B.E. PROJECT ON

HIGH-RADIX ARITHMETIC-LOGIC UNIT (ALU) BASED ON COMPOSITE MEMRISTOR SYSTEMS

Submitted By

SHIVAM RISHI 157EC15

SHREYA SETH 160EC15

SOURAV BHATTACHARJEE 171EC15

TUSHAR KUMAR 187EC15

Under the guidance of

DR. KUNWAR SINGH

A project in partial fulfillment of requirement for the award of

B.E. in

Electronics & Communication Engineering

Department of Electronics & Communication Engineering (NSIT)

Now upgraded to NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

NEW DELHI – 110078

2019

CERTIFICATE

This is to certify that the report entitled "High Radix Arithmetic-Logic Unit (ALU) based on Composite Memristor Systems", being submitted by Shivam Rishi (157EC15), Shreya Seth (160EC15), Sourav Bhattacharjee (171EC15) and Tushar Kumar (187EC15) to the Department of Electronics and Communication Engineering, NSIT, (now upgraded to Netaji Subhas University of Technology), for the award of bachelor's degree of engineering, is the record of bona fide work carried out by them under my supervision and guidance. The results contained in this report have not been submitted by either in part or in full to any other university or institute for the award of any degree or diploma.

Supervisor

Dr. Kunwar Singh

Department of ECE

NSIT, New Delhi, India

(now upgraded to NSUT)

ACKNOWLEDGEMENT

We wish to express and extend our heartfelt gratitude towards Dr. Kunwar Singh, Department of Electronics and Communication Engineering, Netaji Subhas University of Technology, Delhi, for granting us an opportunity to work on a newly emerging research area as a part of our B.E. Project. Without his thoughtful encouragement, careful supervision and unfailing support, this thesis would have never taken shape. The skills and knowledge which we have gained through the project form a valuable component in our future career development.

We would also like to express our acknowledgment towards our seniors for their kind cooperation and encouragement, which helped us in the successful completion of this project. We are grateful to our parents for supporting us through the course of this project. We extend our thanks and appreciation to the staff members and people who helped us during the course of the project.

Shivam Rishi (157EC15)

Shreya Seth (160EC15)

Sourav Bhattacharjee (171EC15)

Tushar Kumar (187EC15)

Bachelor of Engineering

Department of Electronics and Communication Engineering

NSIT, New Delhi, India (now upgraded to NSUT)

ABSTRACT

Historically, the continued dimensional and functional scaling of CMOS technology had been adequate to enable the development of new applications and improved device performance in terms of higher speed, lower power, higher density, more functionality, etc. However, the CMOS technology is approaching the fundamental limits of dimensional and functional scaling, and hence, new fields need to be explored to find alternative information processing and computing devices as well as novel architectures and techniques to sustain the historical IC scaling cadence. The newly emerging area of memristor-based devices offers an alternative approach to extend the IC technology to new applications, beyond those attainable by traditional CMOS. Memristive circuits offer the advantage of having a higher density, lower chip area, non-volatility and lower power consumption as compared to their corresponding CMOS counterparts. Moreover, composite memristive systems can be used for implementing multi-level storage cells (and hence, multi-level memories), which can be used for storing numbers in high-radix format, thereby increasing the amount of information which can be stored per unit area. This work presents a radix-4 Arithmetic Logic Unit (ALU) which utilizes a modified crossbar-based multi-bit memory architecture. Using this hybrid multi-level memory in tandem with CMOS-based peripheral interfacing circuitry enables us to implement efficient and faster arithmetic algorithms, through memory-aided computing i.e. it provides data processing and information storage in the same unit. The operation of multiplication has been implemented to demonstrate the validity of the proposed memristive ALU design. Further, a novel ALU architecture based on memristive memory-aided computing is proposed.

TABLE OF CONTENTS

CERTIFICATE **2**

ACKNOWLEDGEMENT **3**

PLAGIARISM REPORT

ABSTRACT **4**

LIST OF FIGURES **7**

LIST OF TABLES **9**

CHAPTER 1: INTRODUCTION **10**

1.1 MOTIVATION **10**

1.2 MEMRISTOR FUNDAMENTALS **10**

1.3 FINGERPINTS OF A MEMRISTOR **11**

1.4 MEMRISTOR DEVICE STRUCTURE (HP LABS) **12**

1.5 TYPES OF MEMRISTORS **13**

1.6 ADVANTAGES OVER CMOS **15**

CHAPTER 2: MEMRISTOR MODELS **15**

2.1 BRIEF HISTORY OF MEMRISTOR MODELS AND WINDOW FUNCTIONS **15**

2.2 THRESHOLD-TYPE ADAPTIVE MEMRISTOR CIRCUIT MODEL **16**

2.3 LTSPICE COMPATIBLE CODE **17**

CHAPTER 3: COMPOSITE MEMRISTIVE SYSTEMS **20**

3.1 GENERALIZED CONCEPT **20**

CHAPTER 4: RADIX-4 MULTI-STATE SWITCH (MSS) **21**

4.1 BASICS OF MULTI-BIT STORAGE CELLS **21**

4.2 RADIX-4 MSS DESIGN AND SIMULATION **22**

4.3 DATA INTEGRITY TESTING **23**

4.3.1 WRITING DATA **23**

4.3.2 READING DATA **26**

CHAPTER 5: MULTI-LEVEL MEMRISTIVE CROSSBAR **28**

5.1 BASICS OF MEMRISTIVE CROSS-BAR MEMORIES **28**

5.2 RADIX-4 MEMRISTIVE CROSS-BAR **29**

5.3 MODIFIED RADIX-4 MEMRISTIVE CROSS-BAR FOR EFFICIENT MULTIPLICATION **33**

CHAPTER 6: HIGH RADIX ALU **39**

6.1 PROPOSED ALU ARCHITECTURE **39**

6.1.1 INPUT PROCESSING BLOCK **39**

6.1.2 DEDICATED MULTIPLICATION BLOCK **40**

6.1.3 ADDER/LOGICAL OPERATION BLOCKS **42**

CHAPTER 7: CONCLUSION AND FUTURE WORK **43**

REFERENCES **44**

LIST OF FIGURES

Figure 1: The symmetry between the four basic circuit elements: Resistance, Capacitance, Inductance and Memristance **11**

Figure 2: I-V characteristics of a memristor exhibiting pinched hysteresis **12**

Figure 3: (a) Device structure of the memristor. (b) Corresponding equivalent circuit model. (c) Symbol **12**

Figure 4: Advantages of memristive systems over CMOS **14**

Figure 5: Equivalent circuit of the coupled ohmic-tunneling variable-resistor circuit model **16**

Figure 6: Simulation results for the given model (a) Memristance vs time characteristics (b) I-V characteristics of the memristor **18**

Figure 7: General rules for the building composite memristive circuits **20**

Figure 8: Dividing the total memristance range to show (a) single-bit cells (b) two-bit cells **21**

Figure 9: Radix-4 multi-state switch **22**

Figure 10: Simulation output for the radix-4 MSS **23**

Figure 11: Writing data into the radix-4 MSS **24**

Figure 12: Simulation Results for writing data into the radix-4 MSS (a) Writing 01 (b) Writing 10 (c) Writing 11 **25-26**

Figure 13: Radix-4 to binary conversion for reading the MSS **27**

Figure 14: Simulation Results for reading data from the radix-4 MSS (a) Reading 00 (b) Reading 11. **27**

Figure 15: General structure of a 3X4 multi-level memristive cross-bar **28**

Figure 16: The 3X4 multi-level crossbar memory with peripheral interfacing circuitry **29**

Figure 17: Complete structure of the 3X4 multi-level memristive crossbar memory **30**

Figure 18: Radix-4 to Binary Decoder **31**

Figure 19: Simulation results for the multi-level crossbar memory (a) Input Voltages (b) Row Select Signals (c) Read/Write Signals (d) Output Signals **32-33**

Figure 20: The block diagram of modified crossbar **34**

Figure 21: Variation in the control and input signals as compared to a normal crossbar **34**

Figure 22: Complete structure of the 2X4 reconfigurable memristive crossbar nano-architecture for efficient multiplication **35**

Figure 23: Simulation results for the 2X4 hybrid memristive crossbar (a) Input Signals (b) Configuration signals (c)Multiplier Signals (d) Read/Write Signals (e) Row Select Signals (f) Output Levels **36-37**

Figure 24: Block Diagram of the proposed memristive ALU design **41**

LIST OF TABLES

Table 1: MSS compatible output levels 40

CHAPTER 1: INTRODUCTION

1.1 MOTIVATION

Today, semiconductor device production is predominated by CMOS-based information processing and memory devices. The continuous scaling of CMOS technology with respect to dimensions and functionality was traditionally adequate to enable the development of new applications and improved device performance in terms of higher speed, lower power, higher density, more functionality, etc. However the scaling of CMOS is approaching inherent limits and the semiconductor industry is faced with two classes of difficult challenges. The first is related to pushing the CMOS technology beyond its currently achievable functionality and density by combining it with a novel low-power, faster and high-density memory technology. The other class aims to increase the information processing power beyond that achievable by CMOS, by introducing innovative approaches to combine new devices, interconnections and use novel architectures to extend CMOS and eventually invent a new information processing technology [1].

Memristive technology is an alternative approach which is capable of resolving these challenges and extend the IC technology to newer applications, surpassing dimensional scaling of CMOS. Memristor-based devices are faster, consume lower power and offer higher density in contrast to their CMOS counterparts. Moreover, composite memristor systems enable the design of reliable multi-level storage cells i.e. storage of multiple bits of data in a single memory cell. Thus, high-radix numbers can be stored in these multi-level storage units, without the need to double the memory capacity which would have been demanded by CMOS. These high-radix memristive storage elements open the avenue of the development of faster, efficient and robust computation systems.

1.2 MEMRISTOR FUNDAMENTALS

The memristor, a concatenation of “memory” and “resistor” was first proposed by Leon Chua [2] in 1971 to be recognized as the fourth fundamental element after the resistor, capacitor and inductor. This nano-device completed the symmetrical relationship between

the four fundamental circuit elements viz. flux, charge, voltage and current by providing the missing link between electric charge and magnetic flux.

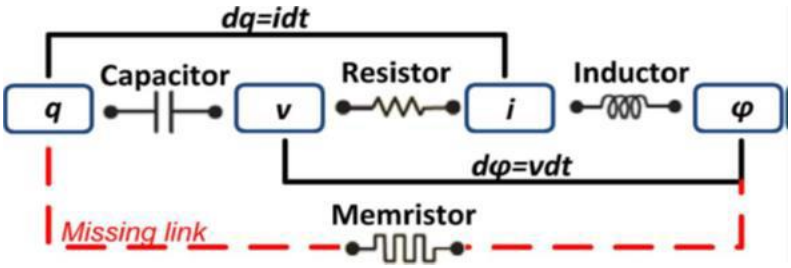


Figure 1: The symmetry between the four basic circuit elements: Resistance (dv = R di), Capacitance (dq = C dv), Inductance (dØ = L di) and Memristance (dØ = M dq)

The origin of the memristor is explained in Figure 1, which shows the four fundamental electrical quantities viz. flux, charge and their respective time integrals voltage and current.

a.) The linear relationship between voltage and current is described by resistance R, ($dv=Rdi$);

b.) The linear relationship between voltage and electric charge is described by capacitance C, ($dq=Cdv$);

c.) The linear relationship between magnetic flux Φ and current is described by inductance L, ($d\phi=Ldi$).

Clearly, before Leon Chua's discovery, the link between magnetic flux and electric charge had not been established. The proposed memristor completed this missing link by establishing the relationship between magnetic flux and electric charge as given in (1).

$$d\phi = Mdq \tag{1}$$

Where M is the memristance.

1.3 FINGERPRINTS OF A MEMRISTOR

There are three characteristic fingerprints which must be exhibited by memristive devices:

a.) Leon Chua said “If it is pinched, it is a memristor”. Therefore, the device must exhibit a pinched hysteresis loop in the voltage–current plane for any periodic signal excitation. This is shown in Figure 2.

b.) The area of the pinched hysteresis lobe should decrease with the increase in the excitation frequency.

c.) As the frequency tends to infinity, the pinched hysteresis loop should reduce to a single-valued function.

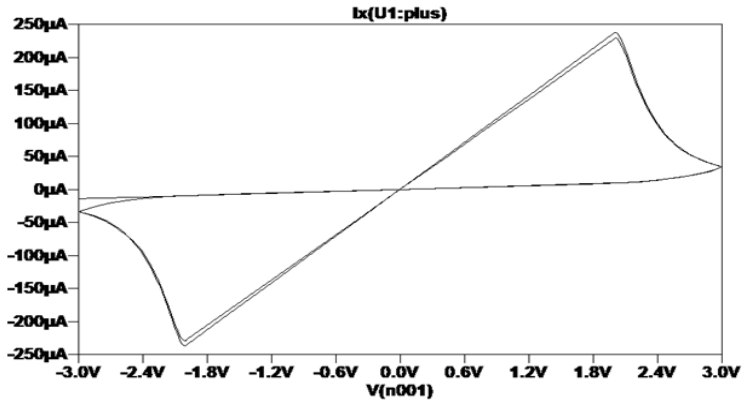


Figure 2: I-V characteristics of a memristor exhibiting pinched hysteresis

1.4 MEMRISTOR DEVICE STRUCTURE

The first basic memristor model and implementation was given by HP Labs in 2008 [6]. The HP memristor was built using Titanium Dioxide, a stable compound. The physical device structure of the memristor device given by HP have been depicted in Figure 3, along with its equivalent circuit model.

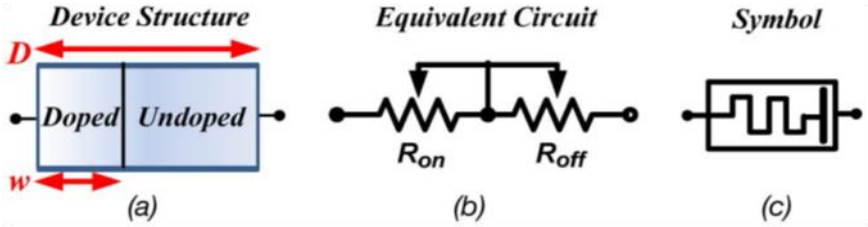


Figure 3: (a) Device structure of the memristor. (b) Corresponding equivalent circuit model. (c) Symbol

The memristor is constructed by sandwiching a thin semiconductor film between two platinum plates. It is an electrically switchable device. Two different layers of titanium dioxide are used in the semiconductor film. The length of the semiconductor film is denoted

by D, which encompasses both the layers of titanium dioxide. One layer is pure TiO_2 (un-doped layer) and is highly resistive. The other layer is highly conductive (doped layer) as it is filled with oxygen vacancies. The width of the doped region (TiO_{2-x} layer) is represented by the state variable w.

When an external bias voltage is applied across the device, the positively charged oxygen vacancies in the doped layer will be repelled by the electric field into the pure TiO_2 layer and hence, the length w will get changed. Hence, the device's total resistivity changes. The memristor state w = D corresponds to the resistance value Ron, i.e. the doped region extending throughout the length. Similarly, when w = 0, the resistance value becomes Roff and the un-doped region extends throughout the length. The device resistance varies between $Ron \leq R(w) \leq Roff$. The memristor has memory effect since the device maintains its resistivity even if the power goes off [3], thereby offering non-volatility.

The mathematical equations as per the above memristor model are described by (2) and (3).

$$M(t) = Ron.\frac{w(t)}{D} + Roff.\left(1 - \frac{w(t)}{D}\right) \quad (2)$$

$$\frac{dw}{dt} = \frac{\mu_v\, Ron}{D}\, i(t) \quad (3)$$

Where μ_v is the average ion mobility.

1.5 TYPES OF MEMRISTORS

Memristors can be of two types: either charge-controlled or flux-controlled. Their type depends on the type of biasing provided. When a current source is connected to the memristor, the current source injects charge into the memristor cell. Such a memristor is conveniently treated as charge-controlled because the state of the memristor changes in accordance with the amount of charge injected, causing the memristance value to change. On the other hand, when a voltage source is applied across a memristor, the memristor is considered as flux-controlled. In this case, the amount of flux injection changes the memristor state and memristance.

Charge-controlled memristors are characterized by the following equations:

$$v(t) = M\big(q(t)\big)i(t) \tag{4}$$

$$M(q) = \frac{d\phi(q)}{dq} \tag{5}$$

M (q) has units of resistance (ohm).

Similarly, for flux-controlled memristors,

$$i(t) = W\big(\phi(t)\big)v(t) \tag{6}$$

$$W(\phi) = dq(\phi)/d\phi \tag{7}$$

W (ϕ) has units of conductance (ohm-1) [4].

1.6 ADVANTAGES OVER CMOS

As compared to traditional CMOS devices, memristor devices possess several desirable advantages including low power consumption, faster switching speed, non-volatility of stored data and higher density. Memristor based circuits are thus area and power efficient. Moreover, they offer CMOS compatibility, i.e. they can be integrated with the existing CMOS circuits, which makes them a promising replacement of the CMOS technology [5]. A summary of these advantages is depicted in Figure 4.

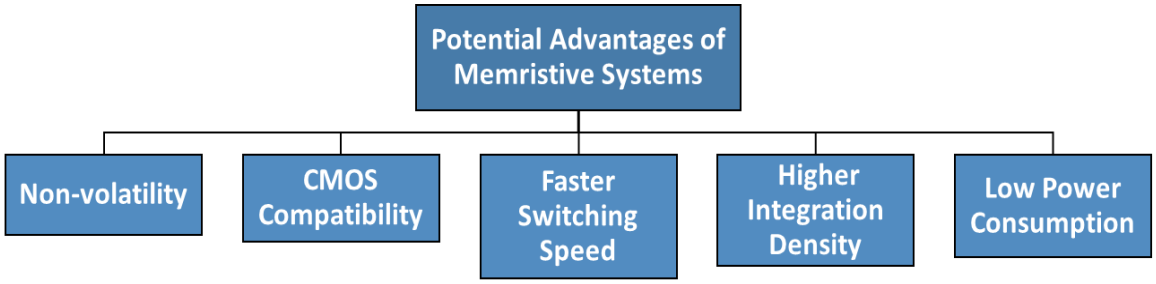


Figure 4: Advantages of memristive systems over CMOS

CHAPTER 2: MEMRISTOR MODELS

2.1 BRIEF HISTORY OF MEMRISTOR MODELS AND WINDOW FUNCTIONS

In order to enhance the understanding of the complex behavior exhibited by memristors, it is essential to develop mathematical models for the nano-device. Proper and accurate modelling of memristors is required for design, analysis, and simulation of memristor-based circuits. Several mathematical models have been proposed for the memristor including the Linear Ion Drift Model (HP Labs 2008; discussed in Chapter 1) [6], the Nonlinear Ion Drift Model [7], Simons Tunneling Barrier Model [8] and the ThrEshold Adaptive Memristor Model (TEAM) [9]. A detailed comparison of these models can be found in [9]. Among these, TEAM is the latest and most generic memristor model, which overcomes the drawbacks of all the previously proposed models and gives the mathematical modelling for a current-controlled memristor. As stated in [9], the TEAM model is very precise with a small value of mean error of about 0.2% and is able to boost the simulation runtime by an approximate value of 47.5%. The model also satisfies a memristive system's physical requirements, the convergence conditions as well as the computational efficiency demanded by simulators. An extended version of TEAM, known as VTEAM [10] or the Voltage ThrEshold Adaptive Model expands on the equations for a voltage-controlled memristor.

Each model has a certain region which can work entirely. For example, the linear ion drift model can work only in the interval of [0, D]. So, the derivative of the state variable is multiplied by a window function to prevent the state variable from getting out of the bound, as well as to add more non-linear behavior near the boundaries. So, the following two things should be given by window functions:

a.) A working interval for the state variable
b.) When the state variable is at the bounds, the non-linearity near the boundaries should make it 0.

Several window functions have been proposed in literature including Joglekar [11], Biolek [12], Prodromakis [13], Piecewise and TEAM. A detailed description and comparison of these models can be found in [15].

2.2 THRESHOLD TYPE ADAPTIVE MEMRISTOR CIRCUIT MODEL

The equivalent circuit of this memristor model is depicted in Figure 5. It is inspired by the basic memristor model proposed by HP Labs.

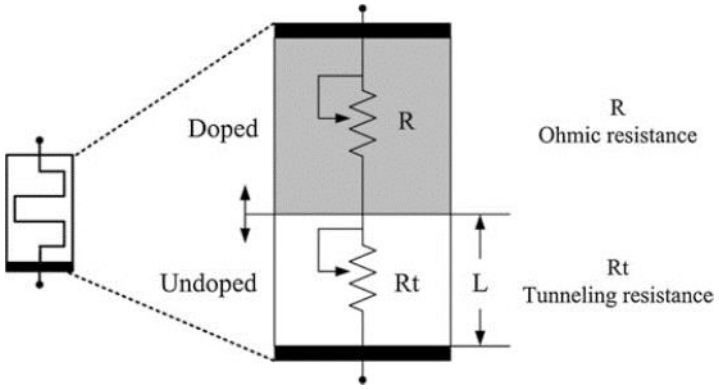


Figure 5: Equivalent circuit of the coupled ohmic-tunneling variable-resistor circuit model

This model is a threshold-type switching model for a two-terminal flux-controlled (voltage) memristor. The following equations give its general definition:

$$I(t) = G(L,t)V_M(t) \tag{8}$$

$$\dot{L} = f(V_M, t) \tag{9}$$

The parameter L denotes system state variable i.e. it indicates the internal memristor state (here, tunnel barrier-width). G denotes the memductance of the device, I represents the current through the device and V_M, the input voltage.

In the coupled ohmic-tunneling variable-resistor equivalent circuit of Figure 5, an ohmic variable-resistor R and a tunneling variable-resistor Rt are considered to be connected in series. R represents the resistance of the doped titanium dioxide layer and Rt represents the tunneling resistance of the un-doped layer. The doped layer acts as a conductor whereas the un-doped layer is a pure insulator. Therefore, Rt $\gg$ R. The tunneling resistance Rt is proportional to the tunnel barrier width L, as the larger the barrier width the higher the resulting resistance. Also, its value changes in accordance with the movement of the boundary between the two layers, because of the transport of oxygen deficiencies under positive or negative applied voltage.

The expected response of L as a function of the time t and the applied voltage V_M is given by the following equation:

$$L(V_M, t) = L_0 \left(1 - \frac{m}{r(V_M, t)}\right) \quad (10)$$

L_0 is the maximum value that L can attain. r (V_M, t) is a voltage-dependent parameter and m, a fitting constant parameter, together determine the boundaries of the barrier width. r lies between rmin and rmax, corresponding to Lmin and Lmax (L_0). Correspondingly, the memristance of the device varies between Ron (most conductive state) and Roff (least conductive state). Values for parameters m and rmin should be selected so that the fraction (m/rmin) < 1. So, the tunnel barrier-width will never be 0.

The threshold voltage (VSET or VRESET) of the device, is viewed as the minimum voltage required to impose a change on the physical structure, and hence the device memristance.

Parameters alpha, beta, and gamma are fitting constants that are used to decide the rate of memristance change, with a≫ b and 0 < c < 1. Setting b = 0 imposes a hard switching behavior, i.e. there is no state change in the memristor unless a certain voltage threshold is exceeded. All unknown material-specific and geometrical issues are contained into the model-fitting constant parameter fo.

2.3 LTSPICE COMPATIBLE CODE

The LTSPICE sub circuit code for the discussed model is given below:

.subckt memristor plus minus PARAMS:

***Parameters' values**

+rmin=100 rmax=390 rinit=390 alpha=2000 beta=0 gamma=0.1 VtR=1 VtL=-1 yo=0.0001

+m=82 fo=310 Lo=5

Gr 0 r value={dr_dt(V(plus)- V(minus))*(st_f(V(plus)-V(minus))*st_f (V(r)-rmin)+

+st_f(-(V(plus)-V(minus)))*st_f(rmax-V(r)))}

Cr r 0 1 IC={rinit}

Raux r 0 1E12

***Current equation Imem = V / R(L)**

Gpm plus minus value={(V(plus)- V(minus))/((fo*exp(2*L(V(r))))/L(V(r)))}

***Func. for non-linear threshold-based behavior**

.func dr_dt(y)={-alpha*((y-VtL)/(gamma+abs(y-VtL)))*st_f(-y+VtL)-beta*y*st_f(y-VtL)*

+st_f(-y+VtR)-alpha*((y-VtR)/(gamma+abs(y-VtR)))*st_f(y-VtR)}

***Smoothing function**

.func st_f(y)={1/(exp(-y/yo)+1)}

***L(V) function**

.func L(y) ={Lo-Lo*m/y}

.ends memristor

The simulation results for testing the validity of this model are given in Figure 6.

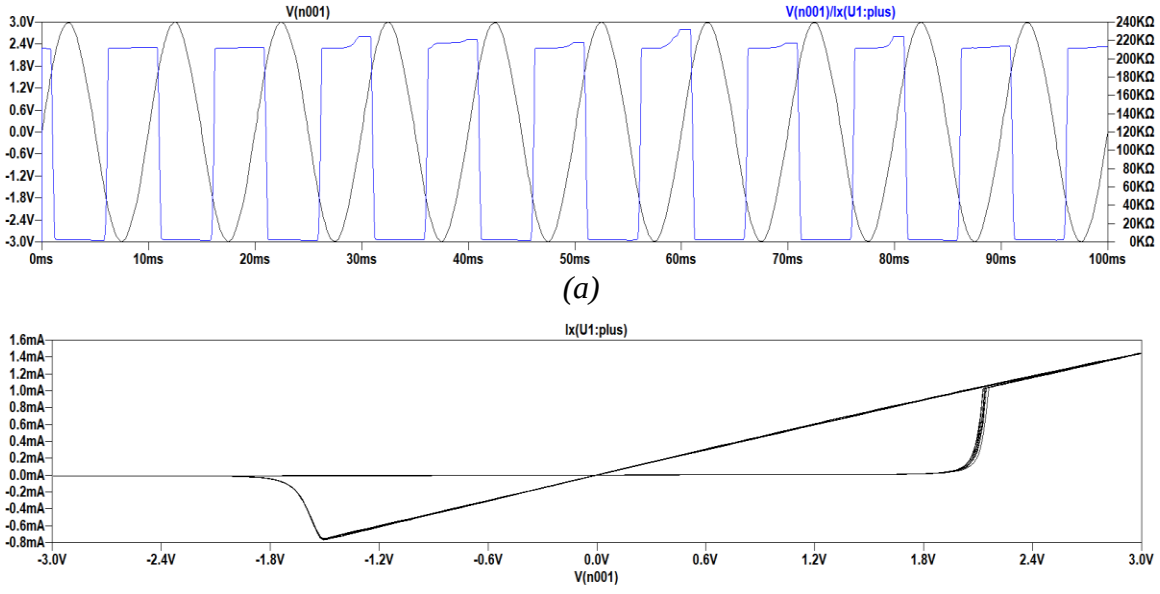


Figure 6: Simulation results for the given model (a) Memristance vs time characteristics (b) I-V characteristics of the memristor

A sinusoidal voltage of 3V amplitude and 100Hz is applied as input. Model characteristics used are alpha = 1e6, beta = 10, VSET = 1.5V, VRESET = -1.5V. Other values are same as given in the code. For all the other simulations, model values used are alpha = 2000, beta = 0, gamma = 0.1, m=82, L_0 = 5nm, Ron = 2k and Roff = 200k.

CHAPTER 3: COMPOSITE MEMRISTOR SYSTEMS

3.1 GENERALIZED CONCEPT

Memristors can essentially be thought of as non-volatile switches with an associated set of voltage threshold values. If a small input voltage is applied, then the memristance changes negligibly. However, if the applied voltage is greater than the voltage thresholds and is applied for an interval larger than the device switching time, then the internal resistive state of the device changes rapidly. Multiple memristors can be connected series or parallel to analyze their composite behavior. However, composite memristor circuits work in a very complex fashion as the operation of each individual memristor is polarity dependent and highly non-linear.

Composite memristor systems also demonstrate threshold-dependent behavior. This means that the composite system starts working only when the applied voltage exceeds the threshold voltage of the composite system. The general behavior of first-level memristor compositions in series and parallel has been shown in Fig 5. The threshold voltages remain unaffected in parallel compositions. The main advantage of composite memristive systems is that by connecting memristors in appropriate polarity and configurations, programmable multi-state switches can be designed, which can in turn be utilized for designing high-radix memories, computation units, oscillators etc.

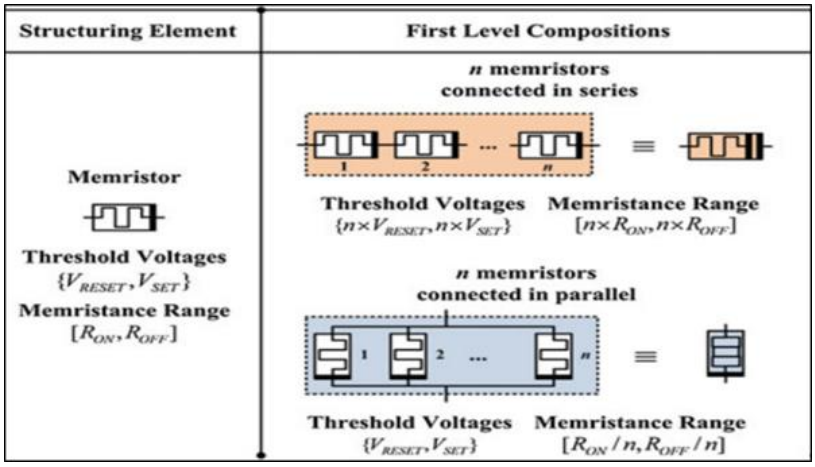


Figure 7: General rules for the building composite memristive circuits

CHAPTER 4: RADIX-4 MULTI-STATE SWITCH (MSS)

4.1 BASICS OF MULTI-BIT STORAGE CELLS

In memristor-based storage cells, the value of the data stored is represented by the internal state of the memristor. When storing 1-bit per cell i.e. in binary format, the total memristance range is divided into three zones and is defined by two by two resistive bounds. The two bounds include the higher bound or logic '0' [High (0)] and the lower bound or logic '1' [Low (1)]. Any memristance value, in-between these boundaries (known as guard bands) is invalid and does not denote any binary value.

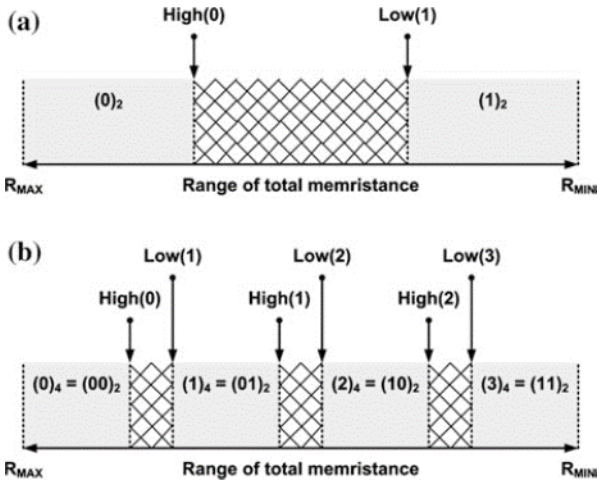


Figure 8: Dividing the total memristance range to show (a) single-bit cells (b) two-bit cells

However, memristor-based memory cells can store multiple bits of data per storage cell. In case of multi-bit memristor cells, the resistance range is divided into several zones. For four states (2-bits per cell), the division of the memristance range is shown in Figure 8 (b). As the number of resistance levels per storage element are increased, a reduction in memory reliability is observed as a result of reduction in the noise margins. Thus, a trade-off between memory density and data reliability exists in multi-bit memories. A higher memory density can be achieved at the cost of reduced reliability or vice versa, depending upon the specific application.

4.2 RADIX-4 MSS DESIGN

In this work, a radix-4 multistate switch, based on composite memristor configurations has been implemented as shown in Figure 9. It is capable of storing data in radix-4 format and can have four states: $(0)_4$, $(1)_4$, $(2)_4$ and $(3)_4$.

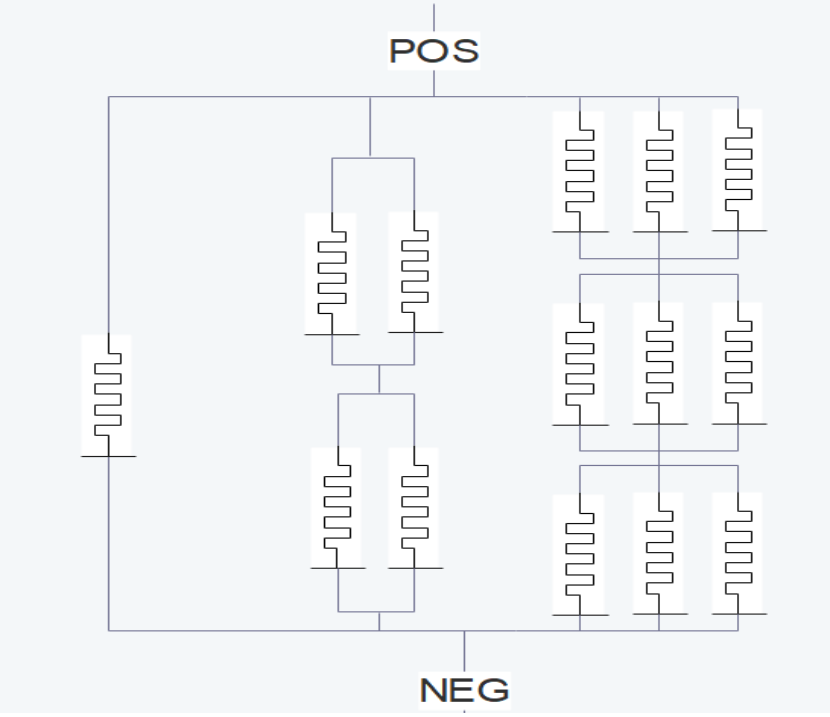


Figure 9: Radix-4 multi-state switch

In this four-state memristive MSS, the range of composite memristance is [Ron/3, Roff/3]. The threshold voltage for the first branch is VSET, the second branch is 2VSET and the third branch is 3VSET. When all the memristors are in OFF state, i.e. applied voltage is below VSET, none of the branches are turned on and the system represents State $(0)_4$ or $(00)_2$. In this case, the system has maximum resistance i.e. Roff/3. When the applied voltage is between VSET and 2VSET, the first branch is turned on and the system represents State $(1)_4$ or $(01)_2$. Similarly, when the applied voltage is between 2VSET and 3VSET, the first and second branches are turned on and the system represents State $(2)_4$ or $(10)_2$. Finally, State $(3)_4$ or $(11)_2$ are stored when all branches are on and this is the state of minimum resistance Ron/3. Therefore, the multi-state switch of Figure 9 can be used to represent four binary states as per the application as it exhibits four possible conducting states. Figure 10 shows the simulation results of the radix-4 MSS. As can be seen clearly,

as the applied voltage exceeds the voltage thresholds of the three branches, each branch gets activated thereby changing the state of the memristor.

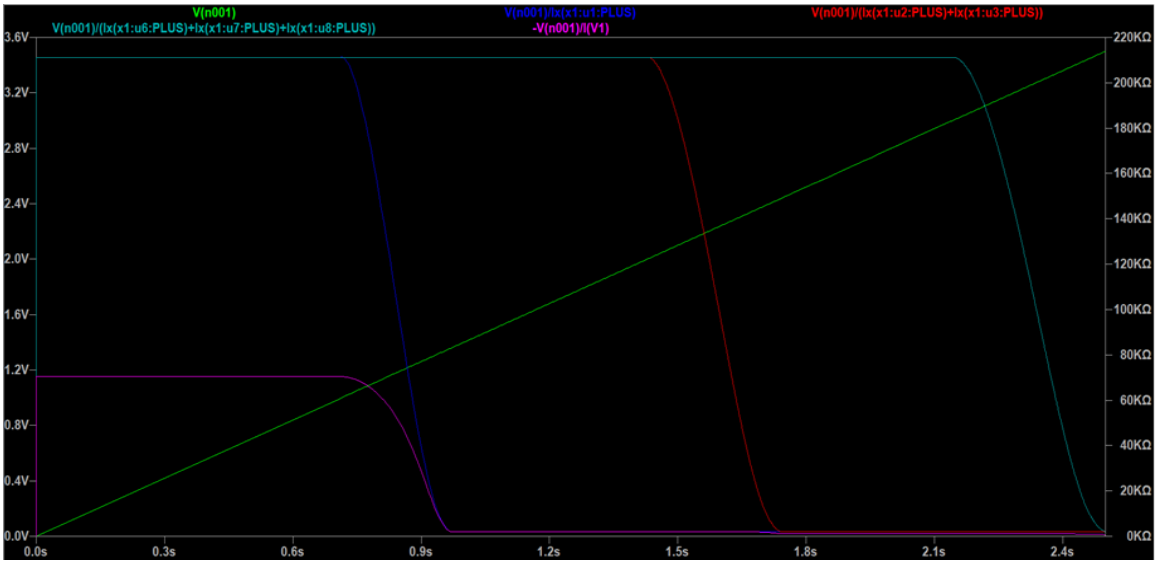


Figure 10: Simulation output for the radix-4 MSS

4.3 DATA INTEGRITY TESTING

4.3.1 WRITING THE RADIX-4 MSS

To program a logic value into the MSS, the memristors are considered as voltage-controlled devices and the state of the MSS is altered by the flux injection. The circuit used for writing has been shown in Figure 11. The input voltage V1 generates a pulse of amplitude 'Va' and pulse width 'Tw'. For programming any state into the MSS, the magnitude 'V' of the applied voltage must exceed the threshold voltage for that state. Also, the pulse width 'Tw' has to be greater than the minimum time in order to ensure state rest after write operation.

This pulse width is computed using the following equations (11), (12) and (13):

$$Tw = \frac{\emptyset\,[\,Roff^2 - R(W)^2]}{|Va|\,Roff^2} \tag{11}$$

Where Ø is the flux injection.

$$\emptyset = \frac{(\beta D)^2}{2\mu_v(\beta - 1)} \tag{12}$$

$$\beta = \frac{Roff}{Ron} \tag{13}$$

μ_v = Mobility of oxygen vacancies

It is important that the applied pulse width must be of appropriate duration. If the pulse width is too short, then the MSS does not reach the desired state. If Tw is too long, then it leads to undetermined switching action in the MSS resulting in corrupt data.

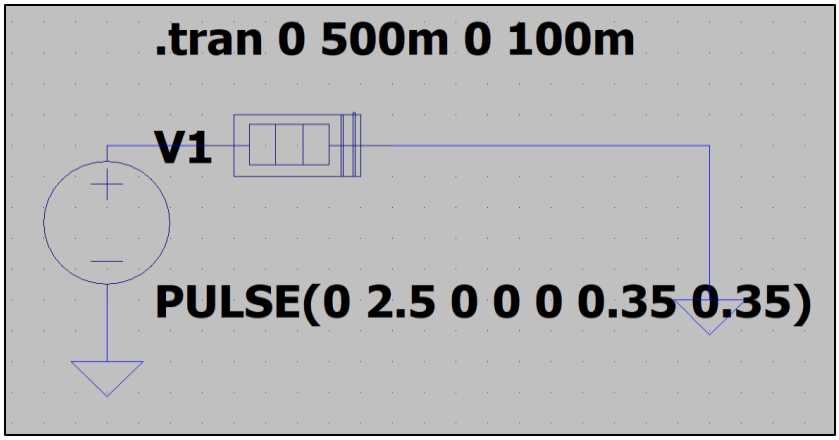
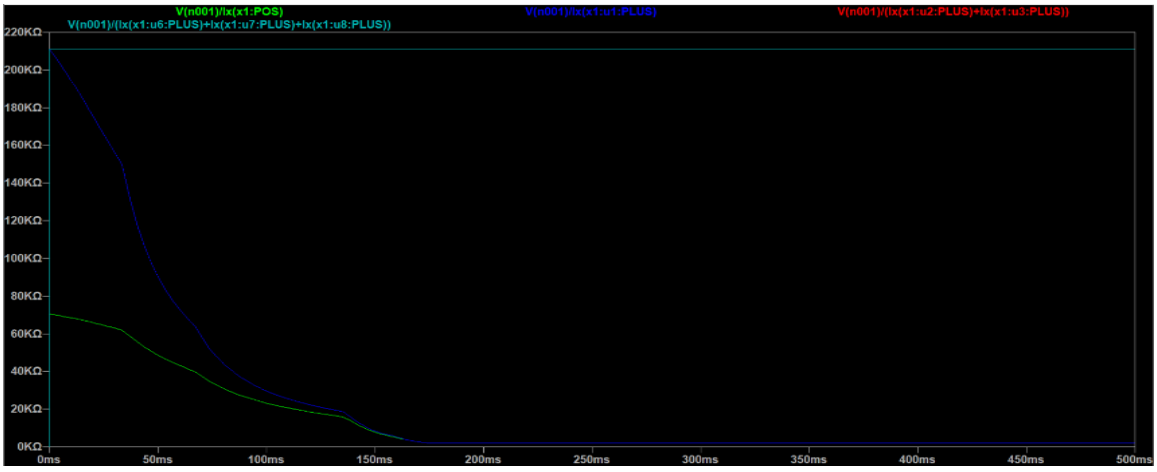
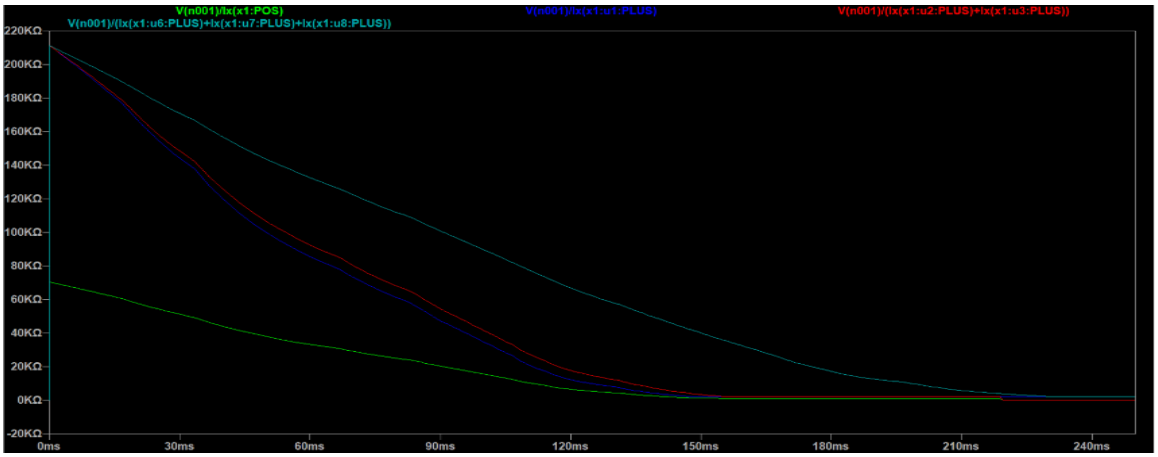


Figure 11: Writing data into the radix-4 MSS

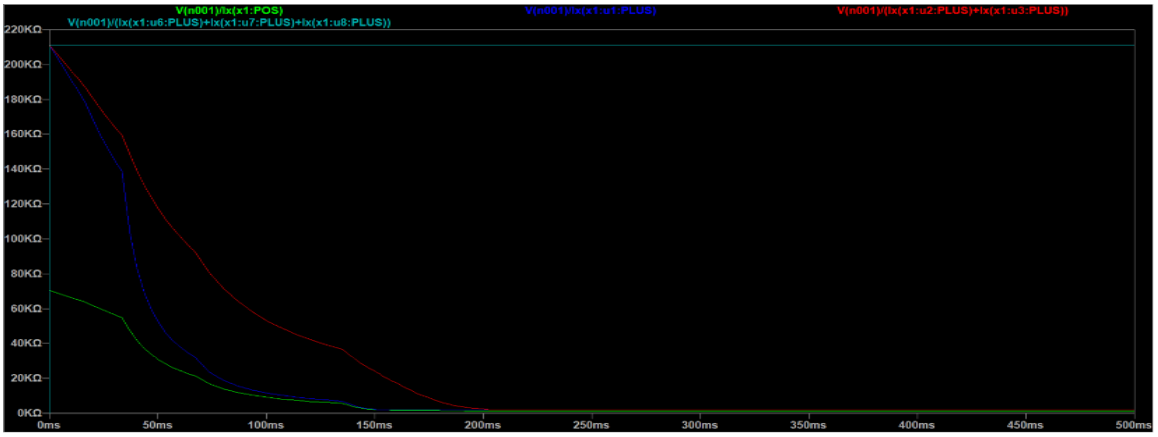
The simulation results for programming the radix-4 MSS have been depicted in Figure 12.



(a)



(b)



(c)

Figure 12: Simulation Results for writing data into the radix-4 MSS: (a) Writing 01 (b) Writing 10 (c) Writing 11

The initial state of the MSS is $(00)_2$. In Figure 12(a), the input voltage signal exceeds the voltage of the first MSS branch and causes it to switch from Roff (200kΩ) to Ron (2kΩ). As a result, the composite memristance falls from around 67kΩ to 1.99kΩ, causing a state change from $(00)_2$ to $(01)_2$. Similarly, in 12(b), the input voltage signal exceeds the voltage of the first two MSS branches. As a result, the composite memristance falls from around 67kΩ to 0.99kΩ, causing a state change from $(00)_2$ to $(10)_2$. Finally, in 12(c), the input voltage signal exceeds the voltage of the all the MSS branches. The composite memristance in this case is (2kΩ/3) = 0.67kΩ.

4.3.2 READING THE RADIX-4 MSS

In order to read the high-radix state stored in the MSS, we use a radix-4 to binary decoder circuit. Any of the four possible composite memristor states need to be converted into two binary signals. For this purpose, the decoding circuit has three levels of implementation as shown in Figure 13. It consists of:

a.) a voltage divider circuit
b.) a comparator network
c.) digital logic components

The first level is the voltage divider circuit formed between the MSS and a series resistor known as the sense resistor. The common node between the MSS and the series sense resistor is called the intermediate node. The value of voltage at the intermediate node will be unique for every composite memristor state. This unique voltage value allows the use of the following additional circuit levels which use this value interpret the exact state stored in the memristor.

The second level of the decoding circuit is used to generate reference voltage levels for inputs into the comparators. This is accomplished by connecting a series resistor network in parallel with the intermediate node. In this stage, all the necessary voltage levels are created a single reference voltage source, corresponding to the boundaries of memristance shown in Figure 8. These reference voltage levels are utilized as one of the inputs into the comparator network and the other input is the intermediate node. For every possible resistive state, two comparisons are needed to identify he state stored in the memristor. For an n-state MSS, 2n comparisons would be required generally to compare the stored state with values of the memristance boundaries (upper and lower). However, we can reduce this number to $2n - 2$, by eliminating the comparisons of the intermediate node with the uppermost and lowermost boundaries, which are the limits of memristancefor the storage cell. Hence, we need 2n-2 comparators and a minimum of 2n-2 reference resistors for designing the second level. Lastly, the third level includes multiplexer circuits for converting the CMOS-compatible digital signals produced by the comparator network into binary signals. Four cascaded 2×1 MUX circuits are used for this purpose, and the last multiplexer circuit produces the corresponding binary word.

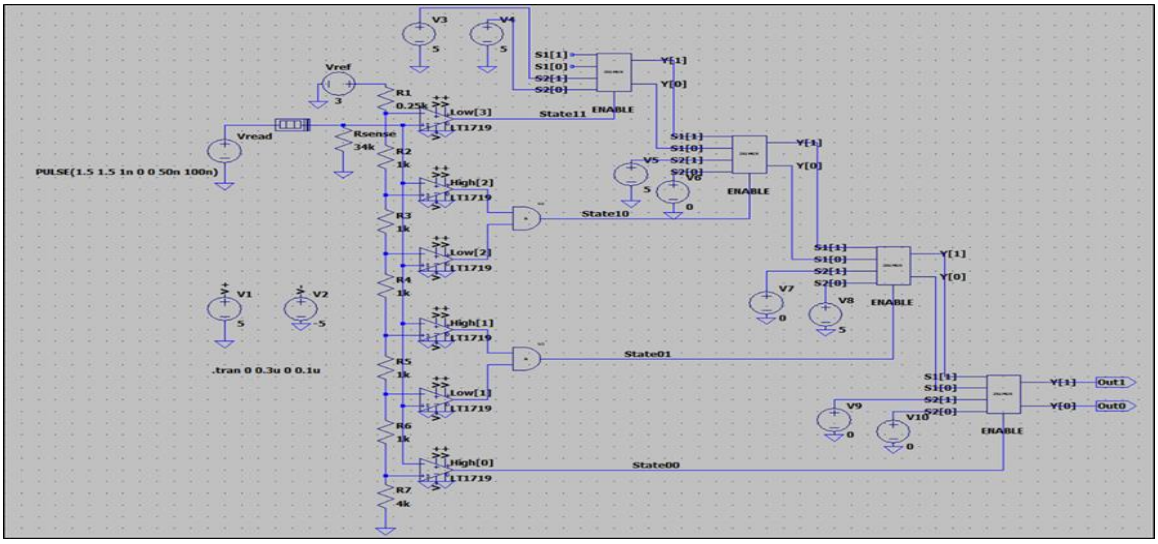
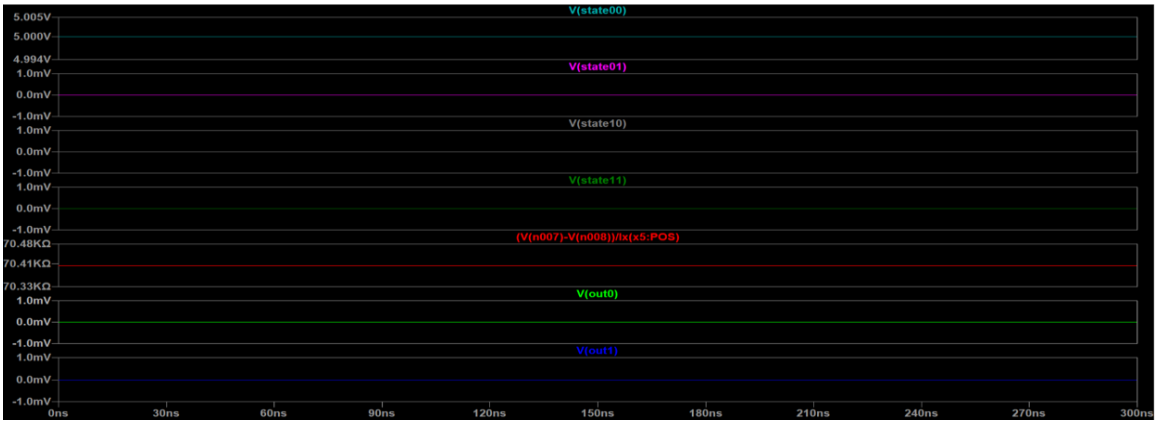
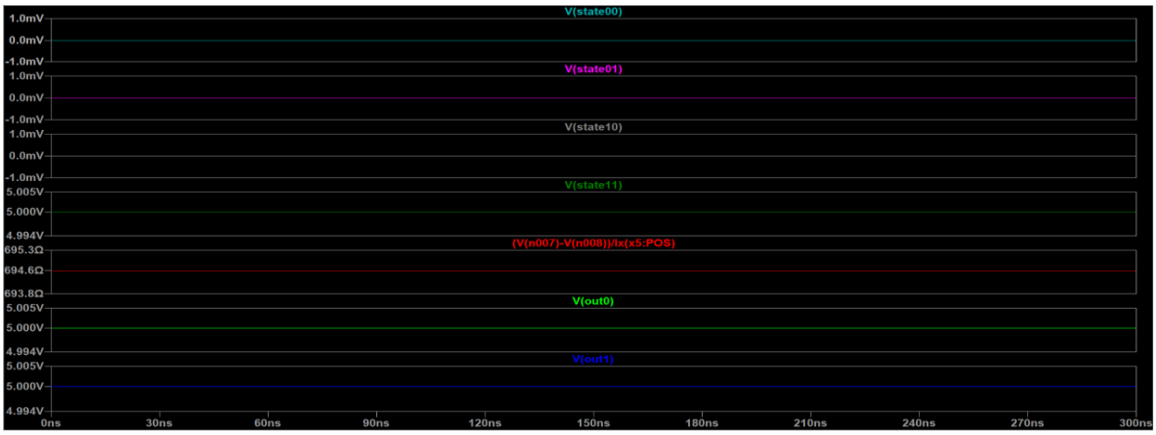


Figure 13: Radix-4 to binary conversion for reading the MSS

The simulation results for reading the radix-4 MSS are shown in Figure 14.



(a)



(b)

Figure 14: Simulation Results for reading data from the radix-4 MSS (a) Reading 00 (b) Reading 11.

CHAPTER 5: MULTI-LEVEL MEMRISTIVE CROSSBAR

Multi-level Memristive crossbar is a storage structure which can store high radix data in a rectangular array of Multi state switches. The general structure of a crossbar is shown in the Figure 15:

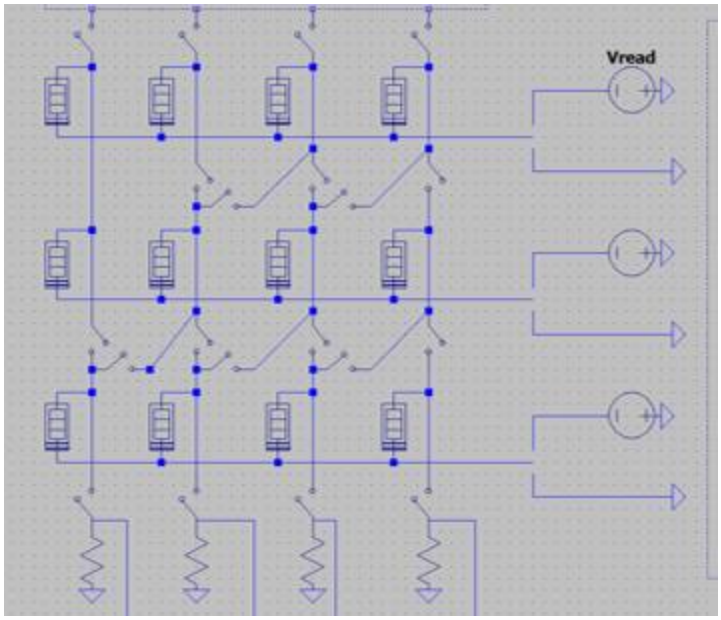


Figure 15: General structure of a 3X4 multi-level memristive cross-bar

As seen in the figure each row of the crossbar stores high radix numbers equal to the columns. Further, the number of rows determine the number of 'words' that can be stored in the crossbar memory. For instance, the figure shows a 3X4 crossbar, which stores 3 words with each row storing a high radix vector of length 4. The proposed ALU architecture aims to utilize memristive crossbars for implementing efficient arithmetic operations. This section elaborates on the details of these components used in ALU with their demonstration using LTSPICE simulations.

5.1 BASICS OF MEMRISTIVE CROSS-BAR MEMORIES

Memristor-based non-volatile memory, also known as ReRAM or Resistive RAM, is the future of memory technology and is a promising candidate to replace the traditional Flash memories. Memristive ReRAM offers several advantages over CMOS memories including lower energy, higher storage density, scalability, compatibility with CMOS, etc. The crossbar architecture is considered to be the best and most area-efficient architecture for

implementing ReRAMs. Crossbar architecture offers the highest possible device density along with several other benefits including manufacturing flexibility, pattern regularity, defect-tolerance and CMOS compatibility.

In this project, a 3X4 high-radix memristive memory has been designed. Every cross-point element is a radix-4 MSS which stores 2 bits per cell. Using multi-bit storage cells in a crossbar structure helps us to exploit the benefits of both compact multi-bit storage, while maintaining low overall complexity of the peripheral CMOS-based interfacing circuitry

5.2 RADIX-4 MEMRISTIVE CROSS-BAR

Radix-4 Memristive crossbar stores High Radix vectors in a conventional memory structure. The block diagram of a 3X4 crossbar is shown in Figure 16, showing the various input and control signals needed to interface the component. The input vector is of length 4 with each element containing a programming voltage from the set {0, 2.5. 4.5. 6.5} for writing {0,1,2,3} on the corresponding row selected by turning on one of the control signals {A0,A1,A2} which selects the row in which data is written. Further, Signals Read, Write help in either reading or writing to the memory. Finally, Vread controls the range of output obtained at the outputs {O1, O2, O3, O4}.

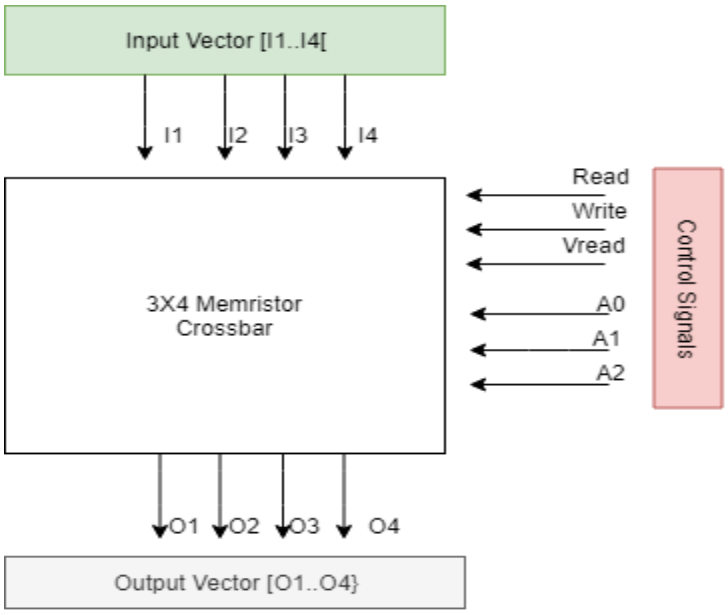


Figure 16: The 3X4 multi-level crossbar memory with peripheral interfacing circuitry

The read and write process are described as follows:

In order to write a crossbar:

1. Apply Input signal {I1, I2, I3, I4}
2. Enable write signal (Write).
3. Select the row in which the data is to be written by enabling one of the signals A0, A1, A2.

Similarly in order to read any row of the crossbar:

1. Select the row whose data is to be read by enabling one of the signals A0, A1, A2.
2. Enable the read signal.
3. Apply a voltage Vread.
4. Sample the output lines O1, O2, O3, O4.

This reading and writing process is derived by the virtue of internal structure of the crossbar. The crossbar circuit is simulated in LTSPICE.

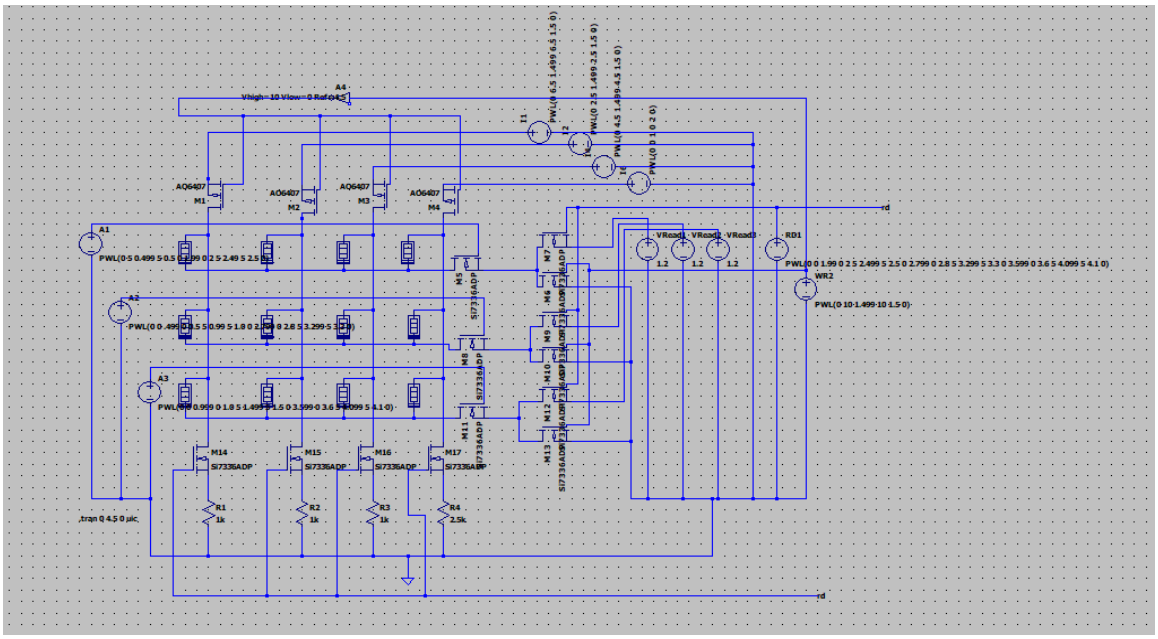


Figure 17: Complete structure of the 3X4 multi-level memristive crossbar memory

The complete circuit for the 3X4 multi-level memristive cross-bar is given in Figure 17. This is a modified crossbar circuit as it includes both horizontal as well as vertical switches among the cross-point elements (the radix-4 MSS). This scheme allows us to determine the

flow of data during memory access operations. A circuit schematic of the 2X4 hybrid crossbar is shown in Figure 19. All the MSS switches corresponding to a memory word or one crossbar row are accessed as well as programmed in a parallel fashion. The row decoder circuit grounds the target row wire and the remaining row wires are left floating. Also, if corresponding row wires are simultaneously grounded, multiple words can be programmed in parallel, and the same input data is copied to all the grounded rows. This parallel programmability of the hybrid crossbar is utilized for the implementation of high speed and efficient hardware multiplication circuit.

The main memristive crossbar memory structure is interfaced in its periphery by CMOS circuits. These include the programming/writing circuitry, the reading circuitry and the decoding/sensing circuitry. As explained earlier, the decoding circuitry includes the network of sense resistors and comparators. This circuit deciphers the high-radix state and convert them into binary to provide memory word stored in the targeted crossbar row. This circuitry operates alternately with the programming inputs; it is activated only during the reading mode, while the programming inputs are properly isolated, and vice versa. For converting the radix-4 outputs of the comparators into binary format, a radix-4 to binary decoding circuit is employed, given in Figure 18.

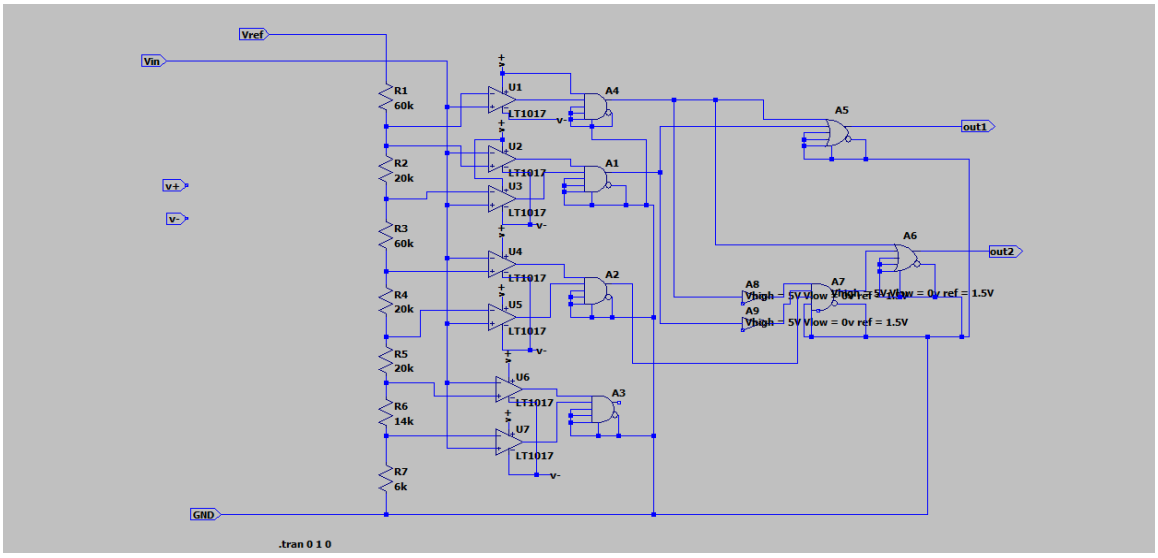
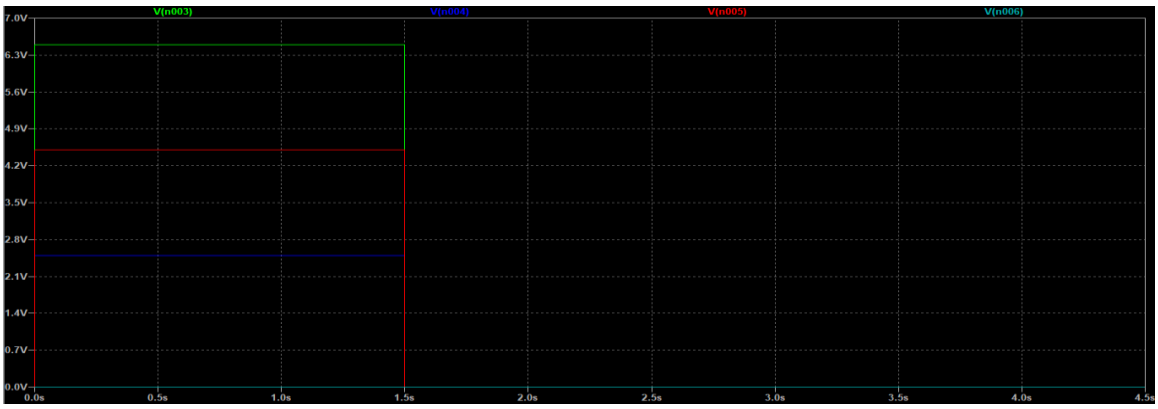
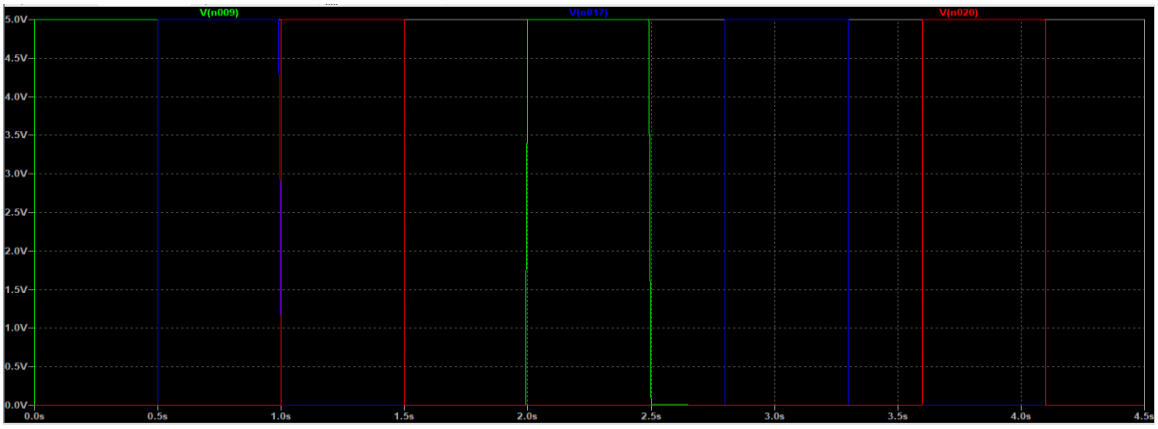


Figure 18: Radix-4 to Binary Decoder

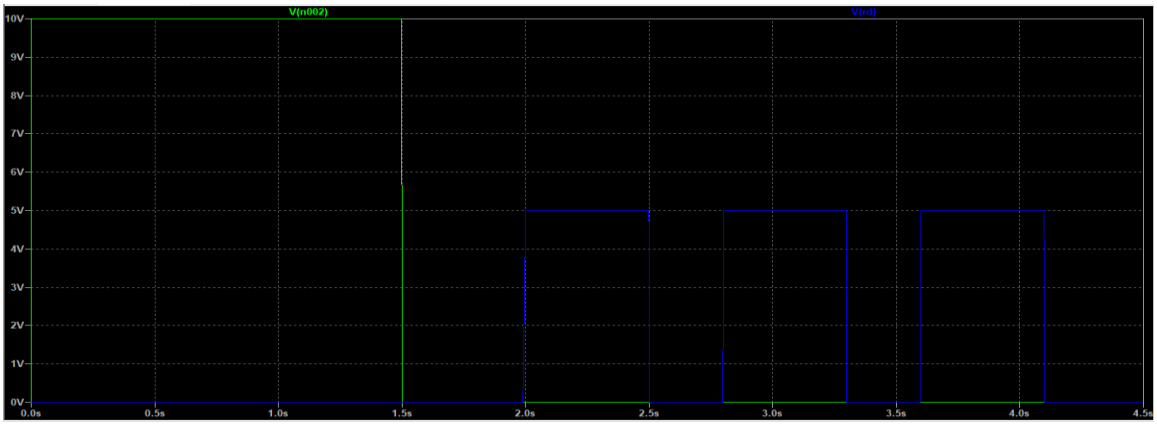
For validating the above memory design, $(3120)_4$ is written into the selected row and is then read by applying appropriate signals. The simulation results for the multi-level crossbar memory are given in Figure 19.



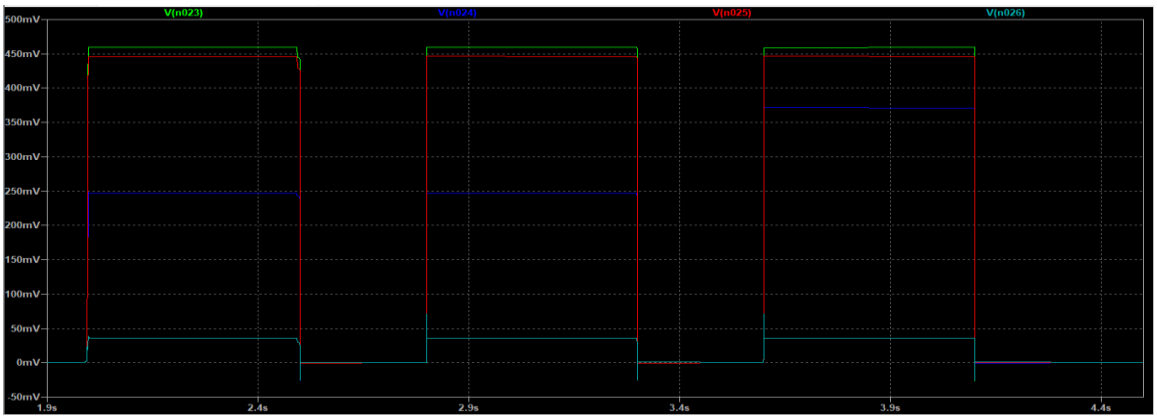
(a)



(b)



(c)



(d)

Figure 19: Simulation results for the multi-level crossbar memory (a) Input Voltages (b) Row Select Signals (c) Read/Write Signals (d) Output Signals

Careful observation of simulation output validates the proposed crossbar circuit and reading and writing process described above.

Further, it was shown by Vourkas et. al. [1] that a slight modification in the crossbar structure can lead to fast calculation of high radix multiplication operation. This modified crossbar is used in the proposed ALU architecture, therefore it is necessary to verify its operation and development of a hardware multiplication algorithm. These objectives are takin into consideration in the next section.

5.3 Modified Radix-4 Memristive Cross-bar for Efficient Multiplication

Commonly, multiplication is carried out by adding shifted partial products obtained by multiplying each bit of the multiplier with the multiplicand. Modified crossbar enables faster calculation of these partial products by virtue of its structure. The block diagram of modified crossbar is shown in Figure 20, describing the external interface of this component which will eventually be used in proposed ALU design.

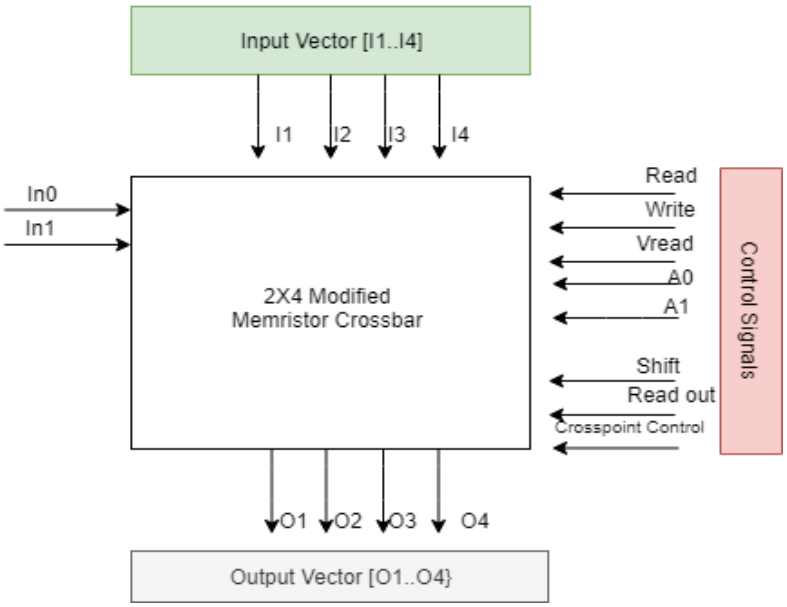


Figure 20: The block diagram of modified crossbar

Majority of control and input signals shown above are same as a normal crossbar described in the above section with the exception of Shift, Read out and Crosspoint Control signals and in0 and in1 input signals. This component calculates partial products by virtue of changes in topology as compared to normal crossbar. To obtain shifted partial products the MSS switches in the consequent rows are connected to the adjacent input line to the right such that during the input phase, the less adjacent less significant input is used to program the switch instead of its own column line, therefore performing a shift operation. This topology variation can be seen in Figure 21, the input is shifted to left in each row.

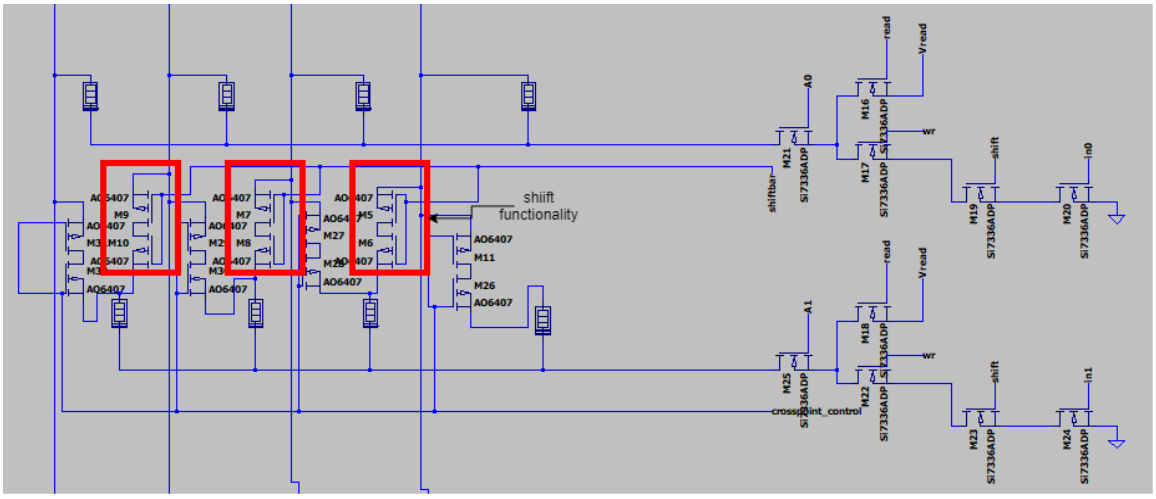


Figure 21: Variation in the control and input signals as compared to a normal crossbar

However, in order to ensure correct operation, it is essential to provide correct control voltages, the order of which is summarized below.

1. Apply high radix vector programming voltage at inputs [I2..I0] (I3 should be 0 during multiplication since the maximum length of partial product is limited by number of columns)
2. Apply binary multiplier at lines in0 and in1
3. Enable write, shift, A0, A1 to write shifted partial products in the modified crossbar
4. Read each row of the crossbar by applying read, read_out and selecting each row one by one.
5. Add the partial products obtained from crossbar output after processing according to multiplication algorithm implemented by ALU.

To verify the operation of the proposed modified crossbar circuit spice simulations performed are described below. In this simulation, a modified memristor crossbar is generated and shifted partial product is obtained at the second row by the application of control signals as shown in the following Figures 22 and 23.

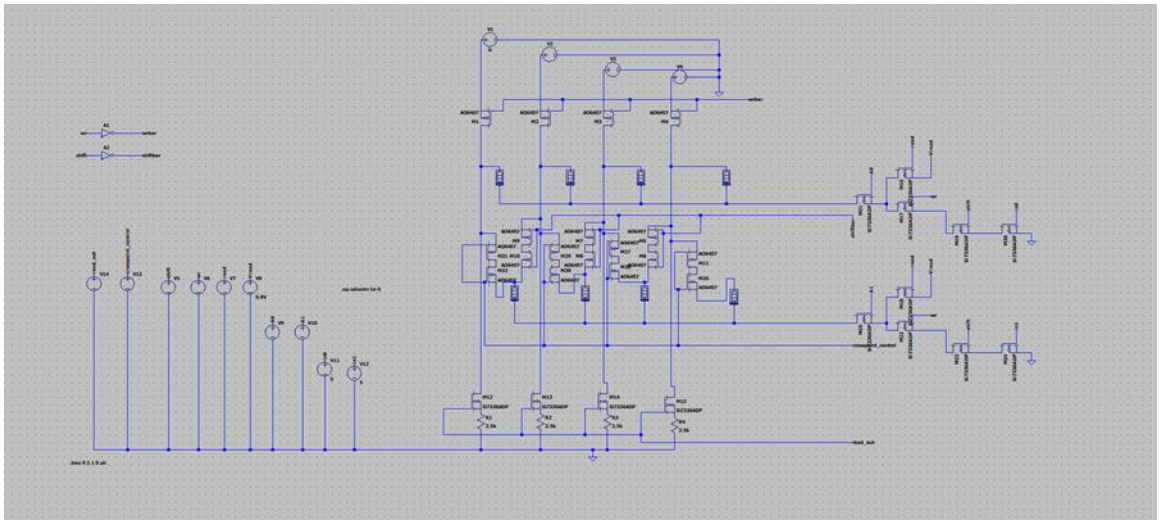
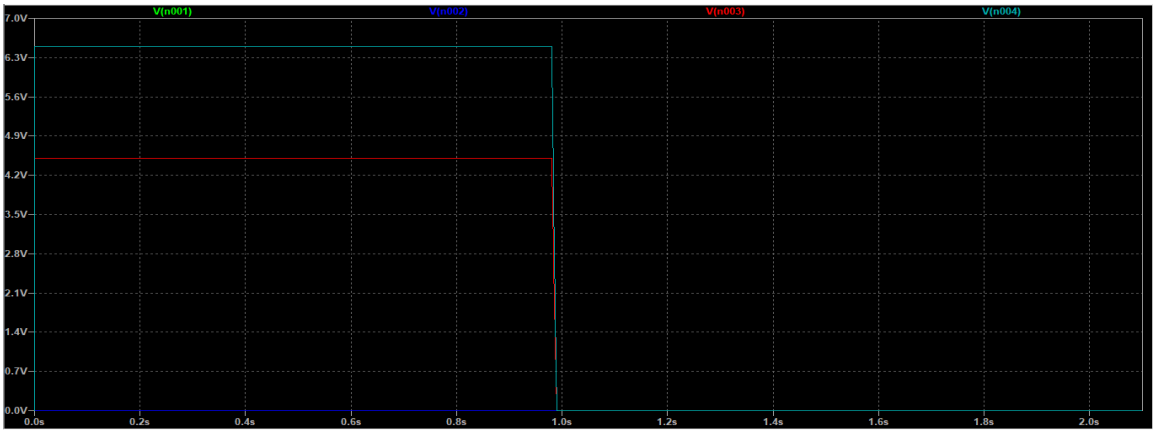
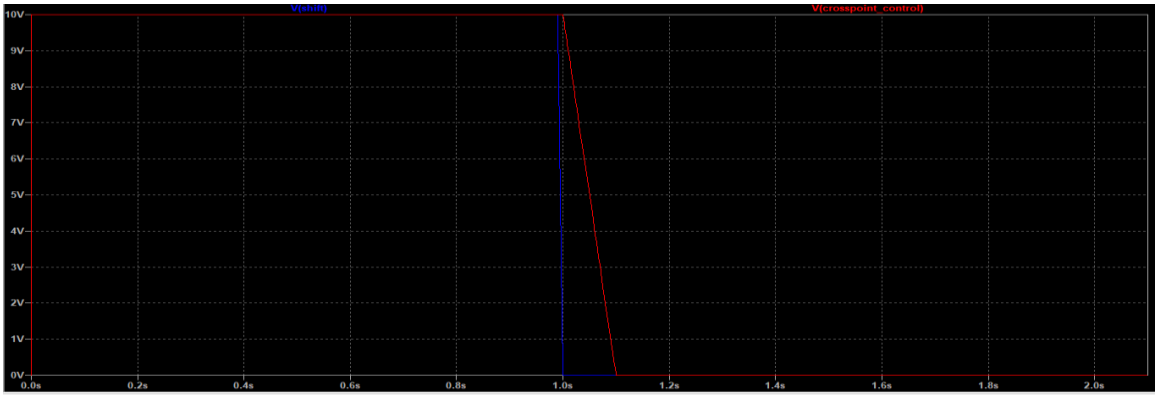


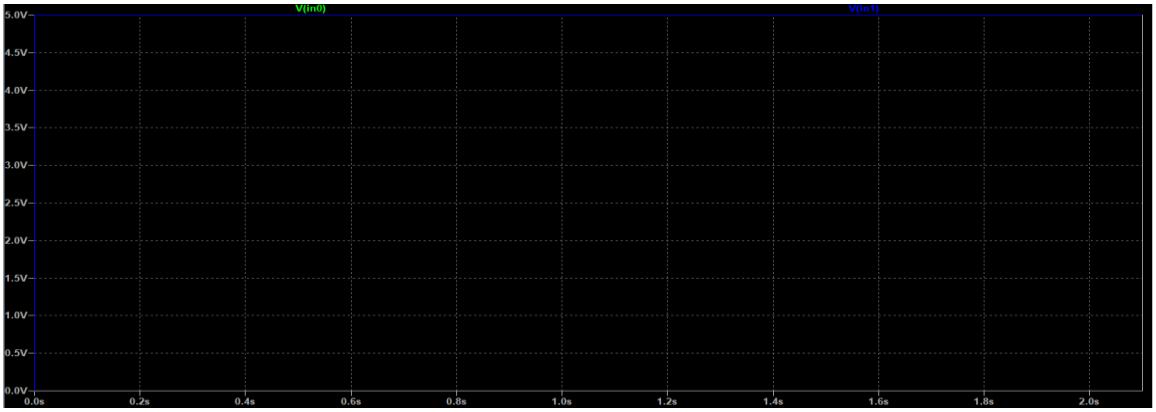
Figure 22: Complete structure of the 2X4 reconfigurable memristive crossbar nano-architecture for efficient multiplication


V(n001)
V(n002)
V(n003)
V(n004)
7.0V
6.3V
5.6V
4.9V
4.2V
3.5V
2.8V
2.1V
1.4V
0.7V
0.0V
0.0s
0.2s
0.4s
0.6s
0.8s
1.0s
1.2s
1.4s
1.6s
1.8s
2.0s

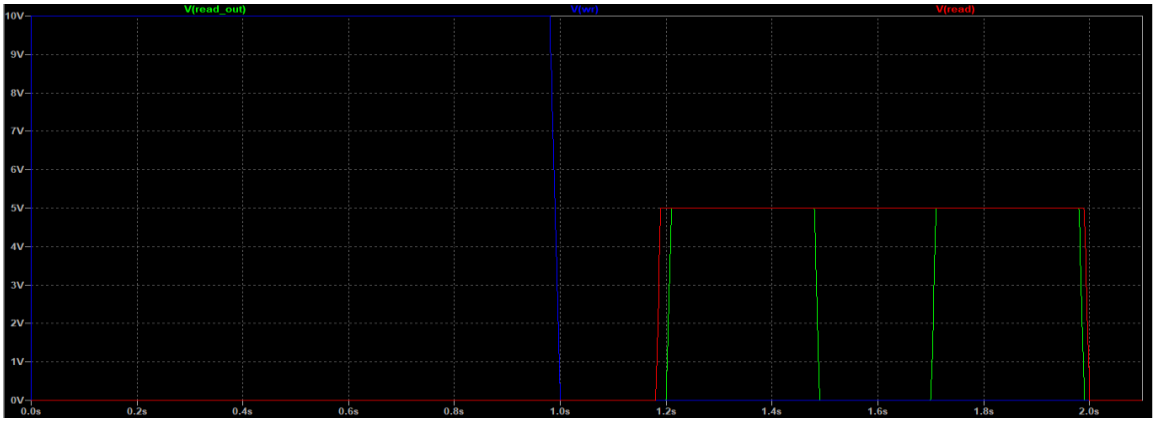
(a)


V(shift)
V(crosspoint_control)
10V
9V
8V
7V
6V
5V
4V
3V
2V
1V
0V
0.0s
0.2s
0.4s
0.6s
0.8s
1.0s
1.2s
1.4s
1.6s
1.8s
2.0s

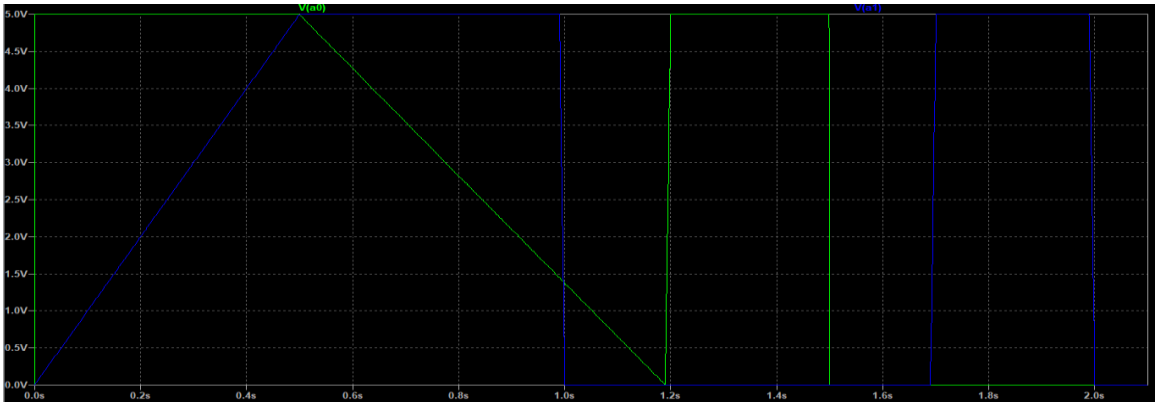
(b)


V(in0)
V(in1)
5.0V
4.5V
4.0V
3.5V
3.0V
2.5V
2.0V
1.5V
1.0V
0.5V
0.0V
0.0s
0.2s
0.4s
0.6s
0.8s
1.0s
1.2s
1.4s
1.6s
1.8s
2.0s

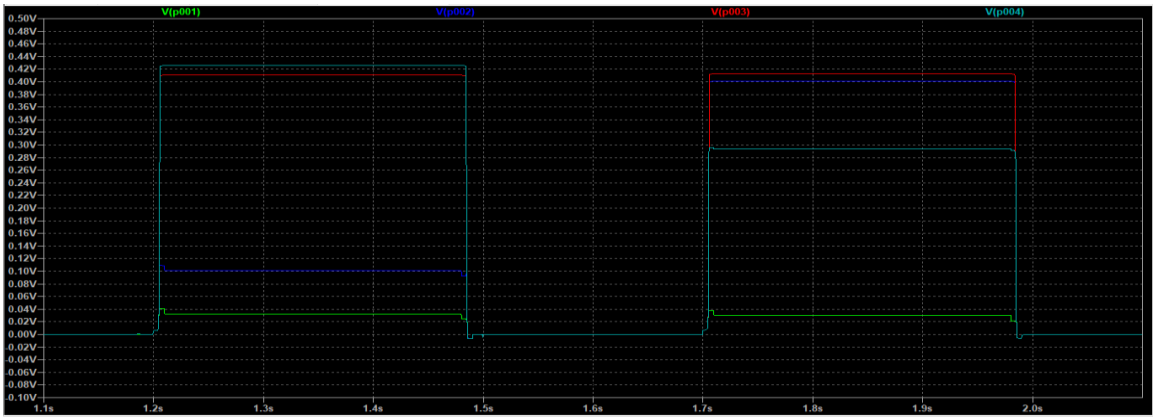
(c)



(d)



(e)



(f)

Figure 23: Simulation results for the 2X4 hybrid memristive crossbar (a) Input Signals (b) Configuration signals (c)Multiplier Signals (d) Read/Write Signals (e) Row Select Signals (f) Output Levels

Using the simulation results shown above, the efficient method to obtain partial products can be validated. After application of valid programming signals, it is observed that the output of consecutive rows are multiplicand shifted by the number of rows prior to the row read if the multiplier is 1 and 0 is programmed in each switch if the multiplier is 0. The usage of the crossbar structures in the proposed ALU is detailed in the following section.

CHAPTER 6: HIGH-RADIX ALU

This chapter gives details about the proposed architecture of a high radix ALU utilizing the Memristor crossbar structures proposed in former sections. The design principles are based on the fact that efficient computations can be carried out using dedicated hardware for each operation. Further, each operation block utilizes high-radix memristor based circuits for facilitating the need for low power consumption and lesser area.

6.1 Proposed ALU Architecture

Data computation as well as its storage within a single unit can be achieved by utilizing an embedded memory architecture. When CMOS decoding and processing circuits are used with a memristive multi-level memory, we can perform memory-aided computation and thus create of memristive Arithmetic and Logic Units (ALUs) with storage capability. Memristive ALUs facilitate the implementation of faster and more efficient arithmetic algorithms as well as provide data storage in the same unit.

In this project, a memristive ALU design has been presented which is based on the reconfigurable hybrid crossbar-based multi-level memory. Using the reconfigurable high-radix crossbar memory, a fast and algorithm for multiplication of radix-4 numbers has been implemented.

The proposed architecture consist of three basic blocks, namely, Input processing block, Dedicated multiplication block and Adder/Logical blocks. Each block performs essential operations with defined external interfaces. These interfaces are elaborated in the following subsections.

6.1.1 Input Processing Block

Input processing block performs binary to high radix conversion of the input operand and loads the output on a bus which is connected to other blocks in the ALU. Consider a N-bit operand with CMOS compatible logic levels, in order to convert the N-bit number into Radix-4, in this instance, 2 adjacent bits are clubbed starting from LSB and the output is obtained in the form of MSS compatible logic levels as shown in the following table:

Table 1: MSS compatible output levels

Input Bits	MSS compatible output level(V)
00	0
01	2.5
10	4.5
11	6.5

Any type of DAC that can provide correct voltage levels corresponding to the input level carries out this conversion. Further, for a N-bit input N/2 DAC's are required to enable parallel conversion of the input signals. Hence, the input processing block has N/2 output for N-bit output and uses N/2 DAC's.

Following the application of input signals, the control signal enable is applied to perform switch on the DAC array and perform conversion. After the conversion is completed, the control signal load enable is applied in order to load the data on the bus interface of the ALU. This N/2 bit bus is connected to other blocks in the ALU, after a specific amount of time when the data is successfully loaded into the other block's storage unit for processing, the control signal load_enable is removed.

6.1.2 Multiplication Processing Block

Multiplication processing block is designed to obtain faster multiplication operations with significantly lesser footprint. Multiplication is carried out in 2 parts, initially, multiplicand is applied to the input processing block and the high radix output is loaded on the bus. At the same time the binary multiplier is directly applied to input port 'multiplier' of the enhanced crossbar. This is followed by the application of control signals Enable block A, Shift, Write which stores the shifted partial products in the memristor crossbar, for instance is the multiplier is 0, the partial product $(000\ldots0)_4$ is stored in the crossbar row and if the multiplier is 1 then high radix multiplicand is stored in crossbar. The shift signal enables an arithmetic shit of 1 bit for each row. Therefore after the propagation of input signal in each row with simultaneous application of multiplier signal, the crossbar stores the partial products for multiplication operation.

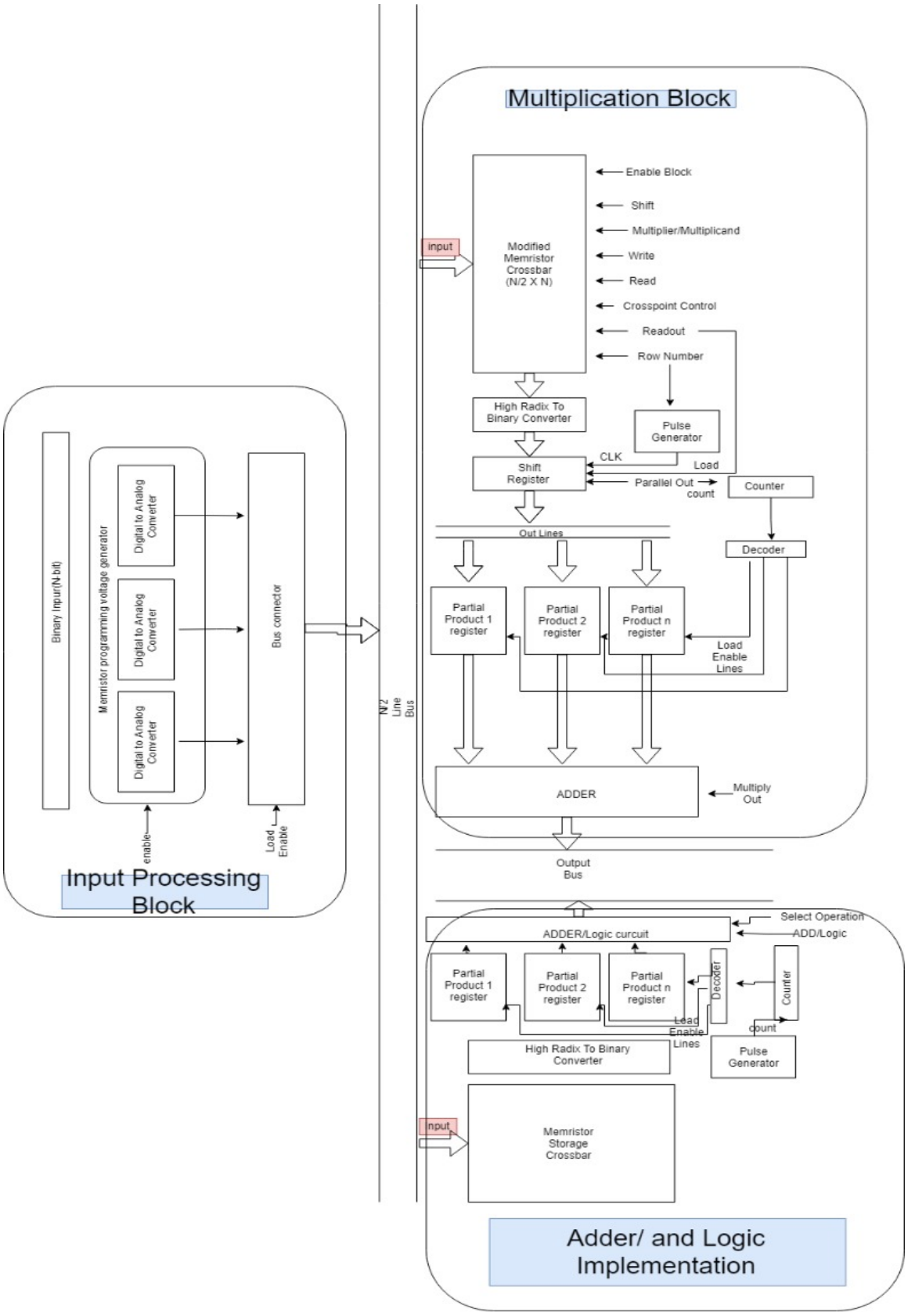


Figure 24: Block Diagram of the proposed memristive ALU design

These partial products must be added to obtain the final output. However, since arithmetic left shift of one bit leads to multiplication by 4, to obtain the final partial products, the output of the crossbar must be adjusted to obtain correct binary partial products. Which is done by performing arithmetic right shifts on binary converted output of the crossbar. This process is carried out by the shit register circuit shown in the ALU architecture. Finally, the partial products are stored in registers and added using a conventional CMOS adder.

6.1.3 Adder/Logical operation blocks

Similar to the multiplier block, the adder/logical block utilizes a crossbar memory for storing high radix operands, which are then read one by one and stored in conventional registers. These registers are then connected to the desired operation block which takes parallel inputs. The operation is performed and the output is loaded onto the common output bus. The advantage of using memory aided computation is that the previously stored operands can be readily used in new computation problems without using CPU time for fetching the operand from the main memory. Thus, the memristor crossbar serves as a local cache where each operand can be stored in memory location.

CHAPTER 7: CONCLUSION AND FUTURE WORK

A high-radix ALU based on composite memristor systems has been presented in this project. The proposed design is efficient in terms of power consumption and area and offer higher density. In order to demonstrate the validity of high-radix memristor-based design, a hybrid multi-level memristive crossbar with built-in memory has been implemented in LTSPICE and the multiplication operation has been successfully demonstrated. This reconfigurable crossbar circuit has been thus verified to enable faster arithmetic operations through memory-aided computing. The area of memristive crossbar circuits has been further explored by the implementation of a normal 3X4 memristive multi-level crossbar memory, along with the implementation of decoding circuitry i.e. the radix-4 to binary convertor. Because of the use of radix-4 cross-point elements, this memory has a smaller area and greater storage density as compared to classic CMOS memories of the same size. Further, the basic unit of all these high-radix systems-the 2-bit memristive storage cell has been designed using the concept of composite memristors, simulated and its data integrity during read and write operations has been verified. Thus, using composite memristor systems, we have been able to achieve storage of multiple bits in a unit memory cell, which enables us to design faster arithmetic circuits via high-radix computations. For all these simulations, we have used the LTSPICE simulator and the Threshold Type Adaptive Memristor Circuit Model.

In future, high-radix circuits can be used for the design and implementation of complex algorithms like the Fast Fourier Transform. Also, multi-level crossbar circuits can be designed for logical operations.

REFERENCES

[1] Vourkas, Ioannis, and Georgios Ch Sirakoulis. *Memristor-based nanoelectronic computing circuits and architectures*. Vol. 19. Cham: Springer International Publishing, 2016.

[2] Chua, Leon. "Memristor-the missing circuit element." *IEEE Transactions on circuit theory* 18.5 (1971): 507-519.

[3] Ho, Yenpo, Garng M. Huang, and Peng Li. "Dynamical properties and design analysis for nonvolatile memristor memories." *IEEE Transactions on Circuits and Systems I: Regular Papers* 58.4 (2010): 724-736.

[4] Zhang, Yang, et al. "A novel design for memristor-based logic switch and crossbar circuits." *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.5 (2015): 1402-1411.

[5] Hamdioui, Said, et al. "Memristor based computation-in-memory architecture for data-intensive applications." *Proceedings of the 2015 design, automation & test in Europe conference & exhibition*. EDA Consortium, 2015.

[6] Strukov, Dmitri B., et al. "The missing memristor found." *nature* 453.7191 (2008): 80.

[7] Roska, Tamás, Marco Gilli, and Ákos Zarándy. "12th international workshop on cellular nanoscale networks and their applications. CNNA 2010. Berkeley, 2010." (2010): 479.

[8] Pickett, Matthew D., et al. "Switching dynamics in titanium dioxide memristive devices." *Journal of Applied Physics* 106.7 (2009): 074508.

[9] Kvatinsky, Shahar, et al. "TEAM: Threshold adaptive memristor model." *IEEE transactions on circuits and systems I: regular papers* 60.1 (2012): 211-221.

[10] Kvatinsky, Shahar, et al. "VTEAM: A general model for voltage-controlled memristors." *IEEE Transactions on Circuits and Systems II: Express Briefs* 62.8 (2015): 786-790.

[11] Joglekar, Yogesh N., and Stephen J. Wolf. "The elusive memristor: properties of basic electrical circuits." *European Journal of Physics* 30.4 (2009): 661.

[12] Biolek, Dalibor, Zdenek Biolek, and Viera Biolkova. "SPICE modeling of memristive, memcapacitative and meminductive systems." *2009 European Conference on Circuit Theory and Design*. IEEE, 2009.

[13] Prodromakis, Themistoklis, et al. "A versatile memristor model with nonlinear dopant kinetics." *IEEE transactions on electron devices* 58.9 (2011): 3099-3105.

[14] Yu, Juntang, et al. "A memristor model with piecewise window function." *Radioengineering* 22.4 (2013): 969-974.

[15] Radwan, Ahmed G., and Mohammed E. Fouda. *On the mathematical modeling of memristor, memcapacitor, and meminductor*. Vol. 26. New York: Springer, 2015.