ANALOG CIRCUIT DESIGN OF SPIKING NEURAL NETWORKS

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIRMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN

VLSI DESIGN & EMBEDDED SYSTEMS

Submitted by

SUNRIKA

(2K22/VLS/19)

Under the supervision of

KRITI SUNEJA & RAHUL THAKUR

Assistant Professor, Department of ECE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)
BAWANA ROAD, DELHI -110042

May, 2024

ELECTRONICS & COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
BawanaRoad, Delh-110042

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude and indebtedness to my high respected and esteemed guides Kriti Suneja and Rahul Thakur (Assistant Professor, Dept of ECE) for having suggested the topic of my project and for giving me complete freedom and flexibility to work on this topic. They have been very encouraging and motivating and the intensity of encouragement has always increased with time. Without their constant support and guidance, I would not have been able to attempt this project. I extend my sincere thanks to all my friends who have been patiently helped me directly or indirectly in accomplishing this project successfully.

Place: Delhi **SUNRIKA**

Date: 31st May, 2024 **2K22/VLS/19**

ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
BawanaRoad, Delh-110042

CANDIDATE'S DECLARATION

I **Sunrika**, Roll No. 2K22/VLS/19 student of M. Tech (VLSI Design and Embedded Systems), hereby declare that the Major project-II titled "**ANALOG CIRCUIT DESIGN OF SPIKING NEURAL NETWORKS**" which is submitted by me to the Department of Electronics & Communication Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not derived from any source without appropriate citation. This work has not previously formed the basis for the award of any Degree, Diploma Associate-ship, Fellowship or other similar title or recognition.

Place: Delhi **SUNRIKA**

Date: 31st May, 2024 **2K22/VLS/19**

ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
BawanaRoad, Delh-110042

CERTIFICATE

We hereby certify that the Major Project-II titled "**ANALOG CIRCUIT DESIGN OF SPIKING NEURAL NETWORKS**" being submitted by **Sunrika**, Roll No. **2K22/VLS/19** of Electronics and Communication Engineering Department, Delhi Technological University, Delhi in partial fulfilment of the requirements for the award of the degree of Master of Technology, is a record of the project work carried out by the student under our supervision. To the best of our knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: 31st May, 2024

KRITI SUNEJA & RAHUL THAKUR
SUPERVISORS

ABSTRACT

Analog circuit design for Spiking Neural Networks (SNNs) is the engineering discipline that can be used to model biological neuron's functionality by using electronic components. This approach bases its functionality on the analogue of SNNs and biochemical neurons that allows designing effective and reliable information systems. They are applied in complex pattern identification, adaptation, motor control, flexibility, immunity to noise, and self-repair mechanisms among others. LTSpice – an effective tool for circuit simulation, has been employed to consider specified neuron models and compare it. Inspired by the model of Hodgkin-Huxley (HH) Neuron that generates an imitation of the ionic currents which traverse the neuronal membrane to create action potentials, a second-order derivative equation of a simple neuron model was proposed by Izhikevich(2003) [6]. Circuit simulations in LTSpice and efficiency studies have been conducted on the implementation of these neuron models. A detailed study of simple model of spiking neuron, and reconfigurable analog version of the piecewise linear neuron model[8] with CCII components is done using AD844 and AD633 in LTSpice. LTSpice has been used to implement many kinds of neuron models to make an extensive analysis about them with special consideration to working with analog circuits for spiking neurons. Subsequently the implementations AD633 analog multipliers and AD844 operational amplifiers are used where these parts are necessary. The firing patterns they exhibit are those that are observed on biological neurons, and these circuits are meant to emulate. In order to enhance the biomedical realism and accuracy of the meant spiking signal following further development, the Integrate-and-Fire neuron concept and, in addition, a membrane recovery variable, will be incorporated. When the second-order derivative equation is in use, the shape is more like the spikes seen in cortical neurons. The first part of the equation is crucial since most mechanisms of spike formation are characterized by the increase in the membrane potential. We have demonstrated the analog circuit of the proposed second-order derivative equation which was earlier proposed in the form of differential equation using AD844 and AD633. It is demonstrated that this second order derivative equation is indeed effective by constructing the neuron models in LTSpice. Also the implementation of CCII using AD844 that constitutes a major part of the membrane

circuit of the neuron has enhanced the area efficiency of the existing reconfigurable analog version of piecewise linear model[8] up to a great extent making its practical implementation more cost effective. Theoretical studies of the analog neuron model depend profoundly on the CCII that is implemented using the help of AD844. It also enables minimal signal attenuation and high-speed net current transfer, which are crucial for emulating neuron functions.

CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1

INTRODUCTION

We must integrate numerical simulation of large-scale brain models with experimental investigations of the human and animal nervous systems to fully comprehend the functioning of the brain. Finding compromises between two seemingly incompatible objectives is necessary when we create these massive brain models with spiking neurons: To depict an individual neuron, the model must be:

1) computationally straightforward and
2) able to replicate the complex firing patterns found in organic neurons.

The study of analog implementation of neuro-systems has been sparked by the discovery that the analog neural computation principles used by the brain are fundamentally different from the digital ones used in classical computing. For these reasons, the principal choice for the direct application of neuro-systems is the use of sophisticated electrical components and analog circuits to simulate neurological functions. Neural models, brain dynamics, network topologies, and learning processes can all be prototyped using very large-scale integration (VLSI) technology to test different hypotheses, provided that a suitable reconfigurable platform for neural structure implementation is available.

1.1 TYPES OF NEURAL NETWORKS

Neural networks are computer programs that use machine learning (ML) techniques to mimic how the human brain functions. Speedier are neural

networks at processing data than traditional computers and are more adept at seeing patterns and solving problems. Neural networks can go by the titles Neural networks, both artificial and simulated (ANNs and SNNs).

1.1.1 Convolutional Neural Networks

Neural networks with convolutions, a very well-liked model of neural networking, use one or more convolutional layers along with a kind of multilayer perceptron. These layers may be fully connected or pooled. This neural network model recognizes and processes patterns in images applying concepts from matrix multiplication in particular and linear algebra. Convolutional layers in this model are able to generate feature maps that identify specific areas within a given visual input. After that, the website is broken down and scrutinized more closely to yield illuminating findings. Convolutional neural networks are useful for AI-based image recognition applications. In complex use cases including image classification, facial identification, optical character recognition (OCR), natural language processing (NLP), this kind of neural network is frequently employed. It's also used in signal processing and paraphrase recognition.

1.1.2 Deconvolutional Neural Networks

The actions carried out by convolutional neural networks are reversed by deconvolutional neural networks. This specific use of artificial intelligence (AI) looks for missing signals or features that the convolutional neural network may have missed while performing its intended task. Neural networks with deconvolution capabilities are helpful for many tasks, including image analysis and synthesis.

1.1.3 Recurrent Neural Networks

A recurrent neural network is one kind of neural network where the output from one step is used as the input for the subsequent step. Every input and output in a traditional neural network is independent of every other one. Remembering the prior words is necessary because in certain situations, it is necessary to predict the word that will come next in a sentence. Consequently, to tackle this issue, RNN was created and a Hidden Layer was used. The primary and most crucial feature of an RNN is its hidden state, which retains some sequence-specific data. Because the state keeps track of the previous input that was provided to the network, it is also known as memory state. To produce the result, it executes the same task on all inputs or hidden layers using the same parameters for each input. This reduces the complexity of the parameters compared to other neural networks. Text-to-speech applications, sales forecasting, and stock market forecasting are three common uses for recurrent neural networks.

1.1.4 Feed-forward Neural Networks

This version of a basic neural network transfers information in a single direction via a number of processing nodes before arriving at the output node. Large amounts of "noisy" data are processed by feed-forward neural networks, which produce "clean" outputs. The multi-layer perceptrons (MLPs) model is another name for this particular kind of neural network. The input layer, one or more hidden layers, and the output layer make up the architecture of a feed-forward neural network. These models, despite their different name, use sigmoid neurons instead of perceptrons, which enables them to handle nonlinear, practical problems. Feed-forward neural networks

provide the foundation for other neural network models, computer vision, facial recognition, and natural language processing.

1.1.5 Modular Neural Networks

Multiple separate neural networks overseen by an intermediary make up modular neural networks. Every independent network functions as a "module," utilizing different inputs to accomplish specific tasks related to the overall goal of the larger network. When computation is happening, the modules don't talk to each other or impede on each other's work. This improves the speed and efficiency of completing lengthy and complicated computational tasks.

1.1.6 Spiking Neural Networks

The Spiking Neural Networks – SNNs are, in fact, a particular type of artificial neural network that is different from a traditional ANNs since it resembles biological neurons. Here are a few crucial SNN features:

- Brain Model:

    1. Spikes: In SNNs, information is communicated between neurons by a series of brief, unique occurrences called spikes. When a neuron reaches a certain level of polarization of membranes, known as the threshold, a spike is created.
    2. Temporal Coding: This implies that the spiking activity does not merely encode information in terms of the presence or absence of a spike, but also the time of the occurrence of these spikes. Such spikes regimes should be accurately timed.

- Biological Realism:

    1. Dynamic Synapses: Two more physiologically realistic models of synaptic behaviour, short-term plasticity and STDP together for SNNs are often incorporated.
    2. Energy Efficiency: SNNs may offer greater efficiency in energy usage because their neurons are activated only when there is a stimulus or spike. Another advantage of progress variables is that it is easily observed and monitored because its value is register-based which is preferred for hardware implementations.

- Calculations:

    1. Event-Driven Processing: SNNs employ event-based processing unlike most other network types, which employ continuous processing. for this reason, SNNs are well suited for such application since neurons do not fire unless activated by a spike thus implying the model uses less power.

1.2 ADVANTAGES & DISADVANTAGES OF NEURAL NETWORKS

1.2.1 ADVANTAGES

- Frequently use online services that lower (but do not completely eliminate) systematic risk.
- Are constantly being expanded into new fields with more challenging problems.
- Can operate more effectively and longer than humans.
- Can be designed to take lessons from past results and use them to improve future computations.

1.2.2 DISADVANTAGES

- The development of the code and algorithms could take a while.
- If a self-learning system lacks transparency, it could be difficult to identify mistakes or changes to the assumptions.
- Usually, it gives an approximate range or sum that might not come to pass.
- It still depends on hardware, which may need to be maintained with effort and knowledge.

1.3 SPIKING NEURAL NETWORKS (SNNs)

Spiking neural networks (SNNs) are artificial neural networks that closely resemble natural neural networks. SNNs incorporate time into their operational model in addition to synaptic and neuronal state. The concept is that, unlike neurons in conventional multi-layer perceptron networks, neurons in SNNs only transmit information when a membrane potential, or the intrinsic property of a neuron linked to its membrane electrical charge, crosses a threshold. When the membrane potential crosses the threshold, the neuron fires, causing the potentials of nearby neurons to change accordingly. A neuron model that fires at threshold crossing is known as a spiking neuron model.

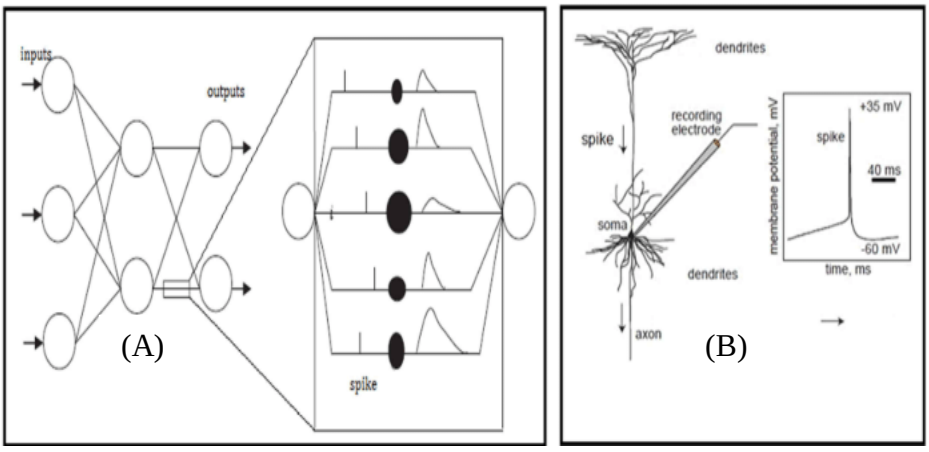


Fig 1. (A)Spiking Neural Nets and their Interconnections in Comparison to (B) Neurones that are biological [2]

Artificial neurons behave differently from biological neurons, even though they look a lot alike. The following are the main differences between artificial and biological NNs:

- General brain structure;
- Brain computations;
- Learning is a rule as opposed to the brain.

In 1952, Scientists Andrew Huxley and Alan Hodgkin created the initial Spiking Neural Network model. The action potential initiation and propagation in biological neurons were described by the model. In contrast, biological neurons do not transmit impulses directly. Neurotransmitters are chemicals that need to be exchanged in the synaptic gap in order for communication to occur.

1.3.1 The advantages of using Neural Networks with Spikes

- Energy Efficiency: SNNs are likely to be able to achieve higher levels of energy efficiency in comparison to more traditional ordinary ANNs due to their event-based operation and relative sparing usage, which makes them fitting for usage in power-constrained environments.

- Temporal Dynamics: SNNs are particularly useful in tasks involving sequences or time series because SNN naturally handles temporally dependent input and temporal relationships.

- Robustness: For some purposes, its asynchronous and event-driven topology may afford SNNs the edge of being less susceptible to noise and hardware failure.

1.3.2 Obstacles and Recent Studies

- Training Complexity: Due to the nonsmooth nature of spike events in SNNs, training of SNNs is a more complex process than that of conventional ANNs. New learning paradigms are proposed by authors, who with a reference to biological learning principles, such as STDP.

- Absence of Standardized Tools: However, there is relatively less number of existing conventional and systematic methodologies and tools when it comes to the construction and training of the SNNs as compared to the case with classic ANNs and this still comprises the current research area.

- Cost of Computing: Alas, to mimic the activity of spiking neurons in these cases, when using general purpose engineering and not SNN specialized technologies, it may cost too much to accurately imitate SNNs.

1.3.3 Uses

- Neuromorphic Hardware: SNNs are a very good fit and complement neuromorphic hardware that was designed based on the architecture and functionality of a human brain.

- Real-Time Processing: SNNs find their application as real-time classifiers – for applications in robotics, sensory motoring and real time decision making.

- BCI individuals are being adopted using SNNs due to its ability to fit the temporal characteristics of neural data.

1.3.4 Prospective Courses

As the fields of neuromorphic electronics and engineering advances, with improvements in training methodologies and the need for highly efficient and real-time processing, Spiking Neural Networks have a great future ahead of them. Research is gradually enhancing their responses toward biological realism, computing capability and adaptability to hard-tasking real-world scenarios.

1.3.5 In summary

The recent development of structures that can be considered as a new line of artificial intelligence staking to make a bridge between AI and BI is the spikey neural networks. SNNs offer an opportunity for alteration in disciplines that need rugged, real-time, low-power architectures using temporal dynamics and high efficiency by emulating biological neural systems.

1.3.6 How Does Spiking Neural Network Work?

The information propagation strategy is what sets an SNN apart from a conventional ANN. As much as is feasible, SNN aims to mimic a biological neural network. Because of this, SNN functions with discrete events that happen at set times as opposed to ANN's constantly fluctuating time values. A spike train is the term used to describe a set of spikes. The SNN receives

a set of spikes and outputs another set of spikes. The primary idea is as follows:

- The electrical potential of biological neurons is the value that each neuron possesses at any given time.
- A neuron's value can fluctuate according to its mathematical model; for instance, if it receives a spike from a neuron upstream, its value could increase or decrease.
- A neuron's value will instantly fall below its average if it exceeds a threshold and fires a single impulse to every neuron downstream that is connected to the first one.
- As a result, the neuron will have a refractory phase that is comparable to that of a biological neuron. Over time, the neuron's value will progressively revert to its average.

1.3.7 Neural Codes based on spikes

Neural computation is the aim of artificial spiking neural networks. Neural spiking must therefore be given meaning, and every computation-related variable must be described in terms of the spikes that spiking neurons use to exchange information. Based on biological understanding, numerous neural information encodings have been proposed:

- **Binary Coding**

In binary coding, an all-or-nothing encoding, an active or inactive neuron will release one or more spikes over a predetermined period of time. This encoding was backed by the discovery that physiological neurons exhibited a predisposition to fire in response to input (a sensory stimulus such as light or external electrical inputs). Since individual neurons are represented as binary units that can only accept two on/off values, this binary abstraction can be advantageous to them. Spike train classification is another use where

it may be used to analyze spike trains from pre-existing spiking neural networks. It uses a binary interpretation of the output spike trains.

- **Rate Coding**

Rate coding removes the temporal aspect of spikes by utilizing only the rate of spikes within an interval as a meter for the information delivered. The mechanism underlying rate encoding is the fact that physiological neurons fire more frequently in response to stronger (sensory or artificial) stimuli. Once more, it can be used to the analysis of spike trains or even individual neurons. As rate neurons, the neurons in the first example transform real-valued input numbers, or "rates," into an output, or "rate," at each time step. The idea of rate coding has served as the foundation for traditional artificial "sigmoidal" neurons in technological settings and cognitive research.

- **Fully Temporal Codes**

Accurate spike timing is necessary for the whole temporal code's encoding. Neuroscience research indicates that spike-timing can be extraordinarily accurate and consistent. A precise (internal or external) event, like the start of a stimulus or the spike of a reference neuron, is associated with a timing in a fully temporal code.

- **Latency Coding**

It's the time, not the number of spikes, that counts in latency coding. Information is encoded using the period of time between a certain (internal or external) occurrence and the first spike. This is based on the observation that upstream neurons spike earlier in response to significant sensory inputs. This encoding, among other supervised and unsupervised learning methods,

has been used with the Spike Prop and Chronotron. The sequence in which neurons in a group fire their initial spikes encodes information about a stimulus and is intimately related to rank-order coding. It's the time, not the number of spikes, that counts in latency coding. The period of time between a certain (internal or external) event to the first spike is utilized.

1.3.8 Architecture of SNN

In an SNN architecture, reconfigurable scalar weights describe spiking neurons and connecting synapses. The data input in analog form is first converted employing a spike train population coding method, a sort of temporal coding, or a rate-based methodology in order to create an SNN. A genuine neuron in the brain receives synaptic inputs from neural network neighbours, just like a simulated spiking neuron does. Network dynamics and action potential creation are both demonstrated by biological brain networks.

Compared to real biological networks, artificial SNNs have far simpler network dynamics. Assuming that the spiking neurons in the model have pure threshold, dynamics is helpful in this situation (unlike aspects of post-inhibitory rebound, hysteresis, resonance dynamics, or refractoriness). An action potential, or spike, is produced when the membrane potential of postsynaptic neurons above a threshold that is influenced by the activity of presynaptic neurons.

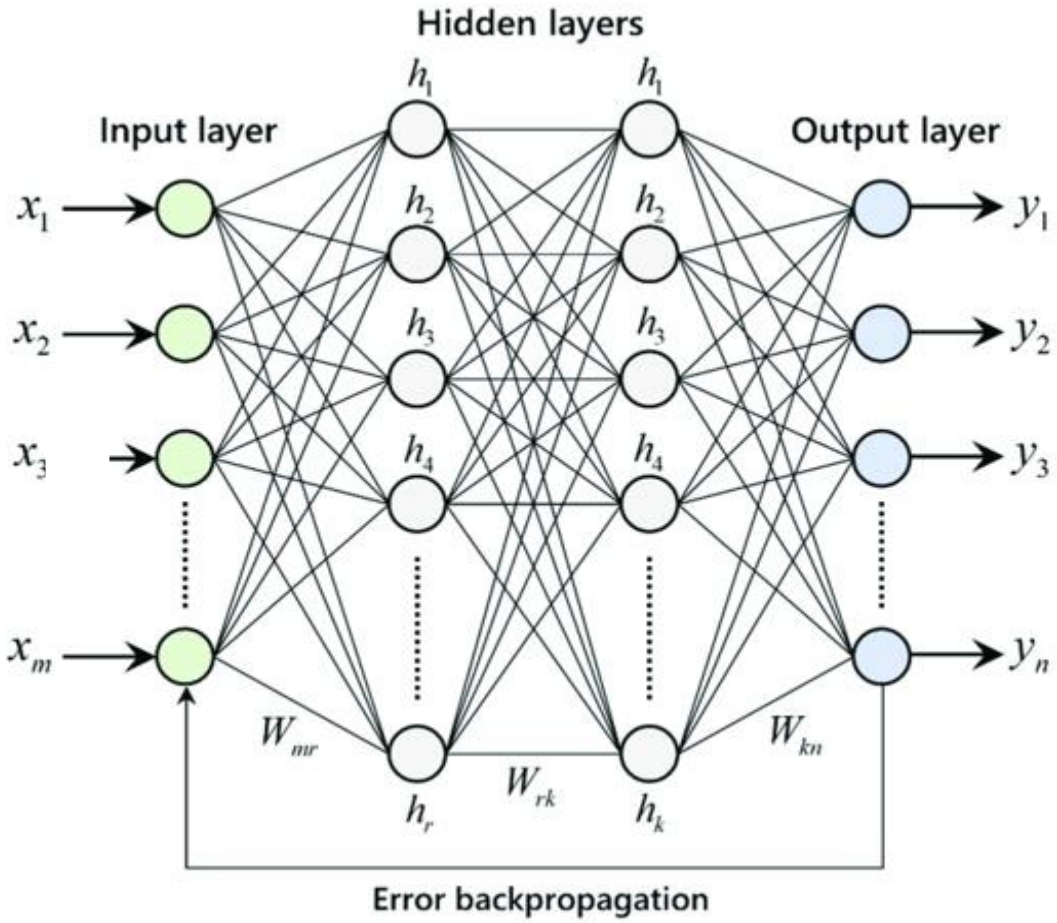


Fig 2: Architecture of Multilayer SNN [1]

1.3.9 Learning Rules in SNN

To achieve learning, use scalar-valued synaptic weights, whether spiking or non-spiking. In non-spiking networks, it is impossible to replicate a specific type of bio-plausible learning rule; spikes make this possible. Neuroscientists have discovered other versions of this learning rule, which they have collectively dubbed spike-timing-dependent plasticity (STDP). Its primary characteristic is that, between time intervals of tens of milliseconds, the weight (synaptic effectiveness) connecting a pre- and post-synaptic neuron changes in response to their respective spike rates. Based on data that is local to the synapse and local in time, the weight is adjusted. Both

supervised and unsupervised learning strategies in SNNs are covered in the upcoming subsections.

- **Unsupervised Learning**

No label is applied to the data, and the network is not given any performance feedback. Finding statistical correlations in data and responding to them is a typical task. A prime example of this is Hebbian learning and its spiking generalizations, such STDP. While correlation analysis can be a goal unto itself, it can also be applied to later grouping or classification of data. By definition, synaptic weight is lowered if a post-synaptic neuron fires later, and strengthened if Shortly after a pre-synaptic neuron fires, a post-synaptic neuron fires. On the other hand, there are several physiological types of STDP, of which this traditional variety is only one.

- **Supervised Learning**

Through the correlation of (classes of) inputs with target outputs a mapping or regression between inputs and outputs is produced by the learning device in supervised learning, where data (the input) is paired with labels (the targets). An error signal is used to update the weights of the network by calculating the difference between the target and actual output. Under supervised learning, we can directly update the parameters using the targets; in contrast, reinforcement learning only gives us a general mistake signal, or "award," which shows the system's effectiveness. Practically speaking, it's difficult to distinguish between the two kinds of supervised learning.

1.3.10 Traditional Neural Networks Vs SNNs

A spiking neural network is a feed-forward, two-layered network with heterogeneous lateral connections in its second hidden layer. Information is sent by biological neurons using brief, sharp voltage spikes. We call these signals action potentials, spikes, and pulses. Because spiking neuron networks have the ability to hold temporal information in their signals, they are more powerful than non-spiking ones. Nevertheless, unique and physiologically more plausible rules for synaptic plasticity are also necessary for spiking neuron networks. It is not possible for spikes to travel randomly across neurons. The synapse, which consists of the beginning of the dendrite, the end of the axon, and a synaptic gap, is the most intricate part of the neuron. Previously believed to be merely a means of transmitting a signal from the axon to the dendrite, the synapse is now recognized as a sophisticated signal pre-processor that plays a crucial role in learning and adaptability. Some vesicles fuse with the cell membrane and release their neurotransmitter content into the extracellular fluid that fills the synaptic gap when a spike reaches the axonal (presynaptic) side of the synapse. The original concepts and models for artificial neural networks were developed more than 50 years ago, making them a relatively ancient computer science technique. McCulloch-Pitts threshold neurons, a theoretically simple model where a neuron transmits a binary "high" signal when the total of its incoming weighted inputs is greater than a threshold, were the first artificial neural networks. These neurons have been utilized in complex artificial neural networks like Hopfield nets and multi-layer perceptrons, despite the fact that they can only provide digital output. A multilayer perceptron with a single hidden layer for digital computations, for instance, is called a universal network because it can compute any function and produce a Boolean output. Second-generation neurons use a continuous activation function instead of a step or threshold function to calculate their output signals, analog input and output is appropriate for them. Activation functions that are frequently used are the sigmoid and hyperbolic tangent.

1.3.11 Implementing Spiking Neural Networks

SNNs have potential applications similar to those of conventional ANNs. Moreover, SNNs have the ability to activate biological organisms' central nervous systems, such as an insect looking for food in a foreign place. Their realism makes them useful for studying how biological brain networks function. A real neural circuit's topology and function can be hypothesized, and its plausibility can be evaluated by comparing recordings of the circuit to the output of the relevant SNN. Unfortunately, there aren't enough training procedures for SNNs, this may pose issues for SNNs in certain applications, such as computer vision.

1.3.12 Pros and Cons of SNN

Pros

- A dynamic system is SNN. It performs exceptionally well in dynamic processes such as dynamic picture identification and speech.
- An SNN can continue to train even after it has begun operating.
- Training an SNN just requires training its output neurons.
- Compared to regular ANNs, which usually have more neurons, SNNs often have fewer neurons.
- SNNs can operate very quickly because impulses, as opposed to continuous values, are sent by the neurons.
- They have improved noise immunity and information processing productivity by utilizing the temporal presentation of data.

Cons

- At the moment, there isn't a learning algorithm created especially for this assignment; SNNs are challenging to train.
- It is not feasible to construct a small SNN.

CHAPTER 2

ANALOG IMPLEMENTATION OF ANNs

2.1 COMPARISON BETWEEN ANALOG & DIGITAL IMPLEMENTATIONS

While it has been established that digitally constructed simulators are helpful and practical for researching neural network behaviour, they are not suitable for large-scale, realistic, real-time simulations of neural systems or for actual biologically plausible systems.

In the future, these kinds of capabilities might be provided by custom digital systems that make use of Field programmable gate arrays (FPGAs) or parallel Graphics processing units (GPUs). However, it's unclear how these systems could potentially get close to the resilience, density of the neurons and synapses they simulate, as well as energy efficiency. This field of analog implementation of neuro-systems has been the subject of research since it was found that the rules of neural computation, which are analogous to those of classical computing but fundamentally different from each other, govern how the brain operates.

For these reasons, the primary technique for the direct building of neuro-systems involves the use of intricate electrical components and analog circuits to simulate neurological activities. Testing can be done by implementing very large-scale integration (VLSI). different ideas via prototyping brain models, neural dynamics, network topologies, and learning mechanisms, provided that a suitable reconfigurable platform for neural structure implementation is available.

Analog techniques are fast, efficient, and capable of simulating brain activity right down to the ion channels in the membrane of each neuron when it comes to VLSI implementation; yet, they are rigid and take a long time to develop. For designers of biologically plausible neuro-systems, reconfigurable platforms can offer small and adaptable solutions as a midpoint in the design space.

2.2 HARDWARE IMPLEMENTATION USING FPGAs

Neural networks are based on massive amounts of multiply accumulate calculations. By simulating the interactions of thousands of neurons, these computations generate a statistical probability of an event. This is a measure of the network's confidence in identifying a specific object in images. Of course, it could be incorrect. People frequently perceive human faces in inanimate objects, for instance! Thus, any system will need to have some margin for error in its results. Since the final answer must stay within the acceptable degree of accuracy specified by the application's tolerance, the statistical nature of these results allows for flexibility in the dynamic range of the computations. As a result, inference offers chances to use various data types creatively.

It is frequently possible to reduce data widths to 8-bit integers, and in certain situations, to just one bit. Almost any size of data can be processed by FPGAs with little to no loss in compute utilization [5]. There are benefits and drawbacks to using ASICs, CPUs, GPUs, and FPGAs for neural network inference. The best performance and lowest cost are provided by custom chips (ASICs), but only for specific algorithms; no flexibility is offered. The most programming flexibility, however, is provided by CPUs, albeit at a lower compute throughput. In general, GPU performance outperforms CPU performance and is further enhanced when a large batch number is used, meaning multiple queries are processed in parallel.

It is not always possible to batch input data in latency critical real-time systems. Neural networks can be optimized for a single query and still achieve high-level compute resource utilization in this domain, which is one area where FPGAs are somewhat unique. FPGAs are perfect for processing neural networks with latency criticality when an ASIC is not available.

CHAPTER 3

DIFFERENT NEURON MODELS

3.1 WHAT IS A NEURON?

A mathematical model is called an artificial neuron or neural node. Most of the time, it calculates the input's weighted average before biasing it. After that, this resultant term is run via an activation function. This activation function is nonlinear, just as the sigmoid function; given a linear input, it generates a nonlinear output.

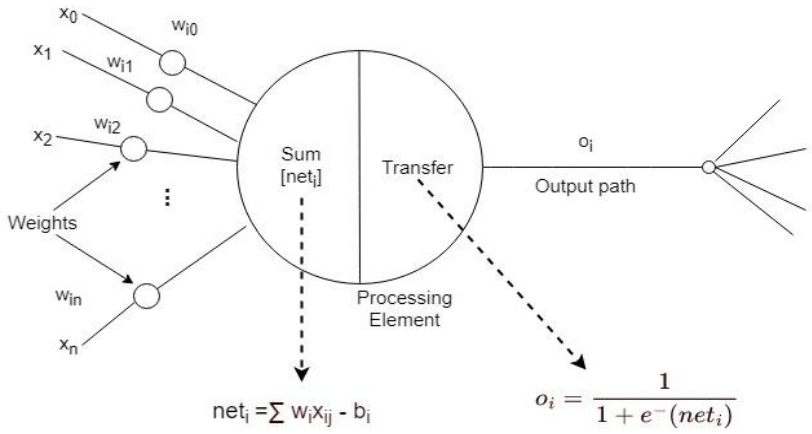
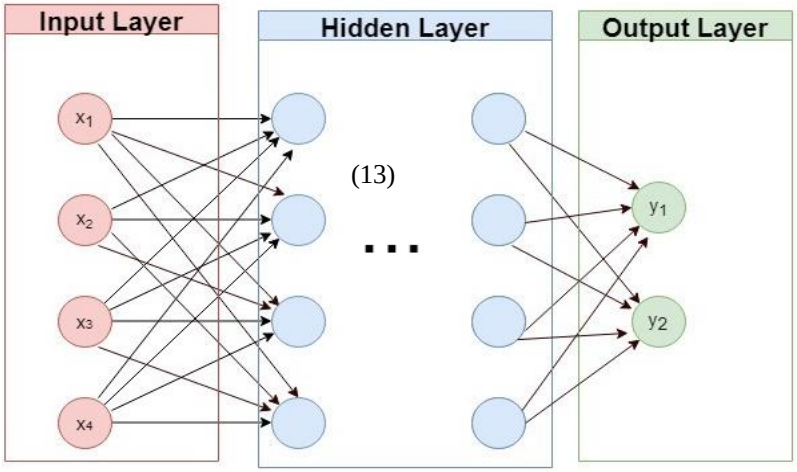


Fig 3: Model of Artificial Neuron [3]

Neural nodes, or layers of neurons, make up a typical network of neuron. These layers come in three main varieties:

- single input layer;
- one or more hidden layers;
- single output layer

Every neural node has a threshold and a weight, and they are all connected to one another. After undergoing some transformation on an input, it sends back an output. Any node that has an output that exceeds the designated threshold value is activated. It then transmits data to the network's subsequent layer. If not, it stays inactive and doesn't send any data to the network's next tier.



(13)

Fig 4: Layers of Neural Network [2]

3.2 TYPES OF NEURON MODELS

3.2.1 LIF MODEL

The LIF model encapsulates the following characteristics of a neuron:

- integrates synaptic inputs both spatially and temporally;
- produces spikes when voltage hits a threshold, converting raw analog impulses into spikes;
- possesses a leaky membrane;
- becomes refractory during the action potential.

The spatial and temporal integration of inputs is assumed to be linear in the LIF model. Furthermore, compared to genuine neurons, the dynamics of membrane potentials near the spike threshold in LIF neurons are far slower.

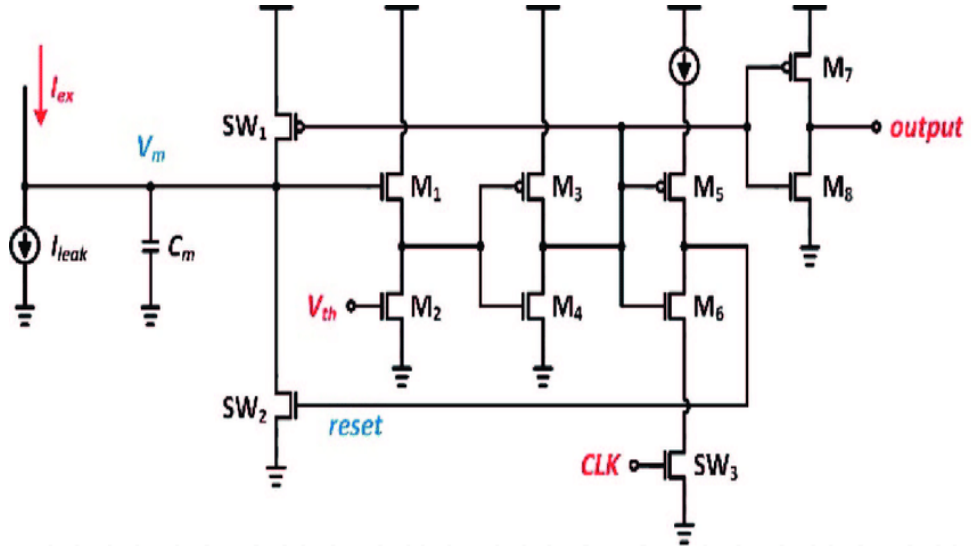


Fig 5: Model of an analog electronic circuit for a LIF neuron.[11]

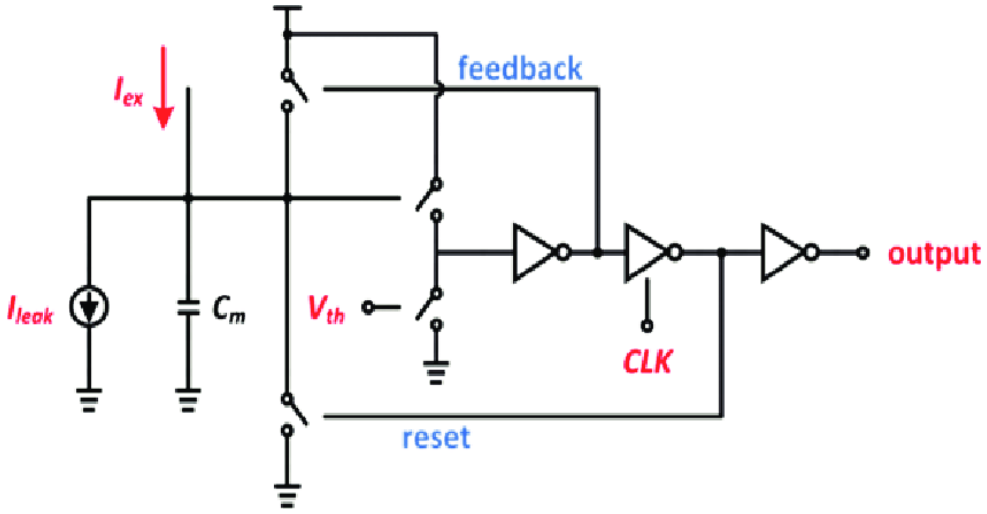


Fig 6: LIF neuron model in simplified analog electronic circuit.[4]

3.2.2 OZ NEURON

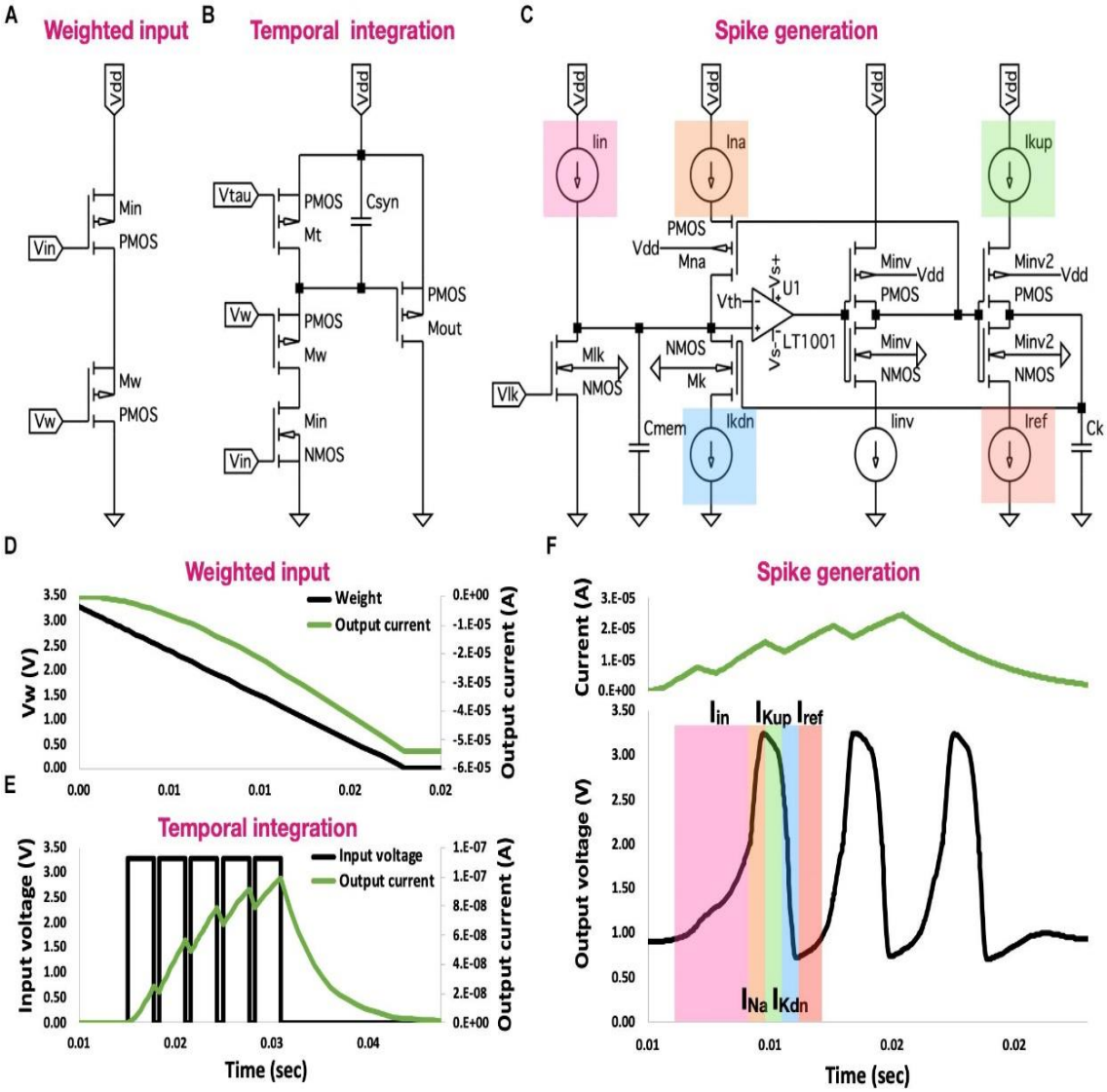


Fig 7: OZ neuron's building blocks. [12]

3.2.3 IF (INTEGRATE AND FIRE) NEURON

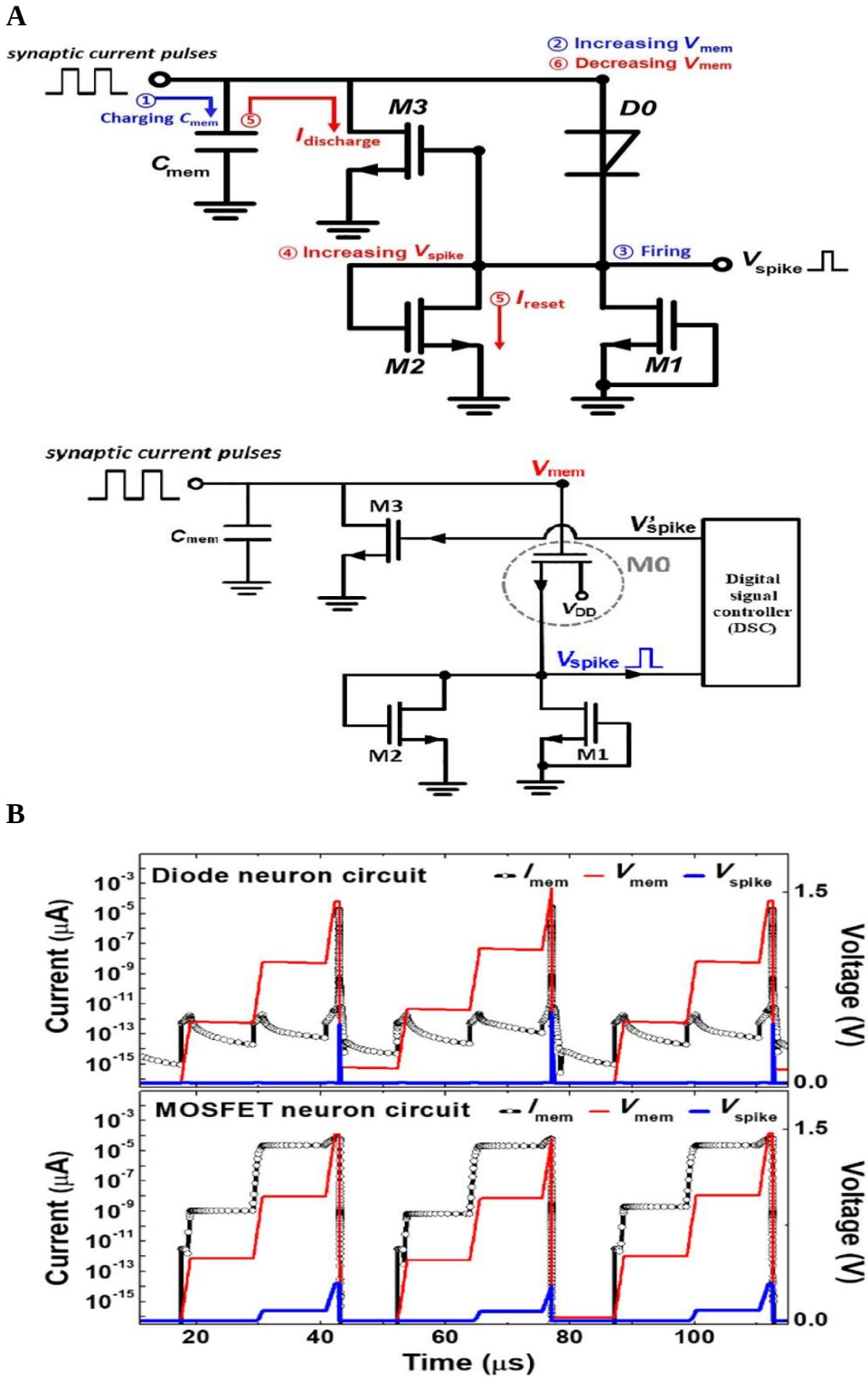
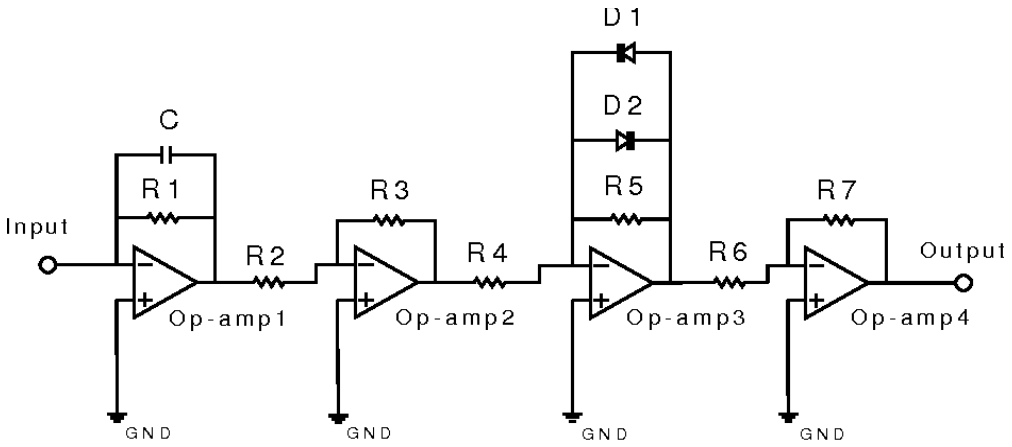
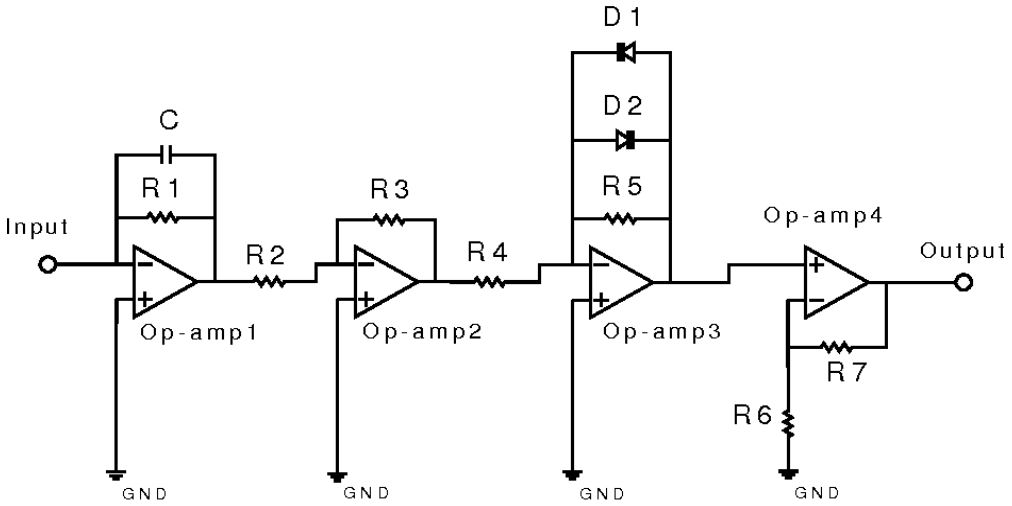


Fig 8: **(A)** Schematic of the diode (with operation mechanism without external bias) & using a digital signal controller in the MOSFET neuron circuit, **(B)** In relation to time, the pulses of the MOSFET and diode neuron circuits in synaptic current, membrane voltage, and spike voltage. [11]

3.2.4 Neuron circuit with smooth nonlinear output function



(a) Excitatory neuron circuit.



(b) Inhibitory neuron circuit.

Fig 9: (a) is an excitatory neuron circuit and (b) is an inhibitory neuron circuit. [14]

CHAPTER 4

SPICE MODELS

4.1 AD844

The junction isolated complementary bipolar (CB) process developed by Analog Devices, Inc. was used to create the high-speed monolithic operational amplifier known as AD844. It combines outstanding dc performance with a wide bandwidth and a rapid large signal response. It is suitable for many noninverting applications, even though it is optimized for use as an inverting mode amplifier and in current-to-voltage applications. The AD844 can replace traditional op amps, despite its current feedback architecture resulting in a very crisp pulse response, excellent linearity, and much improved ac performance. This kind of operational amplifier produces a closed-loop bandwidth that is primarily controlled by the feedback resistor and is essentially independent of the closed-loop gain.

The limitations on slew rate found in other current-feedback operational amplifiers and conventional op amps are not present in the AD844. For an output step of 20 V, the maximum rate of change in production can surpass 2000 V/μs. Generally, settling time ranges from 100 ns to 0.1% and is largely unaffected by gain. The AD844 is short-circuit protected up to 80 mA. It has low distortion, driving 50 Ω loads to ±2.5 V.

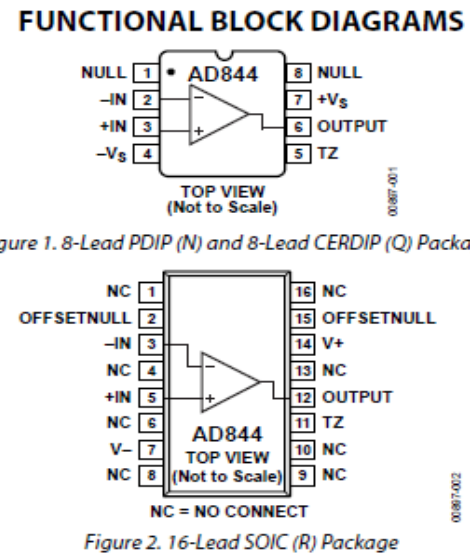


Fig 10: AD844 functional block diagram [12],[16]

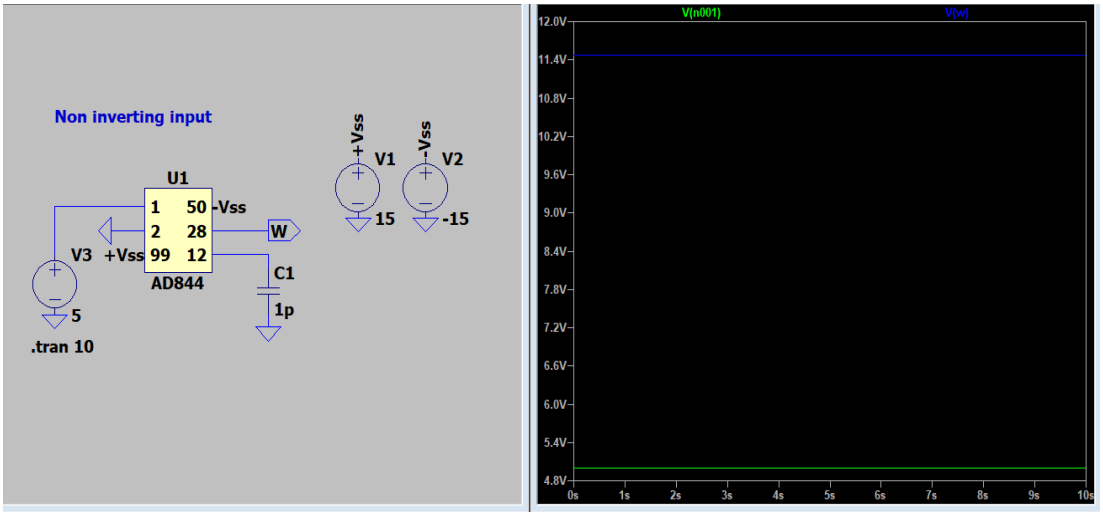


Fig 11: output of AD844 when non-inverting input is given

AD844 spice model has been imported in LTSpice and verified with the desired results of an operational amplifier for a non-inverting input, where port 1 is the non-inverting input port, y.

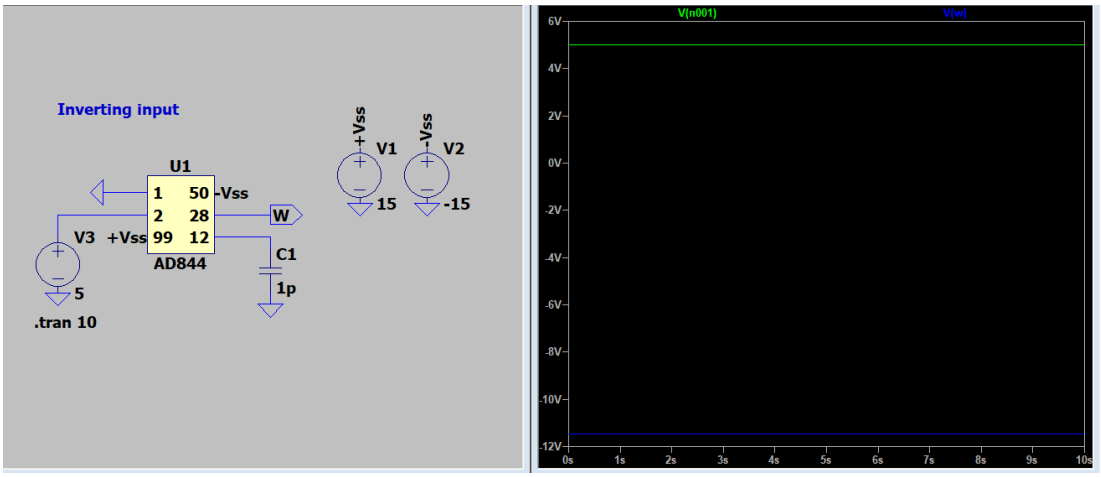


Fig 12: output of AD844 when inverting input is given

AD844 spice model has been imported in LTSpice and verified with the desired results of an operational amplifier for an inverting input, where port 2 is the inverting input port, x.

- The AD844 is a cheap, adaptable part It provides an excellent blend of ac and dc performance
- It is not restricted by slew rates. The output level practically has no effect on the increase and fall time intervals.
- It can drive loads as low as 50 Ω and very large capacitive loads with the use of an external network. It can operate on power supplies ranging from ±4.5 V to ±18 V.

The AD844's offset voltage and input bias currents are laser trimmed to reduce dc errors; the drift of bias current 9 nA/°C and the VOS drift is usually 1 μV/°C. With its outstanding differential gain and differential phase characteristics, the AD844 can be used in a wide range a bandwidth of up to 60 MHz for video based programs. It is an excellent choice for an analog-to-digital converter (ADC) flash input amplifier because it has a wide bandwidth, low noise, low distortion, and low drift.

4.2 AD633

An analog multiplier with four quadrants that operates fully is the AD633. It has a high impedance summation input (Z), differential X and Y inputs, and a high impedance. A concealed Zener provides the nominal 10 V full scale low impedance output voltage. The first product with these features available in reasonably priced 8-lead PDIP and SOIC packaging is the AD633. The AD633 can be laser calibrated to ensure a total accuracy of 2% of full scale. In a 10 Hz to 10 kHz bandwidth, noise at the output typically measures less than 100 µV rms, while nonlinearity for the Y input typically falls between 0.1% and 0.7%. The AD633 is a great option because of its 20 V/µs slew rate, 1 MHz bandwidth, and capacity to drive capacitive loads., For many uses where convenience and affordability are critical, the AD633 is a great tool.

The AD633's versatility is not diminished by its simplicity. The Z input allows the user to set up different applications, raise the multiplier gain, and add the results of two or more multipliers. It also provides access to the output buffer amplifier. There are packages for SOIC and 8-lead PDIP for the AD633. For commercial usage (J Grade), its operational temperature ranges are 0℃ to 70℃; for industrial use (A Grade), they are -40°C to +85℃.

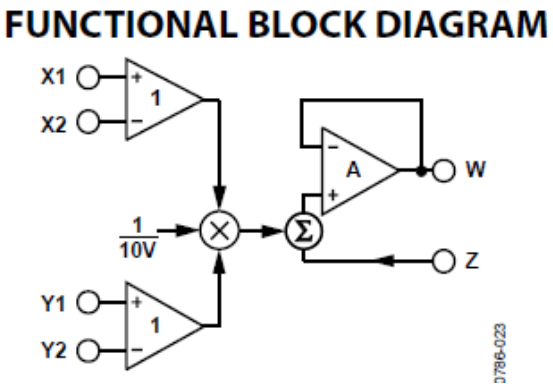


Fig 13: AD633 functional block diagram [12],[17]

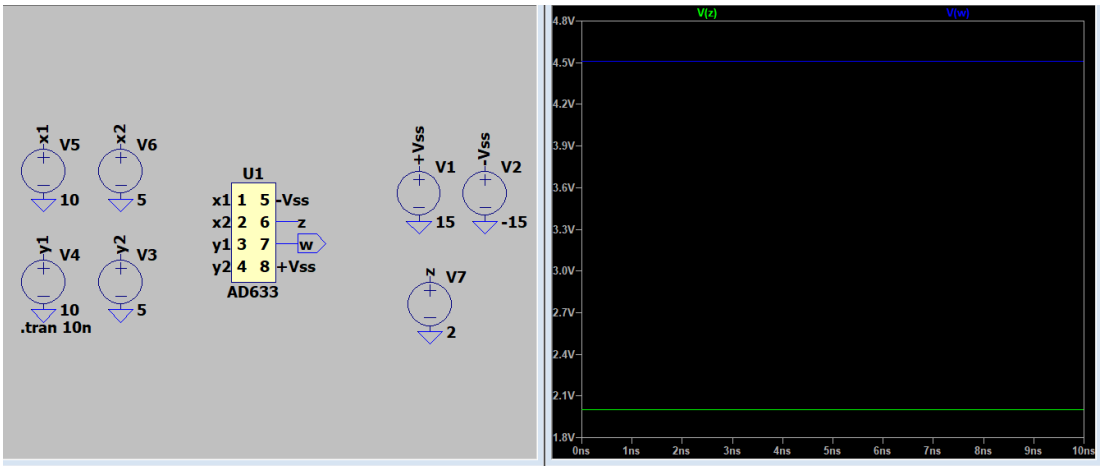


Fig 14: output of AD633, working as an analog multiplier

AD633 spice model has been imported in LTSpice and verified with the desired results of an analog multiplier with four quadrants input x1, x2, y1, y3 and high impedance summation input z.

- The affordable 8-lead SOIC and PDIP packages of the comprehensive four-quadrant multiplier AD633 are available. The end product is simple to use and reasonably priced.
- The AD633 can be used without the need for costly user calibration or additional components.
- The device is stable and dependable due to its monolithic design and laser calibration.
- The signal source loading is insignificant due to the large (10 MΩ) input resistances.
- The voltage range of the power supply is between ±8 and ±18 volts. A stable Zener diode produces the internal scaling voltage; multiplier accuracy is essentially supply insensitive.

4.3 IMPLEMENTATION OF NEURON USING AD844 & AD633

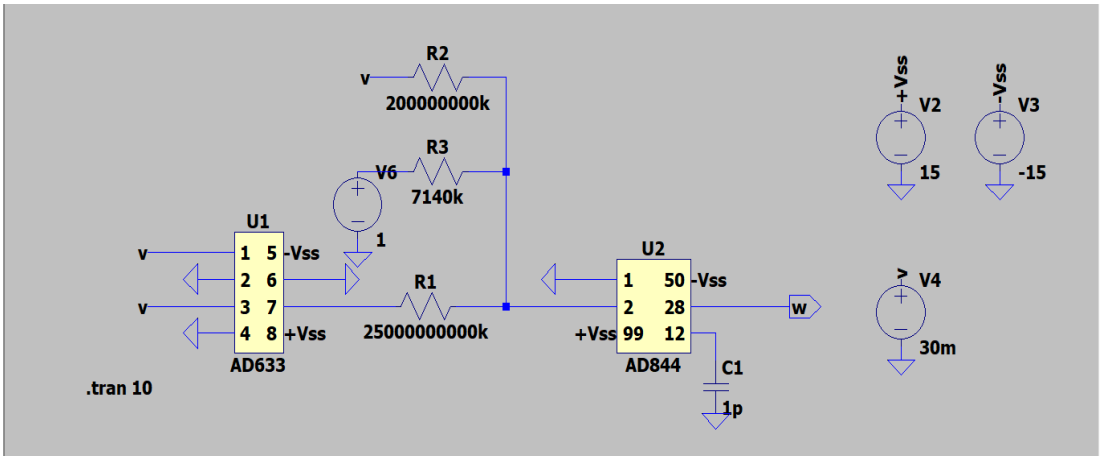


Fig 15: A simple neuron model using AD844 & AD633

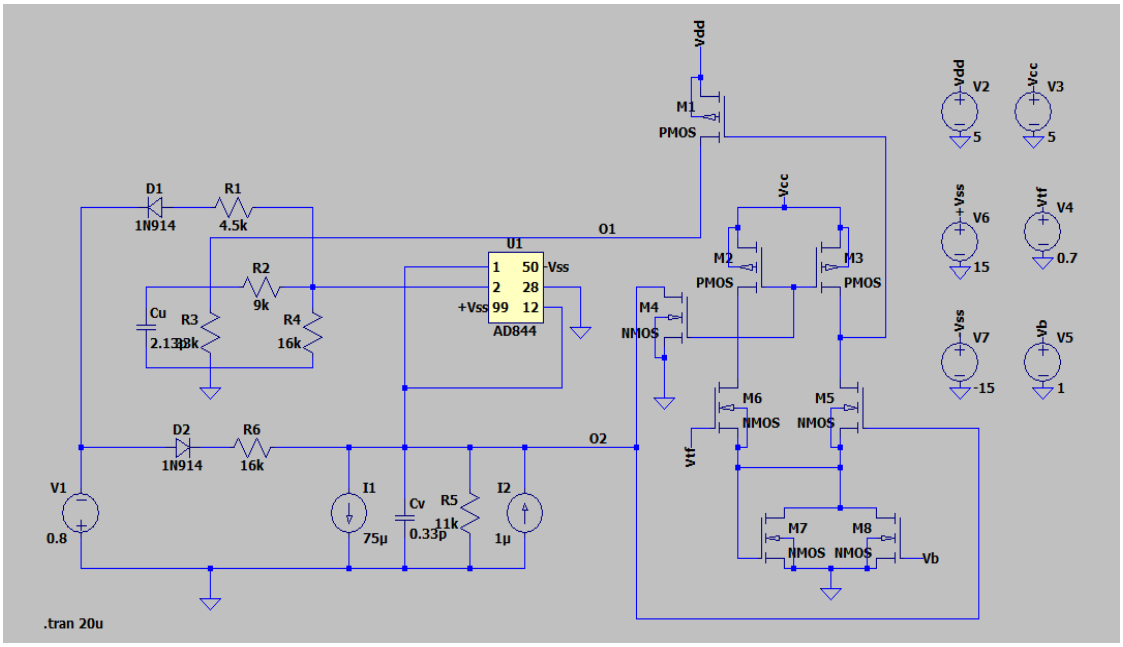


Fig 16: 2nd order piecewise linear neuron model using AD844

CHAPTER 5

SIMPLE MODEL OF SPIKING NEURONS

5.1 IMPLEMENTATION OF NONLINEAR ODEs

Large-scale brain models must be numerically simulated in conjunction with experimental research on the human and animal neurological systems to completely understand how the brain functions. When creating these expansive brain models with spiking neurons, we have to strike a balance between two seemingly incompatible requirements: A single neuron model needs to be:

- Computationally straightforward
- Capable of simulating the intricate firing patterns of real biological neurons.

Due to the fact that real-time neuron simulation is only possible with a small number of neurons, using computationally prohibitive Hodgkin-Huxley-type models that are biophysically accurate is not feasible. Conversely, though, while employing an integrate-and-fire model is more efficient in terms of computation, it is unrealistically basic and unable to replicate the complex spiking and bursting dynamics that are present in cortical neurons. Hodgkin–Huxley-type neuron models to 2- Dimensional system of Izhikevich's form for ordinary differential equations [6].

$$v' = 0.04\, v^2 + 5\, v + 140 - u + I$$

$$u' = a\,.(bv - u)$$

utilizing additional reset equations

$$v >= 30 \text{ mV}$$

$$\text{then } c \rightarrow v$$

$$u+d \rightarrow u$$

here I is the minimal current required for the quadratic IF neuron to fire repeatedly, and v, u, and constant parameters a, b, c, and d stand for the membrane recovery variable and membrane potential, respectively.

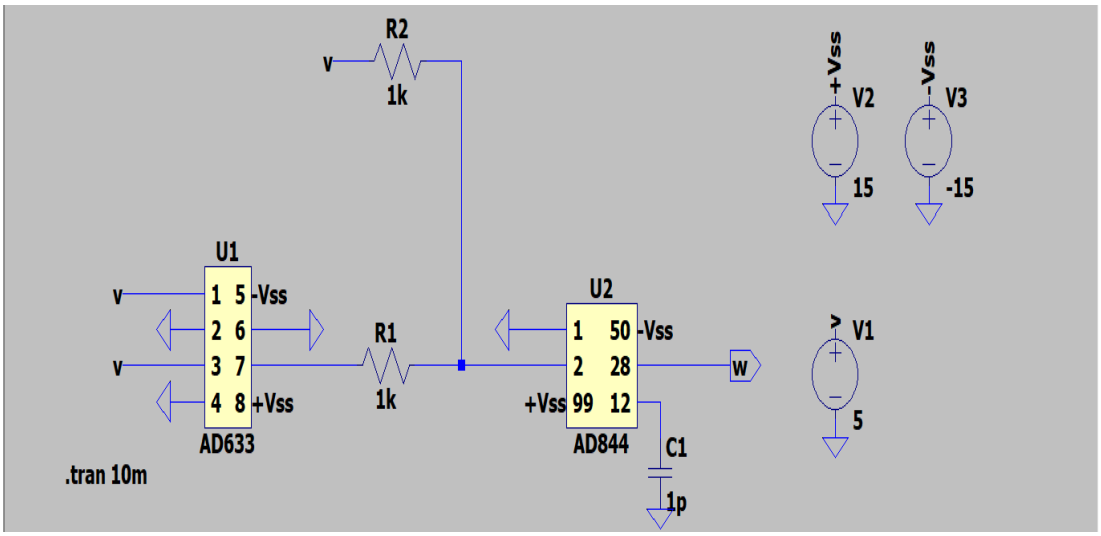


Fig 17: Spiking neuron circuit

5.2 NEURON RESPONSES & SIMULATIONS

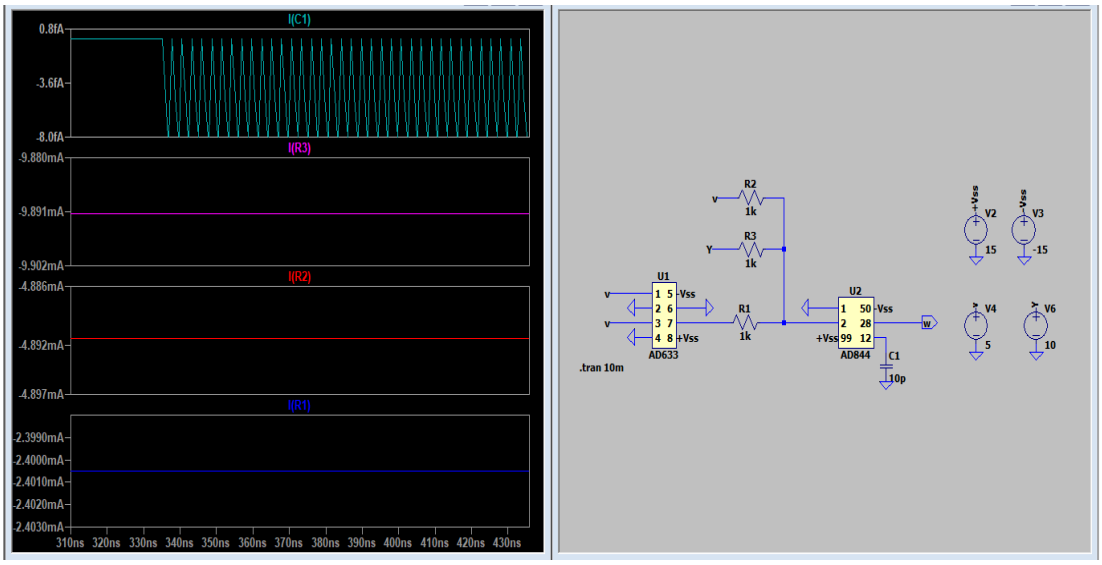


Fig 18: Spiking circuit with Y voltage

Spikes started generating when a basic neuron circuit is designed in LTSpice using AD844 & AD633, here all the three R1,R2 and R3 resistances are kept equal that is 1k ohm.

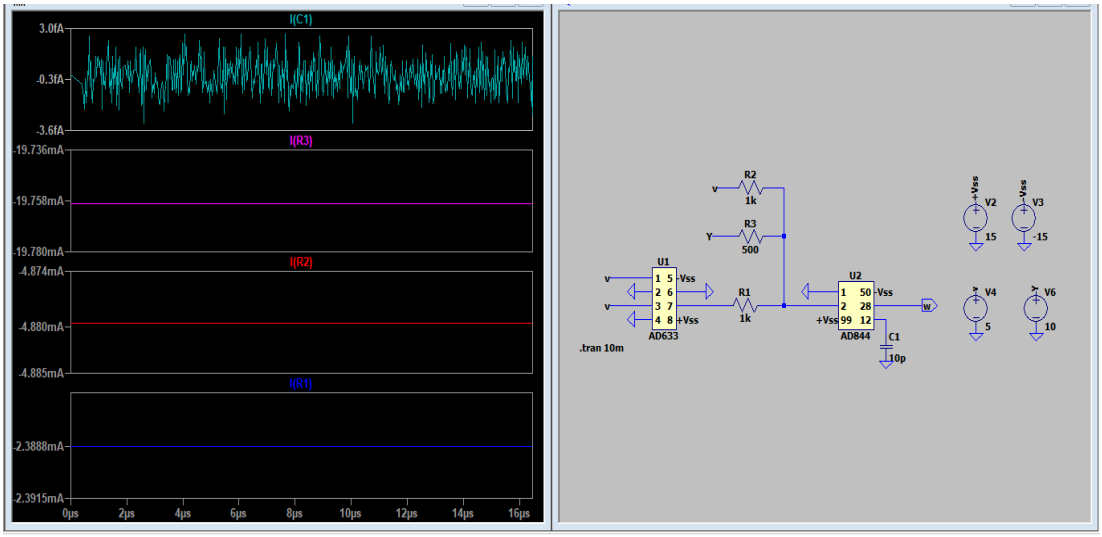


Fig 19: Spiking circuit with Y voltage and varied R3 resistance (R3<R1,R2)

Spikes started generating with greater resembles to the biological spikes, when a basic neuron circuit is designed in LTSpice using AD844 & AD633, here R2 and R1 are kept greater then R3 where R1=R2=1k ohm and R3=500 ohm.

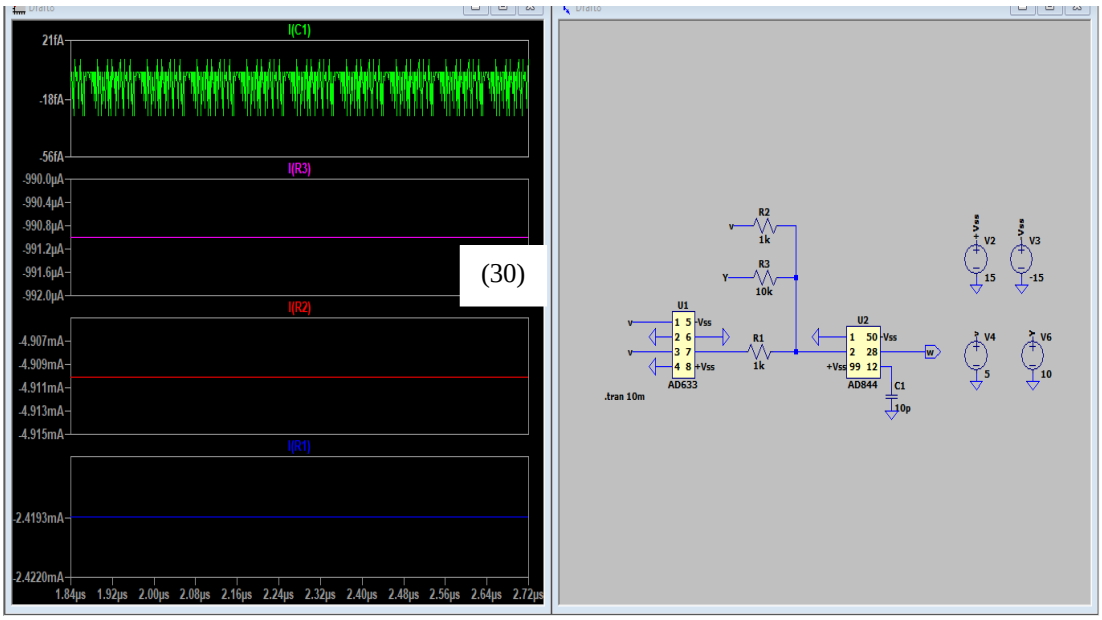


Fig 20: Spiking circuit with Y voltage and varied R3 resistance (R3>>R1,R2)

The frequency of the spikes is getting increased and nature of the spikes is in the resemblance to the biological neural spikes, when R3 resistance is increased up to a large extent in comparison with R1 and R2 which a kept to 1k ohm.

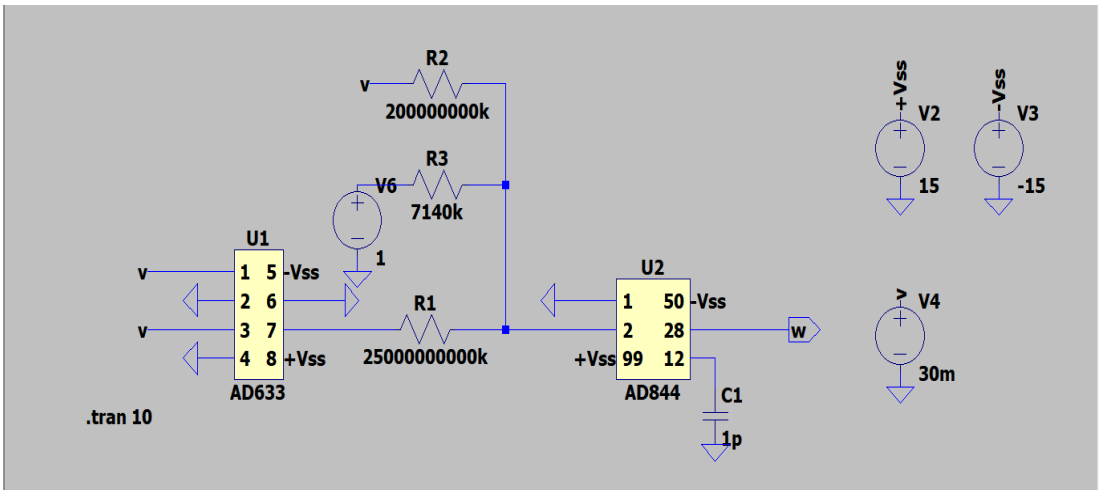


Fig 21: A simple neuron model using AD844 & AD633

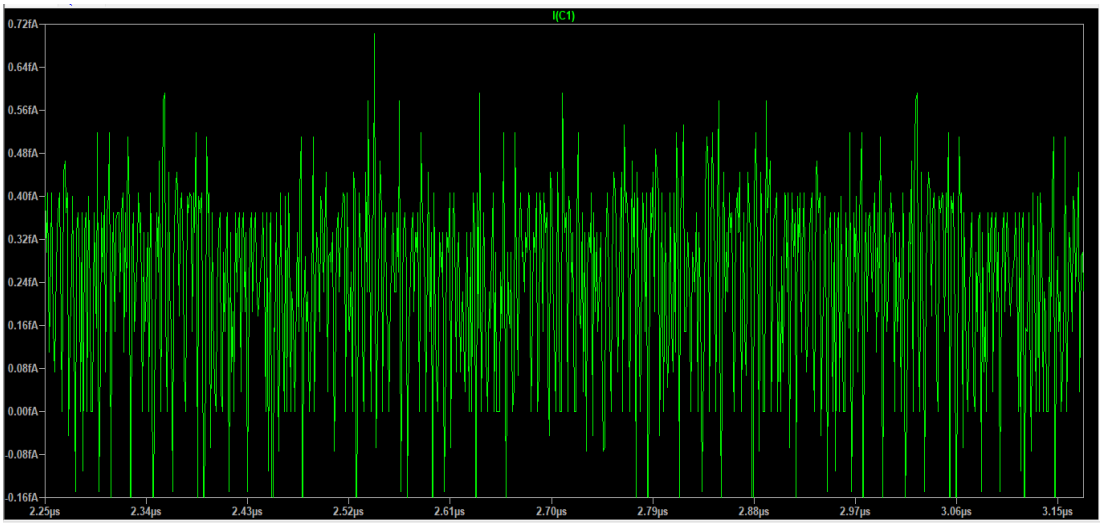


Fig 22: Tens of thousands of Spikes generation

After increasing the R1 and R2 very largely than the R3 resistance we are capable getting the desired output of generating tens of thousands of spikes in the real-time and the nature of the spikes is in the resemblance to the biological neural cells. Here the nature of the spikes is more accurate and realistic biologically in comparison to the above spikes.

CHAPTER 6

ANALOG APPLICATION OF PIECEWISE LINEAR NEURAL NETWORK USING CCII BLOCK CONSTRUCTION

With a contemporary conveyor of the second generation (CCII), a unique implementation of the neuron in a reconfigurable analog circuit has been described [8]. The Izhikevich model [6] has been modified into a new piecewise linear (PWL) circuit that can replicate various Cortical neurons exhibit dynamic characteristics.
Because of its regular form and usage of common building components, the circuit can be built as a reconfigurable analog device that is specifically designed for neural networks. This represents a step toward the development of programmable analog neural integrated circuits. Large-scale neural network implementation in analog is a good fit for this model due to its resistance to matching of the input and output impedance and noise.

6.1 CURRENT CONVEYOR OF SECOND GENERATION (CCII)

- **Conveyor of the First Generation (CCI)**

The three terminals that comprise the CCI are designated by the terminal names X, Y, and Z. Any voltage that is applied to Y is equal to the potential at X. Any current that enters Y also enters X, and as a variable constant current source, it has a high output impedance and is mirrored at Z. Current into Y generates current into Z in a sub-type CCI+, whereas current into Y causes an equivalent current to flow out of Z in a CCI-.The best way to depict its terminal characteristics is with a hybrid matrix that compares the three ports' outputs to their corresponding inputs. This relationship can be expressed for CCI as:

$$\begin{bmatrix} I_Y \\ V_X \\ I_Z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & \pm 1 & 0 \end{bmatrix} \begin{bmatrix} V_Y \\ I_X \\ V_Z \end{bmatrix}$$

Fig 23: Hybrid matrix for CCI [7]

- **Second Generation Current Conveyor (CCII)**

There is no current flow at terminal Y. One way to think of the perfect CCII is as an ideal transistor with flawless properties. Y stands for neither the gate nor the base, where no current flows. The emitter or source voltage (at X) follows the voltage at Y because there is no base-emitter or gate-source voltage drop. The emitter or source has an input impedance of zero (X), but the base or gate has an infinite (Y) input impedance. Even though the output impedance is infinite, any current that leaves the emitter, also known as the source (X), is reflected and flows into the collector, also known as the drain (Z). A CCII− is represented by this ideal bipolar or field-effect transistor because of the reversal of sensing between the X and Z currents. Z would be a CCII+ if the high-impedance current that was exiting it was also exiting X.

$$\begin{bmatrix} I_Y \\ V_X \\ I_Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \pm 1 & 0 \end{bmatrix} \begin{bmatrix} V_Y \\ I_X \\ V_Z \end{bmatrix}$$

Fig 24: Hybrid matrix for CCII [7]

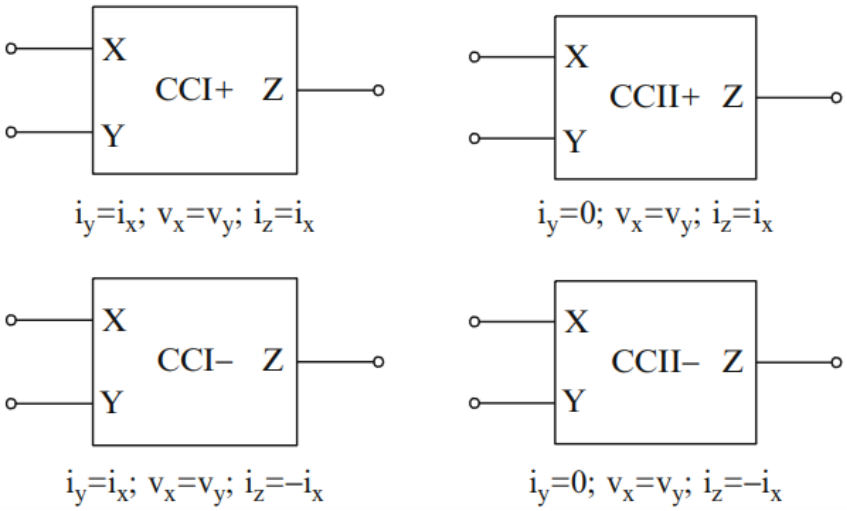


Fig 25: Block diagrams for CCI & CCII [7]

6.1.1 CMOS REALIZATION OF CCII

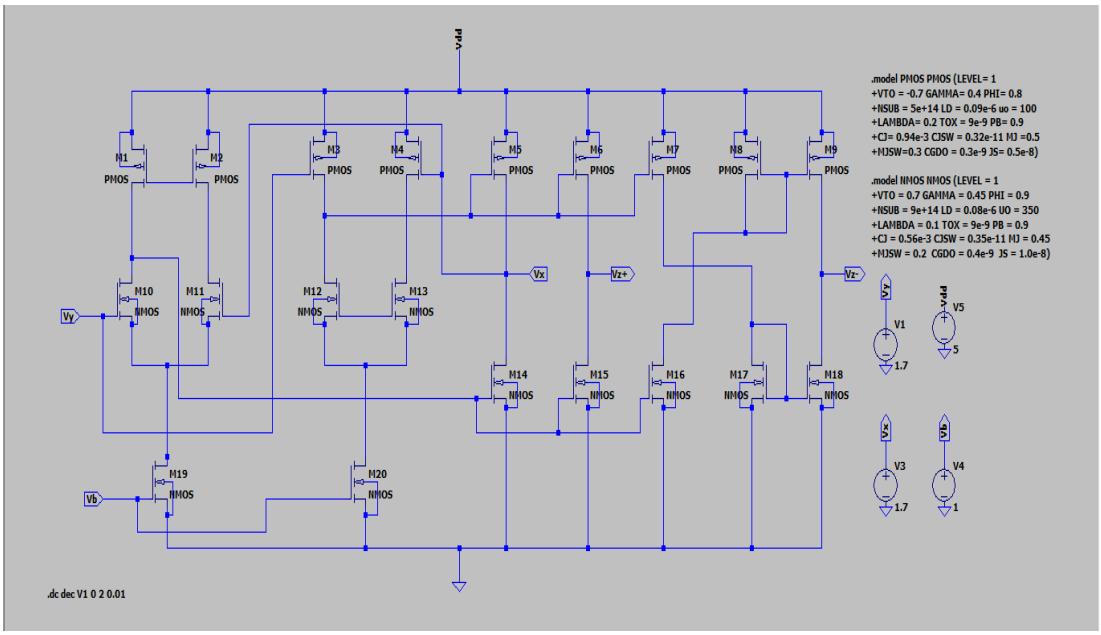


Fig 26: Realization of +CCII/-CCII using 350nm CMOS technology

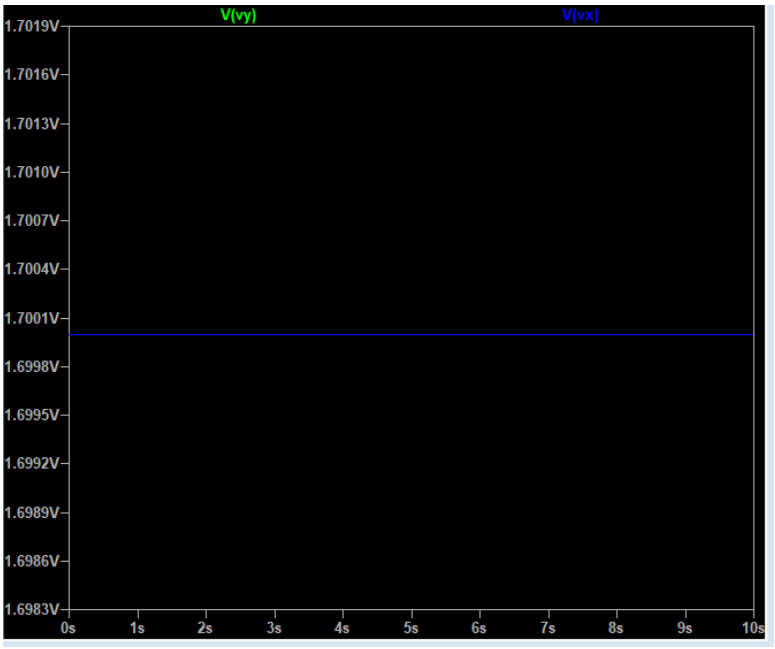


Fig 27: Output of +CCII/-CCII using 350nm CMOS technology, where the input voltages are verified i.e. Vx = Vy.

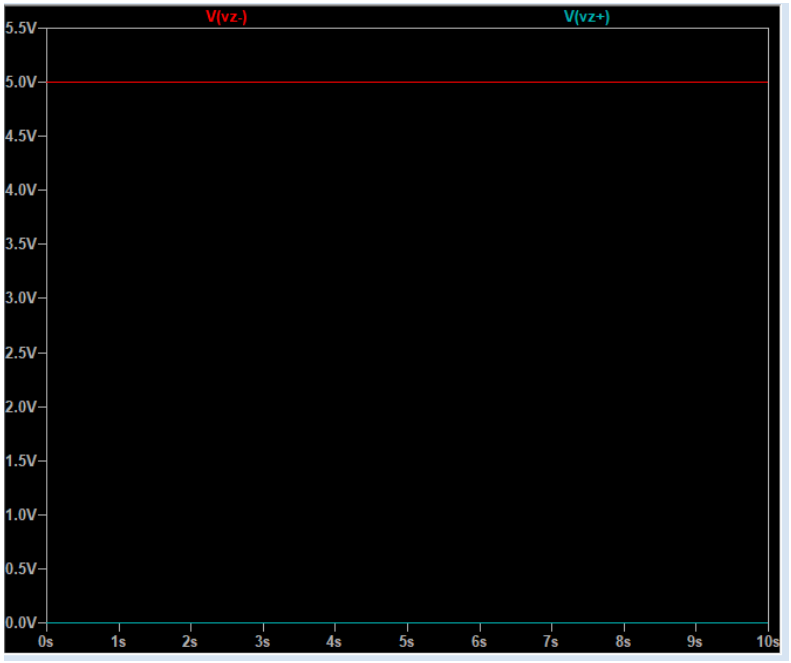


Fig 28: CCII using 350nm CMOS technology is giving an output at V(z-) terminal therefore functioning as CCII-.

6.1.2 CCII USING AD844

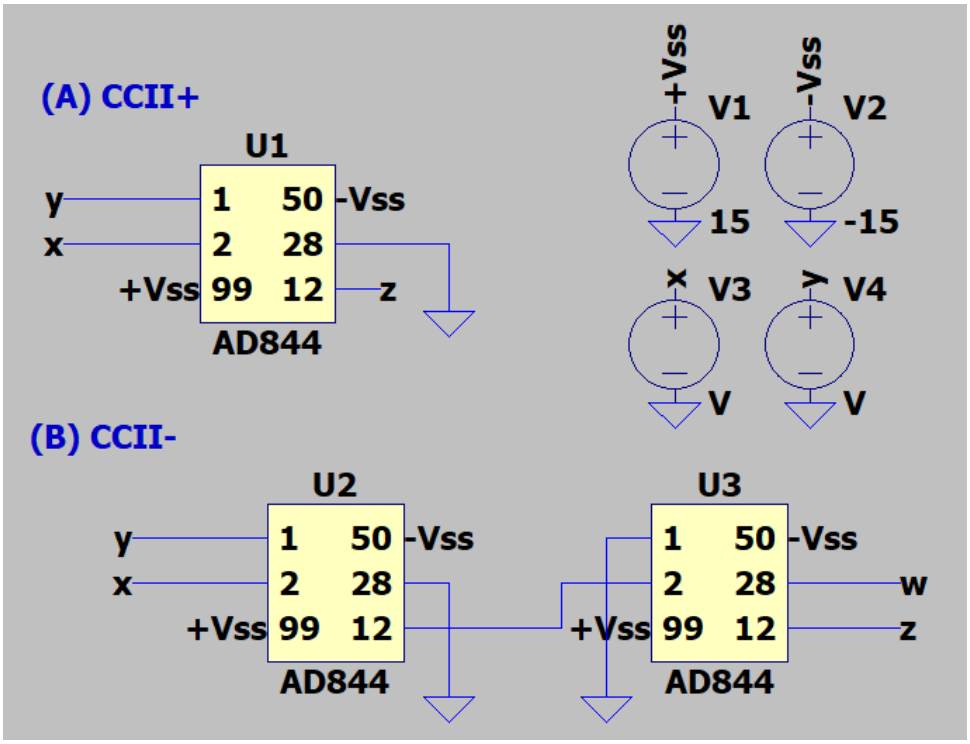


Fig 29: CCII+/CCII- implemented using AD844

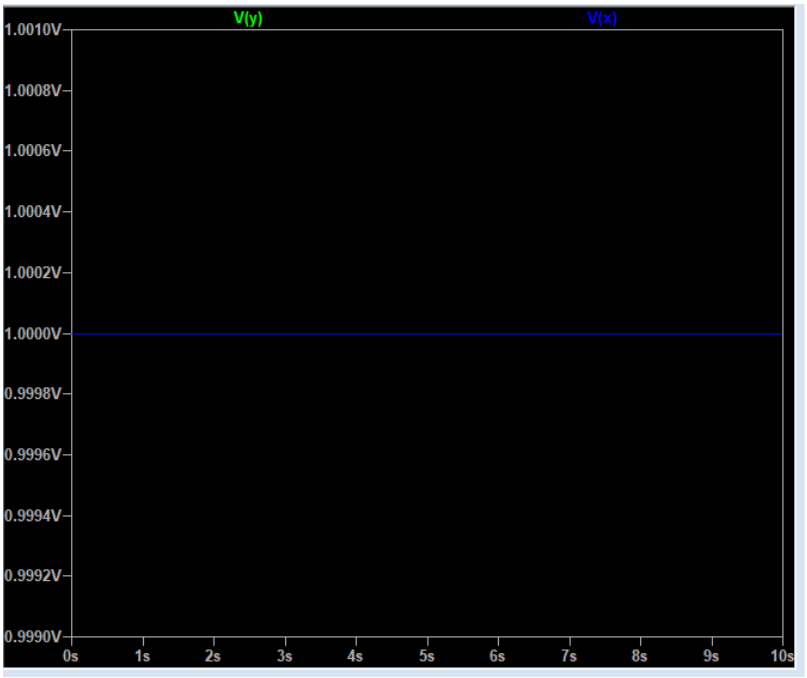


Fig 30: Output verification of the implemented CCII+, where Vx=Vy

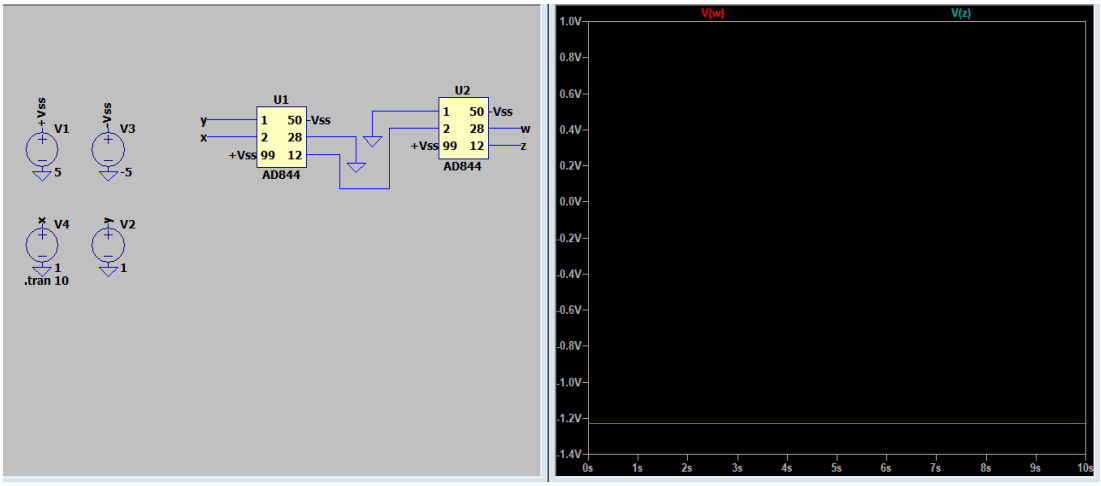


Fig 31: Output verification of implemented CCII-, where Vw=Vz

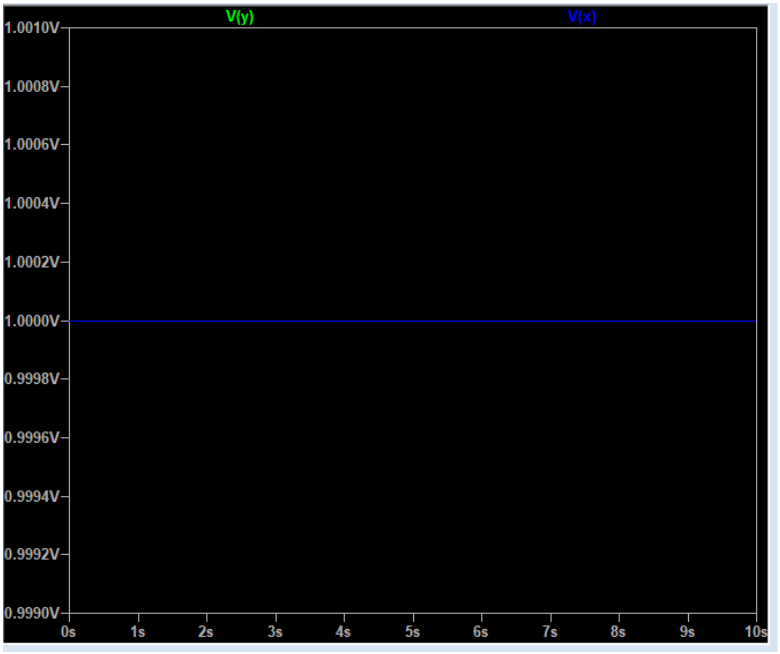


Fig 32: Output verification of implemented CCII-, where Vx=Vy

6.2 PROPOSED NEURON CIRCUIT

6.2.1 MEMBRANE CIRCUIT USING AD844

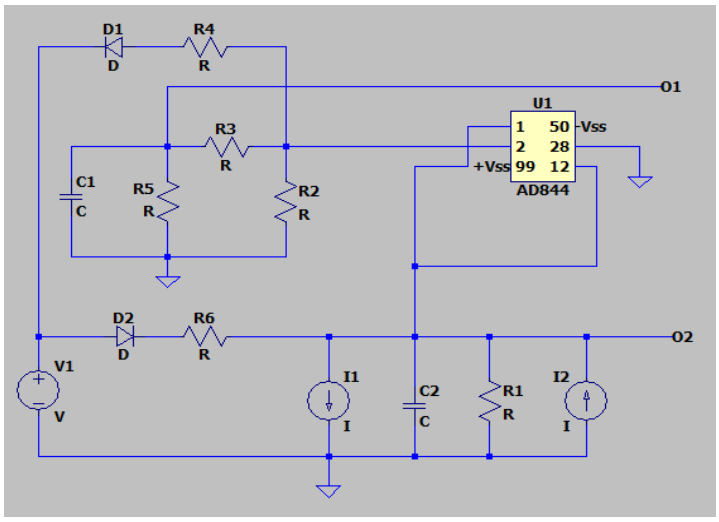


Fig 33: Membrane circuit using AD844

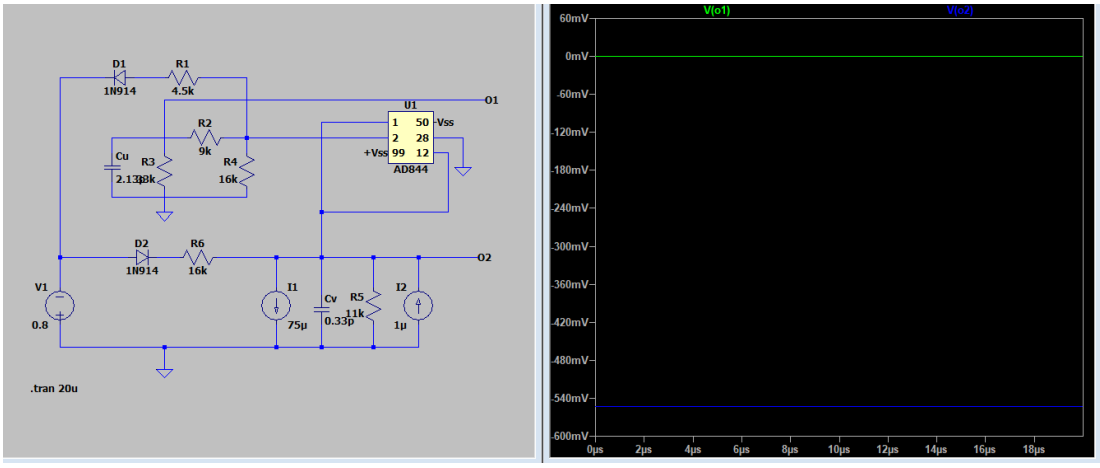


Fig 34: Output of the membrane circuit where the CCII is designed using AD844

6.2.2 AUXILIARY AFTER SPIKE RESTING CIRCUIT

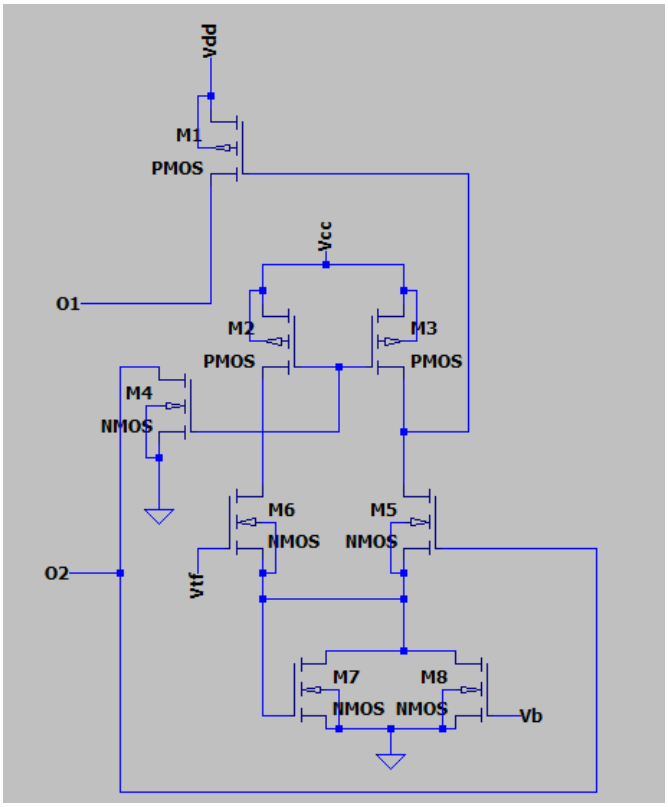


Fig 35: Auxiliary spike resting circuit using open loop comparator

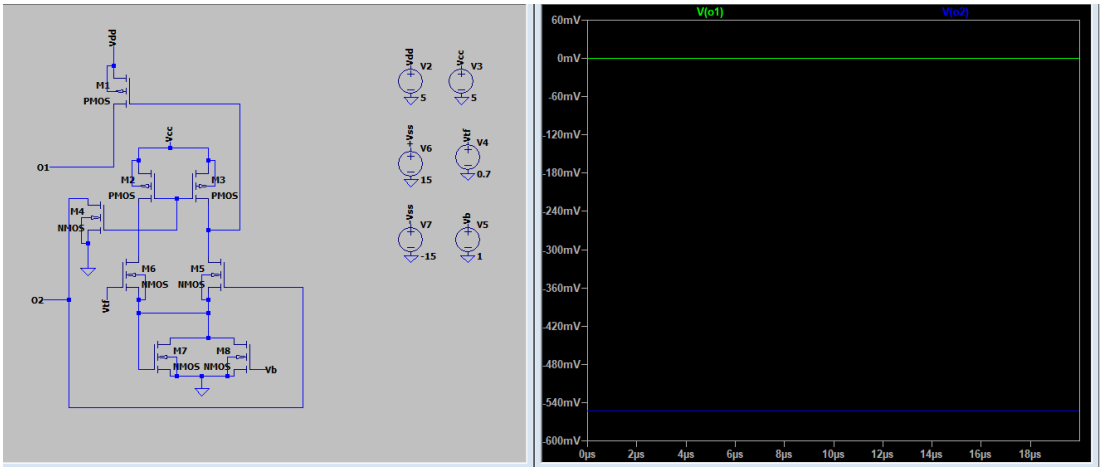


Fig 36: Simulation of the auxiliary spike resting circuit in LTSpice which signifies an open loop comparator.

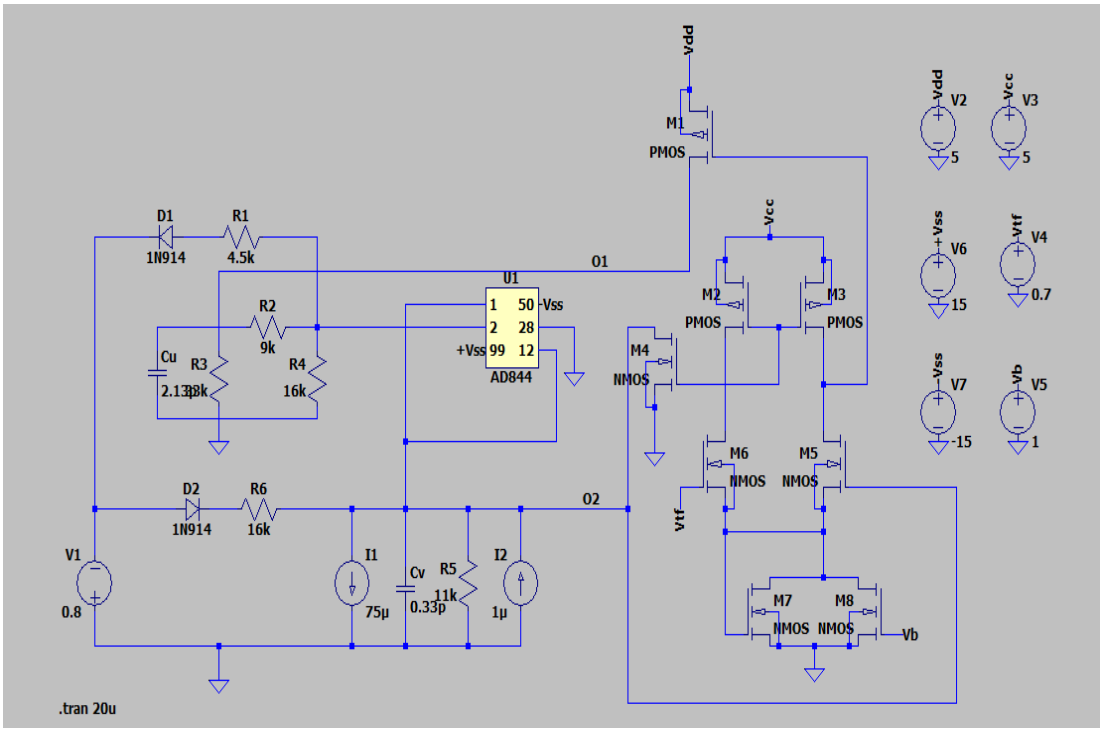


Fig 37: 2nd order piecewise linear neuron model using AD844

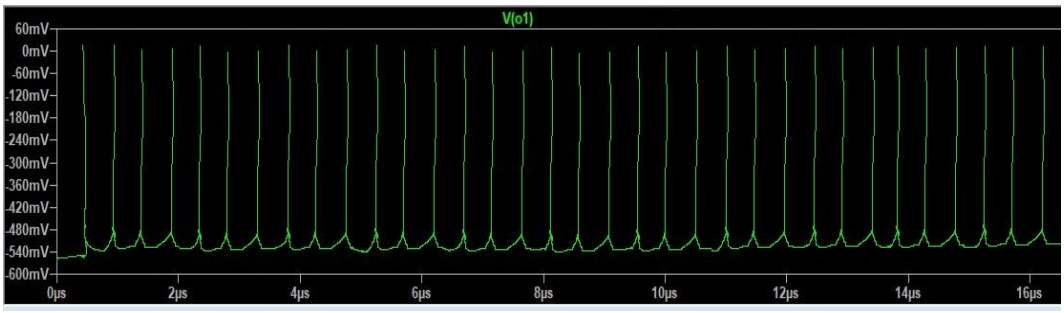


Fig 38: (i) Variation of membrane potential of the 2nd order piecewise linear neuron model

In the above simulation the resistance and capacitances are taken as R1= 4.5k, R2 = 9k, R3 = 33k, R4 = 16k, R5 = 11k, R6 = 16k, Cv=0.33pf, Cu = 2.13pf results in seated spiking kind of waveform.

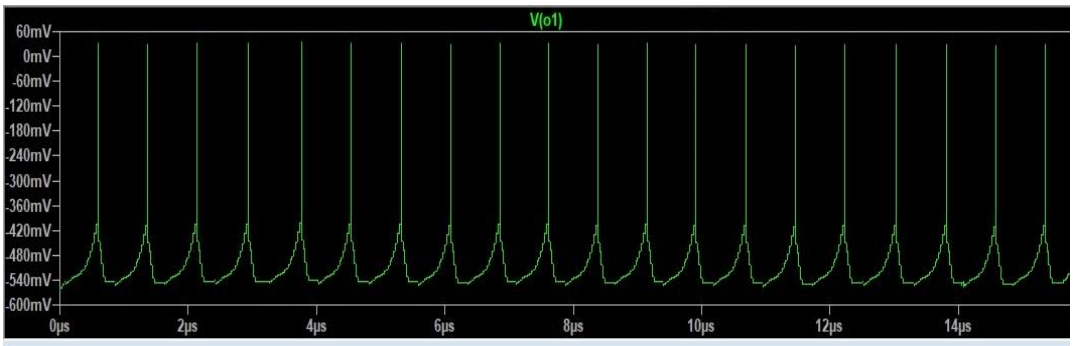


Fig 39: (ii) Variation of membrane potential of the 2nd order piecewise linear neuron model

The output waveform of the membrane potential variation results in consistent tonic spiking with the variation in the resistance and capacitance values.

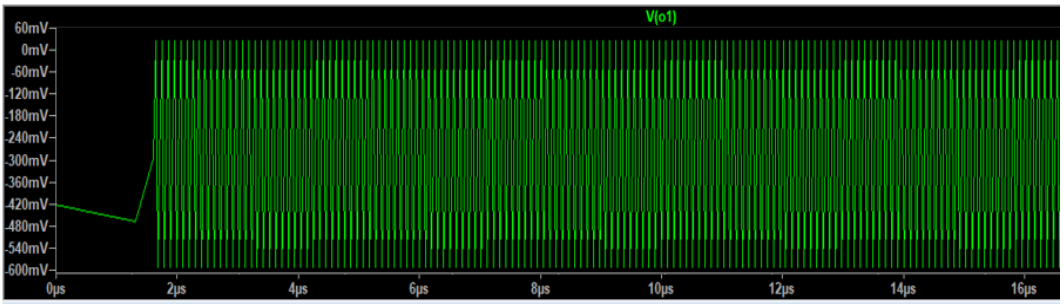


Fig 40: (iii)Variation of membrane potential of the 2nd order piecewise linear neuron model

On further increase in the resistance and capacitor values, more sharper spikes are generating in the waveform with higher frequency.

Neural oscillations and spike trains of diverse types of functions in a neuronal circuit realized in the software. A number of different output waveform shapes of the model neuron circuit are illustrated with the dependency on spikiness. This way, various types of spiking behaviors may be realized without changing size of transistor or connection diagram. With reference to the data derived in the study, it can be deduced that this electrical circuit mimics the behavior of the neural dynamics as postulated by in Izhikevich (2003).

CONCLUSION

Spiking Neurons have been implemented using AD844 and AD644 in LTSPICE. Our model is the most basic model that can replicate various kinds of neural activity: It has only one nonlinear term on the right-side equation which is v^2 and it contains two overall equations. However, the presented model is canonical in a sense that it is equivalent to a large class of biophysically detailed and accurate Hodgkin-Huxley-type models being different from them by the change of coordinates only, this change can be accompanied by incorporating a massive number of equations and all available data on ionic currents. From biological neurons, the phenomena like spiking, burring and mixed mode firing, post inhibitory spikes and bursts, continuous spiking with frequency adaptation, threshold variation, resting and spiking bistable region, subthreshold oscillations and resonance are all created by the model presented. Using the model described here, we show how to build spiking neuronal networks that generate coordinated activity and oscillations comparable to those observed in mammalian brains. It is demonstrated that this second order derivative equation is indeed effective by constructing the neuron models in LTSpice and successfully generating tens of thousands of spikes. In terms of their temporal and spatial organization, the spikes are nearly as complex as the real action potentials that occur in neurons in the cerebral cortex.

The behaviour of biological neurons has been effectively simulated with the help of this careful study and simulations of Spiking Neurons with the help of LTSpice, AD844, and AD633. Synthesis of piecewise linear neuron models with the help of CCII building blocks is presented here only in a generalized form applicable to the analog implementation. Outcomes signify that such circuits are competent of mimicking each style of neuronal activity. It is important to note that this model is highly immune to noise and impedance matching in the input and output, which could be very suitable for large-scale neural networks. This is an important advancement towards building programmable analog neural integrated circuits for actual use. Since high-performance analog computation emanates from the Second-

Generation Current Conveyor (CCII) implementations, these approaches are well suited to the complex biological information processing devices. Second generation current conveyor is implemented with the help of AD844. The generated spikes in this case look realistically similar to the natural action potentials. Therefore, the second-order piecewise-linear model of a neuron's electrical circuit emulates the excitability of the nervous system as described by Izhikevich (2003)[6].

Future development of distinct membrane recovery variables and concept of Integrate-and-Fire neuron will contribute to the realistic and useful models in the future. In future the auxiliary spike resting circuit can be designed using some active element in order to implement the second order piecewise linear neuron model.

REFERENCES

1. S. Kulkarni, S. P. Simon, and K. Sundareswaran, “A spiking neural network (SNN) forecast engine for short-term electrical load forecasting,” Applied Soft Computing, vol. 13, no. 8, pp. 3628–3635, 2013.
2. A. A. Abusnaina and R. Abdullah, “Spiking Neuron Models: A Review,” Int. J. Digit. Content Technol. its Appl., vol. 8, no. 3, pp. 14-21, 2014.
3. H. Abdelbaki, E. Gelenbe and S. E. EL-Khamy, "Analog hardware implementation of the random neural network model," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, 2000, pp. 197-201 vol.4, doi: 10.1109/IJCNN.2000.860772.
4. J. M. Zurada, "Analog implementation of neural networks," in IEEE Circuits and Devices Magazine, vol. 8, no. 5, pp. 36-41, Sept. 1992, doi: 10.1109/101.158511.
5. P. D. Deotale and L. Dole, "Design of FPGA based general purpose neural network," International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, India, 2014, pp. 1-5, doi: 10.1109/ICICES.2014.7033843.
6. E. M. Izhikevich, "Simple model of spiking neurons," in IEEE Transactions on Neural Networks, vol. 14, no. 6, pp. 1569-1572, Nov. 2003, doi: 10.1109/TNN.2003.820440.
7. Sedra, A., & Smith, K. (1970). A second-generation current conveyor and its applications. IEEE Transactions on Circuits System, 17, 132–134.
8. H Soleimani, A Ahmadi, O Bavandpour, O Sharifipoor. A Generalized Analog Implementation of Piecewise Linear Neuron Models Using CCII Building Blocks. *Neural Netw*. 2014; **51**: 26-38.
9. Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of Physiology, 117(3), 500.
10. G. Indiveri, "A low-power adaptive integrate-and-fire neuron circuit," Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03., Bangkok, Thailand, 2003, pp. IV-IV, doi: 10.1109/ISCAS.2003.1206342.
11. Izhikevich, E. M. (2001). Resonate-and-fire neurons. Neural Networks, 14(5),883–894.

12. A Hazan and E Ezra Tsur (2021) Neuromorphic Analog Implementation of Neural Engineering Framework-Inspired Spiking Neuron for High-Dimensional Representation. Front. Neurosci. 15:627221. doi: 10.3389/fnins.2021.627221
13. https://analyticsindiamag.com/a-tutorial-on-spiking-neural-networks-for-beginners/
14. https://www.semanticscholar.org/paper/A-Neuron-Circuit-Model-with-Smooth-Nonlinear-Output-Yamashita-Nakamura/b618f291d493fc88668f9efcbbce2adc1eea6f9a
15. https://www.researchgate.net/figure/Analog-electronic-circuit-model-of-a-LIF-neuron_fig2_331836456
16. https://www.analog.com/media/en/technical-documentation/data-sheets/AD844.pdf
17. https://www.analog.com/media/en/technical-documentation/data-sheets/AD633.pdf