ADDIS ABEBA UNIVERSITY

ADDIS ABEBA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Implementation of the Programmable Memristor based – Long Short Term Memory (LSTM) Circuit for Time Series Prediction

By:

Abrham Ayal

Advisor:

Fetene Mulugeta (Phd)

Micro-Electronics Engineering - Stream

A Thesis Submitted to the School of Electrical and Computer Engineering, Addis Ababa Institute of Technology, Addis Ababa University, in partial fulfillment of the requirements for the Degree of Masters of Science in Micro-Electronics Engineering.

August 2022, Addis Ababa, Ethiopia.

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Implementation of the Programmable Memristor based – Long Short Term Memory (LSTM) Circuit for Time Series Prediction

Presented by Abrham Ayal, a candidate for the degree of Master of Science and hereby certify that it is worthy of acceptance.

Approval by Boards of Examiners

Dr. Bisrate Derebessa	_____	_____
Dean, SECE, AAiT	Date	Signature
Graduate Committee		
Dr. Fetene Mulugeta	_____	_____
Advisor	Date	Signature
Dr. Getachew Alemu	_____	_____
Examiner I	Date	Signature
Mr. Daniel Dilbie	_____	_____
Examiner II	Date	Signature

Declaration

I hereby certify that this report constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another. I declare that the report describes original work that has not previously been presented for the award of any other degree of any institution.

Abrham Ayal _____ _____

Name signature Date of submission

Place: Addis Ababa, Ethiopia

Acknowledgment

I want to start by expressing my gratitude to the omnipotent God. Furthermore, I appreciate Dr. Feten Mulugeta's constant follow-ups and helpful criticism during the seminars.
I want to thank you for letting me have from abroad co-advisor, Doctor Wendemagen. When I had a problem with my thesis, he was always willing to assist. I also want to thank my friends and classmates who were always willing to lend a hand and offer insightful advice.
In addition, I want to thank Habte Aregawi, my best friend, for all of his help. I also want to thank the staff at the School of Electrical and Computer Engineering in the Micro-Electronics stream at the Addis Abeba Institute of Technology for their technical support.
Last but not least, I would like to sincerely thank my family, especially my mother Kassanesh Berhanu, sister Abebayhu Ayal, Rahel Geta (Mitye), and brother Temesgen Ayal for their encouragement and support of my research and academic career.

Abstract

Recent advancements in recurrent deep neural networks with Long Short-Term Memory (LSTM) units have resulted in significant technological advancements in artificial intelligence. Recurrent Neural Network (RNN) and LSTM analysis of the time-series data enables us to recognise long-term trends and make predictions that can improve our way of life. State-of-the-art LSTM models with massively higher ambiguity and a large number of parameters, on the other hand, have a significant advantage. Previously, they tried to solve the time series prediction problems using LSTM based architecture, but the method to solve the problems is more complex, has less accuracy, limited memory capacity, data communication bandwidth and takes high power consumption. This work mainly focused on the implementation of already trained LSTM neural network and fully functional programmable memristor-based LSTM in analog circuitry for the solving time series prediction with increased the performance and to enhance power consumption of the system. Using the voltage-based memristive circuit on the LT-spice circuit simulator and machine learning on python programming with in the combination of the activation function, and multiplier describe the time series prediction of the LSTM architecture. Each state will have some noise, and as the noise level increases, so does the prediction accuracy in analog hardware. As a result we have get results using machine learning based LSTM and programmable memristor based LSTM of the MSE and RMSE value reduced by an average of 0.0101 and 0.100 to 0.00915 and 0.095 respectively. We use real-world regression and prediction problems to demonstrate the capabilities of our system, and the programmable memristor LSTM is a promising easy-configuration, low-power, and low-latency hardware platform for edge inference.

key-word: LSTM, RNN, Memristor, prediction, Programmable, Crossbar Memristor

Contents

List of Figures

List of Tables

ABBREVIATIONS

AI	Artificial Intelligence	4
ANN	Artificial Neural Network	3
BJTs	Bipolar Junction Transistors	1
CMOS	Complementary Metal-Oxide Semiconductor	1
FPGA	Field Programmable Gate Array	16
FVF	Flipped Voltage Follower	31
HP-LAB	Hewlett Packard Laboratory	8
IC	Integrated Circuit	1
LSTM	Long Short-Term Memory	iii
MAE	Mean Absolute Error	43
MSE	Mean Squared Error	42
MOSFET	Metal Oxide Semiconductor Field Effect	1
MIM	Metal-Insulator-Metal	2
NN	Neural Network	12
RMSE	Root Mean Squared Error	43
RNN	Recurrent Neural Network	iii
VMM	Vector-Matrix Multiplication	12

Symbols

ϕ	Magnetic flux
μ	Mobility
σ	Sigmoid function
tanh	Hyperbolic function
β	Beta
i_t	input
f_t	forget
O_t	output
$\sum$	summation
I	Current
Q	Electric Charge
V	Voltage
M	memristor
P	power
C	capacitor
L	inductor

Chapter 1

Introduction

The world has experienced a technological boom ever since William Shockley, John Bardeen, and Walter Brattain created the transistor in 1947. Jack Kilby of Texas Instruments created the first flip flop using two bipolar transistors [1]. Later transistors became an integral part of any electronic device. There was primarily one amendment within the course of technological advancements: the transfer from Bipolar Junction Transistors (BJTs) to Metal Oxide Semiconductor Field Effect (MOSFET). Transistors MOSFET in digital Integrated Circuit (IC) style [2]. Since then, Complementary Metal-Oxide Semiconductor (CMOS) process technologies are scaled down steady by permanent by the "By Moore's Law". However, this electronic transistor shrinking trend cannot continue once the semiconductor industry reaches the purpose wherever any reductions in size are intolerable resulting in unreliable operation of CMOS devices. As a result, various solutions are required to continue producing smaller, faster, and more power-efficient electronic devices. Scaling down of CMOS memory devices (Flash, SRAM, etc.) may be a major concern for the semiconductor industry, consistent with [3].

In circuit theory, the three basic two-terminal devices, namely the resistor, capacitor, and inductor, are well understood. These elements are defined by the relationship between two of the four fundamental circuit variables: current I, voltage V, charge Q, and flux ϕ. The current is defined as the charge's derivative with respect to time. The voltage is defined by Faraday's law as the time derivative of the flux. The relationship between voltage and current $(dv = Rdi)$ defines a resistor, the relationship between charge and voltage $(dq = Cdv)$ defines a capacitor, and the relationship between flux and current (dϕ=Ldi) defines an inductor [4].

In this regard, scientific researchers have planned a brand new physical element called a memristor as a possible answer to switch transistor-based memory cells. A memristor

is a two-terminal 'memory resistor', which performs computation via physical laws at the same location where information is stored. This function completely eliminates the requirement for data transfer between the memory and computer.Memristors have been used successfully in feed-forward fully-connected neural networks when integrated into a crossbar design. Professor Leon Chua predicted the theoretical existence of a memristor in 1971, and it describes the relationship between the integral of the applied voltage and the integral of the flowing current [5, 6]. In fact, memristors have found a wide range of applications in the electronics area. The nano-scale memristor has low power consumption, is nonvolatile, and its resistance can be adjusted by external stimulus. It has been demonstrated that the memristor can mimic the behavior of synapses. Furthermore, a large number of memristors organized in a crossbar architecture can achieve in-memory parallel computing while consuming very little power. Memristors can be used in memory-logic, analog circuits, and neuromorphic systems, among other things. A memristor has several advantages over standard memory technologies, including nonvolatility, good scalability, virtually no leakage current, and electrical and manufacturing compatibility [5, 6]. They're utilized to create neural networks in hardware, including back propagation and Spike-timing-Dependent-Plasticity training algorithms, as well as neuromorphic computer systems [7] inspired by the human brain [8–10]. Memristors have a variety of uses due to their Metal-Insulator-Metal (MIM) structure, compact footprint on a chip, memory in the form of resistance, soft switching time, great technique, and soft switching energy. [3]. As a result, memristors are expected to have a bright future, with memristors substituting transistors in analog and digital IC design where practical, and memristive systems with in-memory computing replacing the Von-Neumann architecture [6, 11, 12].

LSTM is a recurrent neural network RNN architecture developed by Hochreiter and Schmidhuber in 1997 [13] that has been shown to be effective for various learning problems, particularly those involving sequential data [14]. The LSTM architecture consists of blocks, which are a collection of recurrently connected units. The vanishing gradient problem occurs when the gradient of the error function increases or decreases exponentially over time in RNNs. To address this issue, the network units in LSTMs are redesigned. Each LSTM block consists of one or more self-contained memory cells as well as input, forget, and output multiplicative gates. The gates enable memory cells to

store and access data for longer intervals, enhancing performance [15].

1.1 Statement of the Problem

Artificial intelligence is already widely used in a variety of applications, but it faces challenges in terms of implementation by increasing simplicity and performance. As a result, implementing a programmable memristor-based long-short-term memory (LSTM) is the best solution to this problem. Because of this condition and further extension, LSTM-based time series prediction is a subset of an artificial intelligence-based problem.
As if, the majority of AI-based technology requires the fastest memory, which is CPU storage device size, easy configuration, flexibility, feasibility, accuracy, and other characteristics. To overcome such issues, a programmable memristor-based system is one of the best ways to reduce the aforementioned questions for AI redundancy.
Furthermore, it is a question of whether most technologies have no flexibility and are difficult to repair when they are nonfunctional, but using memristor material to implement, the flexibility increases to the point where it is preferable to use memristor technologies over previous CMOS-based technologies. The goal of this thesis is to create a fully functional analogue LSTM circuit using memristive crossbars and test it on machine learning problems.

1.2 Objective

General objective

- To design and simulate the implementation of programmable memristor based (sigmoid and tanh function) long short-term memory for time series prediction in neural network.

Specific objective

- To understand memristor component characteristics.
- To learn how to configure the memristor element.
- To study machine learning algorithm based the memristor technologies for RNN and Artificial Neural Network (ANN) structure .
- To recognize the various activation functions used in this findings

- To Comprehend the time series prediction on analog circuit and machine learning
- To show overall system of programmable memristor based sigmoid and tanh activation function for LSTM.
- to understand various multiplier and operational amplifier.

1.3 Significance and Contribution of the research

More research into long- and short-term memory is required, given recent advances in artificial intelligence. It is incredibly clever, and the programmable memristor functional circuit is the best option for reducing complexity and improving Artificial Intelligence (AI) performance. It believes that understanding long-short-term memory and implementing the programmable memristor-based LSTM as soon as possible is the best approach. Finally, the analog circuit will implement the LSTM.

The cross-bar memristor architecture of the programmable memristor is supposed to make it simpler to understand and improve the performance of the activation function. Additionally, the memristor is the current technology utilized for various memory capacities, such as Re-RAM. Therefore, it can be used for the implementation of LSTM. For this work or research, we design and simulate the programmable or re-configurable memristor element based on long-term memory or artificial intelligence purposes for time series prediction. Therefore, the programmable memristor of the analog circuit is the new trend, and it used to implement the long-short term memory for many applications such as deep learning systems, prediction, audio, video, and image recognition systems. Previously, neural network circuits required a high amount of power, timing delay, complex circuit network configuration, size scale problems, and low flexibility; however, solving such problems by using the best arrangement of memristor electrical elements is now possible.

1.4 Scope of the research

The scope of this research is first to find the research topic, pick out the existing gap in the problem and formulate the general and specific objective. And , select the proper software used for analog circuit implementation and machine learning problems. And also simulate the memristor electrical element and prove if it is work properly or not.

Finally, design the methodology and configure the memristor element properly for time series prediction and also compare the analog circuit with the software based on a machine learning algorithm. At the end of the day, we conclude that this research scope is the implementation of a programmable memristor based on long short-term memory for time series prediction that shows up to the simulation stage.

1.5 Paper organization

The thesis document consists of Five chapters:

- Chapter 1 Covers introduction, the general and specific objectives, the scope, the contribution, significant, Existence gap and thesis Organization of the thesis.
- Chapter 2 Covers the relevant literature review of the research, Background theory, Crossbar memristive array, Long short-term memory and Important literature.
- Chapter 3 Covers the system design, appropriate methodology and all the circuit part of the research, it is includes software part,selected problem,selected model,proposed voltage based circuit for implementation of LSTM,the whole circuit and LSTM architecture.
- Chapter 4 Covers the whole research results, That is mainly includes the results ,the evaluation metrics result and the power consumption of the circuit.
- Chapter 5 Covers Conclusion and Recommendation

Chapter 2

Literature Review

2.1 Background

2.1.1 Memristor Theory

Leon Chua was the first person to notice that there was a missing elementary circuit half and to publish a bit in 1971 regarding the memristor [5]. He advocates that a memristor would complement the following list of elementary circuit elements: resistance, capacitor, and inductance.

In fact, it's typically been best shown by learning the association of elementary circuit components with a try of the fundamental circuit parameters as represented in Figure 2.1. among the figure, the diagonal line contains the basic circuit variables of charge, current, voltage, and flux. each row contains the equations where the corresponding diagonal parameters are described in terms of the alternative circuit variables and so the circuit parts. Whereas, each column contains the equations where the corresponding diagonal variable is a component of the expressions. it's not difficult to see the missing part relationship between charge letter of the q and flux ϕ Chua discovered the next mathematical correlations of charge-controlled memristance with voltage and flux-controlled memristance with the present [6]. If the flux and charge relationship is delineated as a charge operation, it's a charge-controlled memristor; if it is expressed as a flux operation, it is a flux-controlled memristor. A charge-controlled memristor is outlined as:

$$\phi = f(q) \tag{2.1}$$

differentiating (2.1) with respect to time yields

$$\frac{d\phi}{dt} = \frac{df}{dq} * \frac{dq}{dt} \tag{2.2}$$

having the voltage as v(t)= $\frac{d\phi}{dt}$and the current as $i(t)$= $\frac{dq}{dt}$ can be rewritten as:

$$V(t) = M(q) * i(t) \;\; where \;\; M(q) = \frac{df(q)}{dq} \tag{2.3}$$

M(q) is the memristance, and similar to resistor it has the units of ohm. Similar to the equations above we can write for a flux controlled memristor,

$$q = f(\phi) \tag{2.4}$$

differentiating (2.5) with respect to time yields

$$\frac{dq}{dt} = \frac{df(\phi)}{d\phi}.\frac{d\phi}{dt} \tag{2.5}$$

Having the current a i(t)=$\frac{dq}{dt}$ and the voltage as v(t)=$\frac{d\phi}{dt}$, (2.6) can be written as with the analogous to memristance

$$i(t) = W(q)v(t) \;\; where \;\; W(\phi) = \frac{df(\phi)}{d\phi} \tag{2.6}$$

W(ϕ) is called the memductance ,and similar to conductance ,it has units of Siemens. Memristance (short for ”memory resistance”) M(q(t)) and resistance units are used in memristors. Similarly, the unit of a flux-controlled memristor is memductance W(ϕ(t)), which has units of electrical phenomenon. Consequently, a memristor could be a two-terminal passive elementary circuit element that exhibits memory properties by dynamically adjusting its resistance as a function of the fraction of charge that has flowed through the memristor. Chua and Kang extended the memristor speculation to memristive devices in 1976 [6].

Memristive devices differ from memristors within the sense that their resistance changes. A memristive device, in essence, has an interior that influences its resistance. Depending on how much voltage is applied across it and for how long, as well as how much current is flowing through it, the internal state changes. Mathematically, a similar equation for memristors will be expressed as following [11]:

$$V(t) = M(x,i)i(t) \;\; and \;\; i(t) = W(x,v)i(t) \tag{2.7}$$

where M(x, i) denotes the time-invariant memristive device's memristance and W(x,v) denotes the time-invariant memristive device's memductance. Also, $\frac{dx}{dt} = f$(x, i) and is valid in the former case and $\frac{dx}{dt} = f$(x, v) in the latter case. Both memristor devices and circuits exhibit hysteresis in their I-V curves, as seen in Figure 2.2. Although the devices' shapes can vary, the hysteresis always goes through the origin. Memristors with $f(x, i) = i$ are included in the broad category of memristor devices.

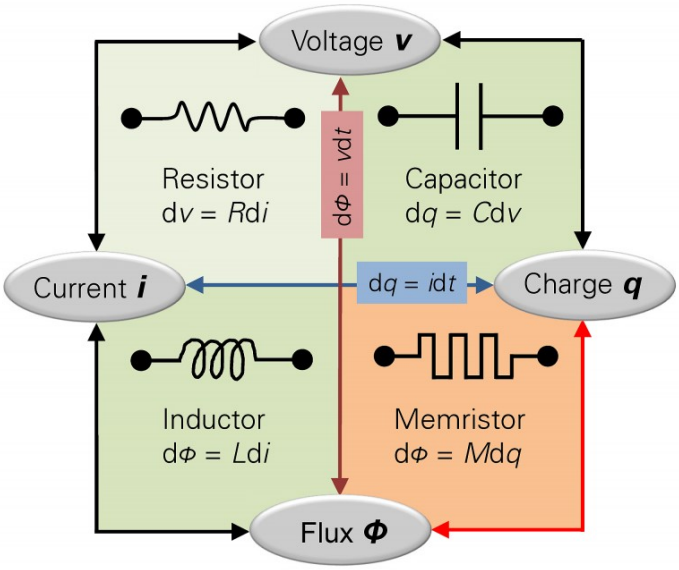


Figure 2.1: Relationships between the basic circuit parts and variables are symmetrical. The memristor symbol is also present in the bottom left corner.

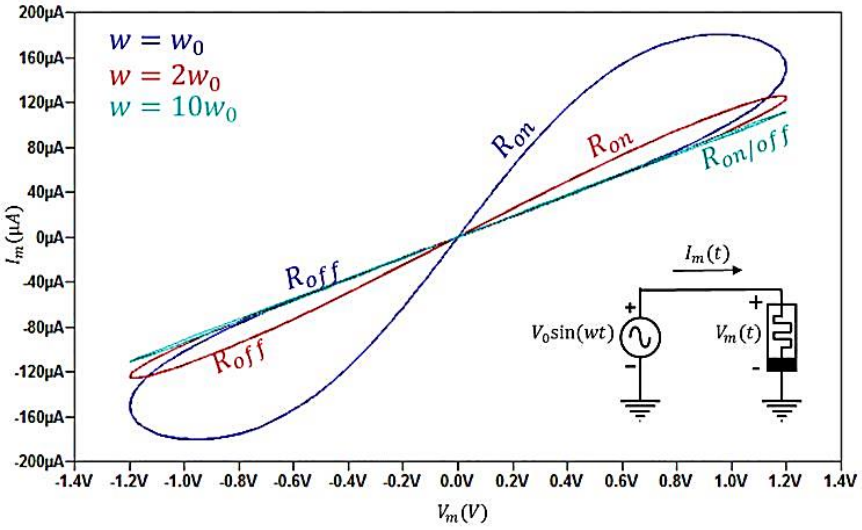


Figure 2.2: The I-V characteristics of a memristor circuit with in different frequency [11].

The physical memristor device is first introduced by Hewlett Packard Laboratory (HP-LAB) 37 years after [12] . The fact that the conceptual Chua's memristor can be realized us-

ing HP Lab's nanoscale device inspired the researchers to use this element in the implementations of high-density systems, such as non-volatile memory implementations and synapse realizations in neural network circuits. The schematic and crossbar circuit symbol of the memristor is describe that in Fig 2.3 The element has two states, one corresponding to the closed state, where the element displays a low resistance with W ≅ D, (R_ON) ;while the other corresponds to the open state, where the element shows a high resistance (W≅ 0), (ROFF) The total resistance R_{MEM} of the memristor can be shown by

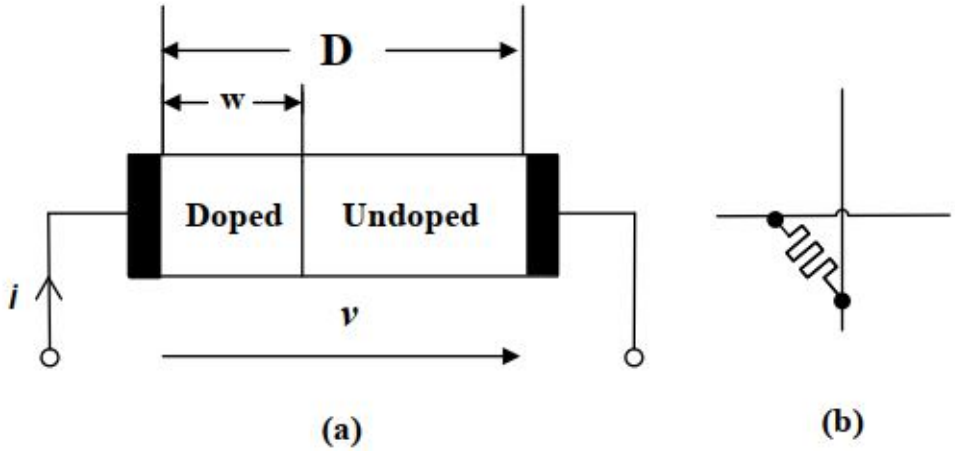


Figure 2.3: The schematic a and crossbar circuit symbol b of the memristor [16]

the following relation:

$$R_{MEM}(X) = R_{ON}(X) + R_{OFF}(1 - X) \quad Where \;\; X = \frac{W}{D} \tag{2.8}$$

The relationship between the current and voltage of the memristor can be defined as follows:

$$V(t) = R_{MEM}(W)i(t) \tag{2.9}$$

The speed of motion of the limit between the doped and undoped regions can be shown as follows, contingent on RON resistance, passing current, and other factors: undoped regions can be shown as follows:

$$\frac{dx}{dt} = ki(t)f(x), \qquad k = \frac{\mu vRON}{D2} \tag{2.10}$$

where μv is called as dopant mobility and the value of μv is approximately is $10^{-14}m^2s^{-1}V^{-1}$.

2.2 Cross bar memristive array

Two parallel Nano wire layers that make up the memristor crossbar serve as the top electrode and bottom electrode, respectively. At each cross point, a memristor is created by sandwiching the memristive material between two nano wire layers [17]. The schematic diagram of memristor crossbar is shown in Figure 2.4. For the architecture of massive neural networks, memristor crossbar is appropriate. First off, it has a high density since each cross point is a memristor and the crossbar may be stacked vertically. Memristor is also multi-state, nano-scale, and nonvolatile. Due to the crossbar's ability to integrate memory and processing [18] [19] and the memristor's nonvolatile nature and low operating voltage, it also has a low power consumption. Because of the memristor crossbar's advantages, a variety of neural networks have adopted this architecture. Memristor crossbar in neural networks has three operations: read, write, and training. We demonstrate how the memristor crossbar reads, writes, and trains with an example in this section.

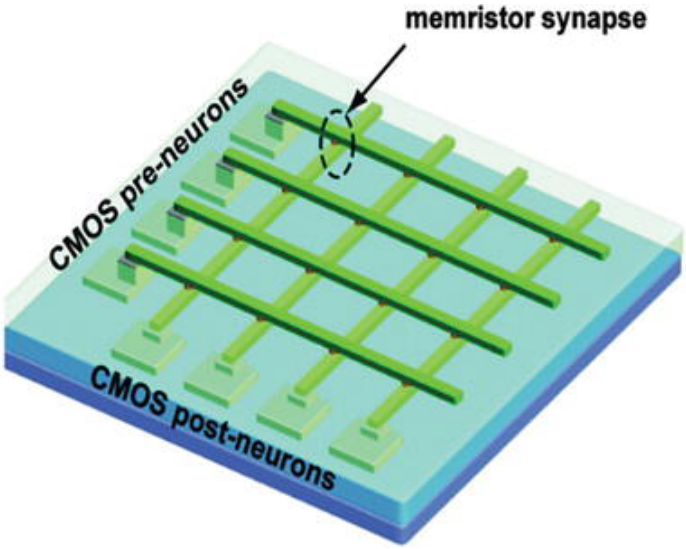


Figure 2.4: Cross bar memristor [19]

2.2.1 Read operation

The conductance of a single memristor can be read separately in a memristor crossbar. We presume that we will read the m_{ij} memristor, which is the intersection of the **i-th** upper wire and the **jth** bottom wire, as illustrated in Figure 2.5. Other top wires and bottom wires are grounded while the voltage V is supplied to the **ith** top wire. Only

the m_{ij} memristor is receiving the V bias in this case, allowing the current to be collected on the **jth** bottom wire. Ohm's law states that M $=\frac{1V}{4I}$ is the formula for calculating the conductance of a mij memristor [17] .

2.2.2 Write Operation

The conductance of a m_{ij} memristor can be written down individually, much like a reading operation. We anticipate writing the M_{ij} memristor. The target memristor will be directly exposed to writing pulses of varying amplitude and length. The j_{th} bottom wire is grounded, and voltage V is applied to the i_{th} upper wire. Only the m_{ij} memristor receives the entire voltage V, which is over the threshold and can alter the conductance of the target memristor, after the other upper wires and bottom wires have received voltage $V/2$. Since there is no voltage delivered to the other memristors, their conductance is unaffected [20]

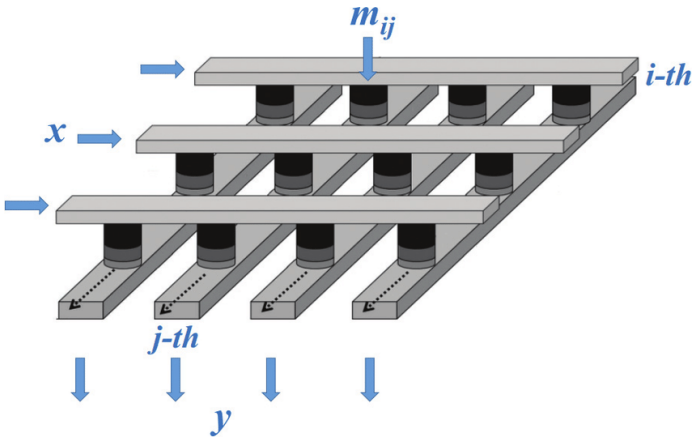


Figure 2.5: Cross bar memristor readout [19]

2.2.3 Vector Crossbar Arrays

As was pointed out in the introduction, there are many different electronic applications for memristors. They are mostly used in crossbar arrangements, as shown in Figure 2.6 (Steinbuch introduced the idea of 2-dimensional resistivity crossbar arrays for the first time in 1961) [21, 22]. In particular, Figure 2.6 shows how to implement a neural network using a memristor crossbar. It operates in a simple manner. The inputs x_1, x_2, and x_3 are reflected by the voltage sources V_1, V_2, and V_3 in the circuit. The weights from W_{11} to W_{33} correspond to the memristor conductance's in the same order from G_{11} to

G_{33}. Finally, the outputs Y_1, Y_2, and Y_3 correspond to the expected currents i_1, i_2, and i_3. In practice, the currents are passed thorough the virtual grounds created by operational amplifiers. The output layer uses linear activation as a function to keep things simple. In essence, the crossbar executes parallel Vector-Matrix Multiplication (VMM) operations quickly. Another advantage arises when the Neural Network (NN) size increases, creating a slowdown in the software's execution and a slowdowns in the data transfer between the processing unit and memory. There is no need to store weight values elsewhere in the case of the memristor crossbar because weight training takes place there [22]. The synaptic weights of neural networks can be efficiently realized using this

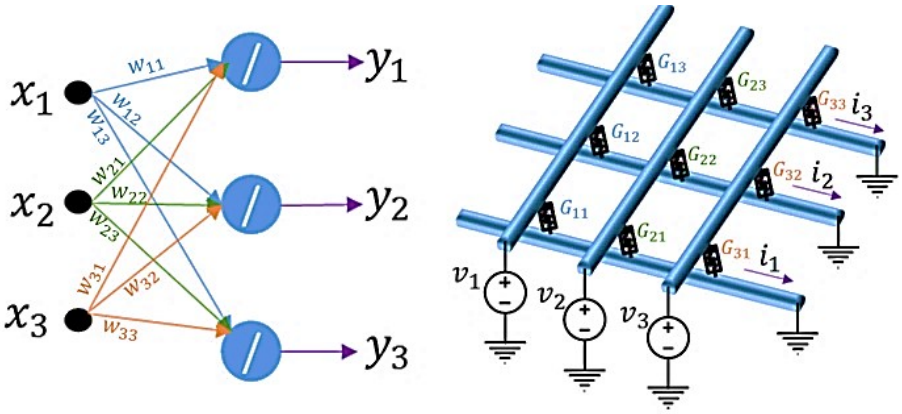


Figure 2.6: A basic neural network (on the left) and its crossbar implementation (right) [21].

circuit. Each memristor's conductivity corresponds to a weight. The usage of the memristive crossbar gives the synaptic weights dynamical properties because the memristor is an element with memory. In addition, the availability of nanoscale components enables the installation of the NN in a very tiny physical space with high efficacy. Analog outputs are produced as voltage pulses, and analog input values are represented by symmetrical current or voltage pulses. Many elementary mathematical operations, such addition and multiplication, may now be accomplished very effectively in a single step thanks to this future [23,24]. As it can be seen clearly from Fig 2.7, By directly applying Kirchhoff's law, the vector multiplication can be naturally processed. The applied voltages V_i are multiplied by the conductance's (G_{i+} and G_{i-}) according to Kirchhoff's rules to produce the

output voltages:

$$V_o^+ - V_o^- = \sum \left(Vi \frac{G_i^+ - G_i^-}{G_o} \right) \quad where \; G_0 = G_1^+ + G_2^+ + G_3^+ + ...G_N^+ = G_1^- + G_2^- + G_3^- ...G_N^- \tag{2.11}$$

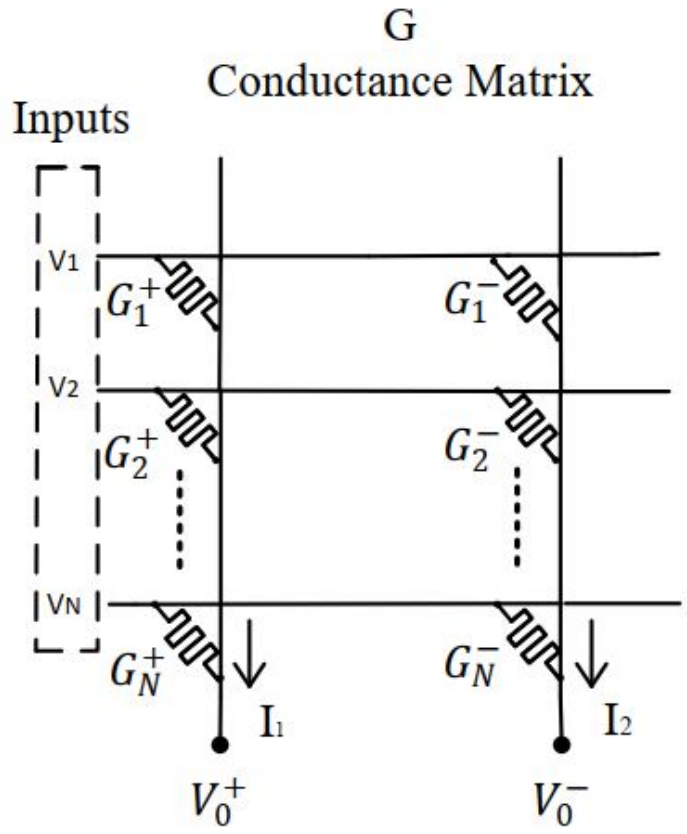


Figure 2.7: The principle of analog vector-matrix multiplication with memristor array [23]

2.3 Long short term memory (LSTM)

In order to address the problem of vanishing and exploding gradients in RNNs, Hochreiter and Schmiduber created LSTM, a form of RNN [13]. The issue appears when RNNs are being trained for patterns with wide time step distributions. Figure 2.8 depicts an RNN and its unrolled equivalent with a decreasing gradient descent problem. The RNN has only one input unit, one recurrent hidden unit, and one output unit in the diagram. Consider the case where the input is non-zero at time t and zero at subsequent time steps, as shown in [25]. The output at time (t+3) will be very little affected by the input at time (t) if the recurrent weight value is less than one. Truthfully, as the time step difference be-

tween the input and the output grows, the influence of the input on the output diminishes exponentially.

The error derivative with regard to the input will therefore decrease. Similar to this, the expanding gradient problem arises when the weight value of the recurrent edge is greater than 1. The vanishing and expanding gradient issues are also influenced by the type of activation function. A sigmoid activation function, for example, will always output values less than one, increasing the likelihood of the vanishing gradient problem. Because its output value is not limited to one, amplify a linear unit with an output equal to max (0, x) tends to explode in contrast. As a result, a simple RNN cell cannot be controlled in

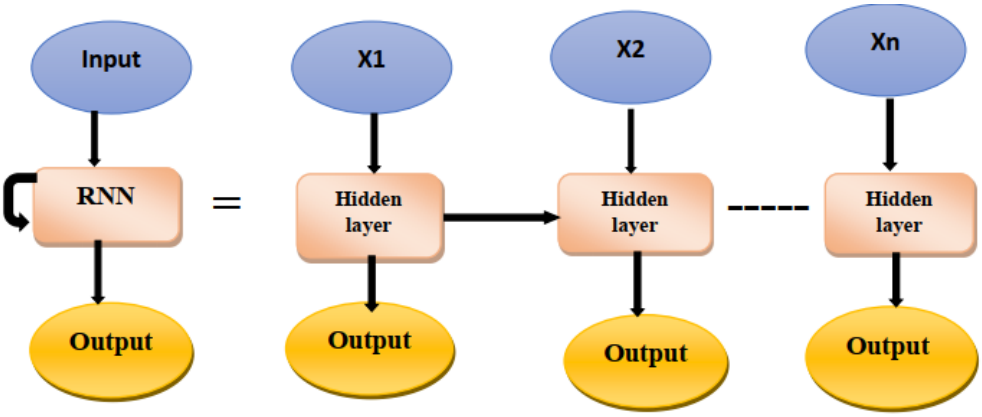


Figure 2.8: RNN schematic representation on the left When RNN approaches a vanishing gradient problem, the equivalent diagram on the right is displayed.

terms of the data that is fed into and output from it. The sigmoid function has a smooth range of zero to one. When a gate's output is one, it is possible to use the Hadamard multiplication operation to send all the information from the current stage to the following stage. The gate similarly prevents passing any information to the following stage when it is zero. As seen in Figure 2.9 each of the three gates has the same structure, including the same activation functions and hadamard multiplication units. The output value at the previous time step, h_{t-1}, and the network input value at the current time step, x_t,, are combined to create the shared input vector. They simply differ in the weights they apply to the input vector[x_t, h_{t-1},1]., as shown below in mathematical form [13].

$$f_t = \sigma\left(b^f 1 + w^f xt + u^f h_{t-1}\right) \tag{2.12}$$

$$i_t = \sigma\left(b^i 1 + w^i xt + u^i h_{t-1}\right) \tag{2.13}$$

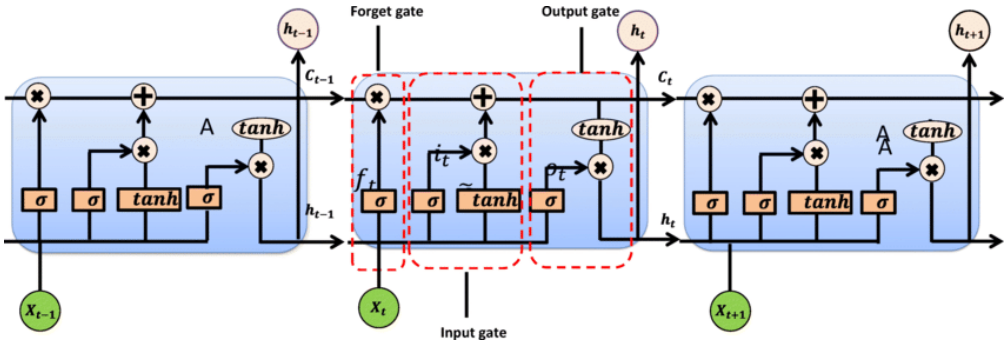


Figure 2.9: Full long short term memory (LSTM) cell architecture [26].

$$o_t = \sigma\left(b^o 1 + w^o xt + u^o h_{t-1}\right) \tag{2.14}$$

The participant worth C'_t, which has a similar structure to the gates but a different activation function, is added to some of the recent cell state C_{t-1} in order to modify the cell state C_t of an LSTM cell:

$$C'_t = \tanh\left(b^{c'} 1 + w^{c'} xt + u^{c'} h_{t-1}\right) \tag{2.15}$$

$$C_t = f_t * C_{t-1} + i_t * C'_t \tag{2.16}$$

Finally, the current cell LSTM output is some portion of the filtered cell state C_t:

$$h_t = o_t * \tanh(C_t) \tag{2.17}$$

The equations (2.12-2.15) relate to the scenario of a single LSTM hidden unit, which is significant to note. In actual use, LSTM hidden layers are much larger because larger weight matrices are used with LSTMs.

Also, keep in mind that the Keras Tensor Flow library employs the LSTM version presented above [27]. Other LSTM architectures will be presented in this thesis' methodology section. It is no surprise that LSTM has grown in popularity in recent years, as it is widely used in the field of machine learning. Image captioning and Natural language processing [28–30] Some LSTM applications (included in this research) include speech recognition [31], time-series prediction, and video processing [32] [32].

2.4 Important Literature's

Memristor crossbar arrays are more effective at implementing NN than computers with their software, as was previously noted in section [2.2.3].

There aren't many chip implementations of NNs that use memristive crossbar arrays because memristos are still in their early phases of development. Instead, Field Programmable Gate Array (FPGA) NN implementations have become a general trend in literature works. Since 2015, the research community has presented FPGA implementations of LSTM each year [28–30, 33].
These analyses compare the performance of LSTM implementations on FPGA and software. Memristor crossbar implementations are expected to perform better than digital crossbar implementations as memristor technology advances [34]. There are, however, a few unpublished works [34–36] that discuss the application of LSTM to memristor and memristive crossbar arrays. offers a completely analogue circuit configuration for a memristors LSTM in 0.18μm CMOS technology [35]. In order to account for the non-idealities connected with analog VMM operation, they introduced various software limitations. Their main finding is that memristors require a symmetric change in conductance values when positive and negative voltages are applied for good performance. In large LSTM networks with sizes of up to 512 hidden units, asymmetry variations as small as 2 percent can significantly affect performance results when compared to fully connected networks with smaller sizes and smaller training datasets. Last but not least, [36] shows a constructed chip that employs a crossbar array of the one-transistor, one-memristor (1T1M) type to implement the VMM operation of the LSTM algorithm. However, software was used to implement the remaining operations in their work. They were successful in solving time-series prediction issues and training their weight matrix of memristors in-situ. Systems based on memristor neural networks must have an effective training procedure. Off-chip training and on-chip training are two training methods. Any learning algorithm may be implemented efficiently and performed on strong computer clusters, which is the main advantage of off-chip training. Memristor crossbars are susceptible to design flaws and variances [37, 38] These differences might exist both within a crossbar's individual devices and in crossbars separating various chips. These analog circuits are tough to emulate in software, making off-chip training for them challenging. The advantage

of on-chip learning is that it can use the component's whole analog range and account for differences between components (as opposed to a set of discrete resistances that off-chip training will likely need to target). On-chip learning has the benefit of utilizing the complete analog range of the device and compensating for variations between various devices (as opposed to a set of discrete resistances that off-chip training will likely need to target). This paper proposes circuits for on-chip training of memristor crossbars that use two memristors per synapses. The most recent memristor crossbar circuit fabrications for neuromorphic computing have utilized two memristors every synapse [39, 40].
Two memristors every synapse offer twice the precision of synaptic weight as matched to a system that only uses single memristor every synapse. This may allow neural networks to be trained more successfully and efficiently while consuming less energy [41]. In order to train using back propagation algorithms, we construct an unique circuit for error back propagation. On-chip training circuits for memristor crossbars have been discussed in previous works . On-chip gradient descent-based training of memristor crossbars with a single memristor per synapse was addressed by Soudry et al. [42, 43]. The training of systems with two memristors per synapse is not properly considered. On-chip training of crossbar systems with two memristors per synapse was addressed by Boxun et al. [43]. Ta/HfO2 memristors were used in this experiment.

Chapter 3

System Design and Implementation

3.1 Introduction

In this section, we describe the overall system of the thesis, which are used the software part of anaconda Jupyter and simulation part of LT-Spice the whole circuit design.

3.2 Software part

In this thesis, we use Jupyter anaconda 3 Notebook for the simulation of international airline passenger prediction using the dateset [44]. We can create and update documents that show the input and output of a Python or R script using the anaconda jupyter notebook application. Notebook comes with Python and R by default, but with some adjusting, it can run a wide range of other kernel environments. As a result, using Jupyter Anaconda 3 (which ended up being the main tool for software simulations of different LSTM architectures in this research), solve the problem using LSTM in Kera's library. LT-spice, which is referred to be a SPICE-based analog electrical circuit simulator software, is an additional program. Makes semiconductors is Analog Devices. It is mostly used and distributed piece of software in the industry, and LT-spice has not been intentionally made less powerful. no limitations on nodes, components, or sub circuits. Additionally, the issues that are resolved utilizing the long short-term memory based on programmable memristors are presented as follows: Utilizing prior sample points at earlier time steps, forecast the sample point at the current time step t, $(t-1, t-d)$. The selection parameter d is prediction delay or a look-back number. look-back define that the number of recent data points to be used when predicted each future value in the time series. A single LSTM cell's output can be expressed as follows:

$$y(t) = f_{LSTM}\left(x(t-1), ..., x(t-d)\right). \tag{3.1}$$

3.3 Select the Problems

3.3.1 Sequential time-series prediction problem

The prediction of international airline passenger count was chosen as the initial stage in the implementation of the LSTM neural network algorithm in analog hardware (in the circuit simulator) utilizing memristive crossbar arrays. In reality, Brownlee's [44]web tutorial was already employing LSTM to resolve the issue in Kera's library (which became the main tool for software simulations of different LSTM architectures in this thesis).
To get acceptable prediction results for this straightforward situation, a larger LSTM neural network is not necessary. Reduced memristive crossbar arrays are employed in the analog version of the algorithm because the LSTM network's smaller size necessitates the use of smaller weight matrices in the network. Additionally, the dataset [45] for the issue is moderate enough to allow circuit simulators to run all of the testing data. It contains 144 sample points, which correspond to the thousands of international passengers carried by airlines each month between 1949 and 1960. Figure 3.1 below displays the dateset plot. This dateset is used in LSTM-based simulations at the circuit- and system-level. Data on the number of passengers flying internationally each month from 1949 to 1960.

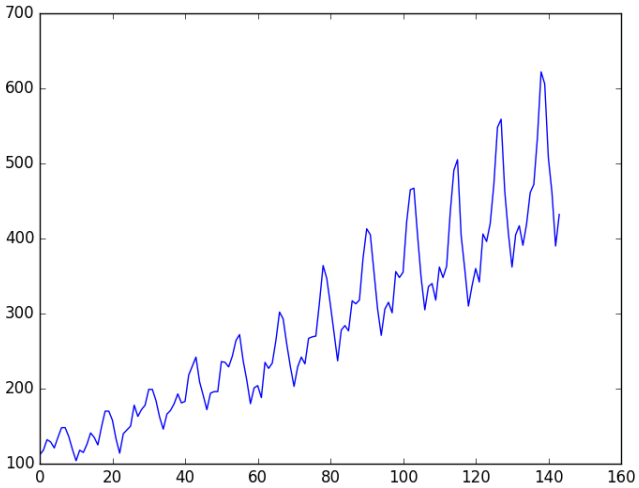


Figure 3.1: Sample data are displayed in the problem of time series prediction.

Monthly data of international airline passenger count from 1949 to 1960 [45].

3.4 Selected Models

As in the lesson, a two-layer neural network composed of fully linked Dense and LSTM layers was utilized to answer the problem. The network's LSTM layer has four hidden units that are neither too small nor too large, and the dense layer has one unit that compresses the LSTM layer's four outputs into a single prediction value output. In contrast to the values in the lesson, a look-back number, or the number of recurrent operations performed before obtaining the last time step output, was set to two. It was chosen empirically because a look-back value of two produced a better result than values of one or three for the same number of hidden units. One could wonder why the LSTM performed poorly with a look-back value of three since it is good at learning extensive patterns of data over time. It can be as a result of a lack of additional features or a poor data pattern. The network was trained using the Adam optimizer [46] with default parameters and the mean square error loss function [47].
However, a 500-epoch period was chosen while preserving a batch size of one, as opposed to the tutorial's 100-epoch period. In order to maintain input voltages into a memristive crossbar array small enough for a specified range of memristances, [Ron, Roff] it was done while having weight limits between 1 to produce almost the same performance results as in the previous point. Then, learned weights are extracted to set memristive crossbar array memristance values. The ratio of training to testing data is 2/1.
Table 3.1 below shows a summary of the models that have been chosen for each problem. It was decided to divide the 144 sample points into 142 datasets, each of which contained two samples and a single target value. Data samples were reduce in linearly to be in the ranges between [0, 1] for the selected problems. This is the ranges where LSTM performs well for the selected problems.

In figure 3.2 the block diagram describe that for the first time (start condition) starts with zero input of hidden layer h_{t-1} and next iteration h_{t1} generate. The first h_{t1} the input of the second iteration So using the hidden output which is feedback to the input and x_1 we can get the second output. That is the predicted value of the system.

Table 3.1: Frame work of selected models for the following selected problem

No	Characters and parameters	Selected problem
1	Network Config. [L1(units)+L2(units)]	LSTM (4) + dense (1)
2	Train/Test data ratio	2/1
3	The Look-back number	two
4	Dataset size (samples,features)	(3,1)
5	Total number of Datasets	142
6	Epoch size (weight extraction/analysis)	500/300
7	Batch size (weight extraction/analysis)	[1/1]

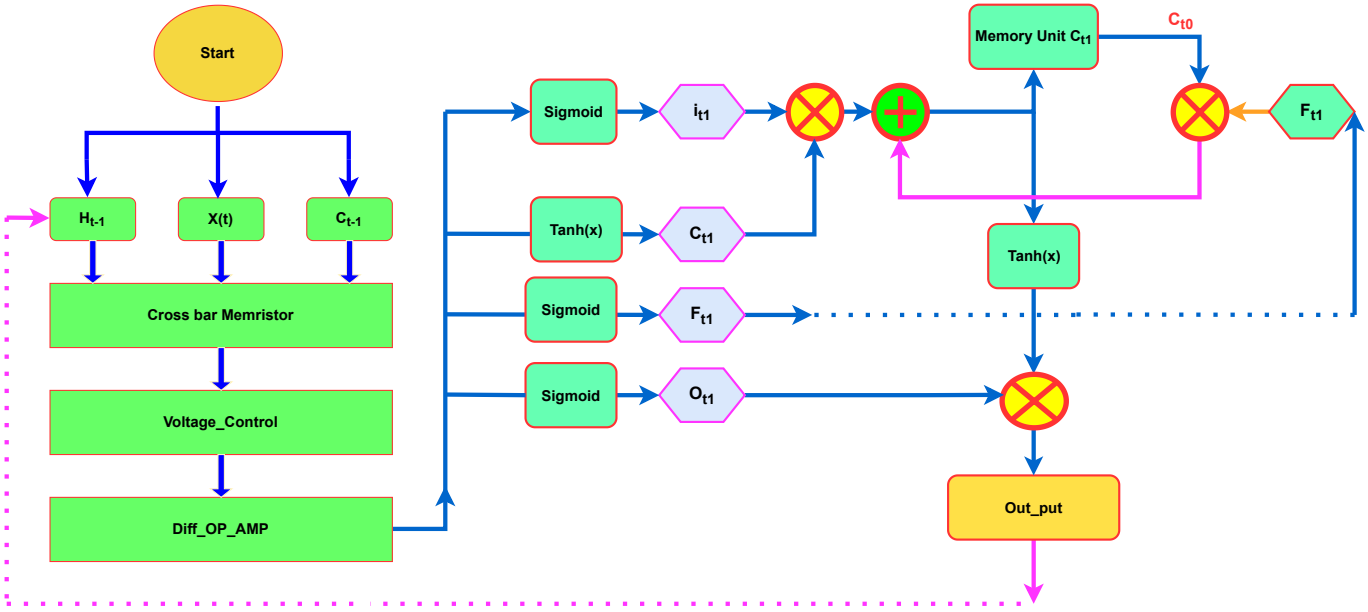


Figure 3.2: Flow chart diagram of programmable memristor crossbar LSTM

3.4.1 Two-Layer Network Method

Brownlee's time-series prediction problem is implemented using LSTM in the proposed circuit design [44]. The dataset provided shows the change in the number of international airline passengers over a 12-year period using 144 observation points.Because of the LSTM's sensitivity to input data, it was split into training and testing sets and normalised between 0 and 1. Figure 3.3 shows a diagram for the implementation of the task.

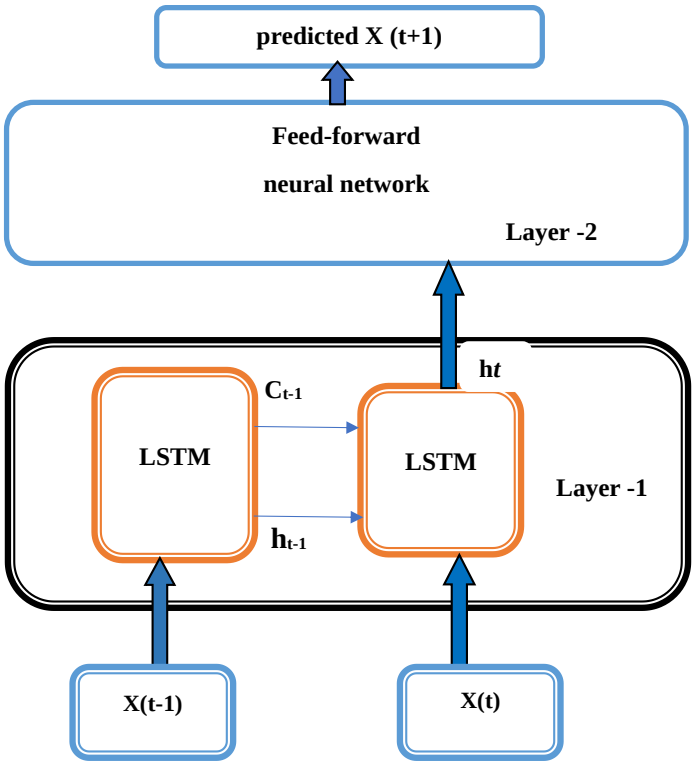


Figure 3.3: Block diagram of a time-series prediction problem using LSTM

3.5 Current-based for implementation of LSTM

The LSTM circuit design in this study is based on current-based activation function circuits, which were the first attempt to implement the LSTM algorithm in analog circuitry using memristive crossbar array [48]. These circuits use currents as inputs to implement sigmoid and hyperbolic tangent functions. Because it necessitates the usage of additional circuit types, such current-mirror [49] and voltage-to-current converters, the total design might be referred to as current-based LSTM.

Figure 3.4 depicts a memristive crossbar implementation of a single LSTM layer with N inputs and M LSTM blocks (cell). The hidden unit values (h1 to hM) are LSTM layer outputs from time step t-1, whereas the features (X1 to XN) are time step t input samples. Bi, bf, and bo are the biases of the input, forget, and output gates, respectively. The bias for the intermediate cell state Ct is bc (also known as candidate cell state). C_t is the actual cell state. Three outputs of the gates i_t, f_t, o_t as well as the intermediate cell state ct, are computed by the four structures separated by dashed blue lines in the diagram. Because the transistors in those structures act as switches, weighted summation can be performed

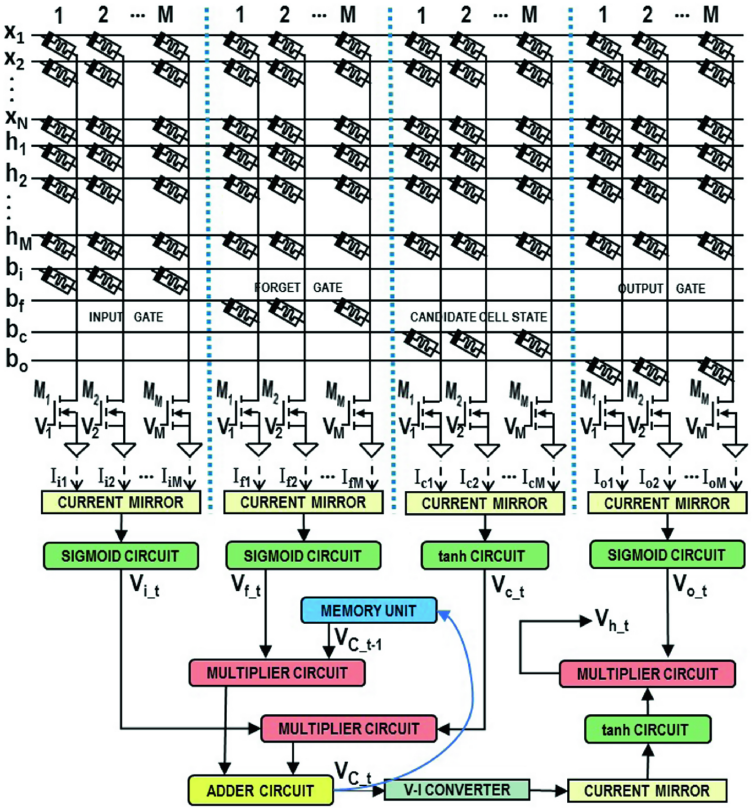


Figure 3.4: Diagram of a memristive current-based LSTM circuit.
Circuits for the activation function, current mirror, and multiplier were lifted from [48]. The values for the memristor's "R_on" and "R_off" are based on [49]. TSMC 0.18 μm CMOS technology was employed in the circuit implementation process.

only once per block. The sum of the currents going to current mirrors is the result of the values. The mirrored currents are then fed into circuits with sigmoid and hyperbolic tangent activation functions.

The activation function circuits then output the voltage values for i_t, f_t, c_t, and ot. Except for Ct1, the cell state from the previous time step, we have computed everything to obtain the cell state Ct at this point. Ct1 is read from memory, and Ct is calculated and saved in the same memory unit. The voltage Ct is then converted to current and filtered with the hyperbolic tangent function. Finally, by multiplying the filtered Ct voltage by the voltage of ot, the current time-step output ht of a current LSTM block is obtained as a voltage. To obtain all hidden unit outputs ht, however, M execution cycles are required. The sequential operation of the circuit can save on-chip area at the expense of execution speed.

3.6 Proposed Voltage-based circuit for implementation of LSTM

This section will show you how to build a voltage-based memristor LSTM circuit. Voltage-based activation function circuits and voltage buffers provide higher precision and more predictable outputs than current-based activation function circuits and current mirrors. Using current-based implementation, a classification problem could be solved. This is because we don't care about the analogue output voltage; as long as it's high or low enough, we know it's either digital 1 or digital 0. In the case of time-series prediction, we care more about the analogue output voltage being much more accurate than it being higher or lower than some threshold value. As a result, high-accuracy sigmoid and hyperbolic tangent operate circuits based on voltage were constructed. A high-accuracy four-quadrant multiplier factor circuit was also adapted from [**?**]. The steerage in obtaining correct values at each stage so as to attain an accurate output value. In addition, a sway circuit has been enforced to carry out the LSTM RNN's multiple-time step feature. statistic prediction can always produce some error, resembling mean square error (MSE) or root mean square error (RMSE). We will conclude that the LSTM neural network was with success implemented in an analog circuit if the RMSE values of the circuit and software system implementations are near enough. Figure 3.5 depicts the voltage-based memristor LSTM circuit diagram for solving the primary time-series prediction problem. The figure's final sign of the ht1 sign corresponds to its real check in the LSTM formula equations. it's performed by employing activation operate circuits that generate values. A nearer checkup on AN LSTM cell structure will be seen in Fig 3.5. The computer file of the LSTM unit may be a concatenated vector of the latest input data xt and data from a previous cell ht-1. Biases bt are accustomed establish zero inputs. The concatenated vector is multiplied by a weight matrix, and also the ensuing outputs are passed via activation functions such as sigmoid or hyperbolic tangent to form gate values. The forget gate f t output values following the sigmoid layer are around zero and 1. Hadamard multiplication of foot with previous cell state Ct-1 is used to make a decision whether or not to stay or partially/completely delete information on Ct-1 in the current cell. Similarly, by choosing whether to dam or pass a new candidate cell state Ct, the input gate output it contributes to the new cell state Ct.

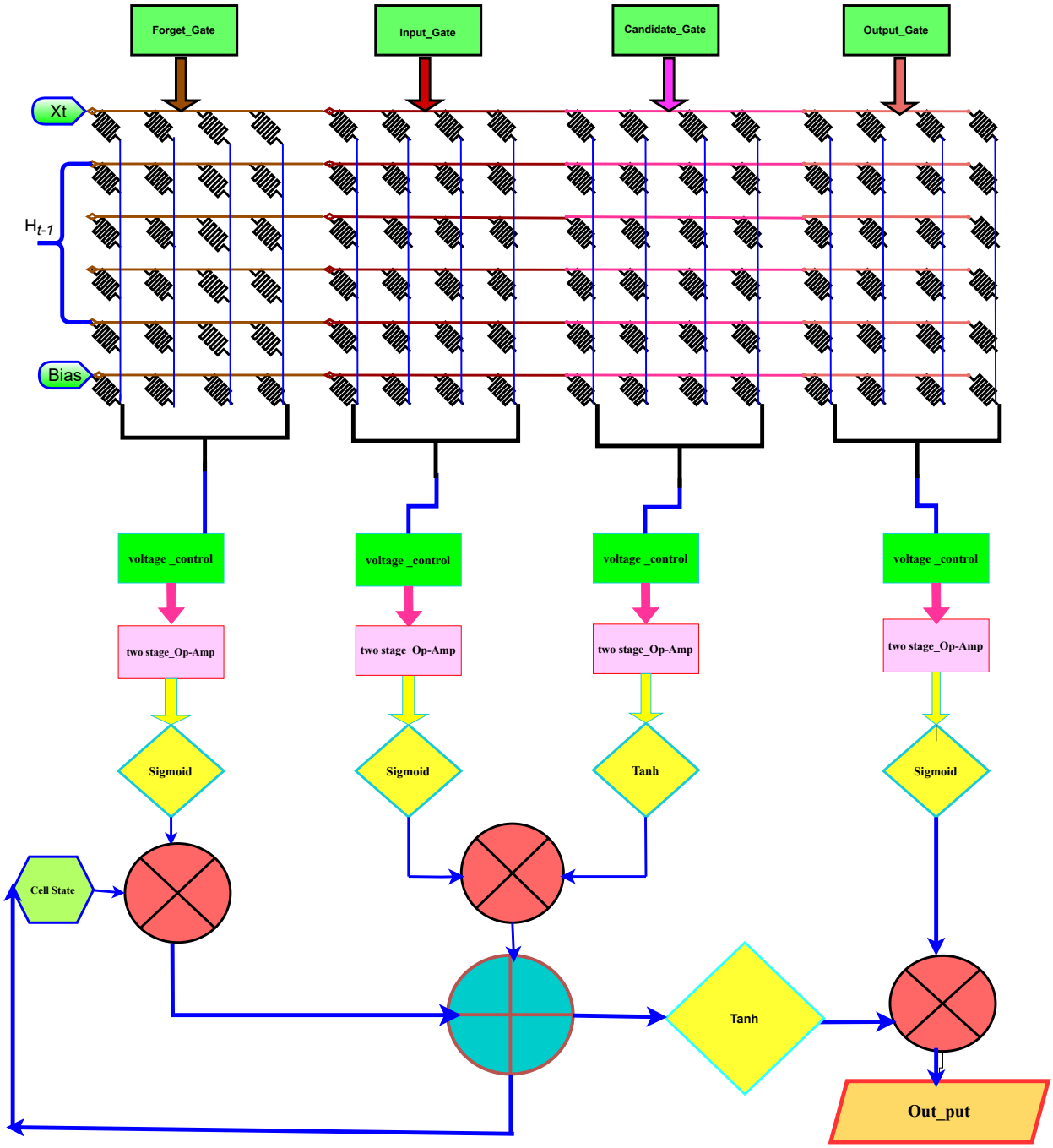


Figure 3.5: The flow chart of the time-series prediction problem is solved using neural network circuit design.
Figure 3.5: The network uses two previous time-steps, which are (x_{t1}, x_{t2}) and to predict x_{t3} which is equivalent to V_{pred}.
It means that the LSTM layer goes through two full cycles of operations before its output is captured at memory unit h_{t2} for subsequent VMM operations in the dense layer. It is not stated on the circuit, but the LSTM layer has four hidden units and the dense layer has one (no activation function). The input voltage range of the network is between [-1, 1] All the functional circuit describe in the section of circuit part and the whole circuits tested independently finally the results depend on the circuit put together.

A integration of some new and some old information results in a new candidate cell state Ct, as shown mathematically in section 2.3 and 3.4.1 Finally, in equation 2.14, output gate Ot determines how much of the filtered version of Ct forms a new cell output ht. Ct is filtered through a hyperbolic tangent function to produce values ranging from -1 to 1. A similar circuit architecture is used to solve the time-series classification problem [33]. The only differences between them are the control voltages and memory unit circuits. Figures 3.6 and 3.7 display the reminiscence unit circuits and manage voltages for the primary design, respectively. The memory units store the previous time-step outputs of the LSTM hidden units as well as the prior time-step cell state values. Because the community structure employs four hidden units, every reminiscence unit incorporates four pattern and keep circuits. The control voltages, as in a current-based LSTM architecture, ensure that each LSTM cell output is obtained sequentially. At the current time step, retrieving the outputs of all four LSTM hidden units takes 40 seconds, and forecasting the target value of a single dateset takes 100 seconds.

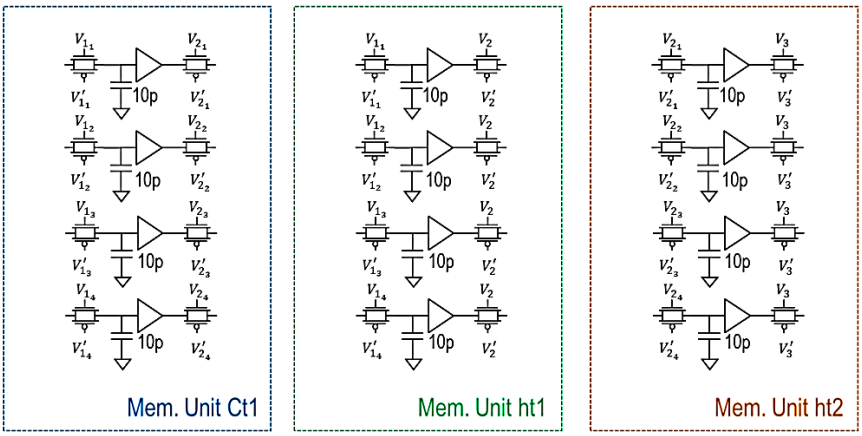


Figure 3.6: Logic circuits with pass-transistor logic and sample and hold memory units are utilized in voltage-based LSTM..
For the final design, the pass logic switches employ NMOS and PMOS transistors with a W/L ratio of 45 um/0.18 um [35].

From figure 3.6 obtaining a single prediction value x_{t3} takes 88us-100us. Max-Amplitude of the pulses is 1.8V. The control voltage blends follow the same pattern but have an amplitude of -1.8V. Voltages $V1_1$ to $V1_4$; and $V2_1$ to $V2_4$ are not describe in this graph, but they are first and second halves of signals $V12_1$ to $V12_4$,respectively.

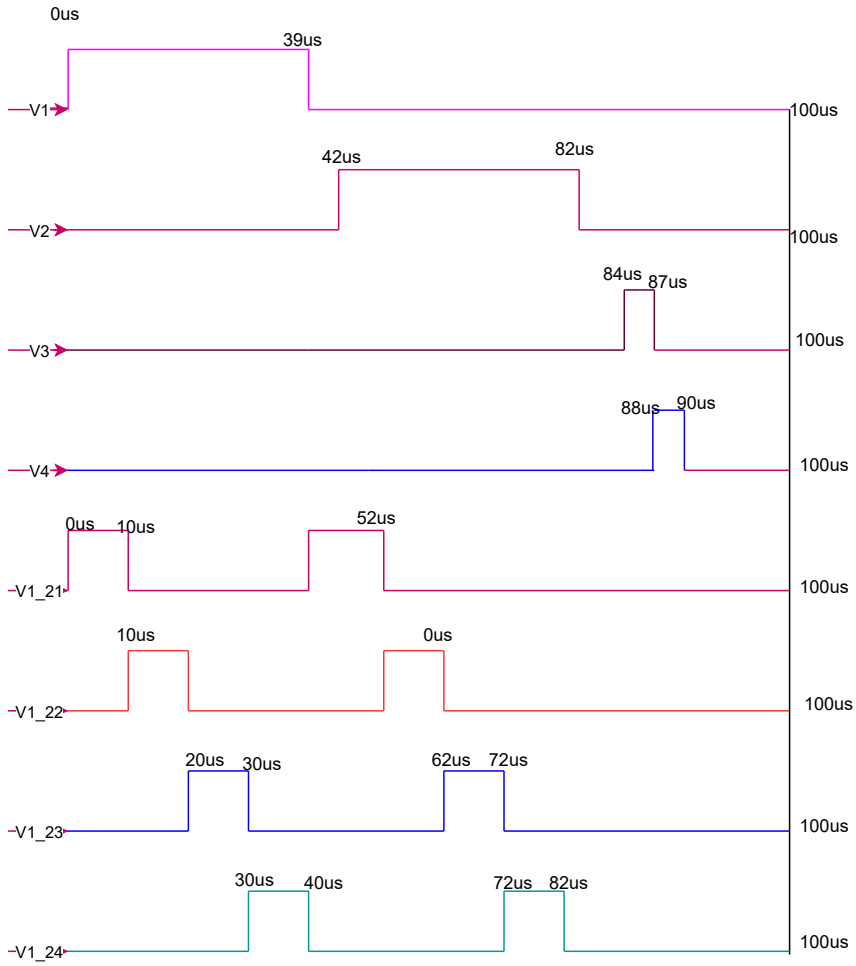


Figure 3.7: Control Voltage signals for voltage-based LSTM solving selected problem .

3.7 The Whole Circuit Section

3.7.1 Circuit For Vector Matrix Multiplication Part

The effective implementation of VMM operation is the strength of memristive crossbar circuits. Op-amps are useful once more, this time as virtual basis for accruing currents in a crossbar column. They also convert total current to voltage at the output. The use of a single memristor and op-amp per synapse and column, respectively, limits the number of weights that can be implemented. In practice, we can only use positive weights in this manner. In the memristive crossbar [50], the issue can be resolved by utilizing two memristors per synapses and two op-amps per columns.

Figure 3.8 depicts how two op-amps are set up in a crossbar. Inverting and summing amplifier functions are performed by the first column op-amp and second column ampli-

fier, respectively. The second column's current can be subtracted from the first one using this configuration. The subtracted current is then multiplied by Rf, yielding the correct voltage at the second operational amplifier's output. According to [51], the voltage-based designs 1 and 2 each use a single pair of op-amps (for a total of four). This is because the design employs sequential mode operation.

In figure 3.8 the two memristors acting as one synapse in a memristive crossbar circuit.

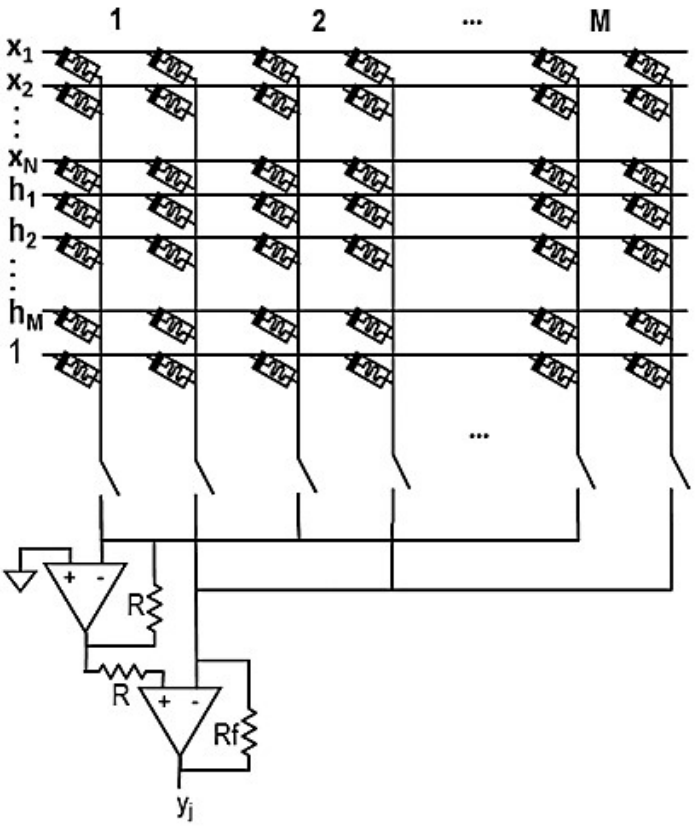


Figure 3.8: Multiplication of vectors Memristive crossbar circuit expressing a single synapses with two memristors [34].

Switches are logical elements with pass transistors. They are employed to carry out VMM operations in a sequential fashion. There would be four of these crossbars throughout the whole circuit of the voltage-based LSTM systems. At node yj, the output voltage is read. For continuous or discrete memristance states, $R = 1.25k and Rf = 1k/1.24k$.

The previous section mentioned that two types of op-amps were used in the LSTM circuit designs, that is describe in figure 3.9 and figure 3.10

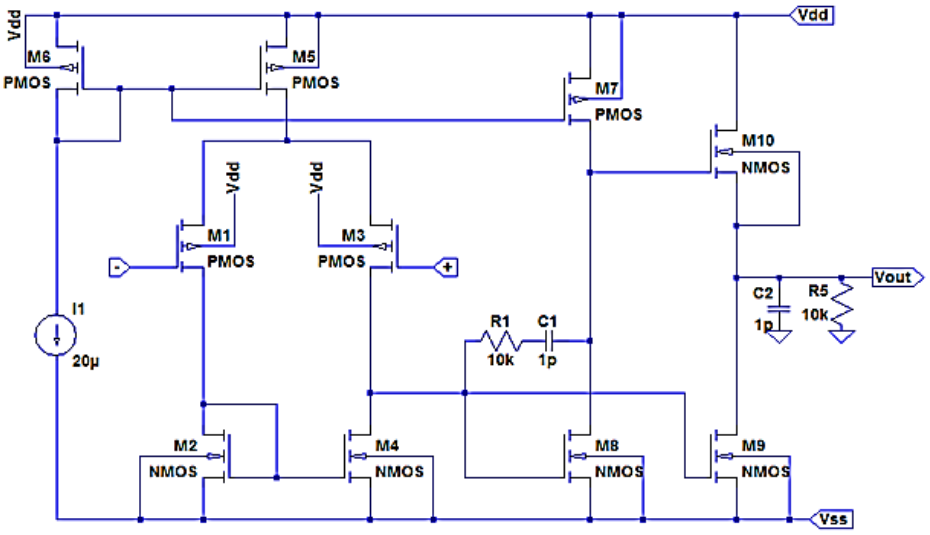


Figure 3.9: The voltage-based and current-based LSTM systems both make use of a two-stage operational-amplifier [48].

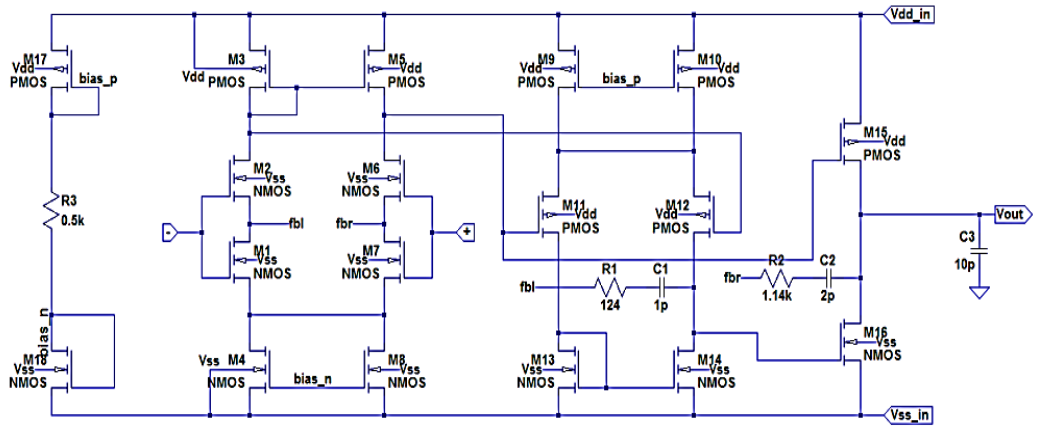


Figure 3.10: Three-stage operational-amplifier – RNIC-1 operational-amplifier based on [52].

3.7.2 Activation Function Circuit

To obtain sigmoid and hyperbolic tangent functions, utilize the circuit shown in figure 3.11. It essentially makes use of a differential amplifier's ability to improve output voltage gradually and smoothly when the differential input is swept through a particular range. The desired output range and form can be achieved by adjusting the supply voltage V_{DD}, current I_1, and the sizes of the NMOS transistors (N1 and (N2). The output values are shifted to fit the graphs of the sigmoid and hyperbolic tangent functions using the voltage source values of V_1, V_2, and V_3. The parameters stated above are different for each of these two functions since they are different. Figure 3.12 and Figure 3.13 show the DC transfer

characteristics for sigmoid and hyperbolic tangent function circuits, respectively. The input and output ranges of the graphs are scaled down by -10 to obtain the operational range of the other circuit components (the negative part is canceled later stages).

$$\left[\sigma(x) = \frac{1}{e^{-x}+1} and \quad \tanh(x) = \left(\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}\right)\right]$$

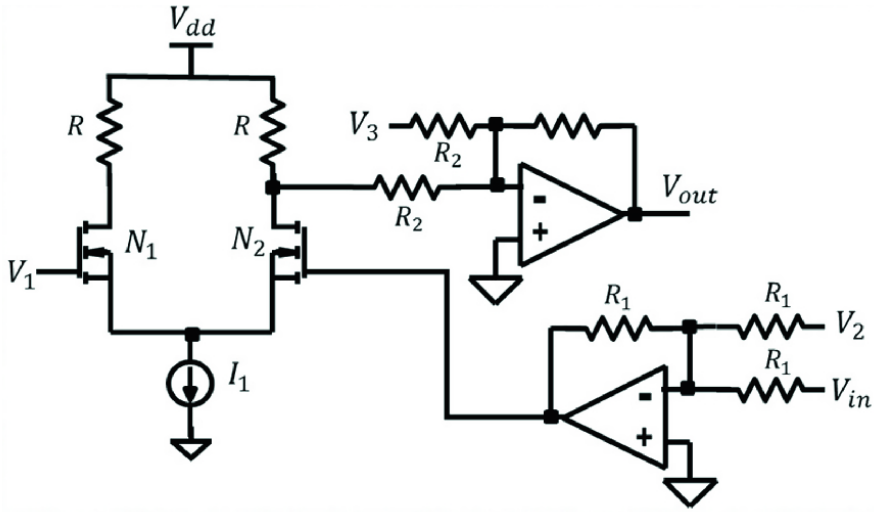


Figure 3.11: Activation function(sigimoid and tanh function) [34]

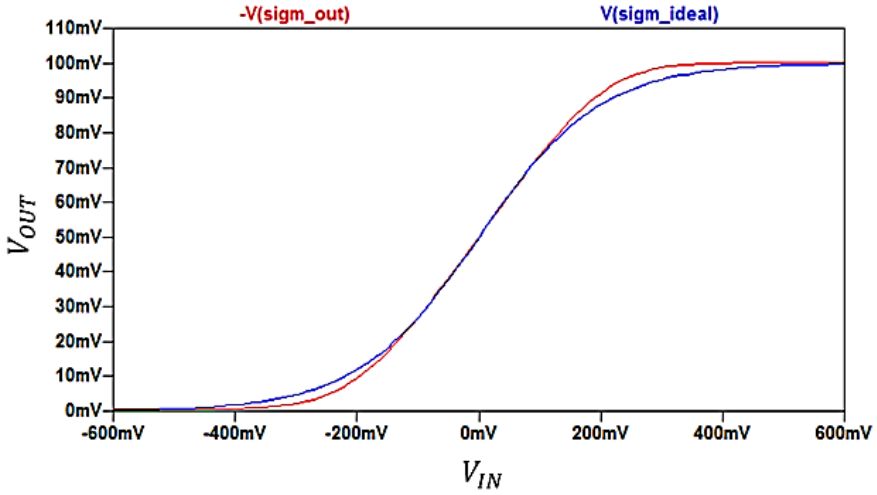


Figure 3.12: sigmoid function simulation.
Red represents circuit implementation, while blue represents software implementation. Inputs and outputs are scaled down by a factor of 10.

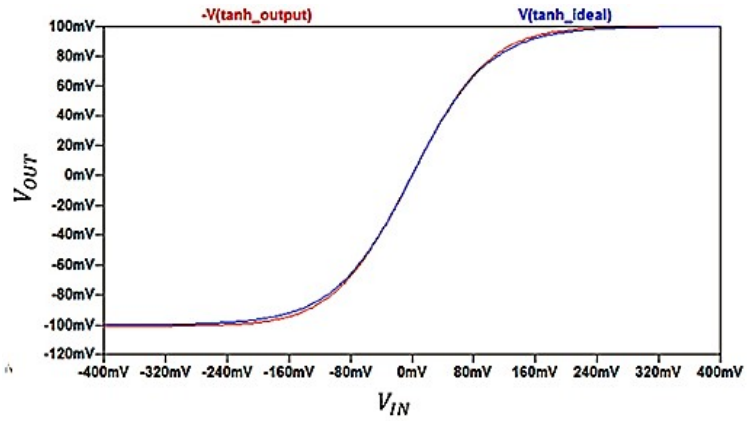


Figure 3.13: tanh function simulation.
Red represents circuit implementation, while blue represents software implementation. Inputs and outputs are scaled down by a factor of 10.

3.7.3 Analog Circuit Multiplier

Analog multiplier of four-quadrant multiplier based on Flipped Voltage Follower (FVF) was used to declar the Hadamard multiplication operation [8]. Figures 3.14 and 3.15 illustrate the FVF-based multiplier's circuit schematic and DC transfer characteristics, respectively. The core of the multiplier is made up of NMOS transistors $M_1 - M_4$. The current source I_b and the transistors M_a and M_b contain an FVF cell. The difference in currents I_E and I_F results in output current ($I_o ut = I_E - I_F = \mu_n C_{ox} W/L$). Only if VE = VF, all transistors are in triode mode, and the sources driving nodes A and B have a very low impedance is this expression true. These requirements can be met by using FVFs as current sensing elements and voltage buffers in a feed-forward structure. The bottom FVFs generate low impedance nodes E and F, allowing currents to be sensed through these nodes. The best FVFs power nodes A and B with very low impedance sources. When particularly in comparison to operational amplifer-based multipliers, the resulting multiplier has good linearity, can operate at high frequency bands, and requires little power . Figures 3.16 and 3.17 depict The latter multiplier's circuit schematic and DC transfer characteristics As shown in the figure, M1 to M6 on the left side form an analogue voltage squarer with a "symmetric complementary push-pull source follower structure," M5 and M6 transistors are available [54]. Triode transistors are used. M5 and M6 drain currents are expressed as follows:

$$I_{D5} = \beta \left[(V_{GS5} - V_{Tn})\, VDS5 - \frac{1}{2} V_{DS5}^2 \right] \tag{3.2}$$

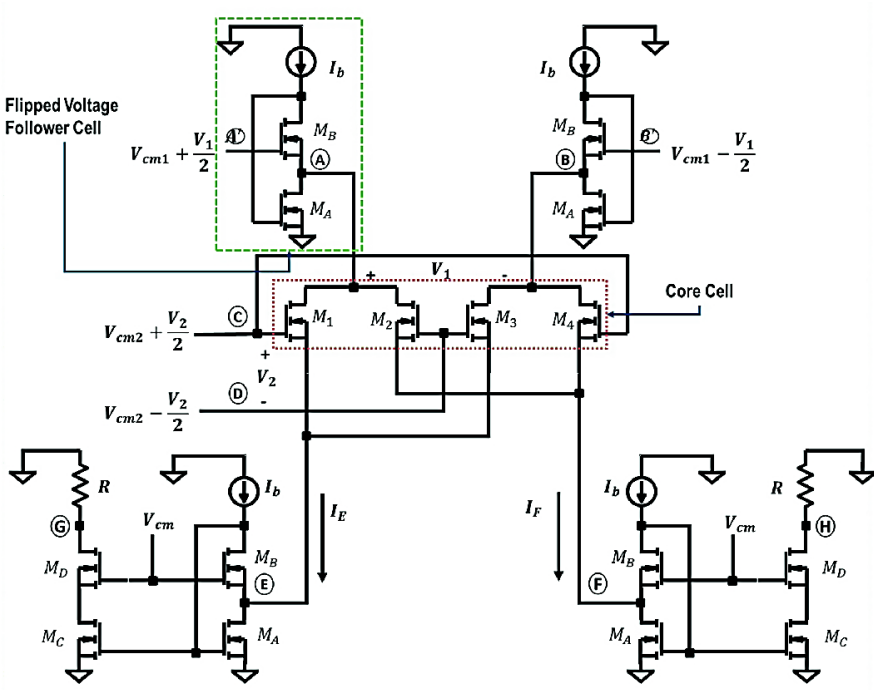


Figure 3.14: Analog four-quadrant multiplier one.
Transistors all function in a linear region. The input range is between -0.4V and 0.4V; Ib = 600uA; R = 2k; and Vcm = Vcm1 = Vcm2 = 1.4V [53].

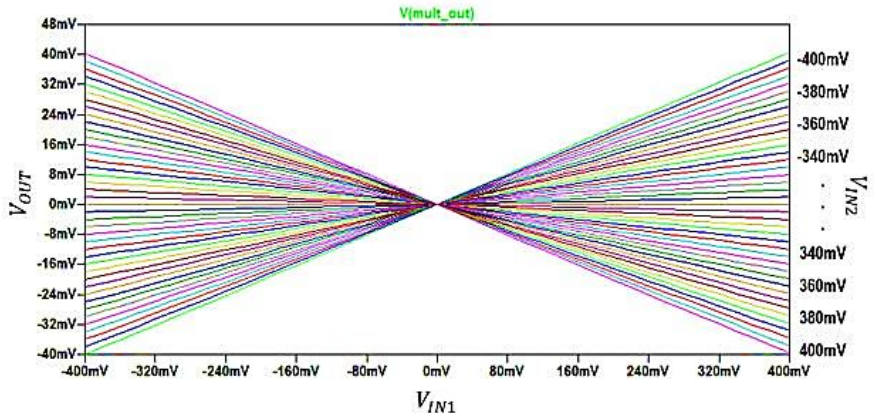


Figure 3.15: Multiplier one's DC transfer characteristics are based on Flipped Voltage Follower cells.
The multiplier circuit performs real multiplication on a scaled-down by -4 basis.

$$I_{D6} = \beta \left[(V_{GS6} - V_{Tn})\, VDS6 - \frac{1}{2} V_{DS6}^2 \right] \tag{3.3}$$

Where β=$\mu_n C_{ox} \frac{W}{L}$. The output currents in the circuit will then be as follows due to symmetry:

$$I_{o+} = I_{D5} + I_{D6} = -\beta (A + B)^2 \tag{3.4}$$

$$I_{o-} = I_{D11} + I_{D12} = -\beta (A - B)^2 \tag{3.5}$$

The difference of the above two currents will be:

$$I_O = I_{o+} - Io- = -4\beta AB$$

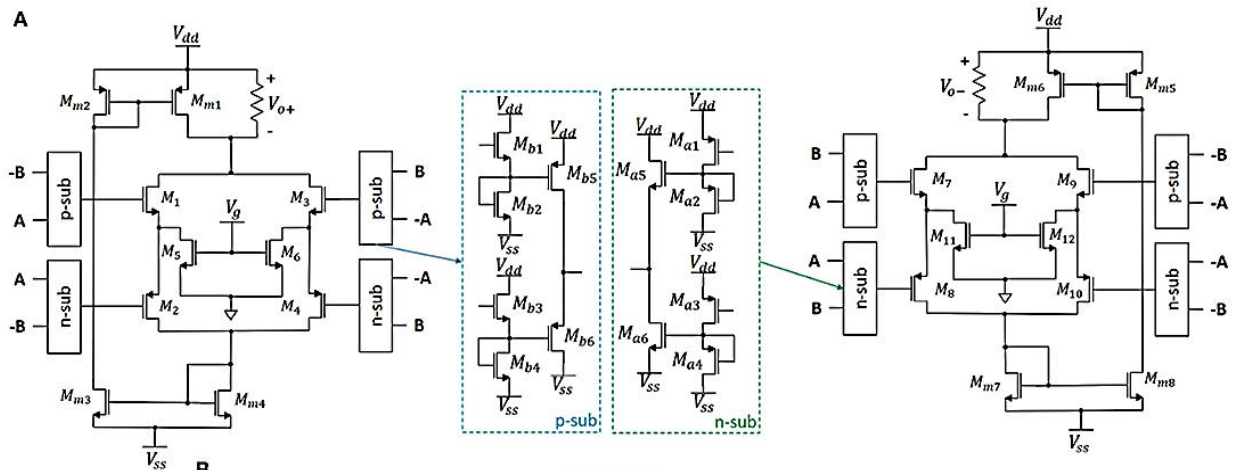


Figure 3.16: Analog four quadrants Multiplier two.
Vg = 1.5V; R = 1kΩ. The input range is between -0.5V and 0.5V, but -0.1Vand 0.1 is sufficient for LSTM implementation. [54].

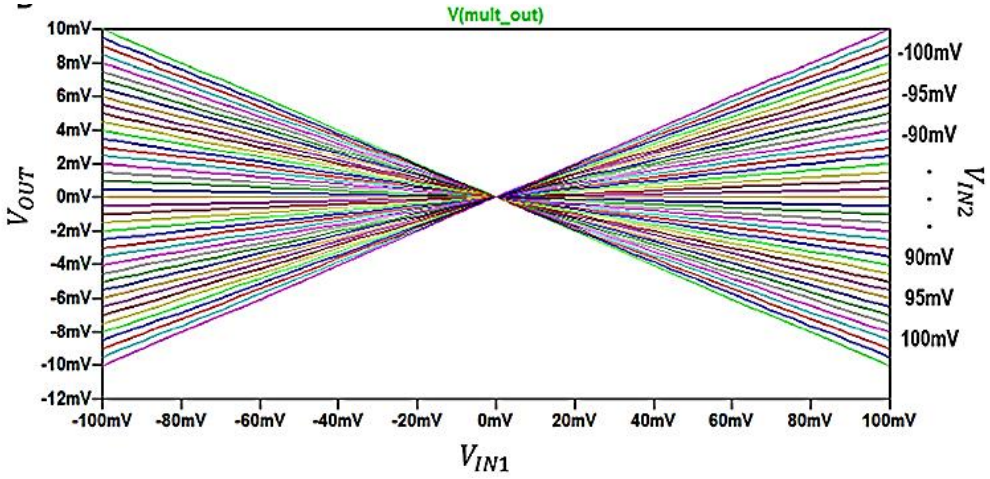


Figure 3.17: The DC acquisition of Multiplier two are based on Collinear Combined Framework squarer circuits..

3.7.4 Ad633 Multiplier

- 4-quadrant multiplication and Low cost, 8-lead SOIC, and PDIP packages 38
- Complete- no external components required and Laser-trimmed accuracy and stability
- Differential high impedance X and Y inputs and Laser-trimmed 10 V scaling reference

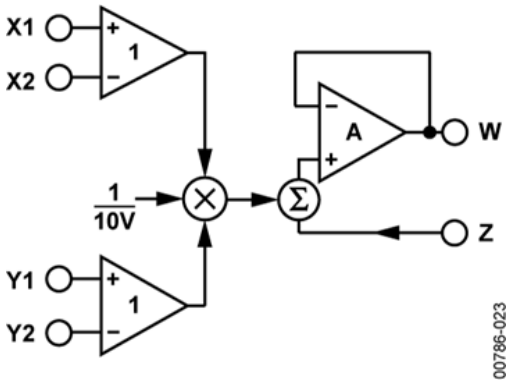


Figure 3.18: Block diagram of Ad633 four-quadrant multiplier

A device with two input ports and an output port is called an analogue multiplier. The two input signals are multiplied to produce the signal at the output. In figure 3.19 and figure 3.20, we describe the simulation of multiplier Ad633 for the two different amplitudes and frequency. So the product of the two signals gives us the multiplier of the two. the result is mentioned in figure 3.20.

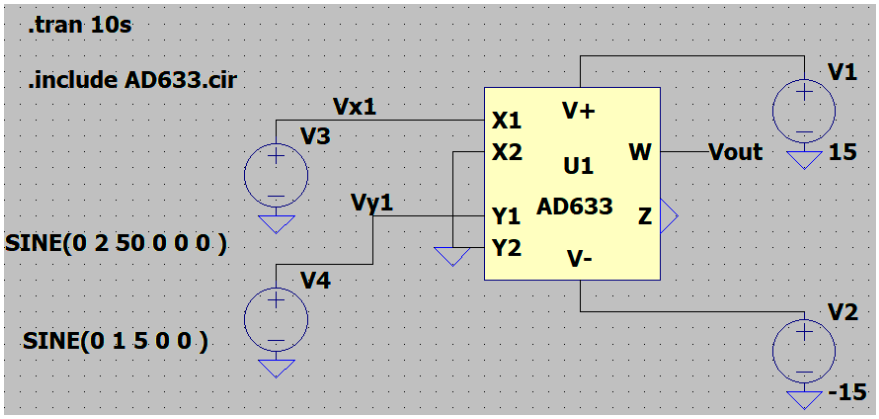


Figure 3.19: Functional circuit of Ad633 multiple two different signals

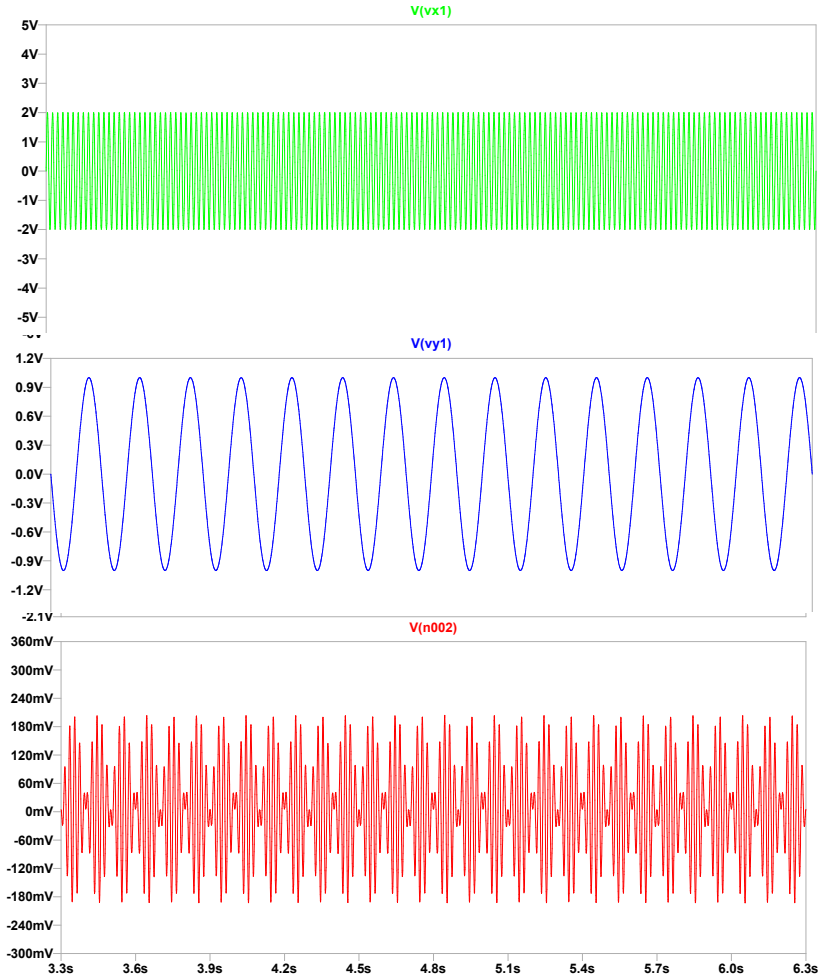


Figure 3.20: Ad633 IC multiplier simulation of two different signal

General Description of Ad633 Multiplier

A hidden zener supplies the low impedance output voltage of the four-quadrant, fully functioning multiplier Ad633 with a nominal 10 v whole scale output voltage [55]. The multiplier has been laser certified with a 2-percent overall full scale accuracy. Non linearity for the Y input is normally less than 0.1 percent, and output noise is typically less than 100 vrms in a bandwidth of 10 Hz to 10 kHz. The AD633 is useful in different of applications where the easy applicable and cost are main factors and it gives recoginized to its 1 Mega-Herthz bandwidth, 20 V/s slew rate, and capacity to drive loads. The AD633's multiplier is adaptable; it is unaffected by its simplicity, as noted by [56]. The user has access to the output buffer amplifier via the Z input, allowing them to construct a range of applications as well as combine the outputs of two or more multipliers, boost the multiplier gain, convert the output voltage to a current [57]

Chapter 4

Result and Discussion

4.1 Software Analysis

This section describes how to simulate the programmable memristor based long short-term memory (LSTM) using anaconda 3 Jupyter notebook (tensor flow) version 2.3 software and LT-Spice circuit simulator software.

In order to make package management and deployment easier, Anaconda Jupyter notebook is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.) It is also well-liked because it includes many of the machine learning and data science tools in a single install, making it ideal for quick and easy setup. Anaconda, like Virtual Environment, makes use of the concept of environments to divide various libraries and versions. The other LT-spice circuit simulation software is high-performance SPICE simulator software with a graphical schematic capture interface. Schematics can be probed to generate simulation results that can be viewed using LT-built-in spice's wave form viewer. Circuit simulation provides a critical view of the behavior of electronic circuits. Given the expense and time involved in fabricating electronic circuits, especially ICs, it's much more practical to validate circuit behavior and performance via circuit simulation before manufacturing. This two software enable the calculation of the mathematical model prepared for the time prediction and reconfigure of the crossbar memristor to arrange the time series prediction using on software.

4.2 Memristor circuit and Simulation

Memristor-based long short-term memory for time series prediction using the LT- spice circuit simulator. A memristor is an electrical component that limits or controls the flow of electrical current in a circuit while also recalling the amount of charge that has pre-

viously passed through it. Memristors are useful because they are non-volatile, which means they can store data without requiring electric source.Memristors were invented by Professor Leon Chua of the University of California, Berkeley, as a nonlinear, passive two-terminal electrical component that linked electric charge and magnetic flux. Since then, the definition of memristor has been expanded to include any type of non-volatile memory based on resistance switching, which increases current flow in one direction while decreasing current flow in the opposite direction [6].

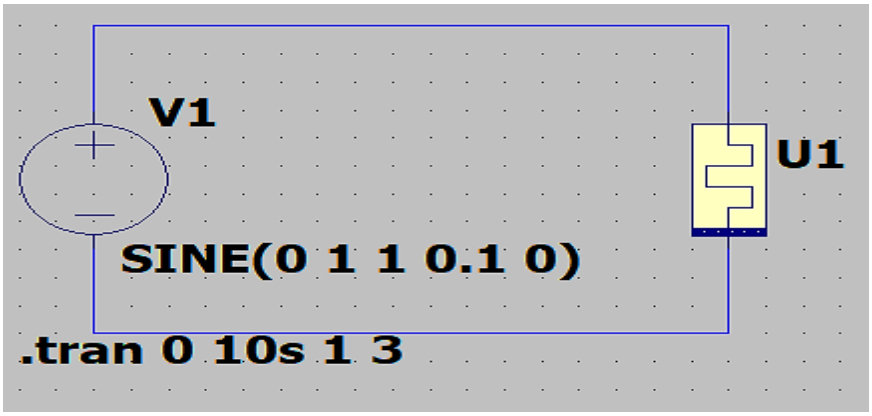


Figure 4.1: Single memristor circuit

Figure 4.1 describes as a single memristor connected with ac voltage which has frequency and memristor connected with in polarity, the device used as resistor but the device connected with reversed biased it holds the previous resistance until the next time comes with in polarity. Therefore due to that condition it creates hysteresis loop.

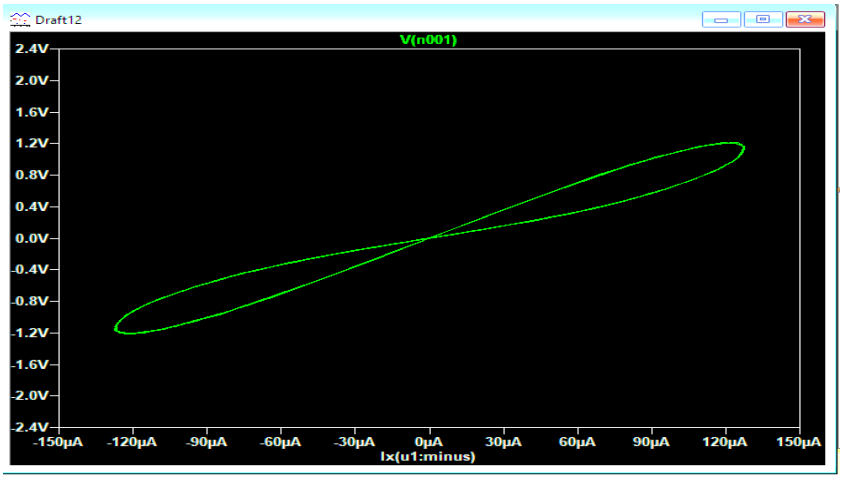


Figure 4.2: Single memristor circuit simulation

4.3 Current-based LSTM

For input voltages between 0 and 1 Volts, the single cell LSTM circuit's total power consumption is 105.901mW. There are no findings comparing circuit level and system level implementations of LSTM since this element doesn't offer complete block experiment resolving the prediction problem. Since it was clear that a voltage-based LSTM implementation would produce superior forecast accuracy, the entire circuit-level simulation was skipped. The comprehensive circuitlevel simulation of current-based LSTM utilized inside a neural network setup, however, might be considered an open problem as unmitigated comparison of the two approaches [34] .

4.4 Actual data set Result

Using the Kaggle, which is data science company, data take as experiment international airline passenger data. The original dataset contained 144 observation points collected over 12 years. From 1949 to 1960, it contains 144 sample points representing the monthly number of international airline passengers in thousands. Therefore, form the international airline passenger we used the first 45 sample points from 1949-1952 and draw the raw data using anaconda Jupyter note book below. And for better understand in this research we use the two types of graph the first is the graph drawn with in x-y coordinate chart and the second one is using the same data plot on bar graph.

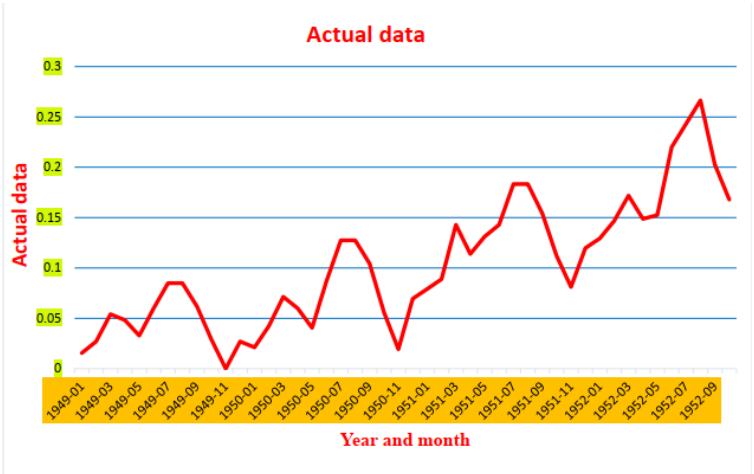


Figure 4.3: Actual data of international airline passenger graph

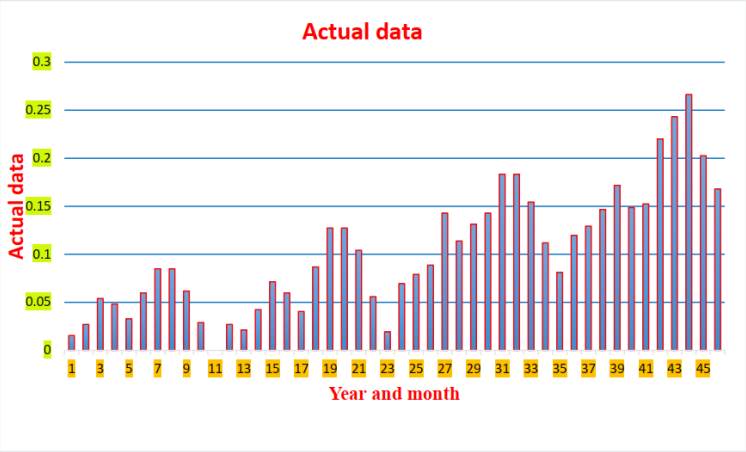


Figure 4.4: Actual data of international airline passenger bar graph

4.5 Predicted data using LSTM model on Software

In this section of the research, using the data set of international airline passengers from data science Kaggle and set the parameters, which is used for long short-term memory (LSTM) to predict the passengers that depend on the previous data or sequential time series. In general, the behavior of neural networks is determined by various factors such as network structure, learning algorithm, activation function used in each node, and so on [58–60]. Therefore, Figure 4.5 and Figure 4.6 show using the LSTM algorithms and machine learning parameters to describe the time series prediction according to the given dataset from the Kaggle dataset. using software to predict the time series problem is less complex rather than memristor crossbar but the time delay and the power consumption is a serious issue.

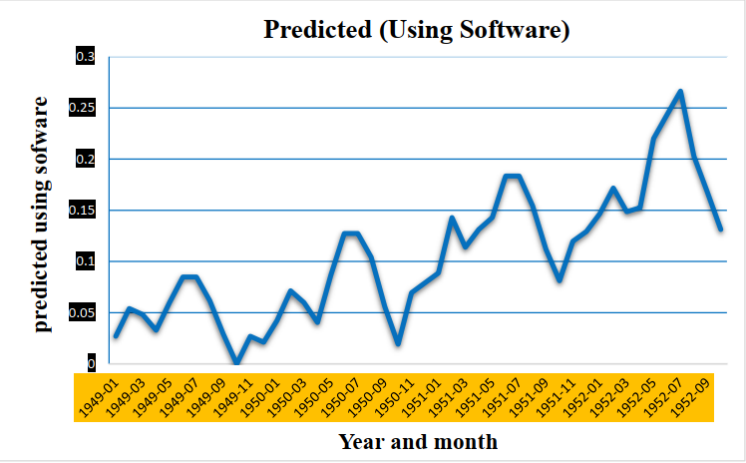


Figure 4.5: Predicted data using software graph

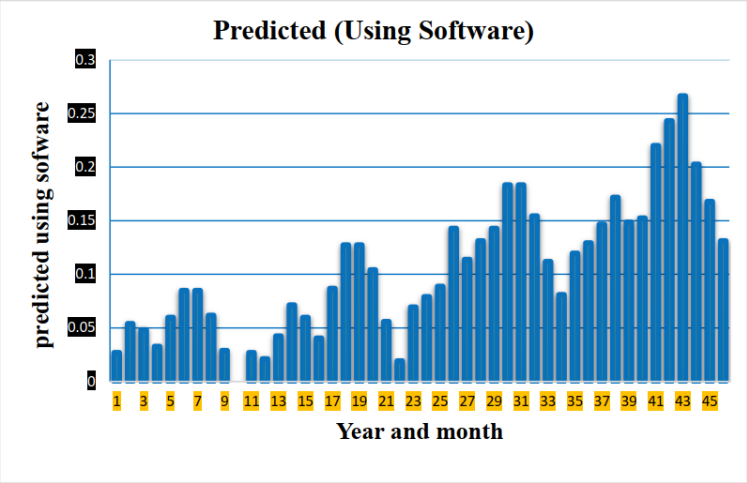


Figure 4.6: Predicted data using software bar graph

4.6 Voltage based using Memristive Analog Circuit Predicted graph

The LSTM problem was used in time-series forecasting to predict the number of airplane passengers. The original dataset included 144 observation points gathered over a 12-year period. The dataset was divided into training and testing sets before being processed by the LSTM neural network. The training set was then converted into a three-column format, with the first two columns serving as input and the third column values serving as a target. A two-layer neural network, as shown in Figure (3.3), was used to carry out the task. In the first layer, LSTM unit was unrolled for two time-steps and a many-to-one model was adopted section 3.4.1. A feedforward neural network with a linear activation function was used to create the second layer. Figure 4.7 and figure 4.7 depicts a graphical representation of predicted values based on memristors.For the same set of weight matrix values, a software-based simulation of the LSTM neural network using Jupyter Anaconda Python and circuit simulation using the LT-spice simulator. From the table 4.1 MSE and RMSE values obtained in Jupyter anaconda Python simulation were 0.0101 and 0.1005, respectively. Memristor circuit simulation produced close results with MSE = 0.0121 and RMSE= 0.1099. The two-layer neural network has a surface area of 108,599m2 (includes sample and hold memory units) Figure 3.5. A single LSTM unit can dissipate up to 225.67mW of power. Finally, the programmable memristor-based lstm is well simulated and set on the graph the results in the figure below.

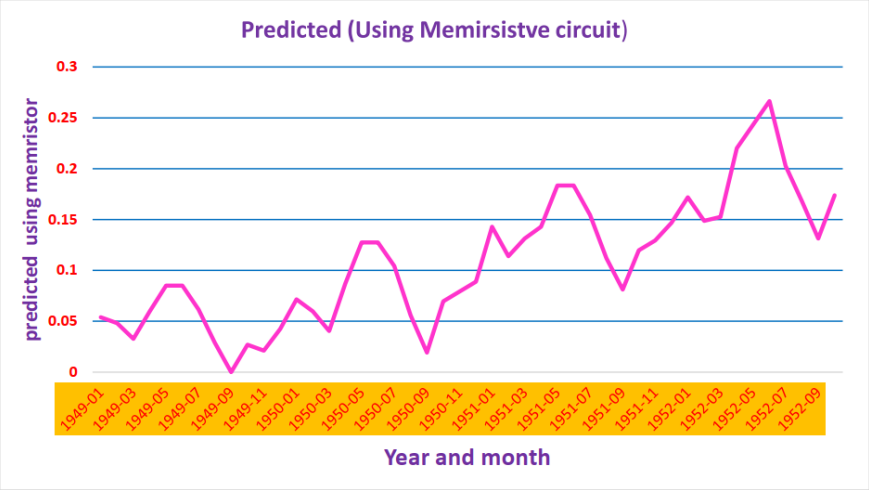


Figure 4.7: Predicted value of using memristive circuit graph

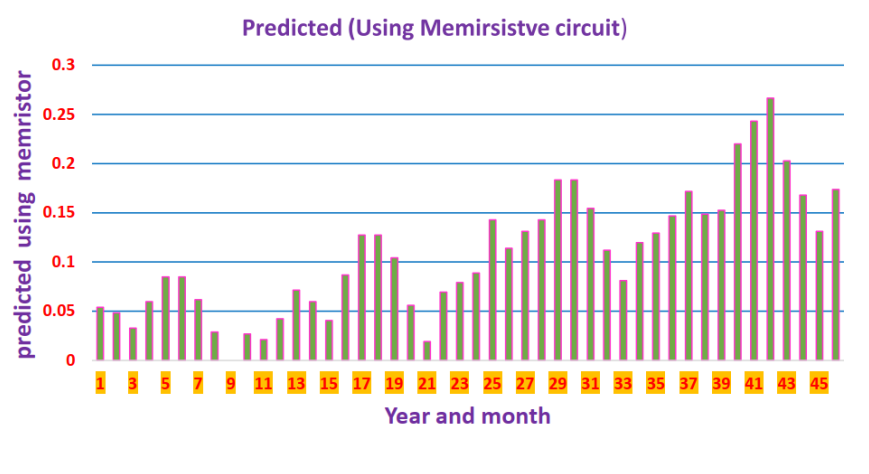


Figure 4.8: Predicted value of using memristive analog circuit and describe in bar graph

4.7 Bench mark Result

In this section we describe as the comparison of three phenomena which is, the actual, software based predicted and the programmable memristor circuit-based prediction value on the Jupyter anaconda python code. All the three environments are almost the same aim, the gap between the blue line and the pink line goes to narrow. This shows that the designed circuit and the software through increase the dense layer of LSTM using memristor configuration iteration, high learning rate on machine learning program and epoch size we get the same result due to that consider the reason of error might be decreased. As we conclude that the implementation of programmable memristor-based long short-term memory is the best approach for neural networks and valid in machine learning.

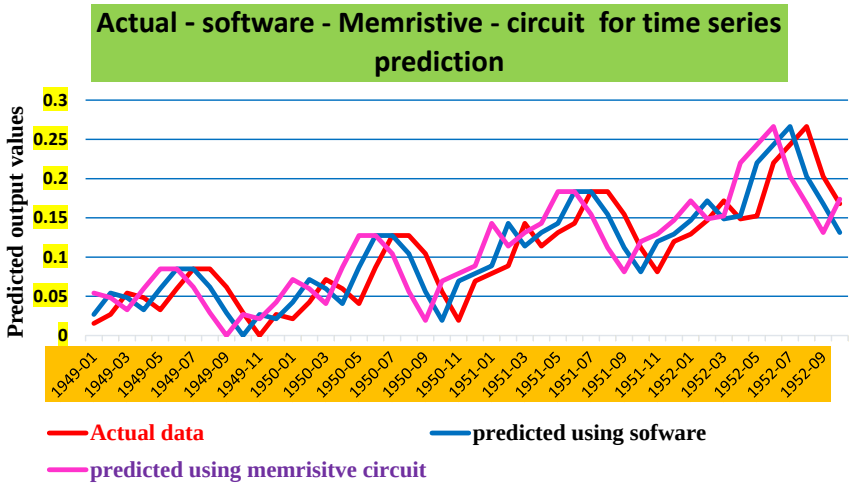


Figure 4.9: Visual comparison of the results of the analog implementation using the LSTM software and memristive (continuous weights) with the target values

4.8 Evaluation Metrics

Four popular evaluation metrics were used in this study to assess the applicability and performance of the prediction model. Prediction algorithms can be affected by a variety of factors, including information quality, datasets class distribution, and the number of instances. The evaluation of the model's accuracy is a critical step in any machine learning model. In regression analysis, the Mean Squared Error, Mean Absolute Error, Root Mean Squared Error, and R-Squared or Coefficient of determination metrics are used to evaluate the model's performance. As a result, the prediction model's performance is evaluated using the four metrics:

4.8.1 Mean Square Error (MSE)

Mean Squared Error (MSE) is a popular metric for evaluating regression problems. Mean Squared Error is the average of the squared difference between the data set's original and predicted values. It calculates the residual variance. This value is represented as a Mean squared error in the formula below, which reflects the average value of squared error per sample.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.1}$$

4.8.2 Root Mean Square Error (RMSE)

The square root of the mean of the squares of all errors is the Root Mean Squared Error (RMSE). The use of RMSE is very common, and it is considered an excellent general-purpose error metric for numerical predictions. RMSE measures the difference between sample predicted and observed (actual) price value. RMSE is calculated using an equation.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.2}$$

4.8.3 Mean Absolute Error (MAE)

Different errors are not weighted differently in Mean Absolute Error (MAE), but the scores increase linearly as the number of errors increases. The MAE score is calculated by averaging the absolute error values. The Absolute is a mathematical function that turns a negative number into a positive number. As a result, the difference between an expected and predicted value can be positive or negative, but it must be positive when calculating the MAE.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|^2 \tag{4.3}$$

4.8.4 R-Squared or Coefficient of determination (R^2)

The coefficient of determination, also known as R-squared, represents the proportion of the variance in the dependent variable that the linear regression model explains. It is a scale-free score, which means that regardless of whether the values are small or large, the value of R square will be less than one.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} \tag{4.4}$$

4.9 Comparison of Op-amp and Multiplier in Voltage-based LSTM

Table 4.1, shows and compares the results with continuous weight values to check the performance of evaluation metrics. Using the two-stage opamps in figure 3.9 [48] and in the figure 3.14 Flipped Voltage Follower multipliers [52]. It is worth noting that the when assessing thresholds using a software implementation of the algorithm, the efficiency results in Table 4.1 are insufficient. An R2 score of 0.41 indicates that the prediction is poor. Higher scores would almost certainly result from more hidden units and LSTM layers. This was not, however, the project's primary concern. The primary goal was to show that the algorithm could be fully applied in analogue hardware using memristor devices. arrays of crossbars So far, this goal has been met with a prediction ranking of R2 of 0.932 to some extent.

Figure 3.10 and Figure 3.16, compares with the result of two-stage and FVF multiplier, and also for both discrete and continuous while using values. Furthermore, for more accurate results three-stage RNIC-1 op-amps [52] and a multiplier based on a symmetric complementary structure [54].

Table 4.1: Evaluation Metrics Result using two stage-op amp and FVF

Performance	Software to Target	Analog to Target	Analog to Software
MSE	0.0101	0.0121	0.00093553
RMSE	0.10052	0.10999	0.030586
MAE	0.079965	0.086229	0.020943
R squared	0.40954	0.3453954	0.931968

Table 4.2: Evaluation Metrics Result using multiplier two,RNIC (three-stage op-amp) and Ad633 multiplier

Performance	Software to Target	Analog to Target	Analog to Software
MSE	0.0101053	0.00915332	6.7539e-0.5
RMSE	0.1005234	0.0956734	0.0082182
MAE	0.079965	0.07734	0.0063793
R squared	0.40954	0.47551	0.99519

In table 4.2 displays the numerical comparison of plots once moreThe table shows that new circuit units have helped within the much excellent circuit level implementation of the complete neural network, that consists of associate degree LSTM layer and a Dense layer.

When using more accurate circuit elements, such as operational amplifiers and multipliers, the R2 score increases from 0.9321 to 0.9951. This is the appropriate score that indicates that real memristor can provide good implementation scores. But when noise in the memristors utilized in the circuit crossbar was ignored, the final score was reached.

4.10 Power consumption and Circuit design size

Table 4.33 shows the first deployments of the LSTM on the current basis, and the second and third coincide to the LSTM network configuration developed for solving the time - series prediction for the chosen issue.

Table 4.3: Data on power for various voltage range LSTM circuit implementations

No	LSTM circuit implementation	Power in (mW)	Input range (0-1V)
1	current-based [34]	105.9	[0,1]
2	voltage- based on two-stage	225.67	[-1,1]
3	voltage- based on three stage	228.11	[-0.1,0.1]

Table 4.4: Power data for the two types of op-amps in config..

stages	Power in (mW) at -0.1 V input	Power in (mW) at 0.1 Vinput	Power in (mW) at -1 Vinput	Power in (mW) at 1 V input
two-stage op-amp	3.111	3.042	3.5093	2.5789
three-stage	3.591	5.540	0.75	10.232

Furthermore, the two-stage op-amp uses an ideal current source for biassing, whereas the three-stage op-amp uses two MOSFETs and a resistor. The supply voltage rails for the two-stage op-amp are (-1.8V, 1.8V) and for the three-stage op-amp are (-1V, 1.8V). The three-stage amplifier, on the other hand, performs well and works well with low-signal voltage values. This precision helps LSTM circuits generate precise intermediate voltage values, which leads to precise final voltage values. In particular, the overall accuracy of

a circuit-level LSTM versus a system-level LSTM is decided by inverting, scaling up or down, summation, and multiplication of voltage signals. The FVF of the cell-based four-

Table 4.5: Area and Power statistics for the three types of multiplier in buffer configuration

Analog Multipliers	Power in (mW) for max inputs of (-0.4,0.4) Volts	R^2 of LSTM + Dense circuit
Multiplier one	35.363	0.93205
Multiplier two	30.123	0.99504
Multiplier three (Ad633)	500	0.99610

quadrant analogue multiplier is the first multiplier in Table 4.5. The second multiplier is built with squarer circuits that have symmetric complementary structures. To generate a single-ended output, the first employs two-stage operational amplifiers for interface circuitry and output extension. The first employed three-stage op-amps for the exact same reason as the second. For comparison, the input voltages in the table were set to the same values of (-0.4, 0.4) Volts. Although it can operate between [-0.5, 0.5] Volts, multiplier two is used in the LSTM circuit implementation with an input voltage range of [-0.1, 0.1]. Multiplier two uses 29.05 mW of power for maximum inputs of (-0.1, 0.1) volts. When multiplier two is used inside the complete neural network circuit of LSTM and dense layers, it is smaller in size, uses less energy and contributes to a high R - squared score. As a result, it is an excellent candidate for future circuit designs beyond LSTM circuit implementation.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

Different researches are conducted in the area of memristor and long short term memory-based time series prediction. A functional circuit-level implementation of programmable memristor on a neural network with an LSTM layer was performed in this study. A number of problems involving time series prediction were resolved using the neural network. As a result, for each problem, custom circuits were designed and tested in SPICE simulation platform using TSMC 0.18 um technique. Furthermore, extensive research on machine learning programs and LSTM architectures was carried out using Kera's library, which was expanded in this work.

Each state will have some noise, and as the noise level increases, so does the prediction accuracy in analog hardware. However, if the circuit was trained, it might be able to keep running epochs until the states, including their noises, contributed to the performance of the circuit's algorithm. Furthermore, time-series predictions are not great and allow for some error tolerance due to real-world circuit errors. Due to the system's flexibility, there is a trade-off between computation time, silicon area, and energy usage, and the final circuit design will vary depending on the application. In that case, it would be beneficial if the base circuit could be easily transformed to meet the requirements of a specific application. In this case, the memristor voltage-based LSTM design comes to solve the problem.

5.2 Recommendation

This thesis focused on using different configurations of memristor for time series prediction. In addition to studying other type of memristor network configuration, you can modify the configuration and minimize the power consumption of the topology.
Furthermore, other configurations for carrying out different types of applications are possible.

- We recommend that you use this research to do various neural network-based topics such as image processing, voice recognition, and other machine learning-based algorithms by using different memristor configurations and adjusting the methods.
- We further recommend that the world always goes to the least expensive, fantastic technology and artificial intelligence are in fashion, so using memristor technology reduces our work load due to speed and power consumption.
- As a result, the memristor component is a new trend, or the memristor element is a new element, and we can replace other electrical components with the memristor component.

REFERENCES

[1] M. Riordan, "The lost history of the transistor," *IEEE spectrum*, vol. 41, no. 5, pp. 44–46, 2004.

[2] W. Brinkman, D. Haggan, and W. Troutman, "A history of the invention of the transistor and where it will lead us," *Solid-State Circuits, IEEE Journal of*, vol. SC-32, pp. 1858 – 1865, 01 1998.

[3] T. Mimura, "The early history of the high electron mobility transistor (hemt)," *IEEE Transactions on microwave theory and techniques*, vol. 50, no. 3, pp. 780–782, 2002.

[4] S. Kvatinsky, *Memristor-based circuits and architectures*. Technion-Israel Institute of Technology, Faculty of Electrical Engineering, 2014.

[5] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[6] S. M. Chua, Leon O Kang, "Memristive devices and systems," *Proceedings of the IEEE*, vol. 64, no. 2, pp. 209–223, 1976.

[7] Kim, G. Kuk-Hwan, W. Siddharth, C.-A. Dana, T. Jose M Hussain, N. Srinivasa, and W. Lu, "A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications," *Nano letters*, vol. 12, no. 1, pp. 389–395, 2012.

[8] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, and S. Kvatinsky, "Memristor-based multilayer neural networks with online gradient descent training," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 10, pp. 2408–2421, 2015.

[9] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "Stdp and stdp variations with memristors for spiking neuromorphic learning systems," *Frontiers in neuroscience*, vol. 7, p. 2, 2013.

[10] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based cmos/nano circuits," in *2009 European Conference on Circuit Theory and Design*, pp. 563–566, IEEE, 2009.

[11] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[12] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: properties of basic electrical circuits," *European Journal of physics*, vol. 30, no. 4, p. 661, 2009.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4, pp. 2047–2052, IEEE, 2005.

[15] A. Graves, "Supervised sequence labelling," in *Supervised sequence labelling with recurrent neural networks*, pp. 5–13, Springer, 2012.

[16] H. A. Yildiz, M. Altun, A. D. Gungordu, and M. R. Stan, "Analog neural network based on memristor crossbar arrays," pp. 358–361, 2019.

[17] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology*, vol. 24, no. 38, p. 384010, 2013.

[18] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola, *et al.*, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, no. 1, pp. 89–124, 2017.

[19] A. Huang, X. Zhang, R. Li, and Y. Chi, "Memristor neural network design," *Memristor and Memristive Neural Networks*, pp. 1–35, 2018.

[20] B. Liu, *Neuromorphic system design and application*. PhD thesis, University of Pittsburgh, 2016.

[21] K. Steinbuch, "Die lernmatrix," *Kybernetik*, vol. 1, no. 1, pp. 36–45, 1961.

[22] Z. Li and R. Bock, "Replication of bacterial plasmids in the nucleus of the red alga porphyridium purpureum," *Nature communications*, vol. 9, pp. 1–8, 2018.

[23] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nature electronics*, vol. 1, pp. 52–59, 2018.

[24] M. Hu, H. Li, Q. Wu, G. S. Rose, and Y. Chen, "Memristor crossbar based hardware realization of bsb recall function," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, IEEE, 2012.

[25] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[26] H. Li, X. Chen, Z. Guo, J. Xu, Y. Shen, and X. Gao, "Data-driven peer-to-peer blockchain framework for water consumption management," *Peer-to-Peer Networking and Applications*, vol. 14, 09 2021.

[27] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[28] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[29] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

[30] K. Chen, L. Huang, M. Li, X. Zeng, and Y. Fan, "A compact and configurable long short-term memory neural network hardware architecture," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 4168–4172, IEEE, 2018.

[31] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5934–5938, IEEE, 2018.

[32] Y. Zhang, C. Wang, L. Gong, Y. Lu, F. Sun, C. Xu, X. Li, and X. Zhou, "Implementation and optimization of the accelerator based on fpga hardware for lstm network," in *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pp. 614–621, IEEE, 2017.

[33] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[34] K. Smagulova, K. Adam, O. Krestinskaya, and A. P. James, "Design of cmos-memristor circuits for lstm architecture," in *2018 IEEE international conference on electron devices and solid state circuits (EDSSC)*, pp. 1–2, IEEE, 2018.

[35] K. Smagulova, O. Krestinskaya, and A. P. James, "A memristor-based long short term memory circuit," *Analog Integrated Circuits and Signal Processing*, vol. 95, no. 3, pp. 467–472, 2018.

[36] T. Gokmen, M. J. Rasch, and W. Haensch, "Training lstm networks with resistive cross-point devices," *Frontiers in neuroscience*, p. 745, 2018.

[37] S. Yu, Y. Wu, and H.-S. P. Wong, "Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory," *Applied Physics Letters*, vol. 98, no. 10, pp. 103–514, 2011.

[38] G. Medeiros-Ribeiro, F. Perner, R. Carter, H. Abdalla, M. D. Pickett, and R. S. Williams, "Lognormal switching times for titanium dioxide bipolar memristors: origin and resolution," *Nanotechnology*, vol. 22, no. 9, p. 095702, 2011.

[39] F. Alibart, E. Zamanidoost, and D. B. Strukov, "Pattern classification by memristive crossbar circuits using ex situ and in situ training," *Nature communications*, vol. 4, no. 1, pp. 1–7, 2013.

[40] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.

[41] R. St. Amant, A. Yazdanbakhsh, J. Park, B. Thwaites, H. Esmaeilzadeh, A. Hassibi, L. Ceze, and D. Burger, "General-purpose code acceleration with limited-precision analog computation," *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 505–516, 2014.

[42] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, and S. Kvatinsky, "Memristor-based multilayer neural networks with online gradient descent training," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 10, pp. 2408–2421, 2015.

[43] B. Li, Y. Wang, Y. Wang, Y. Chen, and H. Yang, "Training itself: Mixed-signal training acceleration for memristor-based neural network," pp. 361–366, 2014.

[44] J. Brownlee, "Time series prediction with lstm recurrent neural networks in python with keras," *Machine Learning Mastery*, vol. 18, 2016.

[45] A. I. McLeod, K. W. Hipel, and W. C. Lennox, "Advances in box-jenkins modeling: 2. applications," *Water Resources Research*, vol. 13, no. 3, pp. 577–586, 1977.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] K. Adam, "Lstm neural network implementation using memristive crossbar circuits and its various topologies," 2018.

[48] O. Krestinskaya, K. N. Salama, and A. P. James, "Learning in memristive neural network architectures using analog backpropagation circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 719–732, 2018.

[49] S. Xiao, X. Xie, S. Wen, Z. Zeng, T. Huang, and J. Jiang, "Gst-memristor-based online learning neural networks," *Neurocomputing*, vol. 272, pp. 677–682, 2018.

[50] R. Hasan, T. M. Taha, and C. Yakopcic, "On-chip training of memristor based deep neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3527–3534, IEEE, 2017.

[51] K. Adam, K. Smagulova, and A. P. James, "Memristive lstm network hardware architecture for time-series predictive modeling problems," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 459–462, IEEE, 2018.

[52] V. Saxena and R. J. Baker, "Indirect compensation techniques for three-stage fully-differential op-amps," in *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*, pp. 588–591, IEEE, 2010.

[53] J. Ramirez-Angulo, S. Thoutam, A. Lopez-Martin, and R. Carvajal, "Low-voltage cmos analog four quadrant multiplier based on flipped voltage followers," in *2004 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, pp. I–681, IEEE, 2004.

[54] S. C. Li, "A symmetric complementary structure for rf cmos analog squarer and four-quadrant analog multiplier," *Analog Integrated Circuits and Signal Processing*, vol. 23, no. 2, pp. 103–115, 2000.

[55] A. Devices, "Low cost analog multiplier ad633," *EN: Special Linear Manual Reference*, pp. 2–47, 1992.

[56] A. Devices, "Ad633 low cost analog multiplier, rev," 2015.

[57] A. Devices and M. Norwood, "Usa. datasheet, ad633, low cost analog multiplier," 2011.

[58] G. S. d. S. Gomes and T. B. Ludermir, "Optimization of the weights and asymmetric activation function family of neural network for time series forecasting," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6438–6446, 2013.

[59] W. Duch and N. Jankowski, "Survey of neural transfer functions," *Neural computing surveys*, vol. 2, no. 1, pp. 163–212, 1999.

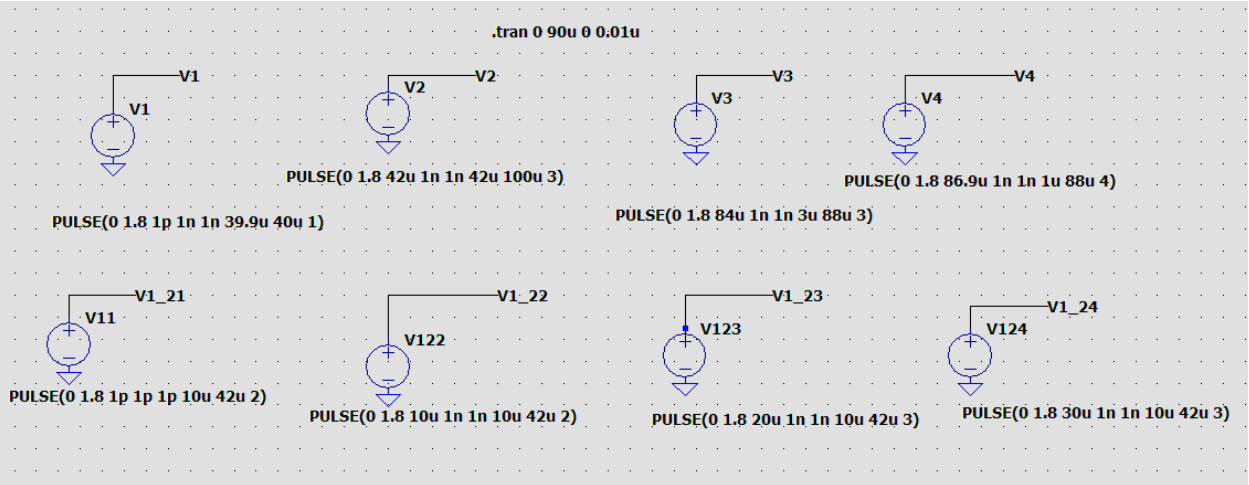
[60] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

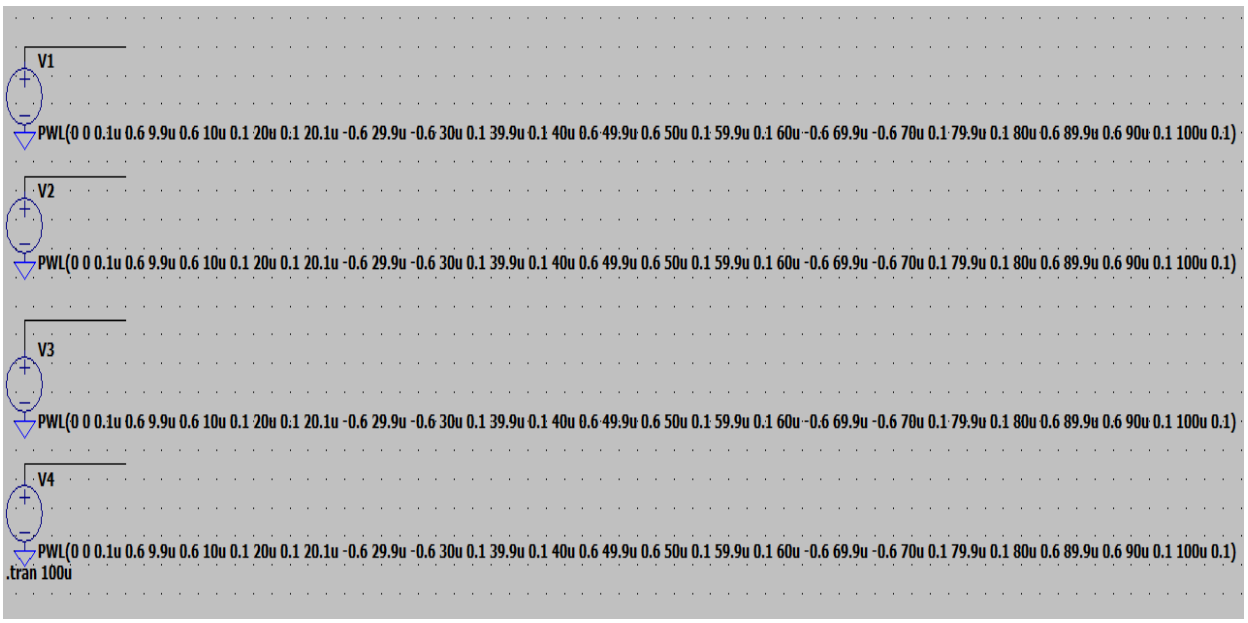
APPENDIX

Remaining circuits

Voltage Control

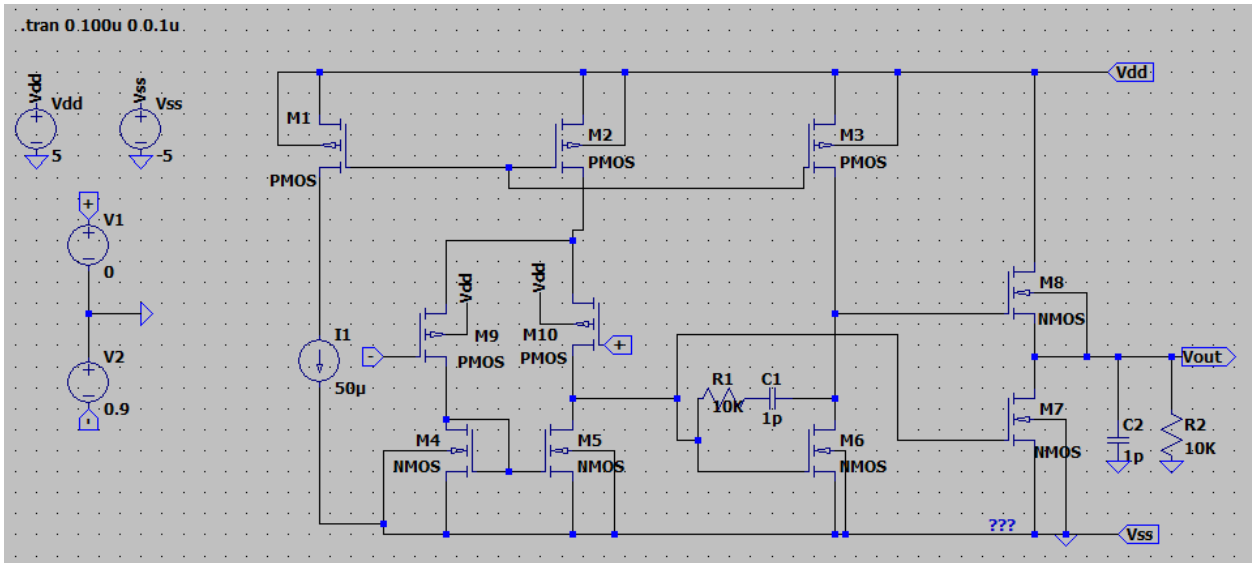
Function of voltage already mention on the topic, it controls the out put of voltage that change fro current to voltage and feed to the system.





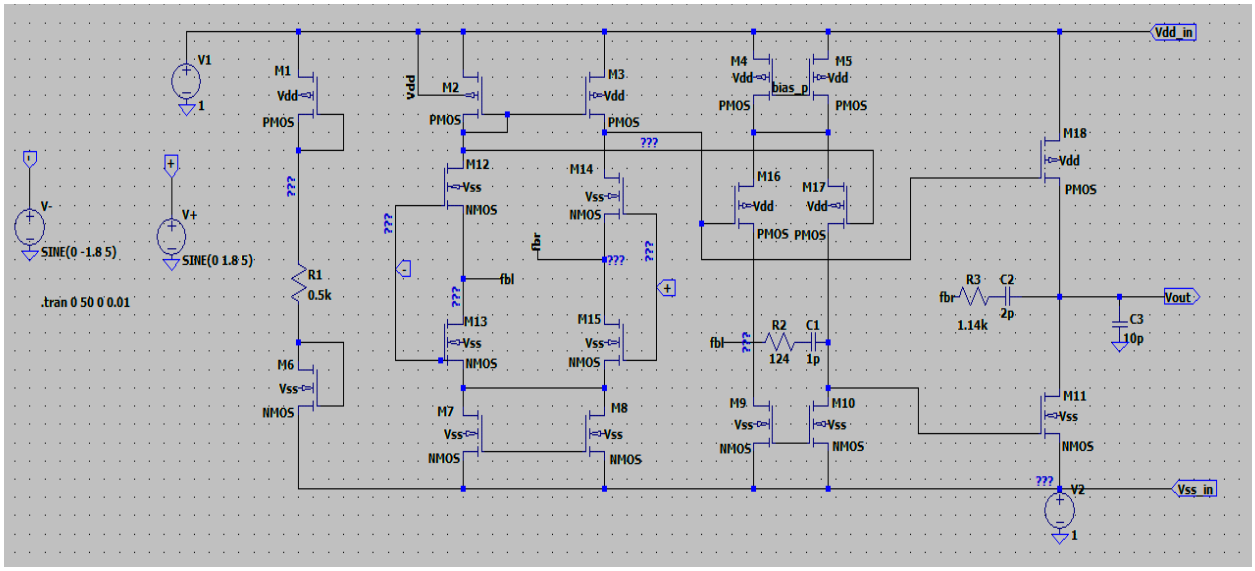
Two-Stage operational amplifies

The function of the two stage operational amplifier change the current in to voltage



Three stage operational amplifier

The function of the three stage operational amplifier or RINC with the same function of two stage ,but the difference is accuracy.it consumes less power with compered to the cascaded op-amps.



Full circuit voltage based LSTM

Figure shown below mention that full functional circuit for implementation time series prediction. According to structure 96 memristor, 32 voltage control , 3 sigmoid function, 2 tanh function , 3 multiplier and 1 addition.