 W/L µm/µm**	**D2 W/L µm/µm**
100.77	7.43	1.17	5.32	3	40	3.8/1.6	15.5/8

Table 4.2: Parameters of each component used in the schematic

The neuron needs external supply for it to function, like the voltage supply, excitation current, leakage voltage, clock and V_{in}. In this work, we also present a Band Gap Reference circuit, Current Generator, Clock Generator and voltage divider. But before going on to the analysis of the circuit, it is very important to understand the individual transistor's I_{DS} equation and operating region. Sections 4.1 discusses the current equation and transistor's operating regions of it. Section 4.2 discusses the detailed implementation and explanation for all the modules used in the neuron.

4.1 IDS EQUATION AND OPERATING REGIONS OF A TRANSISTOR [43]:

MOSFET-Metal Oxide Semiconductor Field Effect Transistor, is a four-terminal device. Drain, Gate, Source and Body are its four terminals. There are two types of MOSFET's NMOS and PMOS. In CMOS technologies, PMOS and NMOS are widely used. In general, NMOS MOSFET's are referred as NFET and PMOS MOSFET's as PFETS.

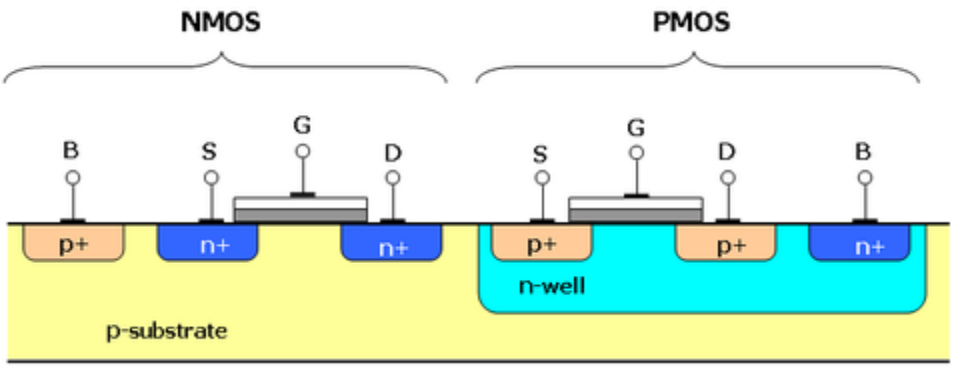

Figure 4.2: Cross-section of NMOS and PMOS transistors (*Source:* [44])

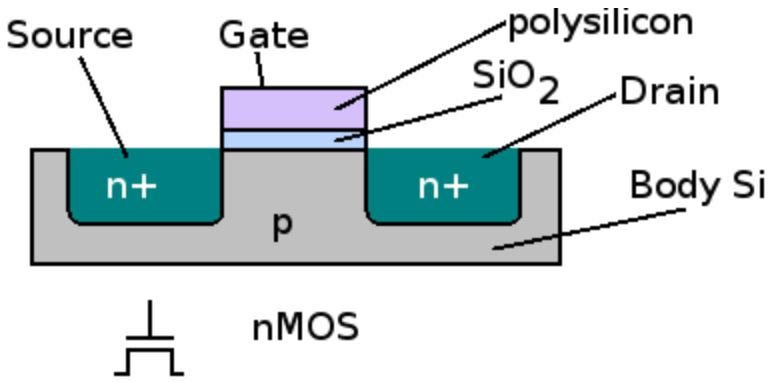

Figure 4.3: NMOS Transistor (*Source:* [45])

Before going to the I_{DS} equation, understanding the transistor's operation is important. The following is a brief understanding of the NFET.

Initially, no voltage is applied to any terminal. But as the gate voltage starts increasing, the gate and substrate form a capacitor. The holes in the p-substrate are repelled from the gate leaving negative ions such that the charge of the gate is mirrored. This is the formation of depletion region. However, due to the absence of charge carriers there is no flow of current.

As, the gate voltage is increased further the width of the depletion region increases as well. The interface potential (Potential at the oxide interface, due to structure of the transistor in silicon.

More details in [43]) gains enough positive charge causing electrons to flow from the source to the interface and eventually to the drain. A channel of charge carriers is formed between the source and drain (under the gate oxide). This causes the transistor to turn ON. The value of the gate voltage at which this inversion takes place is called the threshold voltage, V_{th} of the transistor.

$$V_{Th} = \emptyset_{MS} + 2\emptyset_F + \frac{Q_{dep}}{C_{ox}}$$

Where,

$$\emptyset_F = \frac{K\,T}{q}.\ln\left(\frac{N_{sub}}{n_i}\right)$$

$$Q_{dep} = \sqrt{4\,q\,\varepsilon_{si}|\emptyset_F|N_{sub}}$$

Where, ϕ_{MS} = difference between the work functions of the polysilicon gate and the silicon substrate

q=electron charge

N_{sub}=doping concentration of substrate

Q_{dep}=charge of the depletion region

C_{ox}=gate oxide capacitance per unit area

ϵ_{si}= dielectric constant of Silicon

If the gate voltage is further increased the depletion layer remains constant but the density of the charge carriers increases which causes more current to flow from the source to the drain.

Let us suppose that we have a semiconductor bar carrying current I.

$$I = Q_d \, \vartheta$$

Where, Q_d = charge density along the direction of current

υ=velocity of charge in meter/sec

Now let us connect the source and the drain terminals of the transistor to the ground. We only have supply at the gate voltage. As mentioned earlier, the inversion layer is only formed when this gate voltage is at least equal to the threshold voltage. Now, when the gate voltage starts to increase (more than the threshold voltage) the charge at the gate is the mirror of the charge in the channel. Now the charge density is,

$$Q_d = W \, C_{ox} \, (V_{GS} - V_{Th})$$

We can say that $W.C_{ox}$ is the total capacitance per unit length. Now, if we started changing the drain voltage to a value greater than zero, the local voltage difference between the gate and the channels varies from V_G to $V_G - V_D$. This is because the potential in the channel varies from zero at the source to the drain voltage.

Now at point x, the channel density can be re-written as,

$$Q_d(x) = W \, C_{ox} \, (V_{GS} - V(x) - V_{Th})$$

Where, V(x) is the potential of the channel at point x.

Now using this charge density equation in our initial current equation. The current equation is transformed into:

$$I_D = -W \, C_{ox} \, (V_{GS} - V(x) - V_{Th}) \, \vartheta$$

(negative sign here is to indicate electrons as the charge carriers)

In semiconductors, the velocity of charge is given as

$$\vartheta = \mu\, E$$

Where, μ = mobility of charge carriers

E = electric field

Electric field at point x can be written as

$$E(x) = -\frac{dV(x)}{dx}$$

Where, V is voltage varying from 0 to V_{DS}

Now, υ becomes

$$\vartheta = -\mu \frac{dV(x)}{dx}$$

Use this in the current equation, we get

$$I_D = W\, C_{ox}\, (V_{GS} - V(x) - V_{Th})\, \mu_n \frac{dV(x)}{dx}$$

μ_n is used because we are doing this analysis for an NMOS.

Re-arranging the above current equation,

$$I_D\, dx = W\, C_{ox}\, \mu_n\, (V_{GS} - V(x) - V_{Th})\, dV$$

Applying integration, we get,

$$\int_{x=0}^{L} I_D\, dx = \int_{V=0}^{V_{DS}} W\, C_{ox}\ \mu_n (V_{GS} - V(x) - V_{Th})\, dV$$

Now the equation becomes,

$$L\, I_D = W\, C_{ox}\, \mu_n \left[(V_{GS} - V_{Th})\, V_{DS} - \frac{1}{2}\, {V_{DS}}^2 \right]$$

(I_D is constant along the channel)

Re-writing we get,

$$I_D = \mu_n\, C_{ox} \frac{W}{L} \left[(V_{GS} - V_{Th})\, V_{DS} - \frac{1}{2}\, {V_{DS}}^2 \right]$$

This equation sees a maximum current for increasing V_{DS} when $V_{DS} = V_{GS} - V_{Th}$ which is,

$$I_{D_{max}} = \frac{1}{2} \mu_n\, C_{ox} \frac{W}{L} (\, V_{GS} - V_{Th})^2$$

This region until $V_{DS} = V_{GS} - V_{Th}$ is called the triode region.

Now what if $V_{DS} > V_{GS} - V_{Th}$? The drain current in this condition is relatively constant and this operating region is called the saturation region. As V_{DS} increases further, the point at which the charge density equals zero gradually moves towards the source and the local potential is not sufficient to support the inversion layer. The current is relative constant and now no longer depends on V_{DS}.

The current equation in the saturation region is,

$$I_D = \frac{1}{2} \mu_n\, C_{ox} \frac{W}{L} (\, V_{GS} - V_{Th})^2$$

In this whole process, we ignored the channel length modulation. However, we need to consider the channel length modulation most times. The current equation now becomes,

$$I_D = \frac{1}{2} \mu_n\, C_{ox} \frac{W}{L} (\, V_{GS} - V_{Th})^2 (1 + \lambda\, V_{DS})$$

As a conclusion,

REGIONS	VOLTAGE EQUATION	CURRENT EQUATION	APPLICATIONS
Cut-off region	$V_{DS}=0$ $V_{GS}=0$	$I_D = 0$	Digital
Sub-threshold region	$V_{DS}>0$ $V_{Th}>V_{GS}>0$	$I_D = I_0\, exp\left(\frac{V_{GS}}{\zeta\, V_T}\right)$	Extremely low-voltage, low-power applications
Triode/Linear region	$V_{GS} - V_{Th} \geq V_{DS}>0$ $V_{DS}> V_{Th}$	$I_D = \mu_n\, C_{ox} \frac{W}{L} \left[(V_{GS} - V_{Th})\, V_{DS} - \frac{1}{2}\, {V_{DS}}^2\right]$	Voltage controlled resistor, linear OTA's, multiplier, switches
Saturation region	$V_{DS}> V_{GS} - V_{Th}$ $V_{DS}> V_{Th}$	$I_D = \frac{1}{2}\mu_n\, C_{ox} \frac{W}{L} (\, V_{GS} - V_{Th})^2$	Amplifiers

Table 4.3: Different operating regions and each region's current equations [46, 43]

4.2 NEURON IMPLEMENTATION:

4.2.1 BAND GAP REFERENCE [43]:

Our neuron needs different DC voltages for its function. We can use a voltage divider to solve the issue, but the output of the voltage divider will vary with DC voltage. In some of the circuits we need constant DC voltage. We can overcome this issue by using a Band-Gap Reference circuit. This circuit outputs 1.2 V constant voltage even with varying supply DC voltage. These references work in our favor by being temperature independent. We know that when two quantities having inverted temperature coefficients will give a zero result. We chose α_1 α_2 such that $V_{REF}=\alpha_1 V_1 + \alpha_2 V_2$ will result zero temperature coefficient. BJT's (Bi-polar Junction transistors) are known to have the ability to show both positive and negative temperature coefficients. The forward voltage of a pn junction is known to show negative temperature coefficient. Unlike the

MOSFET, the BJT is a three-terminal device. Base, Emitter and Collector are the three terminals of the BJT.

$$I_C = I_S \, exp\left(\frac{V_{BE}}{V_T}\right)$$

From this we can write, $V_{BE} = V_T \ln\left(\frac{I_C}{I_S}\right)$

Where, $V_T = \frac{k\,T}{q}$

The saturation current I_S is proportional to $\mu\ k\ T\ n_i^2$.

The temperature dependence of μ (mobility of minority charge carriers) can be denoted by $\mu\ \alpha\ \mu_0 T^m$. Here m is approximately equal to -3/2.

The temperature dependence of n_i^2 (intrinsic minority carrier concentration) can be denoted by $n_i^2\ \alpha\ T^3 \exp[-E_g / (k.\ T)]$. Here E_g is the band gap energy of silicon and is approximately equal to 1.12 eV.

Now, we can write saturation current by combining the above two statements as follows.

$$I_S = b\ T^{4+m} \exp\left(\frac{-E_g}{k\ T}\right)$$

Here b is a proportionality constant.

Now taking the derivative of the above equation with respect to T we have,

$$\frac{\partial I_S}{\partial T} = b\ (4+m)\ T^{3+m} \exp\left(\frac{-E_g}{k\ T}\right) + \ b\ T^{4+m} \exp\left(\frac{-E_g}{k\ T}\right)\left(\frac{E_g}{k\ T^2}\right)$$

Therefore, multiplying the above equation by V_T / I_S we get,

$$\frac{V_T}{I_S}\frac{\partial I_S}{\partial T} = (4+m)\frac{V_T}{T} + \left(\frac{E_g}{k\,T^2}\right)V_T$$

Now taking the derivative of the V_{BE} equation with respect to T we have,

$$\frac{\partial V_{BE}}{\partial T} = \frac{\partial V_T}{\partial T}\ln\left(\frac{I_C}{I_S}\right) - \frac{V_T}{I_S}\frac{\partial I_S}{\partial T}$$

From the above two equations, we can write,

$$\frac{\partial V_{BE}}{\partial T} = \frac{V_T}{T}\ln\left(\frac{I_C}{I_S}\right) - (4+m)\frac{V_T}{T} - \left(\frac{E_g}{k\,T^2}\right)V_T$$

$$= \frac{V_{BE} - (4+m)V_T - \frac{E_g}{q}}{T}$$

In 1964 scientists discovered that if two BJT's operate at different current densities, the difference between their V_{BE} is proportional to absolute temperature.

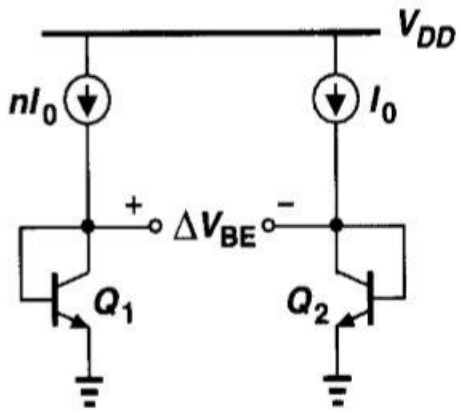

Figure 4.4: Generation of Positive To Absolute Temperature voltage (*Source* [43])

In figure 4.4, the two transistors are biased at the collector and their base currents are negligible. Then,

$$\Delta V_{BE} = V_{BE_1} - V_{BE_2}$$

$$\Delta V_{BE} = V_T \ln\left(\frac{n\, I_0}{I_{s_1}}\right) - V_T \ln\left(\frac{I_0}{I_{s_2}}\right)$$

$$\Delta V_{BE} = V_T \ln n$$

Applying differentiation to the above equation we get,

$$\frac{\partial \Delta V_{BE}}{\partial T} = \frac{k}{q} \ln n$$

V_{BE} difference exhibits a positive temperature coefficient.

Now combining positive and negative temperature coefficient voltages, we have,

$$V_{REF} \approx \alpha_1 V_{BE} + \alpha_2 V_T \ln n$$

At room temperature,

$$\frac{\partial V_{BE}}{\partial T} \approx -1.5\ mV\ /^0K$$

$$\frac{\partial V_T}{\partial T} \approx 0.087\ mV\ /^0K$$

If $\alpha_1 = 1$ and chose α_2 such that (α_2 ln n)(0.087 mv / ^{0}K = 1.5 mv / ^{0}K causing α_2 ln n ≈ 17.2 to get zero temperature coefficient.

$$V_{REF} \approx V_{BE} + 17.2\, V_T \approx 1.25\ V$$

[43] has detailed description of all the different types of Band-Gap reference designs.

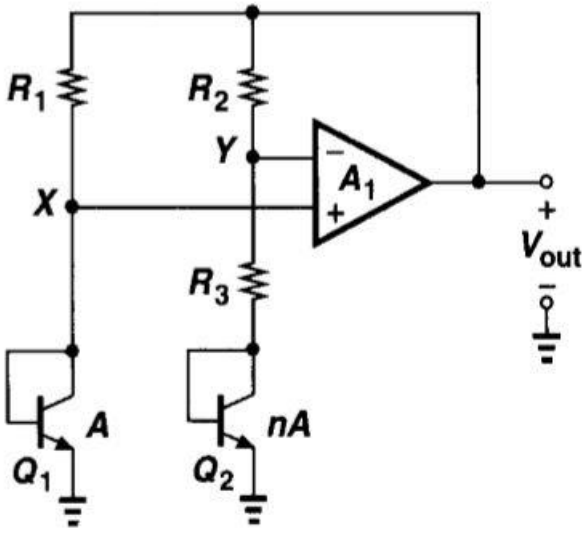

Figure 4.5: Typical BGR circuit (*Source* [43])

$$V_{out} = V_{BE_2} + \frac{V_T \ln n}{R_3}(R_3 + R_2)$$

$$V_{out} = V_{BE_2} + (V_T \ln n)\left(1 + \frac{R_2}{R_3}\right)$$

In our design we use an operational amplifier and diodes. These are explained in detail in [43].

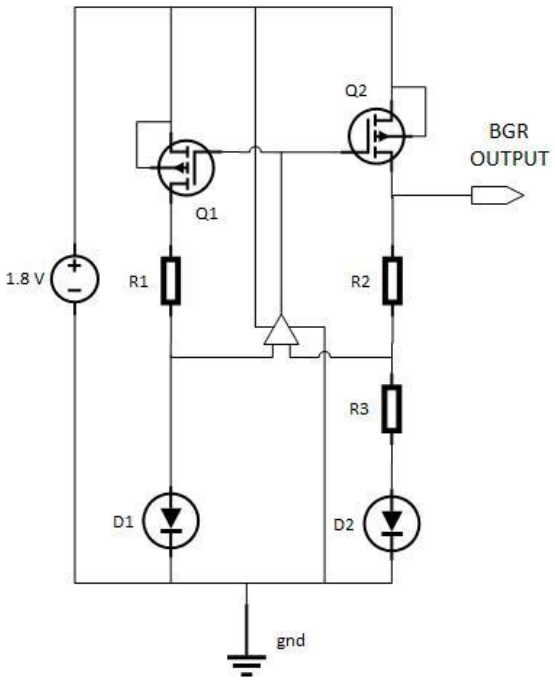

Figure 4.6: Schematic of Band Gap Reference

OP-AMP DESIGN [47] [42]:

The following explains the design of an operational amplifier:

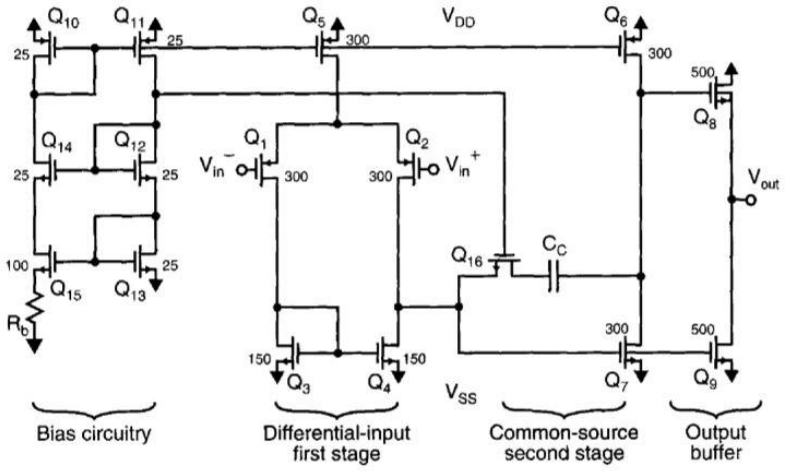

Figure 4.7: CMOS realization of a two-stage amplifier

Operational amplifier gain:

For low frequency applications, the overall gain is one of the most critical parameters.

The gain of the first stage can be derived as follows:

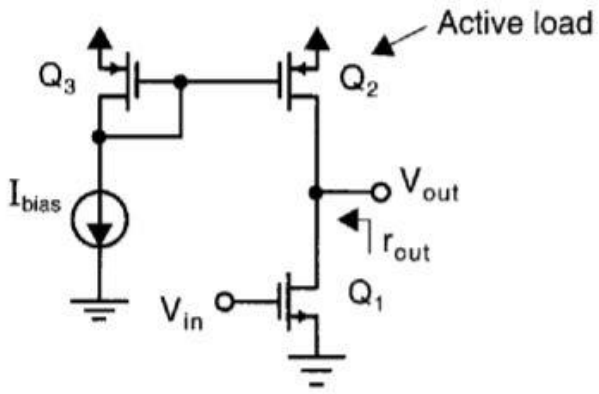

Figure 4.8: Common-source amplifier with a current mirror active load (*Source:* [47])

A small signal equivalent circuit for low frequency analysis for the common source amplifier is as follows:

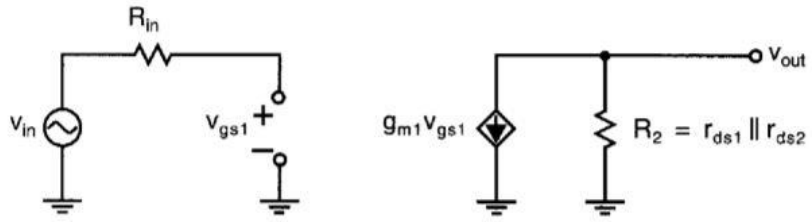

Figure 4.9: Small signal equivalent circuit for the common-source amplifier (*Source:* [47])

V_{in} and R_{in} are the Thevenin equivalent of the input source. It is assumed that the bias voltages are such that both the transistors are in active region. The output resistance, R_2, is made up of the

parallel combination of the drain-to-source resistance of Q_1, that is, r_{ds1}, and the drain-to-source resistance of Q_2, that is r_{ds2}.

Using small-signal analysis, we have $V_{gs1}=V_{in}$,

$$Av = \frac{V_{out}}{V_{in}} = -g_{m_1} R_2 = -g_{m_1} (r_{ds1} \| r_{ds2})$$

gm1 is given by,

$$g_m = \frac{\partial I_D}{\partial V_{GS}}$$

I_d in the active region is given by,

$$I_D = \frac{1}{2}\mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{Th})^2$$

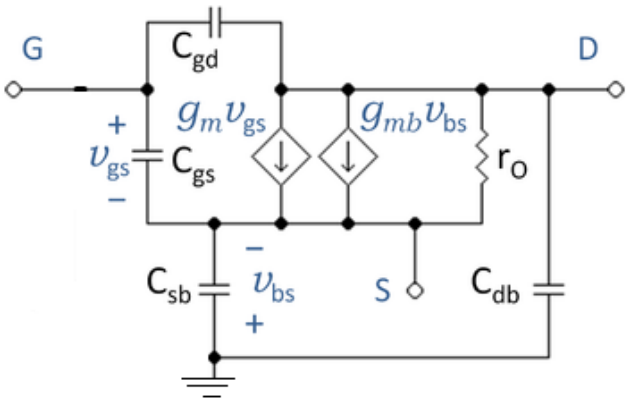

Figure 4.10: Small signal model for a MOS transistor in active region (*Source:* [48])

$$g_m = \frac{\partial I_D}{\partial V_{GS}} = \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{Tn}) = \mu_n C_{ox} \frac{W}{L} V_{eff}$$

Equivalently, we get:

$$g_m = \mu_n C_{ox} \frac{W}{L} V_{eff}$$

Where, the effective gate-to-source voltage, Veff is defined as,

$$V_{eff} = V_{GS} - V_{Tn}$$

We now understand that the trans-conductance is directly proportional to V_{eff}.

It is desirable to express g_m in terms of I_D rather than V_{GS}, so we have:

$$V_{GS} = V_{Tn} + \sqrt{\frac{2.I_D}{\mu_n C_{ox} \frac{W}{L}}}$$

From the above, we get:

$$V_{eff} = V_{GS} - V_{Tn} = \sqrt{\frac{2.Id}{\mu_n C_{ox}(W/L)}}$$

Substituting V_{eff} in g_m and simplifying, we get:

$$g_m = \sqrt{2\,\mu_n C_{ox}.\frac{W}{L}\, I_D}$$

Thus, the transistor trans-conductance is proportional to $\sqrt{I_D}$ for a MOS transistor.

Therefore, gm1 in gain of the first stage is given by,

$$g_{m_1} = \sqrt{2\,\mu_n C_{ox}\left(\frac{W}{L}\right).I_{D_1}} = \sqrt{2\,\mu_n C_{ox}\left(\frac{W}{L}\right)\frac{I_{bias}}{2}}$$

Also, an approximation to the finite output impedance, r_{ds1}, of transistor Q1 is given by,

$$r_{ds_1} \approx \alpha \frac{L_1}{I_{D_1}} \sqrt{V_{DG_1} + V_{t_1}}$$

where, α is a technology-dependent parameter of around 5 x 10^6 $\sqrt{V}/m$.

The second gain stage is simply a common-source gain stage with a p-channel active load Q6. Its gain is given as follows:

$$A_{\vartheta_2} = -g_{m_7}(r_{ds_6} || r_{ds_7})$$

The third stage is a common-drain buffer stage. This stage is often called a source follower, because the source voltage follows the gate voltage of Q8, except for a level shift. The gain of this source-follower is given by,

$$A_{\vartheta_3} \approx \frac{g_{m_8}}{G_L + g_{m_8} + g_{DS_8} + g_{DS_9}}$$

Where, G_L is the load conductance being driven by the buffer stage. When it is not possible to tie the substrate of Q8 to its source, as is the case when an n-well process is used, then the gain of the buffer stage is given by,

$$A_{\vartheta_3} \approx \frac{g_{m_8}}{G_L + g_{m_8} + g_{DS_8} + g_{DS_9} + g_{s_8}}$$

Where, g_s is a body-effect conductance and is given by,

$$g_s = \frac{g_m \cdot \gamma}{2.\sqrt{V_{SB} + 2\emptyset_F}}$$

Voltage V_{SB} is the source-to-substrate voltage and γ is the body-effect constant and $2\phi_F$ is twice the difference between Fermi level in the bulk and the Fermi level of intrinsic silicon. Thus, g_s is around $g_m/5$.

Systematic offset voltage:

When designing the two-stage operational amplifier, it is possible that the design will have an inherent (or systematic) input-offset voltage. To ensure that no systematic input-offset voltage exists, when the differential input voltage is zero (i.e., when $V_{in}^{+}=V_{in}^{-}$), the output voltage of the first stage, V_{GS7} , should be that which is required to make I_{D7} equal to its bias current, I_{D6}.

V_{GS7} is given by,

$$V_{gs_7} = \sqrt{\frac{2.I_{d_6}}{\mu_n C_{ox}(W/L)7}} + V_{tn}$$

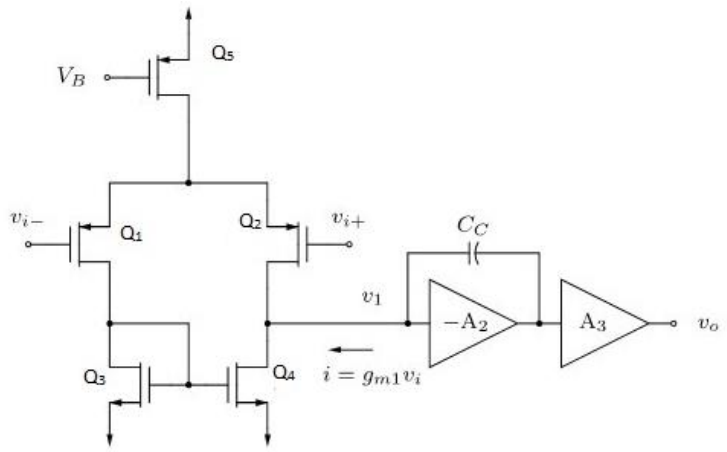

Figure 4.11: Input and gain stages of the two-stage op-amp (*Source:* [42])

When the differential input voltage is zero, the drain voltages of both Q_3 and Q_4 are equal by arguments of symmetry. Therefore, the output voltage of the first stage, V_{GS7}, is given by:

$$V_{GS7}=V_{DS3}=V_{GS4}$$

This value is the voltage necessary to cause I_{D7} to be equal to I_{D6}. If this is not achieved, then the output of the second stage (with Q_6, Q_7) would clip at either the negative or positive rail since this stage has such a high gain. However, the gate-to-source voltage of Q_4 is given as follows:

$$V_{gs_4} = \sqrt{\frac{2.I_{d_4}}{\mu_n C_{ox}(W/L)_4}} + V_{tn}$$

Equating V_{GS7} and V_{GS4}, we get:

$$\sqrt{\frac{2.I_{d_4}}{\mu_n C_{ox}(W/L)_4}} = \sqrt{\frac{2.I_{d_6}}{\mu_n C_{ox}(W/L)_7}}$$

Simplifying, we have:

$$\frac{I_{d_4}}{(W/L)_4} = \frac{I_{d_6}}{(W/L)_7}$$

This equality, when the current density of Q_4 is equal to the current density of Q_7, guarantees that they both have the same effective gate-source voltages.

Since,

$$\frac{I_{d_6}}{I_{d_4}} = \frac{I_{d_6}}{I_{d_5}/2} = \frac{(W/L)_6}{(W/L)_5/2}$$

The necessary condition to ensure that no input-offset voltage is present it,

$$\frac{(W/L)_7}{(W/L)_4} = 2\frac{(W/L)_6}{(W/L)_5}$$

The op-amp used our design is slightly different from the above design but almost has the same working.

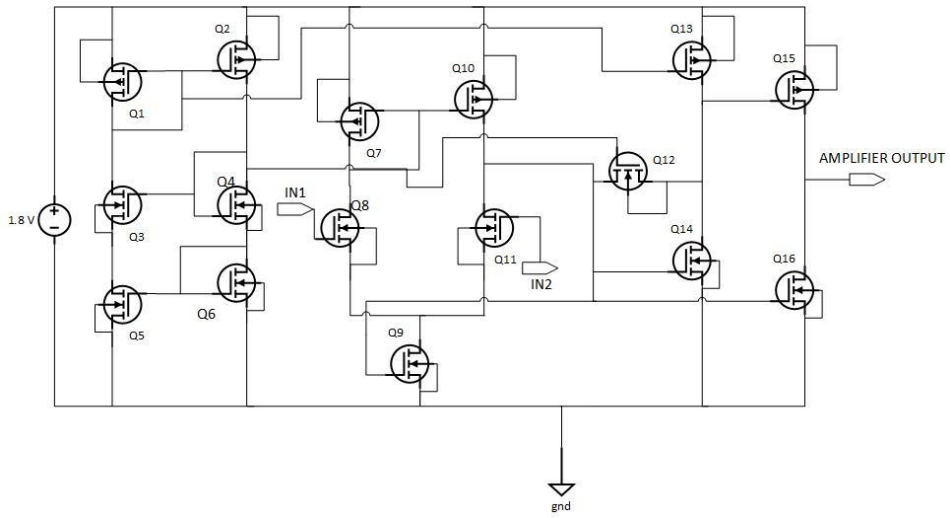

Figure 4.12: Schematic of Operation amplifier

4.2.2 CLOCK GENERATOR:

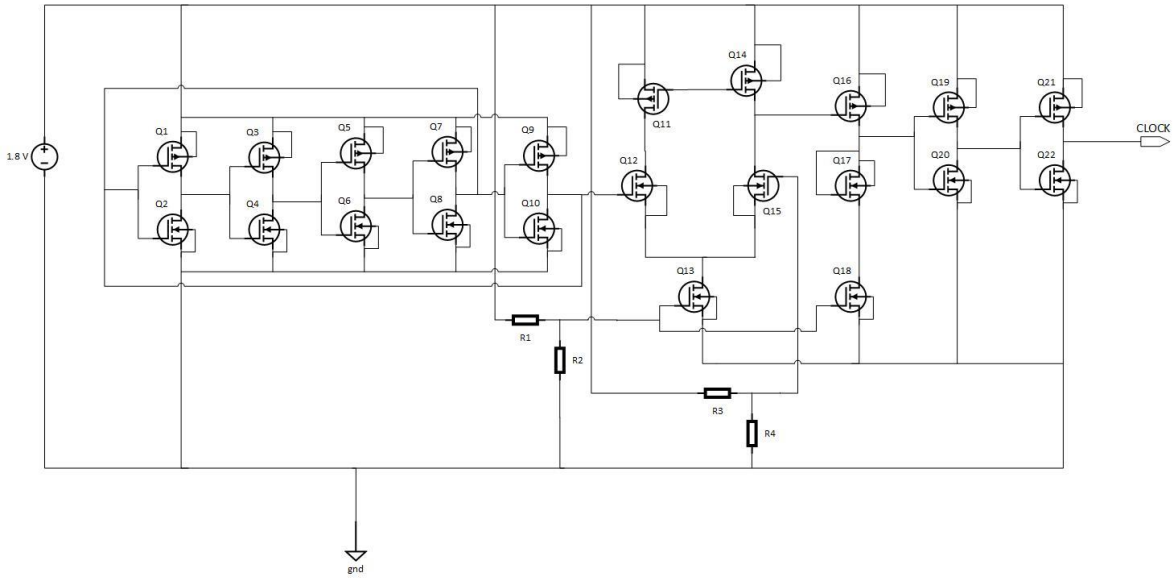

Figure 4.13: Schematic of Clock Generator

The clock generator shown above has five inverters connected back to back in a feedback loop. This back to back placement of the inverters is the ring oscillator. Post the ring oscillator we have a comparator. The reason we have a comparator is because, the output of the oscillator is not

an ideal square wave. Ideally, in theoretical analysis it should be a square wave but, in practical attempts, this doesn't happen. For obtaining a perfect square wave, we use the comparator. The purpose of the feedback capacitor is to force the circuit to have a starting point, to break the symmetry of the circuit (not shown in the picture). The frequency of this clock generator can be altered by using a resistor in series with a capacitor after every inverter connecting it to ground.

COMPARATOR:

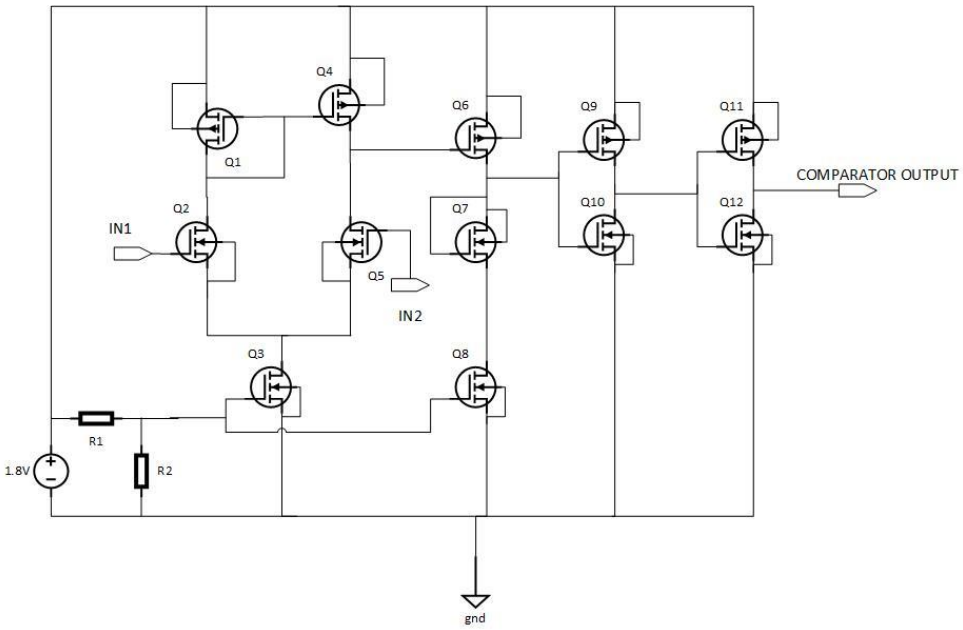

Figure 4.14: Schematic of Comparator

The comparator is a simple two stage amplifier connected to a buffer. Observation can be made from the test bench of the comparator that we have a DC bias voltage which is 400mV for the comparator and sine input at the second input and a DC of 1.1 V to the first one (just to test). The 400mV bias acts as a reference voltage. Anything higher than that is ON and lower than that is OFF. We can change this accordingly to give a reference of your choice. Here input 1 is our

inverting input and input 2 is the non-inverting input. If we give the inputs vice-versa to what we gave here, we will have an inverted waveform because the input is given to the inverted signal.

4.2.3 CURRENT GENERATOR [43]:

The main objective of this current generation is to output constant that is independent of the supply, process and temperature. The basic current mirror is shown in the figure 4.15.

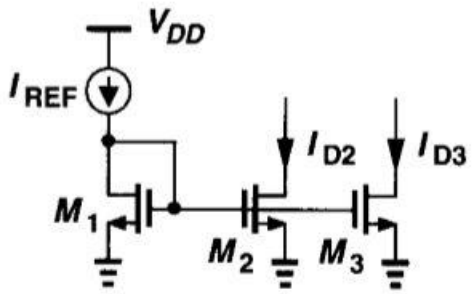

Figure 4.15: Current-mirror biasing using ideal current source (*Source:* [43])

In the current-mirror, it is assumed that an ideal current source is available in reality. Also, the channel length modulation is neglected. Due to this issue, an alternate to this design was introduced using a resistor.

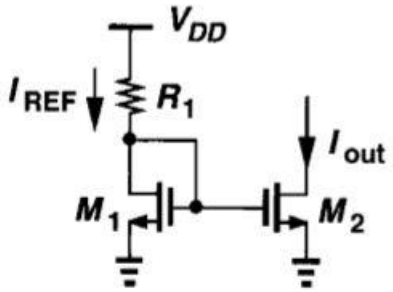

Figure 4.16: Current-mirror biasing using a resistor (*Source:* [43])

The current produced at the output depends on the value of the resistance. Also, the current produced depends largely on the supply voltage.

$$\Delta I_D = \frac{\Delta V_{DD}}{R_1 + \frac{1}{g_{m1}}} \frac{(W/L)_2}{(W/L)_1}$$

To avoid the dependence of output current on the supply voltage another design was postulated. This circuit self-biases, making I_{out} independent of the supply.

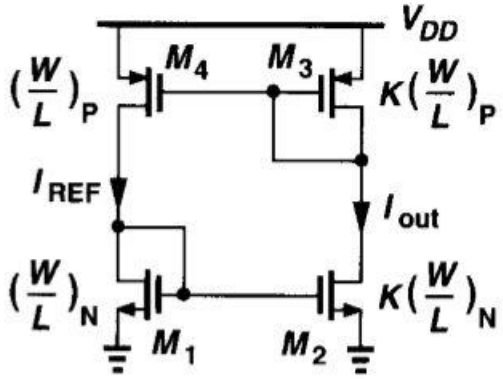

Figure 4.17: Self-biased current mirror (*Source:* [43])

$$I_{out} = K\, I_{ref}$$

(when channel length modulation is neglected)

If the channel length modulation is considered, the above design shows some dependence on the supply. To uniquely define the output, current an extra element is added. This additional element is a resistor.

From figure 4.18, the resistor R_s decreases the current of the transistor M_2. $I_{out} = I_{ref}$ because the PFET's in the design have identical dimensions.

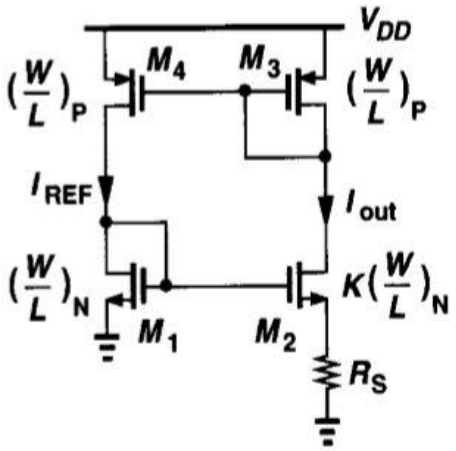

Figure 4.18: Additional resistor added to the current mirror (*Source:* [43])

$$V_{GS_1} = V_{GS_2} + I_{D_2} R_s$$

Finding V_{GS} from I_{DS} equation and substituting in the above equation, we get,

$$\sqrt{\frac{2\, I_{out}}{\mu_n C_{ox} (W/L)_N}} + V_{Th_1} = \sqrt{\frac{2\, I_{out}}{\mu_n C_{ox}\, K\, (W/L)_N}} + V_{Th_2} + I_{out}\, R_s$$

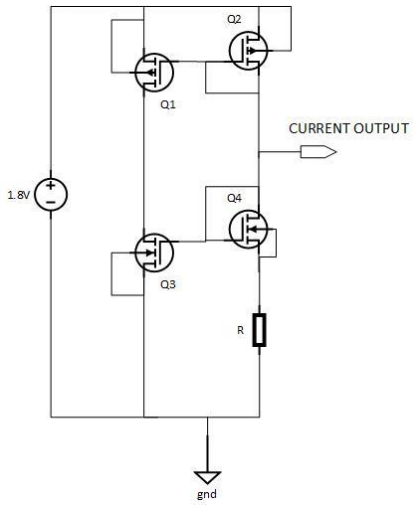

Figure 4.19: Schematic of the current generator

4.2.4 NEURON DESIGN [25] [37]:

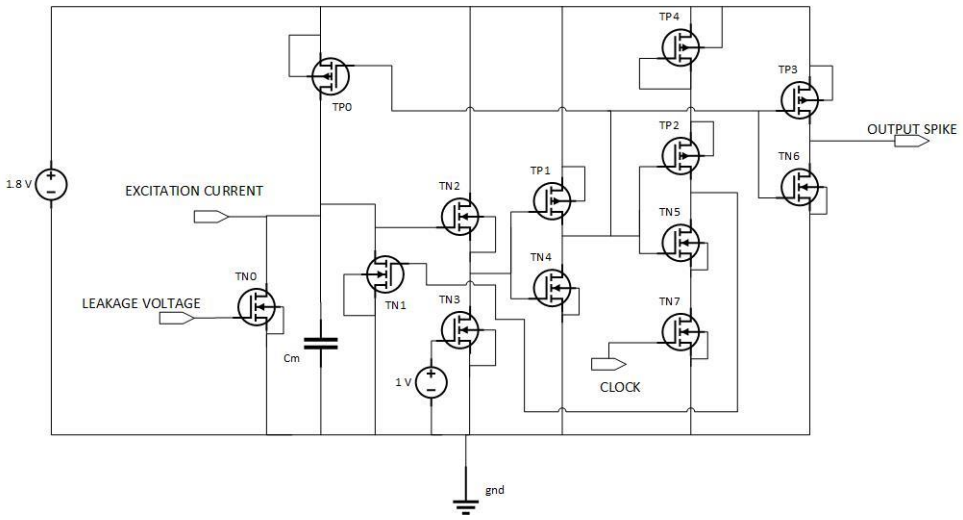

Figure 4.20: Neuron Schematic (*Source* [37])

The Capacitor C_M in the circuit is analogous to the membrane in the biological neuron. This membrane is controlled by the excitation current, the leakage current, the ionic current due to sodium ions and the ionic current due to potassium ions. This capacitor charges when the excitation current is provided. This excitation is usually in the order of nano-amps and this small current is enough for the operation. This current starts charging the capacitor and when the membrane capacitor's voltage exceeds the threshold voltage ($V_{Threshold}$) we see a spike at the output.

$$V_{Spike} = \begin{cases} 1 \; when \; V_m > V_{Threshold} \\ 0 \; when \; V_m < V_{Threshold} \end{cases}$$

Where,

$$V_{Threshold} = V_{DD} - V_{in} + V_{ThTN3} - V_{ThTN2}$$

Where,

$$V_{ThTN2} = V_{ThN} + \gamma \left(\sqrt{|2\emptyset_F + V_{SB}|} - \sqrt{|2\emptyset_F|}\right)$$

When $V_{SB} \neq 0$ V_{ThTN2} is greater than V_{ThN}

The transistor TP0 generates the ionic current due to the sodium ions and transistor TN1 generates the ionic current due to the potassium ions. These ionic currents are representations of the Na^+ and K^+ ions.

When the excitation current is applied, source voltage of TN2 or the drain voltage of TN3 change with respect to the membrane potential. This voltage is fed to the first inverter. When the membrane potential is more than $V_{Threshold}$, the output of this inverter is low. Activating the TP0 transistor such that I_{Na} increases. When this happens, the output of the second inverter is high causing I_K to increase. Transistors TP0 and TN1 are now in saturation. However, during the firing time, the gate voltage of transistor TP0 decreases and that of TN1 increases forcing the membrane potential to a low level. This is called the refractory period.

As mentioned before, the charging of the membrane capacitor increases the voltage on transistor TN2. From the circuit, we can say that the drain to source current in transistor's TN2 and TN3 is the same. As the gate voltage on TN2 increases so does the drain to source current of the transistor

$$I_{TN2} = I_{TN3}$$

$$I_{TN3} = K\,(V_{th} - V_{ThTN3})^2(1 + \lambda V_{DSTN3})$$

Where K is the process parameter

From the above equation, we can confidently say that, as the drain to source voltage of TN3 transistor increases, the current increases as well. When this voltage reaches the threshold, transistors TP1, TN4, TP3 and TN6 work such as to produce a spike.

Re-writing a few of the above analysis, when the spike is generated, the output of the first inverter is low implying the output of the second inverter is high. Through feedback this is connected to the transistor TN1. When the spike is generated, a high voltage signals is generated by the transistors, TP2, TP4, TN5 and TN7 causing the membrane capacitor to discharge.

The purpose of the clock here is controlling the resting period and the discharge time. The discharge of the membrane capacitor results in causing transistor TN2 to go into cut-off region affecting the drain current going to TN3 making the current zero. When the spike is fired and the output of the second inverter is high, the output of this inverter depends on the clock signal. When a high voltage signal is provided to TN3 in feedback, the membrane capacitor has a path to discharge. The spike generated after the second inverter is provided to the third inverter and now we see a proper spike.

CHAPTER 5: PRE-LAYOUT SIMULATION RESULTS

5.1 BANDGAP REFERENCE SIMULATION:

5.1.1 OP-AMP SIMULATION:

The simulation results of an Operational amplifier are done by two tests. The transient analysis and the AC analysis. The transient analysis gives us information about the output waveform, more specifically how it looks over time. The AC analysis gives gain and phase which are important to know in circuits like amplifiers.

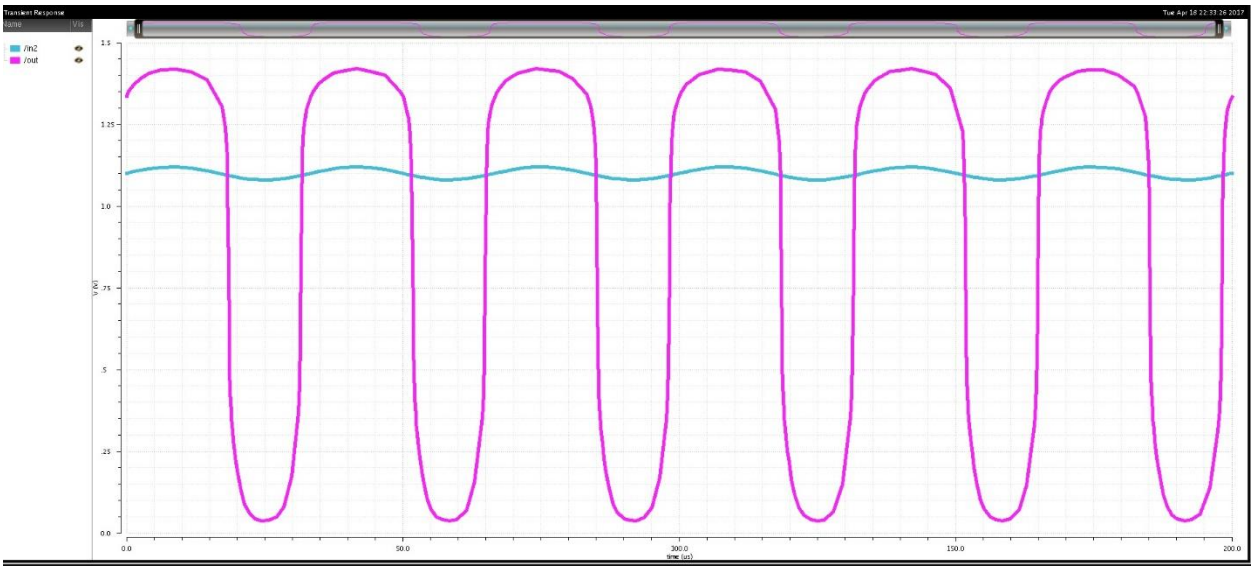

Figure 5.1: Transient simulation of an operational amplifier

In the above image, the blue wave is the input waveform and the pink one is the output waveform. We can see and conclude that the input waveform is getting amplified. But, by how much? To know that, we use the AC analysis.

Figure 5.2: AC Analysis for the operational amplifier

The blue wave indicates the phase. The black waveform is the gain. The gain of this waveform is about 30 dB. Since our operational amplifier is self-biased stabilizing it for lower gain was more important, and this op-amp serves the purpose with no issues.

5.1.2 BANDGAP REFERENCE SIMULATION:

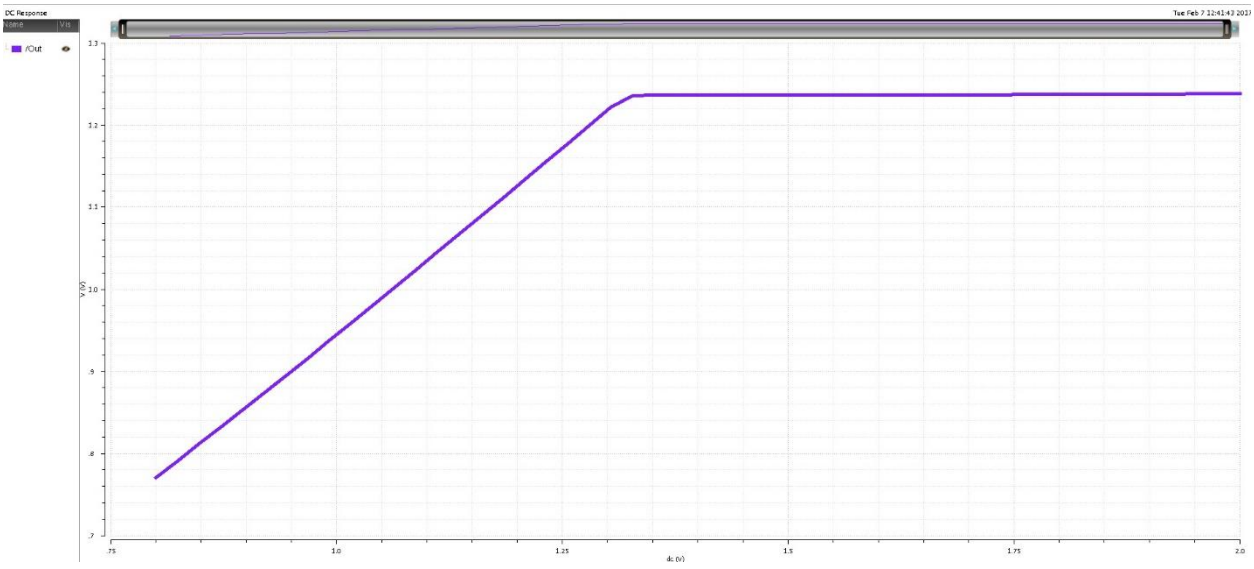

Figure 5.3: Simulation result for Band Gap Reference

The above simulation result is for the output against the DC supply. The input supply is varied from 0V to 2V and until 1.3 V the output doesn't show 1.24V because the circuit by itself needs a minimum voltage to function. Beyond that the output voltage is almost constant, it is approximately 1.24 V.

5.2 CLOCK GENERATOR:

5.2.1 COMPARATOR SIMULATION:

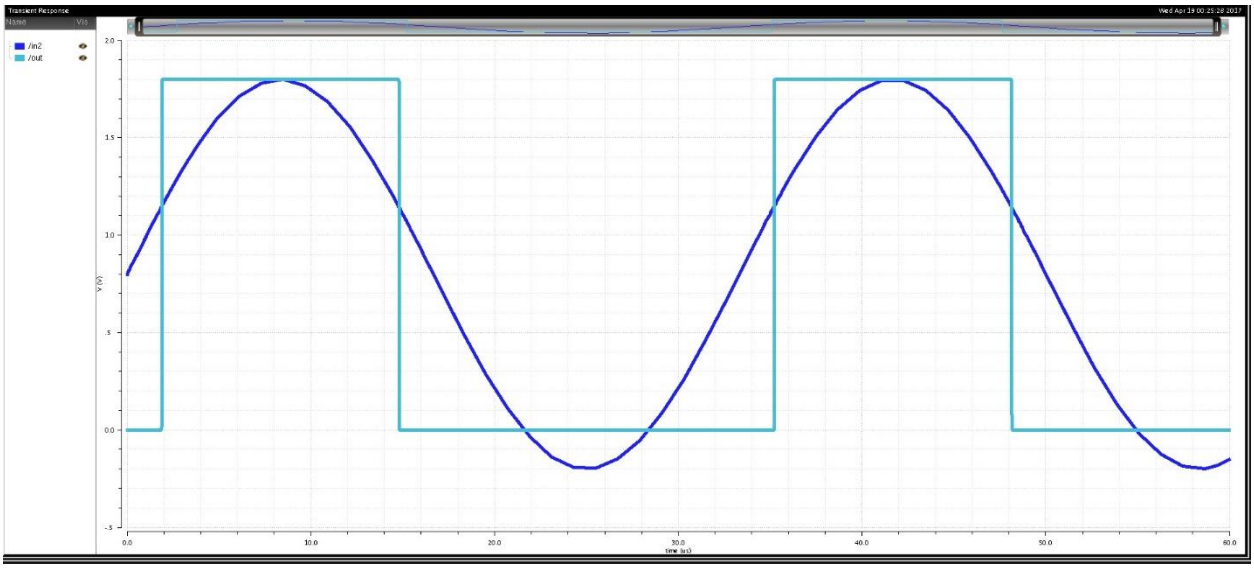

Figure 5.4: Simulation results of comparator with one waveform over the other

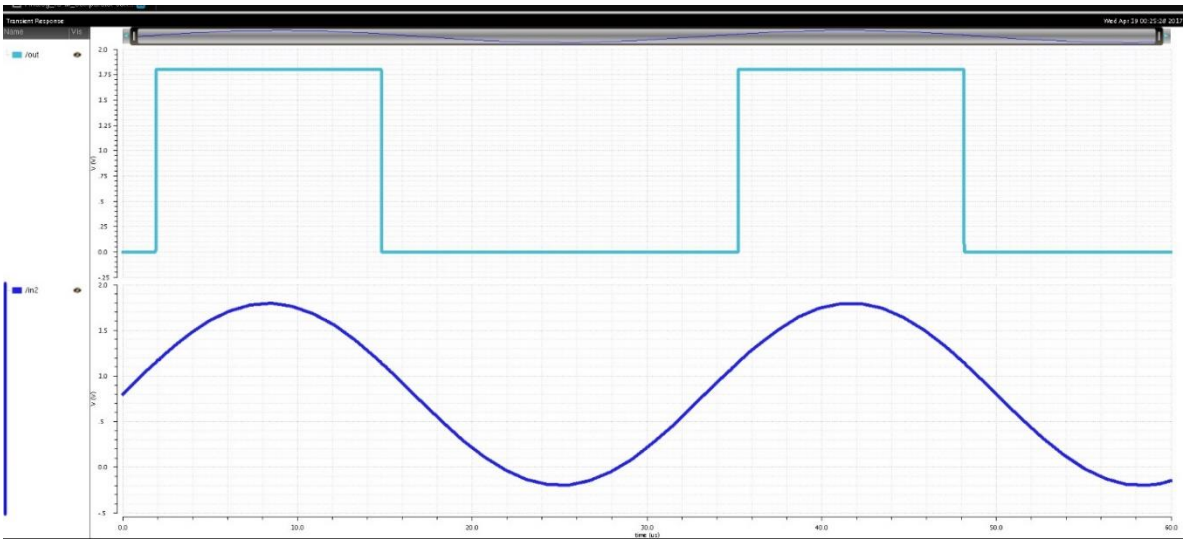

Figure 5.5: Simulation results of the comparator

The dark (royal) blue waveform is the input waveform. Our reference voltage is 400mV. When these are compared, for input waveform greater than 400mV, the output of the comparator goes high. For input value, lower than 400mV the output of the comparator is low. Thus, giving a very smooth rectangular waveform for a sine-wave in.

5.2.2 CLOCK GENERATOR SIMULATION:

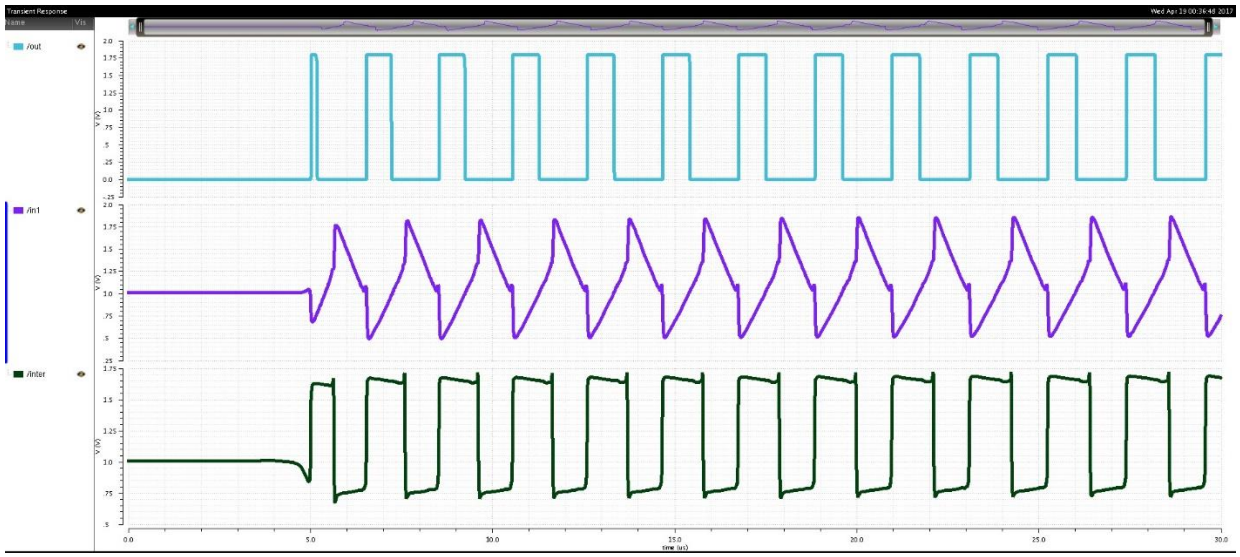

Figure 5.6: Simulation results of the clock generator

The first blue waveform in the picture above is the clock generator shows a clean rectangular waveform. The purple waveform is the output of the ring oscillator, this is the waveform fed as input to the comparator. The reference voltage of our comparator used in the clock generator is 1.1V. The purple waveform is compared to 1.1V and the output is given accordingly. The black or the last waveform in figure 5.6 is the output of the fourth inverter in the ring oscillator of the clock generator. This looks almost like the waveform we need but it is not smooth. The reason why we use another inverter.

5.3 CURRENT GENERATOR:

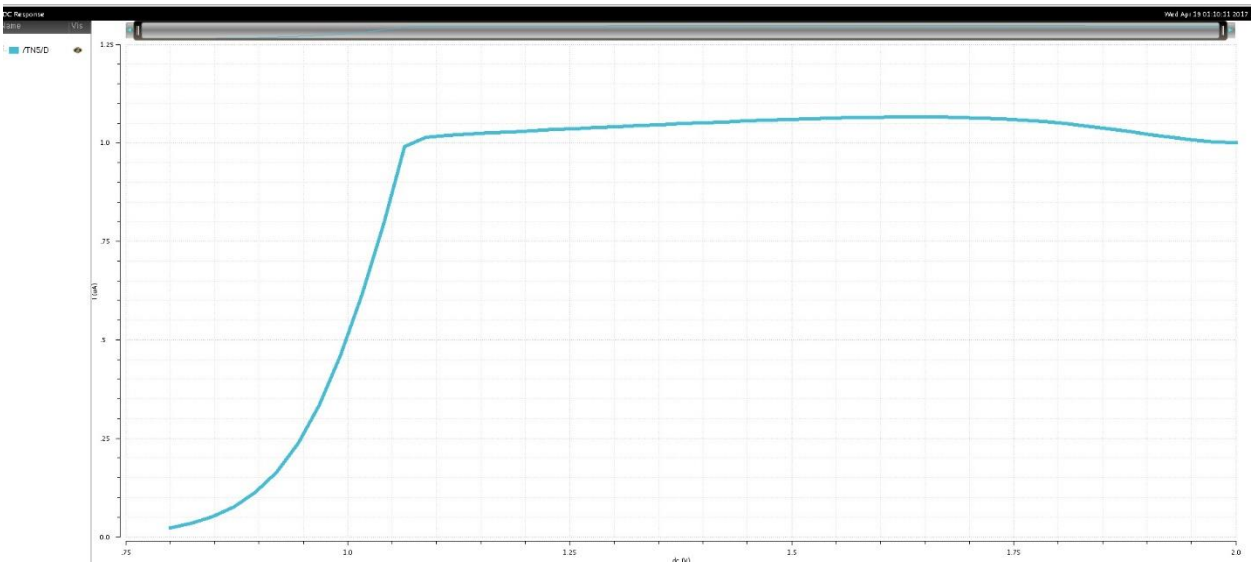

Figure 5.7: Simulation results of the current generator

From the figure 5.7,we can say that the output of the current generator is almost constant with respect to the supply voltage. It is slightly higher when closer to 1.8 but it is not a significant variation. By changing one PMOS transistor's parameters we can get different current values using the same circuitry.

5.4 NEURON SIMULATION:

Results are simulated on a time scale of 20 µ sec on the X-axis and amplitude on Y-axis.

Figure 5.8: Simulation results of the neuron with excitation current =120 nA

From figure 5.8, spikes(blue) for a DC excitation current(green) is shown. Whenever the membrane potential is greater than the threshold we see a spike fire. The spike has a duration of 15 ps.

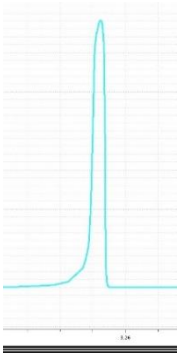

Figure 5.9: Individual Spike

CHAPTER 6: CONCLUSION

The comparison of different encoding schemes used by the biological neuron was clearly explained. The neuron was successfully implemented and an extensive test report was submitted in this thesis. This neuron can be used to build the temporal encoder. From the investigation of the neuron models, Leaky-Integrate-and-Fire model was used to implement for its simplicity and decent scalability. This neuron model used the rate encoding scheme because rate encoding scheme is simpler and good accuracy.

Neuromorphic chips can be used in different fields. In medicine, it can be used for better "therapeutic procedures", improve "medical imaging" and might help reconstruct the retina. These chips can help us monitor the environmental changes very easily. These chips can also be used in military, science, research and surveillance to implement smart techniques that will change our current life style. But this is possible only with the complete understanding of the brain [5].

Our design here consumed on 0.22µW of power. As a target in the future, the design should target much lesser area and power for wider applications. The neuromorphic chips available now focus on simple applications or in some situations a single or very less applications. Brain's mystery does cause some difficulty in building a more general system. However, "Life finds a way" [49] and so will scientists to give us better and more robust computers.

REFERENCES

[1] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Moore%27s_law.

[2] I. K. Schuller and R. Stevens, "Neuromorphic Computing: From Materials to System Architecture," U.S. Department of Energy, Gaithersburg, Maryland, 2015.

[3] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Neuromorphic_engineering.

[4] J. Markoff, "New York Times," [Online]. Available: http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html.

[5] N. A. Rodhan, "Global Policy," [Online]. Available: http://www.globalpolicyjournal.com/blog/18/02/2016/neuromorphic-computers-what-will-they-change.

[6] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Von_Neumann_architecture.

[7] J.-s. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha and D. J. Friedman, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *IEEE*, IEEE Custom Integrated Circuits Conference 2011.

[8] S. Anthony, "MIT," [Online]. Available: http://www.extremetech.com/extreme/105067-mit-creates-brain-chip.

[9] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Neurogrid.

[10] N. Sieslack, "Scientific Computing," [Online]. Available: http://www.scientificcomputing.com/article/2014/04/brain-derived-computing-beyond-von-neumann.

[11] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/BRAIN_Initiative.

[12] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/China_Brain_Project.

[13] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Dharmendra_Modha.

[14] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/TrueNorth.

[15] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Action_potential.

[16] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Neuron.

[17] S. Furtak. [Online]. Available: http://nobaproject.com/modules/neurons.

[18] Denise. [Online]. Available: https://denisezannino.wordpress.com/2014/09/30/neural-circuitry/.

[19] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Neuron.

[20] W. Gerstner, W. M. Kistler, R. Naud and L. Paninski, Neuronal Dynamics : From Single Neurons to Networks and Models of Cognition, Cambridge University Press.

[21] "Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Biological_neuron_model.

[22] D. Mishra, A. Yadav, S. Ray and P. K. Kalra, "IIT Khanpur," [Online]. Available: http://www.iitk.ac.in/directions/feb2006/PRINT~KALRA.pdf.

[23] L. F. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron," *Elsevier Brain Research Bulletin.*

[24] A. L. Hodgkin and A. F. Huxley, "A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve," *The Journal of Physiology,* 1952.

[25] C. Zhao, J. Li and Y. Yi, "Making Neural Encoding Robust and Energy Efficient: An Advanced Analog Temporal Encoder for Brain-inspired Computing Systems," in *2016 IEEE/ACM International Conference on Computer-Aided-Design (ICCAD)*, 2016.

[26] C. Eurich , *Neural Dynamics of Neural Coding: Two Complementary Approaches to an Understanding of the Nervous System,* Christian Eurich, 2003.

[27] E. T. Rolls and A. Treves, "Science Direct," 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030100821100147X.

[28] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Neural_coding.

[29] H. Fairhead, "i-programmer," [Online]. Available: http://www.i-programmer.info/babbages-bag/325-mcculloch-pitts-neural-networks.html.

[30] R. Rojas, "Neural Networks: Threshold Logic".

[31] K. Kawaguchi. [Online]. Available: http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node12.html.

[32] C. Curto, A. Degeratu and V. Itskov, "Encoding Binary Neural Codes in Networks of Threshold-Linear Neurons".

[33] G. Cruz and G. Lowe, "Nature.com Scientific Reports," 2013. [Online]. Available: https://www.nature.com/articles/srep01220.

[34] R. B. Stein, "The Information Capacity of Nerve Cells Using a Frequency Code," *Biophysical Journal,* vol. 7.

[35] A. K. Seth, "Current Biology," 2015. [Online]. Available: http://www.cell.com/current-biology/abstract/S0960-9822(14)01646-7.

[36] D. Ferster and N. Spruston, "Cracking the Neuronal Code," *Science,* 1995 .

[37] C. Zhao, B. T. Wysocki, Y. Liu, C. D. Thirm, N. R. McDonald and Y. Yi, "Spike-Time-Dependent Encoding for Neuromorphic Processors," *ACM Journal on Emerging Technologies in Computing Systems,* vol. 12, 2015.

[38] S. Wu, S.-i. Amari and H. Nakahara, "Population Coding and Decoding in a Neural Field: A Computational Study," *MIT Press Journal,* 2002.

[39] N. Gupta and M. Stopfer, "A Temporal Channel for Information in Sparse Sensory Coding," *NCBI,* 2014.

[40] B. A. Olsahusen and D. J. Field , "Sparse Coding of Sensory Inputs," *Science Direct ,* 2004.

[41] S. J. Thorpe and J. Gautrais, "Rapid Visual Processing using Spike Asynchrony," *Advances in Neural Information Processing Systems,* pp. 901-907, 1997.

[42] L. S. Koutha, S. K. Dharmasagar and H. Omkar, *Design and Implementation of Instrumentation Amplifier for EEG using 180nm CMOS Technology,* 2015.

[43] B. Razavi, Design of Analog CMOS Integrated Circuits, McGraw-Hill.

[44] [Online]. Available: https://courses.engr.illinois.edu/ece110/content/courseNotes/files/?MOSFETs.

[45] [Online]. Available: https://www.quora.com/What-is-the-difference-between-NMOS-PMOS-and-CMOS-transistors.

[46] Y. Yi. [Online]. Available: http://www.ittc.ku.edu/~yangyi/teaching.html.

[47] D. Johns and K. Martin, Analog Integrated Circuit Design.

[48] "Citizendium," [Online]. Available: http://en.citizendium.org/wiki/Hybrid-pi_model.

[49] *Jurassic Park.* [Film].

[50] K. Kress, "Indybay," [Online]. Available: https://www.indybay.org/newsitems/2006/05/18/18240941.php.