M. Tech (VLSI Design and Embedded Systems) NISHANT SHARMA 2024

MEMRISTOR – ANN FOR ENERGY EFFICIENT PATTERN RECOGNITION

A DISSERTATION

SUBMITTED IN THE PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

Master of Technology

In

VLSI Design and Embedded Systems

Submitted By:

Nishant Sharma (2K22/VLS/09)

Under the Supervision of:

Mr. Vinay Kumar (Assistant Professor)
&
Ms. Lavi Tanwar (Assistant Professor)

Department of Electronics & Communication Engineering

Delhi Technological University, Delhi

(formerly Delhi College of Engineering)

Bawana Road, Delhi- 110042

May, 2024

Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Nishant Sharma, Roll No. 2K22/VLS/09 student of M. Tech. (VLSI Design and Embedded Systems), hereby declare that the project Dissertation titled "Memristor – ANN For Energy Efficient Pattern Recognition" which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi NISHANT SHARMA

Date: 2K22/VLS/09

Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Memristor – ANN For Energy Efficient Pattern Recognition" which is submitted by Nishant Sharma, Roll No. 2K22/VLS/09 student of M. Tech. (VLSI Design and Embedded Systems) Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:

MR. VINAY KUMAR (SUPERVISOR)
ASSISTANT PROFESSOR
Department of ECE
DTU, Delhi

MS. LAVI TANWAR (SUPERVISOR)
ASSISTANT PROFESSOR
Department of ECE
DTU, Delhi

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude and indebtedness to my high respected and esteemed guides Mr. Vinay Kumar (Assistant Professor, ECE) Delhi Technological University, Delhi & Ms. Lavi Tanwar (Assistant Professor, ECE) Delhi Technological University, Delhi for their constant support and guidance throughout the course of our work. Their sincerity, thoroughness and perseverance have been a constant source of inspiration for me. I would also like to thank all other faculty members of Electronics and Communication Engineering Department for their valuable suggestions and co-operation at every step in this work. I would also like to thank my parents and batchmates for their constant motivation throughout this work. Last but not the least we would like to thank Almighty for His invisible presence and constant support.

Place: Delhi

Date:

NISHANT SHARMA

2K22/VLS/09

ABSTRACT

This work explores the application of Spiking Neural Network (SNN), Memristor (MR) crossbar arrays for efficient digit recognition (0-9). Leak Integrate and Fire (LIF) Neuron is also implemented to generate spikes which can be fed into memristive crossbar arrays 4X4, final decision is made based on the highest current obtained from a particular column designated for the digit. For circuit level implementation TSMC gpdk180nm library file is used and LT Spice is used to carry out the simulation work. The MNIST Dataset is used and accuracy of 88% is obtained when simulated the same in Pycharm using Keras from Tensor flow library which is used to map activation functions of the Artificial Neural Network (ANN).

Table of Contents

List of Figures

List of Tables

List of Symbols, Abbreviations and Nomenclature

AI: Artificial Intelligence

ML: Machine Learning

NN: Neural Network

ANN: Artificial Neural Network

SNN: Spiking Neural Network

MR: Memristor

Chapter 1: Background and Literature Review

Memristors eliminate the bottleneck caused by data movement in von Neumann architectures by storing data and performing computations in the same location. This is known as in-memory computing. **Parallel Processing:** A task like digit recognition could be accelerated by using the crossbar architecture, which enables parallel computations across multiple memristors. **Lower Power Consumption:** Compared to von Neumann architectures, which process data continuously, event-driven SNNs and in-memory computations can result in significant power savings. It's likely that the von Neumann architecture will always be the foundation of general-purpose computing. Nonetheless, studies such as this one on memristor-based SNNs show promise for particular applications requiring effective, low-power information processing. Memristor crossbar arrays with SNNs will have a great role in shaping the future of computing, collaborating with conventional architectures to address an ever-expanding array of computational challenges, as material science and SNN training algorithms progress.

In this chapter, we are going to study the current research that is going on in the Neuromorphic World for the same we have referred to various research papers mentioned in the reference section and the roadmap published in 2022 in Neuromorphic Engineering [2].

1.1 Background

The majority of computers we use today are working on Von Neumann Architecture as shown in Fig. 1.1, which is like powerful machines that can easily handle tonnes of data. Nevertheless, we have seen we need faster access time, and the current era demands faster processing as we know in Von Neumann's design the memory and processing units are separate which while the data processing happens it takes a hell of an amount of time just going TO and FRO from memory to processor which in returns drain a lot of power. As

the energy issues are rising and, the amount of data increasing at an exponential rate, we need to delve deeper into the neuromorphic World as brain-inspired computing promises high speed and can handle tonnes of data efficiently due to its highly parallelized structure. To explore more, you can refer to the roadmap [2].

The current research in few years is giving more attention to how we can process fast without using much power and one promising solution is being offered by the Neuromorphic community, the research is rooted in the theoretical demonstration of the fundamental 4th element which is the Memristor as described by L. Chua [1].

For more than three decades the memristor remain the theoretical concept unless it was developed in 2008 by the HP Labs in their famous paper missing memristor found [3]. Once the discovery happened numerous research papers started publishing where the mathematical models were developed and simulations were carried out on SPICE. One of the earliest models was shared by Joglekar and Wolf [4].

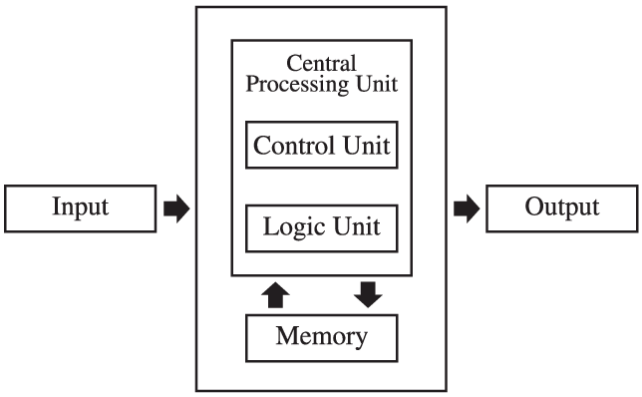

Figure 1.1: Von Neumann Architecture

Neuromorphic Architecture gives significant advantages when we compare it with the traditional Von Neumann architecture as described in Fig. 1.2. The Artificial Intelligence (AI) industry is booming these days. Machine Learning (ML), Deep Learning, and Neural Networks (NN) are hot topics of discussion in the research community. Various ML models have been developed to test the potential of what a NN can do. Artificial Neural Network (ANN) and Spiking Neural Networks (SNN) being the subset of NN offer great potential to Mimic the biology of the Human Brain. Numerous Applications exist where SNN is found useful some of the applications of NN are as follows:

1. **Neuromorphic Computing** is utilized in **Robotics**, Sensory data like images for **Image Processing**, **Cognitive computing,** and much more.
2. **Edge Computing** where the hardware is implemented on edge devices, reduces dependency on central processing which gives greater security and privacy to the end user.
3. **Optimization Problems** which include huge datasets.

To summarize SNN is an evolving field that promises us the potential to revolutionize the way we compute by offering us Low power with High-performance solutions for numerous applications.

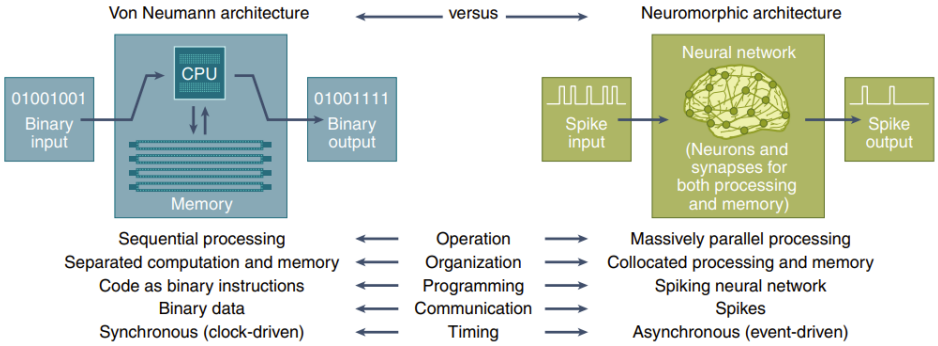

Figure 1.2: Comparison between Von Neumann and Neuromorphic architectures [7]

1.2 Fourth Fundamental Element: Memristor

Apart from the 3 common Fundamental elements Resistor(R), Capacitor(C), and Inductor(L), the fourth one(MR) was introduced by L. Chua in 1971 and is today most commonly known as **Memristor** (Memory + Resistance) [1].

In this project, we are utilizing a Memristor (MR) as a Memory element that can store 2 states basically 1 or 0 by its resistive property Ron of the order of Kilo Ohms and Roff as high as 1 Mega Ohms. The MR is being utilized as a **CROSSBAR ARRAY** [6]. The crossbar offers a highly parallelized structure that offers matrix multiplication.

In Neural Networks the key component of computation is the Matrix multiplication which is done by implementing a Crossbar array. Digit Classification is the subset of Pattern

Recognition. So, suppose we have an image of dimensions (5 X 3) Pixels i.e. 5 rows and 3 Columns. Then we require a Crossbar array of size (15 X 10) where 15 denotes the number of pixels and 10 is the 10 different digits. i.e. to test which image is fed into the system of the crossbar array we will need 150 memristors. The Crossbar implementation and simulation are explained in detail by working on a (4 X 4) Crossbar array [5,6]. We will be discussing in detail about the Memristor in the upcoming chapters.

1.3 Research Motivation

For decades, the von Neumann architecture is being used for computing, but due to its memory bottleneck as a huge limitation, we need to move to different architecture which is promised by **memristor crossbar arrays** for specific tasks such as **digit recognition**. Factors that make this project still relevant is provided below:

1. **Bottleneck of Von Neumann:** Transferring the data back and forth slows down processing, which makes it power hungry and less efficient for the data intensive applications.
2. **Power Usage:** The movement of data takes place all of the time. So, devices which powers on battery or the applications which needs real-time processing, this becomes an issue.
3. **Restricted Scalability**: It's not easy to scale it as heat issues and leakage increases as no of components involves increases.

1.4 What's inside?

We will be performing digits (0-9) classification by implementing a crossbar array on LT SPICE by utilizing memristor as the key element where we have utilized the model given by Joglekar [4].

Chapter 2: Neural Networks

Whenever this term Neural Network coins, we wonder what do we mean by it. In Laymen terms it's just a program which is inspired by brain. It processes information with the help of nodes all connected to each other in various layers. It learns from the data and helps is various applications like recognizing patterns in images or making predictions based on a huge data set.

Let's define the simplest Neural Network. The simplest Neural Network is known as a **perceptron.** A perceptron consists of a single layer of artificial neurons that mimic how the brain cells are connected to each other. For an e.g. supposed you have been given a task to classify a set of images into two different groups, group 1 contains images of cats and group 2 contains the images which doesn't have a cat such that perceptron is basically making a simple decision whether the image contains cat or not.

As the perceptron is making two different classes it is also referred as Linear binary Classifier. A basic Perceptron is shown in Fig. 2.1

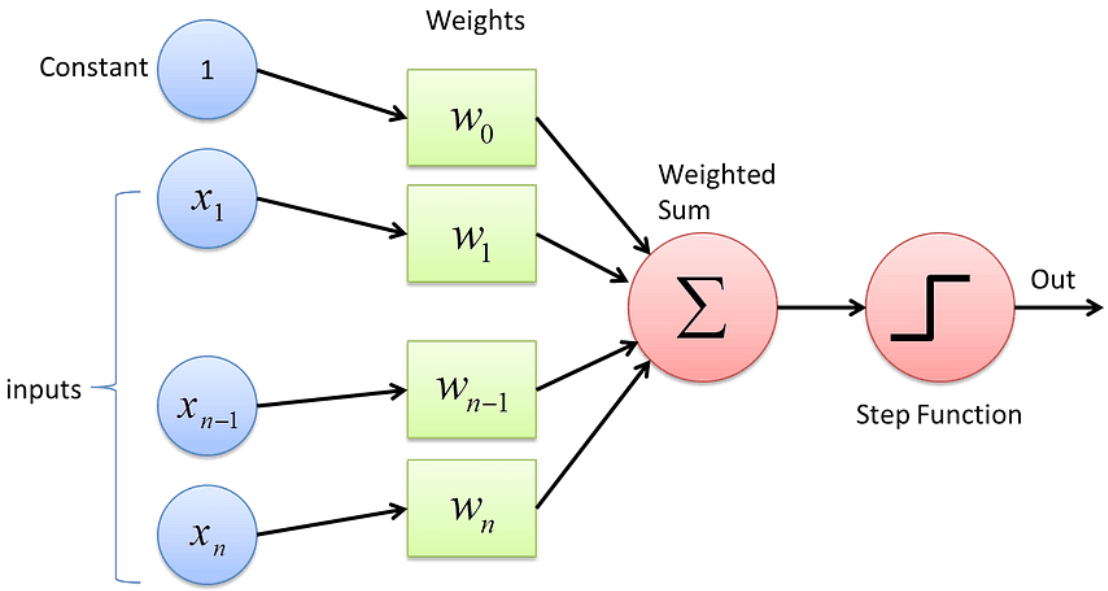

Figure 2.1: Perceptron: Simplest Neural Network

2.1 Various types of Neural Networks

There are various types of NN which includes perceptron, MLP(Multi-layer perceptron), CNN (Convolution Neural Network),RNN (Recurrent Neural Network),SNN(Spiking Neural Network), and ANN (Artificial Neural Network) to name a few as they are among the most popular ones. Describing them all here may distract from the main theme so the reader is advised to learn them from various resources available online. We will be focusing on ANN and SNN only in this text.

2.1.1 Common terminologies associated with NN

1. Layer: A Layer is fundamental building block which serves as the processing unit. The whole network consists of numerous layers and for the network to learn complex situations we have to increase the number of layers but this also gives us the tradeoff which we need to bear in terms of training that particular layer.

2. Node: Node is a structure inside layer which consists of input, activation function and output. They perform the essential computation in a neural network.

3. Weight and Baises: This is just the Numerical value given to an activation function for efficient processing of the data. Wrongly decided weights may leads to poor accuracy.

4. Activation function: There are a variety of activation functions exists to name a few **Sigmoid, tanH, ReLU, Leakey ReLU, and Softmax**, the work of an activation function is to determine the neuron's output. Analogous to guard gatekeeping at the society gate this also tells the neuron whether to get active and fire or stay inactive.

Few Activations function is described with their mathematical expressions in Table2.1 and their plot is also shown in Fig. 2.2.

Table 2.1: Activation function mathematical expressions

S.No.	Function	Equation
1.	Sigmoid	$y = \frac{1}{1+e^{-x}}$
2.	Tan Hyperbolic	$y = \frac{2}{1+e^{-2x}} - 1$
3.	ReLU	$y = \max(0, x)$

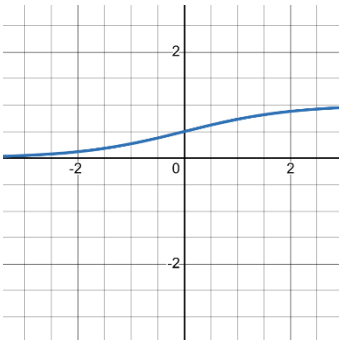

Figure 2.2: Sigmoid Activation function

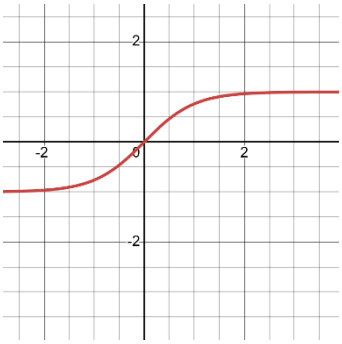

Figure 2.3: Activation function tan Hyperbolic

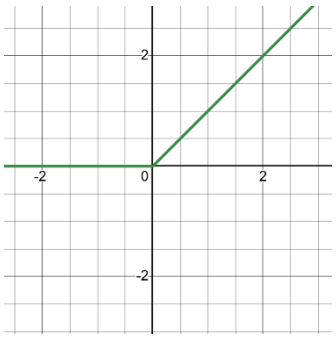

Figure 2.4: ReLU Activation Function

2.2 Artificial Neural Network and MNIST Dataset

Imagine you're teaching someone to recognize handwritten numbers. You show them examples and point out key features. Artificial Neural Networks (ANNs) do something similar, but they learn from a massive dataset on their own.

2.2.1 The MNIST Dataset:

ANNs use a special collection of data called MNIST. It has 70,000 tiny pictures (28x28 pixels) of handwritten digits (0-9) and few sample is shown in Fig. 2.5. Each picture is labelled with the correct number, like flashcards for the ANN to learn from.

Figure 2.5: Sample Images from MNIST Dataset [9]

1. **Getting the Data Ready:** Imagine each picture as a grid of 784 tiny squares. The darkness of each square is turned into a number (0 for black, 1 for white). This creates a long list of numbers for each picture. We do this for all the pictures in the dataset.

2. **Building the ANN: The Learning Machine:** Brain Cells in a Box (Artificial Neurons): These are the building blocks, inspired by how brain cells work. They're like tiny decision-makers.

3. **Layers of Learning:** The ANN has different layers: an input layer, hidden layers, and an output layer. The input layer gets the long list of numbers from the picture. Hidden layers are where the real learning happens. The output layer gives the final answer (what digit it thinks it sees).

4. **How the ANN Learns:** Feeding the Machine: We give the processed pictures (long lists of numbers) to the input layer. Weighted Sums and Firing Up: Each hidden layer neuron gets info from the previous layer, with connections having

weights (importance). These weighted numbers are added up, and a special function decides if the neuron "fires" and sends a signal forward.

5. **Fine-Tuning (Adjusting Weights):** At first, the weights are random. By comparing the ANN's guess (output layer) with the actual digit, we adjust the weights. If it gets it wrong, the weights are tweaked to improve future guesses. It's like correcting a student and helping them learn the right features of each digit.

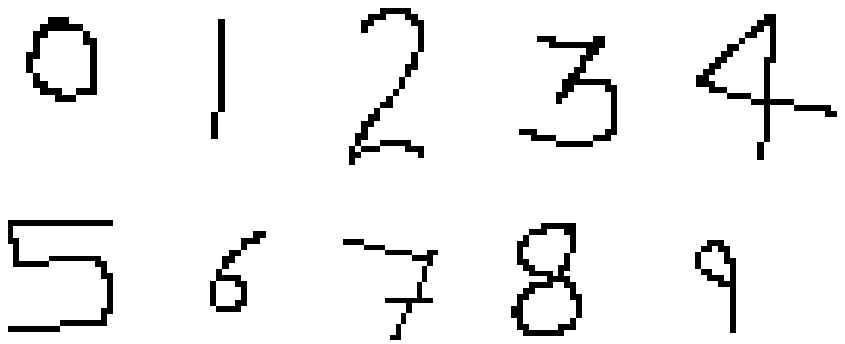

Figure 2.6: Test Data 28x28 size created using Paint tool on Windows.

6. **Practice Makes Perfect (Training and Testing):** We use some of the MNIST data (training set) to train the ANN by repeatedly showing it pictures and adjusting weights. The rest of the data (testing set) is used to see how well the trained ANN does on new examples. We have also used our test set which is prepared used paint in Windows as shown in Fig. 2.6. The images are enlarged and will appear blurry because of low resolution of 784 pixels per image.

7. **Recognition Time:** Once trained, the ANN can see a new handwritten digit picture, process it through the layers, and predict the digit based on what it's learned. It's like the student applying their knowledge to recognize a new digit they haven't seen before.

8. **Why ANNs are Great:** Learning the Tough Stuff: ANNs can learn complex patterns in the data, allowing them to tell similar-looking digits apart.

9. Ready for More (Scalability): They can handle a lot of data, making them useful for real-world tasks. Always Getting Better (Continuous Improvement), As more data becomes available, ANNs can be further refined for even better accuracy.

10. **Brain Power Needed (Computational Cost):** Training large ANNs can require a lot of computing power. Data is Key (Data Dependence): How well they perform depends heavily on the quality and quantity of data used for training.

ANNs, along with datasets like MNIST, are powerful tools for digit recognition. By mimicking the brain's learning process, they can achieve impressive accuracy and pave the way for even more advanced image recognition tasks.

Chapter 3: Circuit Design

We have used TSMC GPDK 180nm library file for implementing our design on LT SPICE for the part where CMOS are necessary. We have used memristor inside the crossbar arrays. We have trained the ANN using Py Charm Software by coding in Python language and tested with MNIST dataset along with the 10 sample test images (0-9) as shown earlier in chapter 2 Fig. 2.6.

Basic Circuit design of Memristor has been studied on LT SPICE it's symbol Fig. 3.1 and circuit is shown in Fig.3.2. The 4X4 Crossbar array is also shown in Fig. 3.4.

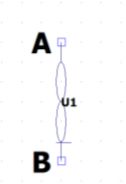

Figure 3.1: Memristor Symbol

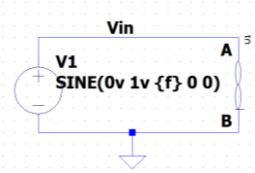

Figure 3.2: Memristor Circuit to study I-V relationship

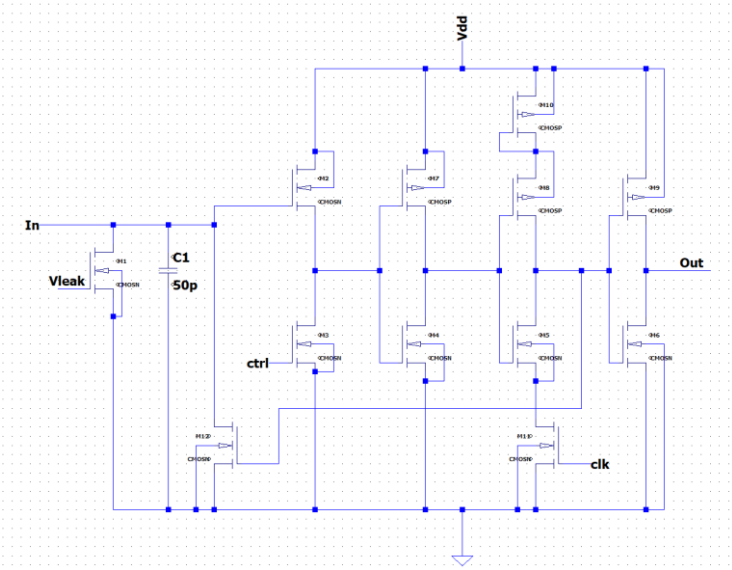

Figure 3.3: LIF Neuron Circuit

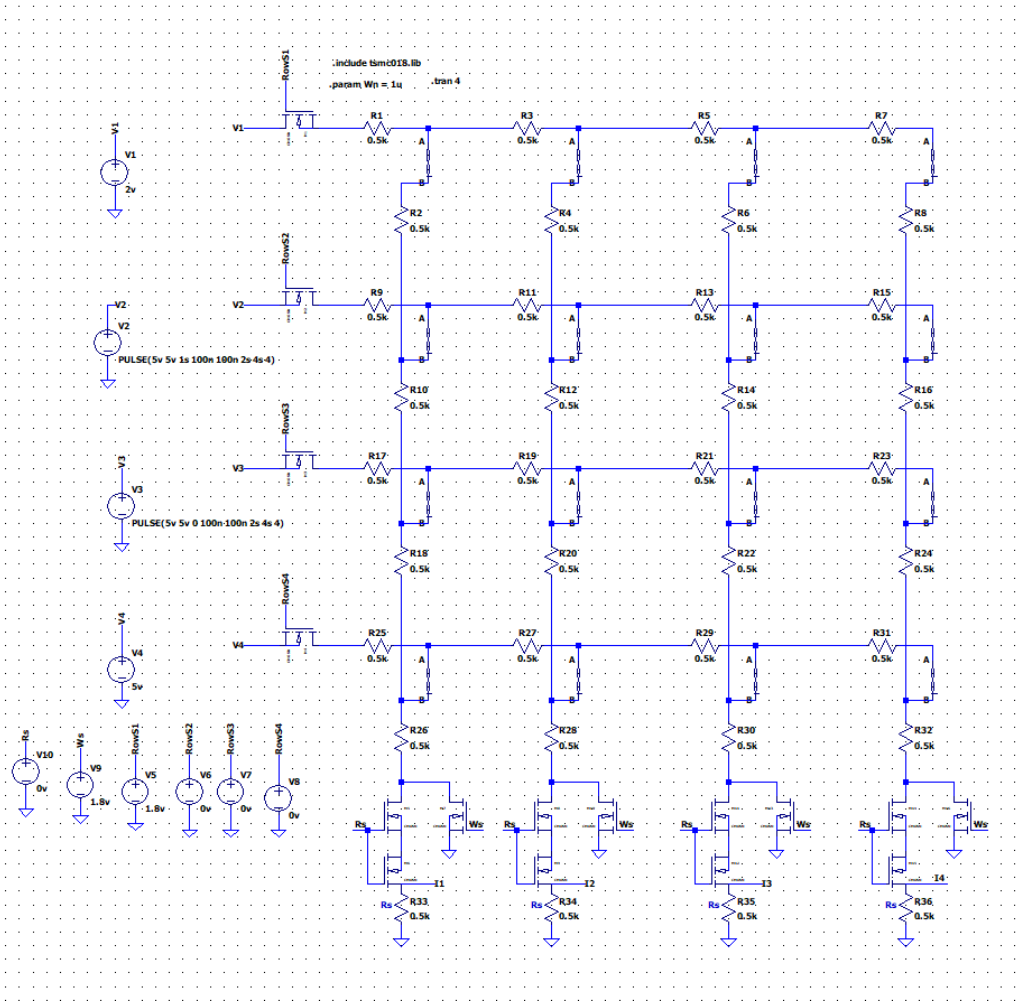

Figure 3.4: 4X4 Crossbar array implemented on LT SPICE

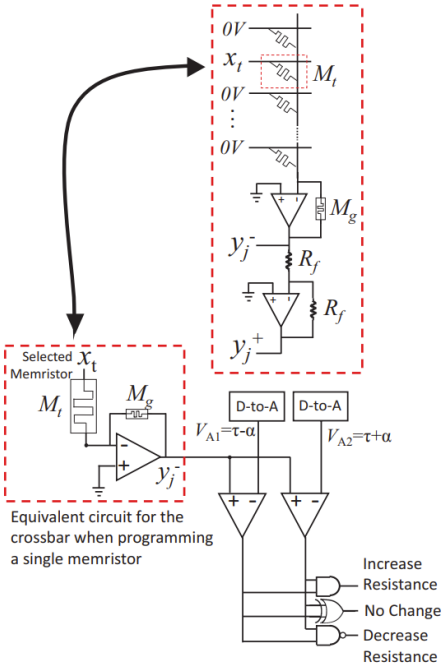

Figure 3.5: Circuit to Programme the Crossbar Column[5]

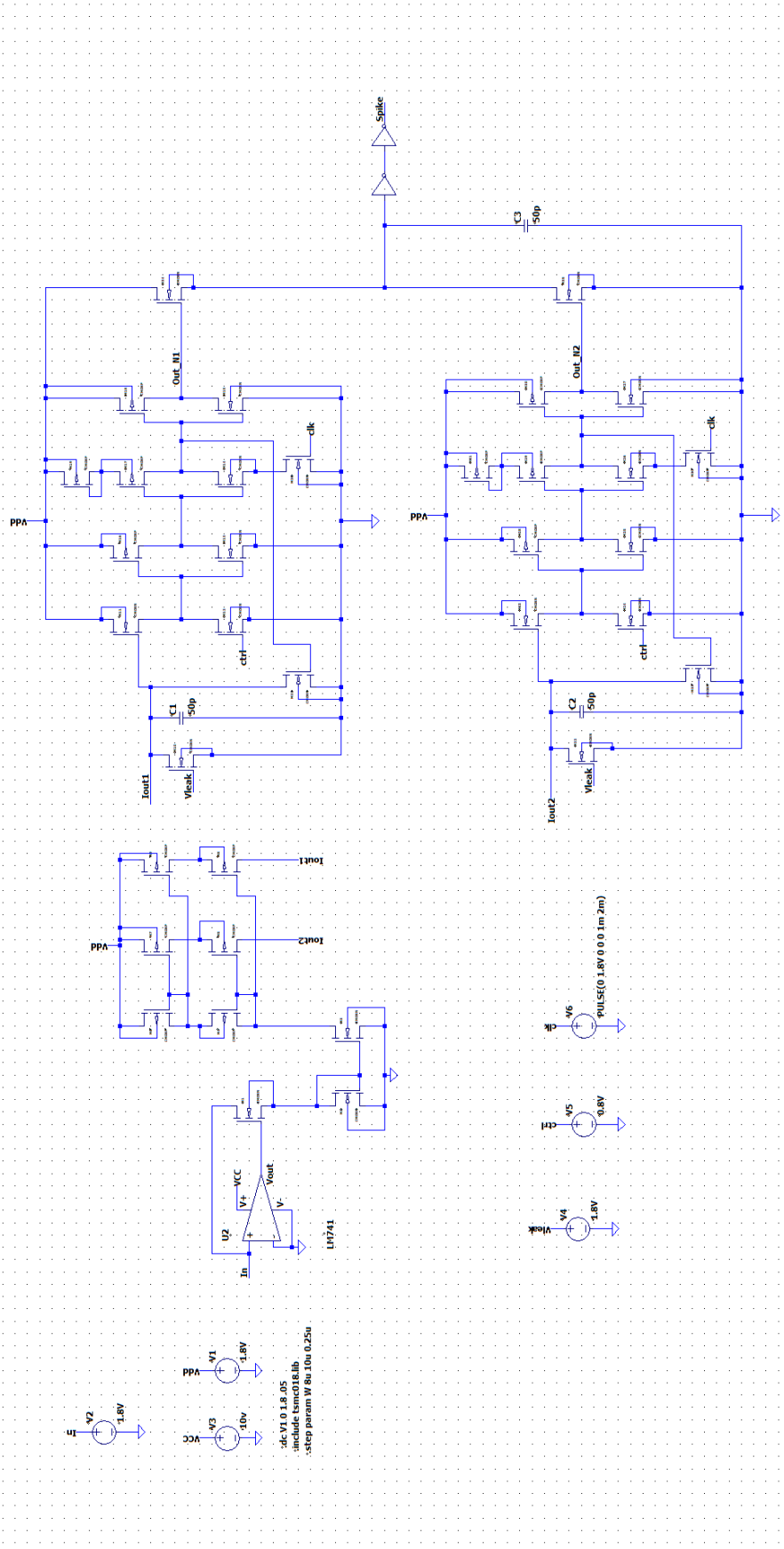

Figure 3.6: Input Processing using LM741 OP-Amp, Current Mirror, 2 LIF Neurons and Spikes Extractor

Figure 3.6: Input Processing using LM741 OP-Amp, Current Mirror, 2 LIF Neurons and Spikes Extractor

Chapter 4: Overview of Memristors

Let's delve deeper into world of memristors. Memristor made up of two words Memory and Resistor hence named as memristor. If we know it's structure it will look like a sandwich where top and bottom bread are metals and in between the breads butter acts as an insulator in this sandwich analogy. Any device is measured by its performance metrics which include On Resistance (R_{on}), Off Resistance (R_{off}), Set/Reset voltage which you can obtain from I-V characteristics as shown in Fig. 4.3, various other parameters are also importance such as power consumption, switching speed to name a few.

As we all know until now that the memristor is theorized by L. Chua [1] and it is a 2-terminal device, we measure its resistance by observing the amount of charge passing through it.

Fig. 4.1 depicts the relationship among the four fundamental elements namely Inductor(L), Capacitor(C), Memristor (MR) and Resistor(R). The relationship that all of them holds with each other can be described using basic equations which involves Charge(q), Voltage(v), Current(i) and flux (Ψ) given as follows:

$$dv = Rdi,\ dq = Cdv,\ d\Psi = Ldi,\ d\Psi = Mdq,\ d\Psi = vdt,\ dq = idt$$

It can be observed from the above formulas that the Memristor has the same dimensions as of resistance but they both re different by the fact the Resistance is not dependent on external circuit variables whereas MR changes with it, as it can be identified with the material itself.

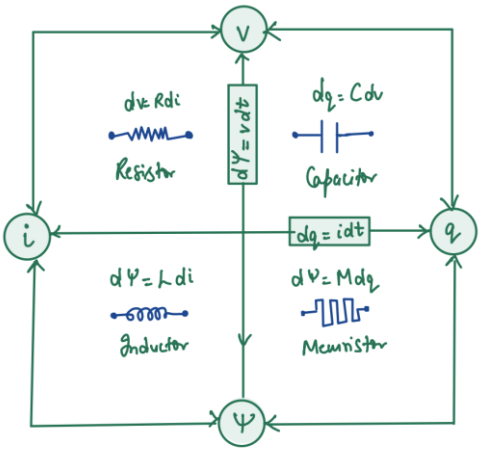

Figure 4.1: Relationship between R, L, C and Memristor

Neural Networks are making the full use of MR devices as they can be used as binary classifier which helps in image recognition tasks such as digit recognition for instance as they offer Low power consumption, scalability which ensures high integration density which is quite useful for recognition in applications based on AI.

In this work, memristor is used as memory at circuit level, its performance is analysed on software by using PyCharm tool and building an ANN and We utilized MNIST dataset along with the self-created test data by coding in python and using Tensor Flow for Activation function implementations [10]. On a smaller scale we have studied the Crossbar made of Memristor which is ideal for performing Matrix Multiplicaions. Each column performs the dot product as described in equation 4.1,

$$V.G = I \tag{4.1}$$

The dot product essentially happens between the Voltage(V) applied to the Memristor with its Conductance (G) and there is current start flowing through it.

The HP labs have discovered the memristor way back in 2008 [3]. It structures and a closer look at microscopic view is shown in Fig. 4.3. It's made of two Platinum (Pt) Metal plates and Titanium Dioxide is sandwiched between them as the insulator layer.

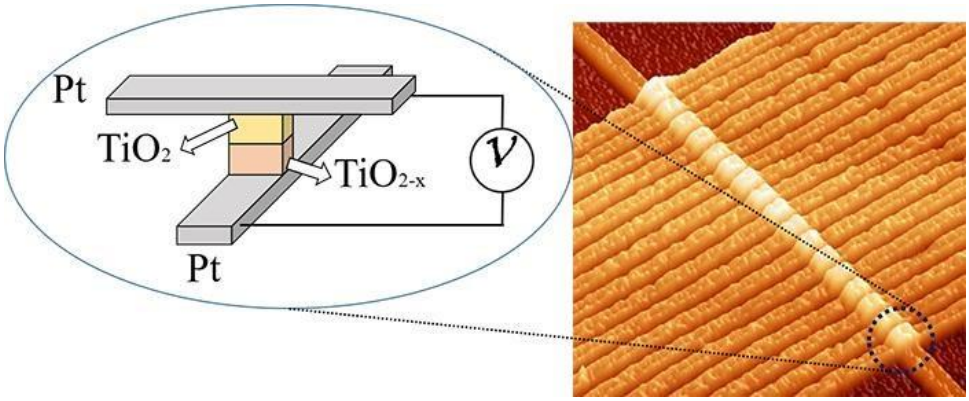

Figure 4.2: The Memristor found by HP Labs [3]

4.1 Features of Memristor:

- It is a 2-terminal Non-Linear (NL) resistor
- Being a passive element and it doesn't store energy like C and L.
- It effectively remembers what was the last amount of current/Voltage has flowed by it.
- It is also known as resistor with a memory associated with it.
- Pinched Hysteresis Loop, L. Chua Said if it's pinched it's a memristor [1].
- It Exhibits NL(Non-Linear) relationship between voltage(V) and current(I).
- No phase shift introduced between current and voltage at zero crossing
- Hysteresis reveals the Non-volatile memory effect

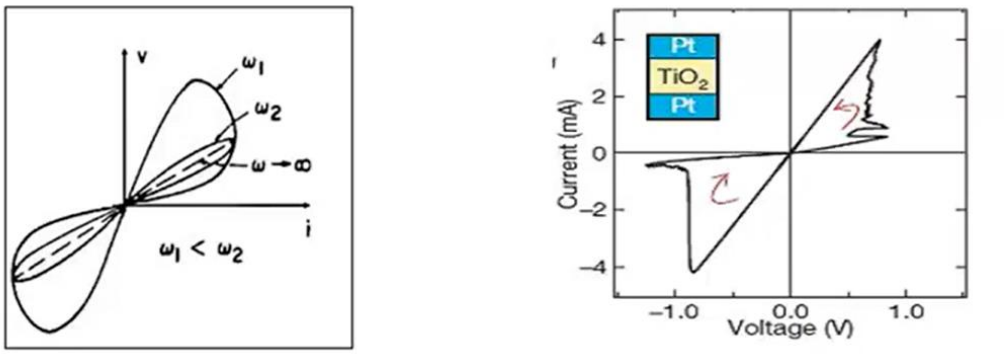

Figure 4.3: I-V Characteristics of Memristor[1]

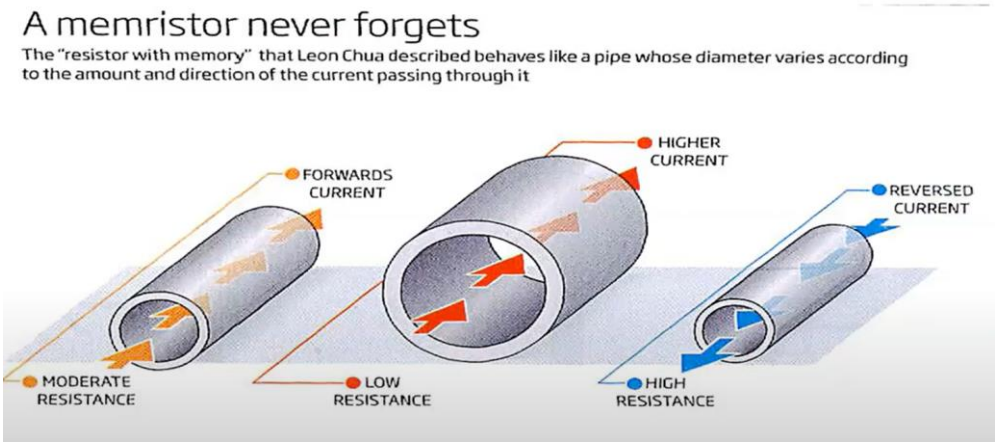

Figure 4.4: Pipeline Analogy of Memristor

A memristor is a device which can be trusted as a person who never forget. We can understand the Memristor by looking at the Pipeline Analogy as shown in Fig.4.5.

4.2 Device Performance:

After going through the Roadmap, we have found that the MR device performance can be determined by its On Resistance (R_{on}), Off Resistance (R_{off}), Set/Reset, switching speed and power consumption to list a few [2]. We have measured the metrics of how memristor is behaving for various Ron values and implemented it using 4x4 Crossbar as shown in Fig.4.6.

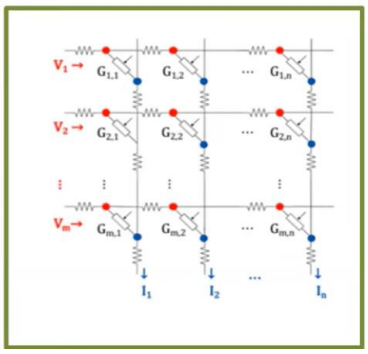

Figure 4.5: Memristor Crossbar array[12]

4.3 Future Direction:

Neuromorphic computing is an amusing field with various levels of abstraction where we can research on as described in the roadmap, if you want to explore them further, they are nicely depicted in Fig 4.7. In our study we have focused on the circuit level implementation of the NN by simulating an LIF Neuron Circuit. The areas are vast we can work more on materials, Architecture, device level, Applications or Algorithms to list a few [2].

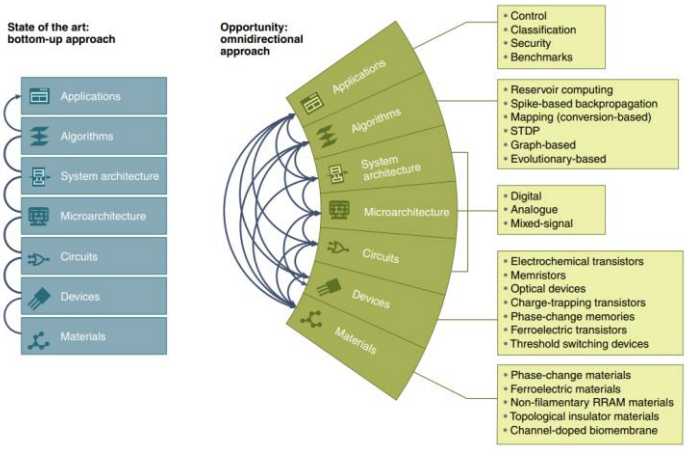

Figure 4.6: Scope and Future Direction [13]

Chapter 5: Results and Analyses

In this Chapter we will summarizing on various results obtained by the simulation of our circuits and will conclude by analyses of the same.

5.1 Memristor I-V Characteristics

All the designs are implemented using LT SPICE. Let's share some results when memristor subcircuit module is operated at various frequencies with two Different Ron Values of 100 Ω and Ron of 1KΩ as shown below in Fig. 5.1 and Fig. 5.2.

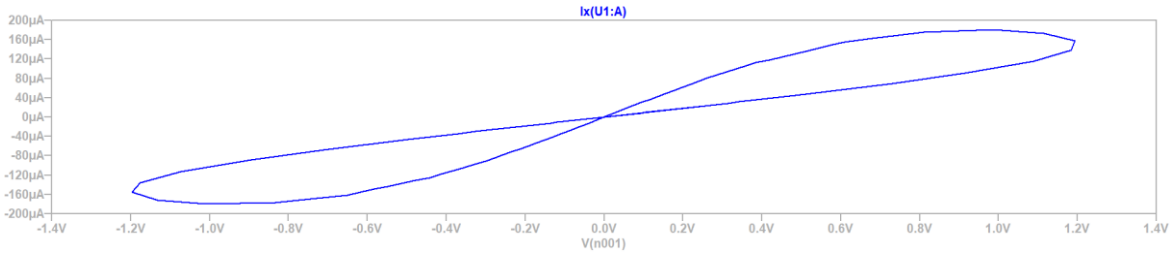

Figure 5.1: Memristor IV characteristics at Ron of 100Ω.

The Current passing when moving from HRS (High Resistance state) to LRS (low Resistance state) can be observed as 160µA when Ron is 100Ω.

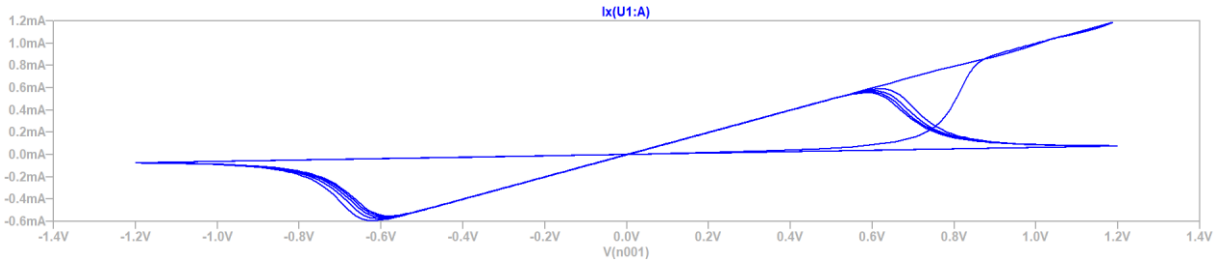

Figure 5.2: Memristor IV characteristics at Ron of 1KΩ.

The Current passing when moving from HRS (High Resistance state) to LRS (low Resistance state) can be observed as 1mA when Ron is 1KΩ.

5.2 LIF Neuron – Leaky Integrate and Fire Neuron

The Input is applied at the drain terminal of transistor M_1, When the voltage across the sensing capacitor C1 exceeds the threshold set by the CTRL signal, the corresponding cascaded inverters ($M_4M_{5\&}M_7$ M_8) generate an output spike which is shown in Fig.5.4. Once a spike is generated, the sensing capacitor is reset by the positive feedback loop from the transistors M_{11}.

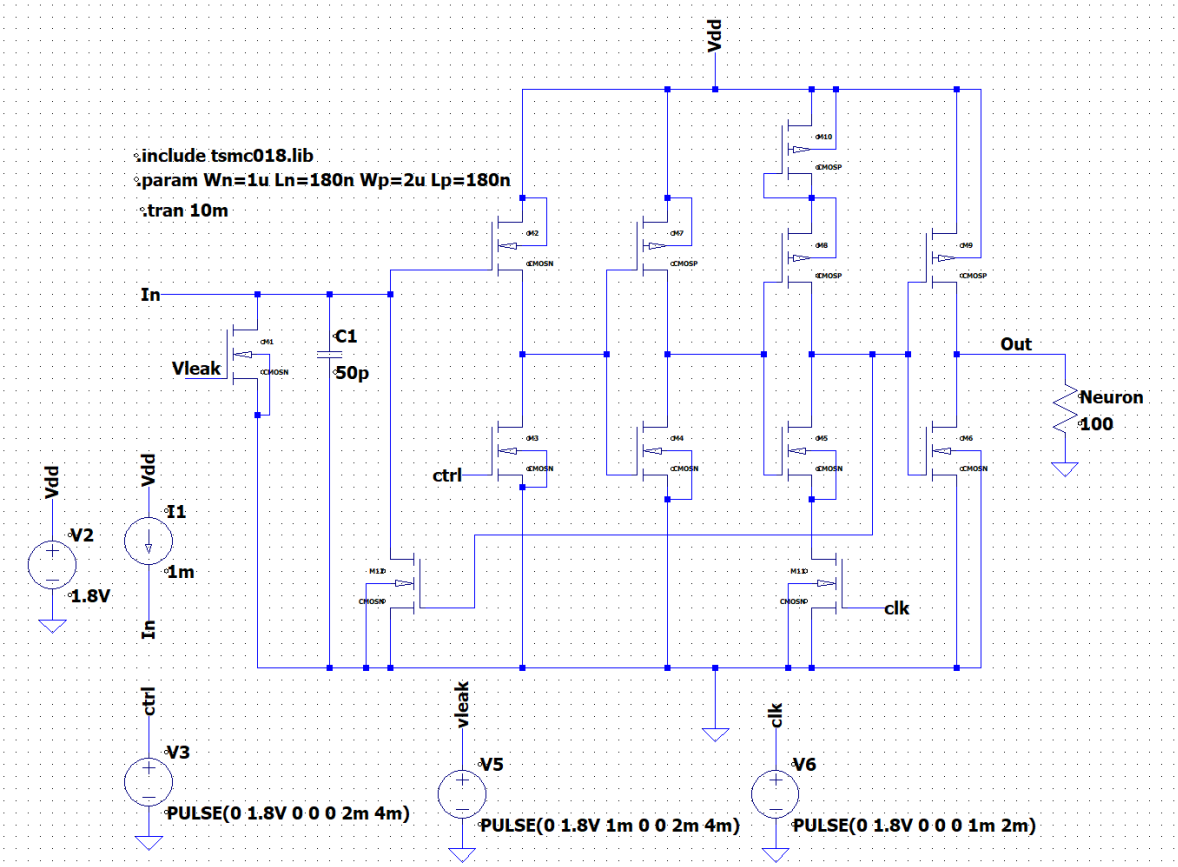

Figure 5.3: Circuit Diagram of LIF Neuron

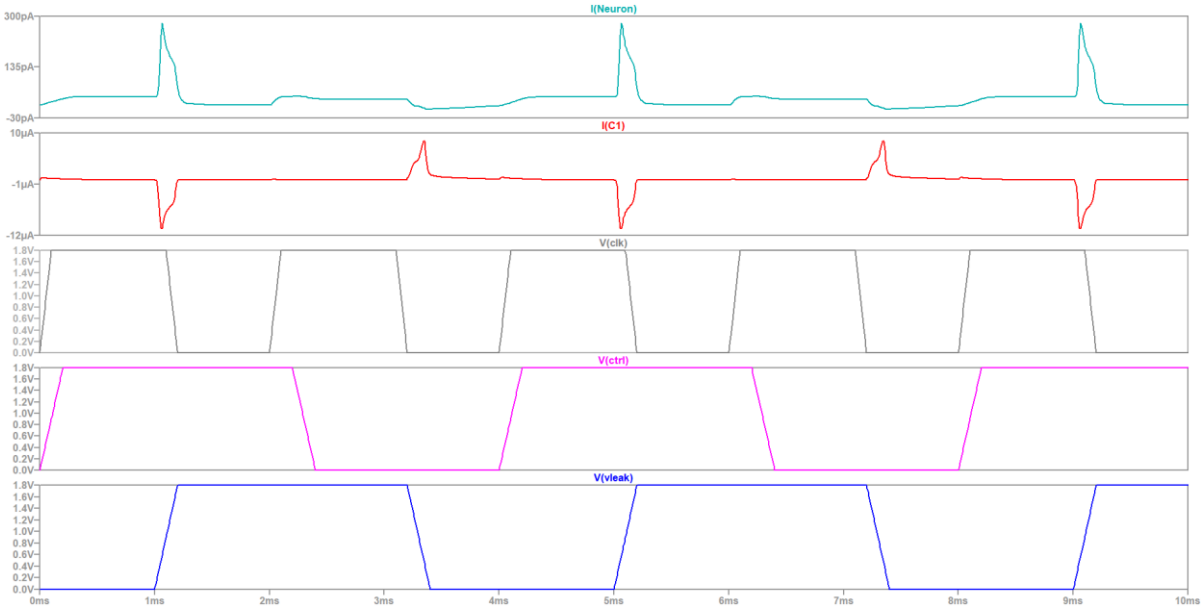

Figure 5.4: Simulation Results of Neuron Spiking

5.3 Storing "0101" in Crossbar Circuit 1 X 4

We will be utilizing Memristor circuit as a memory element to store 0101 in binary. Circuit implementation is shown in Fig.5.5 and simulations results shows that for storing "1" Memristor has High current across it and for storing "0" it passes low current through it. Simulation results are shown in Fig.5.6.

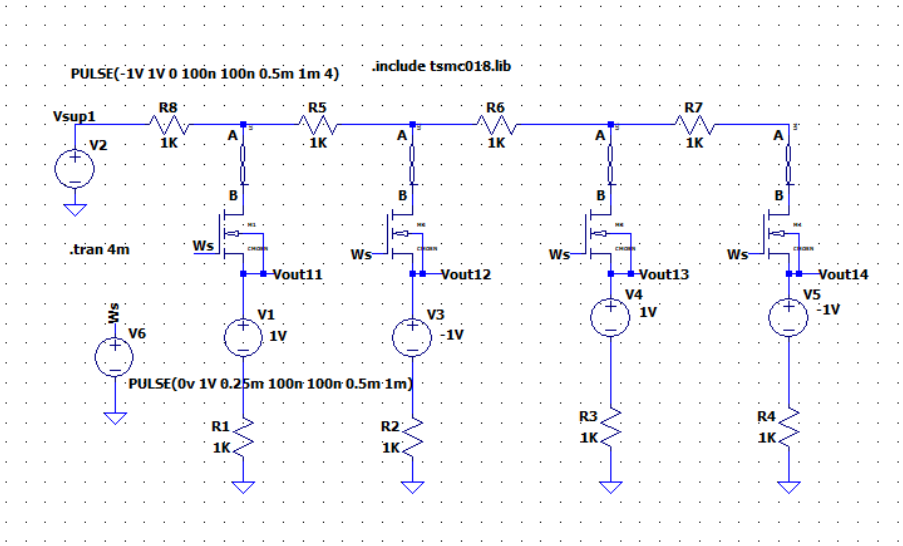

Figure 5.5: Circuit Diagram of 1X4 Cross bar

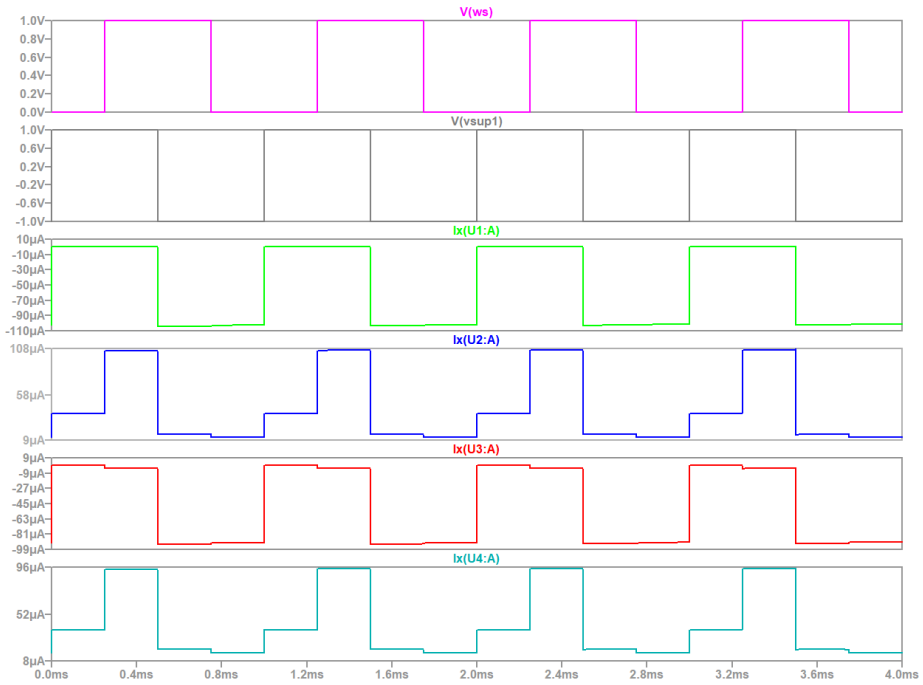

Figure 5.6: Simulation results of 1X4 Crossbar

5.4 Pre - Processing Stage

We are assuming that we already have the stream of 0's and 1's which constitute the array of pixels based on 4X4 array we have a list of 1's and 0's which we write on the crossbar and will obtain the final result as the highest current corresponds to the column.

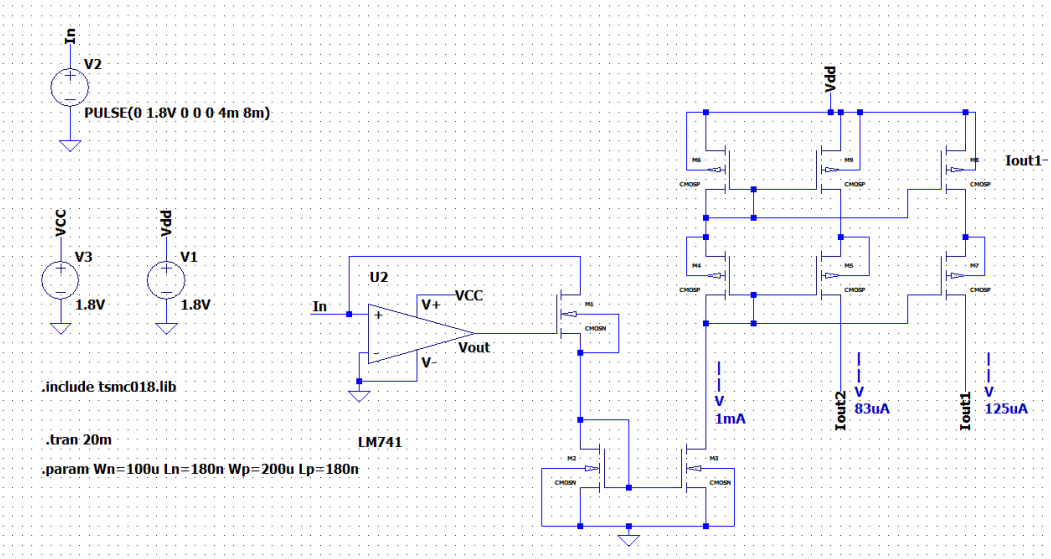

Figure 5.7: Circuit Diagram of Pre- processing stage

The Current mirror circuit is used to amplify the current and the two different currents Iout1 and Iout2 will be fed into further LIF Neuron stages. The Reference Current is 1mA and Iout1 and Iout2 are observed as 128µA and 86 µA respectively.

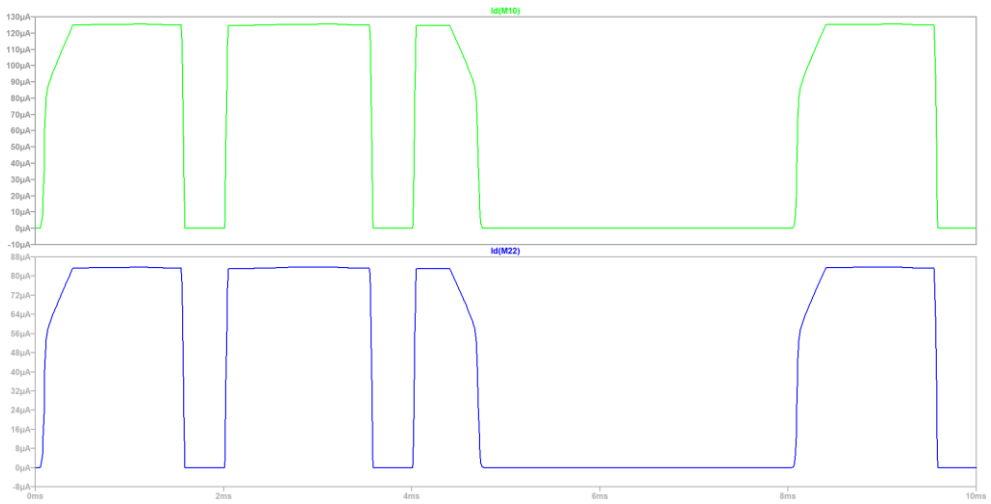

Figure 5.8: Iout1 and Iout2 scaled version of Iref

5.5 Conclusion and Future Work

We have simulated LIF Neuron and the spikes are generated which shows the peak current as 300pA. 1x4 Crossback is shown which stores “1001” where 1 high state is represented by Current of 108µA and 0 as 8 µA. In order to store an image which contains 28x28 pixels we require 784 memristors in a single column. Each digit is represented by a single column hence we require 10 columns each donating the digits from (0-9).

Future work includes the implementation 784X10 Memristive Crossbar along with WTA (Winner take all circuit) which can tells us by taking its input from all the 10 different columns of the crossbar and can readily identify the input image applied to it in range (0-9). Different Memristive models can also be studied for comparative analysis of SNN network.

References

[1] L. Chua, "Memristor-The missing circuit element," in *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507-519, September 1971, **DOI**: 10.1109/TCT.1971.1083337.

[2] Dennis V Christensen *et al,* 2022 “roadmap on neuromorphic computing and engineering” *Neuromorph. Comput. Eng.* **2** 022501 **DOI** 10.1088/2634-4386/ac4a83

[3] Strukov, D., Snider, G., Stewart, D. *et al.* The missing memristor found. *Nature* **453**, 80–83 (2008). https://doi.org/10.1038/nature06932

[4] Yogesh N Joglekar and Stephen J Wolf 2009 “The elusive memristor: properties of basic electrical circuits” *Eur. J. Phys*. 30 661 **DOI** 10.1088/0143-0807/30/4/001

[5] C. Yakopcic, M. Z. Alom and T. M. Taha, "Memristor crossbar deep network implementation based on a Convolutional neural network," *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, 2016, pp. 963-970, doi: 10.1109/IJCNN.2016.7727302.

[6] C. Yakopcic, T. M. Taha, G. Subramanyam and R. E. Pino, "Memristor SPICE model and crossbar simulation based on devices with nanosecond switching time," The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 2013, pp. 1-7, doi: 10.1109/IJCNN.2013.6706773.

[7] Schuman, C.D., Kulkarni, S.R., Parsa, M. et al. Opportunities for neuromorphic computing algorithms and applications. Nat Comput Sci 2, 10–19 (2022). https://doi.org/10.1038/s43588-021-00184-y

[8] The Fundamentals of Neural Networks, https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53

[9] http://yann.lecun.com/exdb/mnist/ (MNIST dataset website)

[10] https://www.tensorflow.org/datasets/keras_example (TensorFlow tutorial on MNIST with ANN)

[11] https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/ (Blog post explaining CNNs for MNIST classification)

[12] F. Nowshin and Y. Yi, "Memristor-based Deep Spiking Neural Network with a Computing-In-Memory Architecture," 2022 23rd International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 2022, pp. 1-6, doi: 10.1109/ISQED54688.2022.9806206.

[13] Schuman, C.D., Kulkarni, S.R., Parsa, M. *et al.* Opportunities for neuromorphic computing algorithms and applications. *Nat Comput Sci* **2**, 10–19 (2022). https://doi.org/10.1038/s43588-021-00184-y

PAPER NAME

nish_v3.pdf

WORD COUNT

4508 Words

CHARACTER COUNT

26513 Characters

PAGE COUNT

28 Pages

FILE SIZE

1.4MB

SUBMISSION DATE

May 31, 2024 1:55 PM GMT+5:30

REPORT DATE

May 31, 2024 1:56 PM GMT+5:30

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 11% Internet database
- 9% Publications database
- Crossref database
- Crossref Posted Content database
- 12% Submitted Works database