Memristor-based Digital Logic Circuit Design

MS (Research) Thesis

By

SHRUTI SANDIP GHODKE

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2024

Memristor-based Digital Logic Circuit Design

A Thesis

Submitted in fulfillment of the

requirements for the award of the degree of

Master of Science (Research)

by

SHRUTI SANDIP GHODKE

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2024

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Memristor based Digital Logic Circuit Design** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF SCIENCE (RESEARCH)** and submitted in the **Department of Electrical Engineering, Indian Institute of Technology Indore,** is an authentic record of my own work carried out during the time period from August 2022 to June 2024 under the supervision of Prof. Shaibal Mukherjee, Professor, Department of Electrical Engineering, Indian Institute of Technology Indore, India.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

(27/08/2024)

Signature of the Student with Date

(Shruti Sandip Ghodke)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

27/8/2024

Signature of Thesis Supervisor with Date

(Prof. Shaibal Mukherjee)

Shruti Sandip Ghodke has successfully given his MS (Research) Oral Examination held on 27/8/2024

Signature of Chairperson, OEB with Date

27/8/2024

Signature of Thesis Supervisor with date

Signature of Convener, DPGC with date

Signature of Head of Department with date

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere appreciation to my thesis supervisor **Prof. Shaibal Mukherjee**, Professor, Department of Electrical Engineering, IIT Indore, his generous guidance, support, mentoring, constant encouragement, and suggestions without which this work would not have been possible.

I am extremely grateful to my PG student progress committee members **Prof. Mukesh Kumar** and **Prof. Somaditya Sen** for their encouragement, valuable advice, questions, insightful comments and co-operation throughout the work.

My sincere thanks go to **Prof. Suhas S. Joshi**, Director, IIT Indore for providing the essential lab facilities and supporting me for betterment of my academic career.

I would also like to express my gratitude to **Dr. Sandip Vilasrao Ghodke** (Father), and **Mrs. Sanjivani Sandip Ghodke** (Mother), for their excellent guidance throughout my life. Thank you for being the pillar of my strength and always believing in me. I am thankful to my brother **Mr. Atharv Sandip Ghodke** for continuous encouragement.

I would like to express my whole-hearted thanks to **Dr. Sanjay Kumar** and **Mr. Mayank Dubey** for their trustworthy help, discussions and unwavering support during my challenging times. I would like to thank **Mr. Saurabh Yadav** for assistance in my research work, for understanding me and being with me in all difficult conditions. Thanks to **Mr. Bramhadatta Mahapatra** for his wholesome company, discussions and continuously inspiring me. My thanks also go to other lab members **Dr. Chandrabhan Patel**, **Mr. Sumit Chaudhary**, **Ms. Kumari Jyoti**, **Mr. Vikas Kumar Verma**, **Mr. Lokesh Kumar Hindoliya**, **Mr. Ajit Yadav**, **Mr. Animesh Paul**, **Ms, Poonam Bajoria** and all pass out seniors at Hybrid Nanodevice Research Group (HNRG) for their motivation whenever required.

Shruti Sandip Ghodke

This Thesis is Dedicated

to

The Divine Power of the World
and My Parents

Abstract

The limitations of existing complementary metal-oxide-semiconductor (CMOS) technology in terms of area and power efficiency have catalyzed advanced research into alternative nanodevices. Memristors, with their unique switching properties, offes a promising avenue for replacing traditional CMOS technology in edge computing devices. This thesis presents the design, implementation, and performance evaluation of memristor-based combinational logic circuits, including adders, subtractors, and decoders, utilizing MATLAB Simulink and Cadence Virtuoso.

We proposed an optimized design for memristor-based combinational logic circuits and conducted a comparative study with conventional CMOS methods. The memristor model, experimentally validated for a high-density Y_2O_3-based memristive crossbar array which demonstrates ultralow device-to-device and cycle-to-cycle variability. Our findings indicate a power reduction of over 90% compared to CMOS technology implemented in Cadence Virtuoso, alongside a significant reduction in the number of components, enhancing area efficiency and facilitating the design of complex logic circuits on a micrometer scale.

Various logic gates were designed and implemented using MATLAB Simulink and industry-standard Verilog-A coding within the Cadence Virtuoso platform. The nonlinear analytical memristor model shows stable switching responses with minimal variability. Following a comprehensive understanding of the underlying principles, a new memristor-based logic circuit architecture was proposed, implemented in Verilog-A, and simulated using Cadence Virtuoso. Simulation results affirmed the feasibility of these logic circuits, highlighting significant improvements in power consumption, area utilization, and delay over traditional CMOS logic circuits.

The proposed architecture was rigorously tested for various logic functions, demonstrating accurate outputs for combinational logic, sequential logic, and complex circuits such as multiplexers, comparators, multipliers, and encoders/decoders. The

outcome of proposed work is implementation of analytical model for the advancement of memristor technology, demonstration and optimisation of logic circuits over the traditional CMOS-based logic circuits for low power, area and performance of logic circuits.

LIST OF PUBLICATIONS

- **Journal Publications:**

1. **S. S. Ghodke** , S. Kumar, S. Yadav, N. S. Dhakad, and S. Mukherjee, "Combinational logic circuits based on a power- and area-efficient memristor with low variability," Journal of Computational Electronics, vol. 23, no. 1. Springer Science and Business Media LLC, pp. 131–141, Dec. 13, 2023. (IF 2.1)

- **Conference Publications:**

1. Saurabh Yadav, Chandrabhan Patel, Sumit Chaudhary, Kumari Jyoti, **Shruti Ghodke** and Shaibal Mukherjee, "Bilayer MoS2 Based (2×2) Memristive Crossbar Array for Neuromorphic Applications", IEEE Nano 2024. (Accepted)

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

RRAM	-	Resistive Random Access Memory
NVRAM	-	Nonvolatile Random Access Memory
DRAM	-	Dynamic Random Access Memory
STT-MRAM	-	Spin-Transfer Torque magnetoresistive random access memory
MTJ	-	Magnetic Tunnel Junction
RS	-	Resistive Switching
LRS	-	Low Resistance State
HRS	-	High Resistance State
IMPLY	-	Material Implication
MAGIC	-	Memristor Aided Logic
MRL	-	Memristive Logic Proportional Circuit
MCA	-	Memristive Crossbar Array
D2D	-	Dvice-to-Device
C2C	-	Cycle-to-Cycle
MIM	-	Metal Insulator Metal
CMOS	-	Complementary Metal Oxide Semiconductor
FF	-	Flip Flop
SR	-	Set Reset Flip Flop
D FF	-	Data Flip Flop

Chapter 1

Introduction

1.1 Motivation

The onset of the Big Data era is driving the miniaturization of CMOS technology, but this transition presents challenges for non-volatile flash memory, which must navigate technical hurdles to keep pace with Moore's law inside the Von-Neumann computer framework. The reduction in device structures in CMOS technology leads to closer proximity between the floating gates of adjacent cells, heightening the risk of read errors and unintended shifts in threshold voltage due to charge floating between gates. Furthermore, diminishing oxide thickness exacerbates charge transfer issues, potentially decreasing the reliability of flash devices' programming, erasing, and retention capabilities. Addressing these scaling challenges requires alternative memory devices. Additionally, to meet the demands of high computing throughput and low-power applications, modern memory devices must offer either lower latency or increased capacity [1]. Although significant strides have been made in research on data retention, device size, endurance and power consumption over the past decade, traditional Von-Neumann computer architecture faces hurdles

in scaling memory devices, prompting exploration of alternative memory solutions like resistive random-access memory (RRAM).

In conventional computers, the processing and memory units are physically segregated, necessitating the constant shuffling of data during computation. This leads to a performance bottleneck known as the "von Neumann Bottleneck" Figure 1.1 [1]. This physical partitioning and the subsequent data transfers represent a significant challenge in traditional computing systems, as memory access uses 100–1000 times as much energy as processor activities. In contrast, in-memory computing conducts computation directly within the memory devices themselves, organized as a "computational memory" unit. For instance, if data A resides within a computational memory unit and we wish to perform function f(A), there's no need to transfer A to the processing unit Figure 1.1. This approach is more energy and time-efficient compared to conventional computing methods [2].

In contemporary computing, tasks like artificial intelligence (AI) and scientific computation require processing vast datasets simultaneously. However, in the Von-Neumann architecture, where memory and computation units are segregated, considerable time and energy are expended in data transfer, rendering this conventional setup inefficient for data-intensive operations. To address this memory bottleneck [3], parallel designs such as graphic processing units (GPUs) and specialized systems like tensor processing units (TPUs) have been introduced. Currently, metal oxide semiconductor field effect transistor (MOSFET) and complementary metal oxide semiconductor (CMOS) technology plays pivotal roles in supporting advanced functionalities in digital electronics and underpin mainstream computational methods.

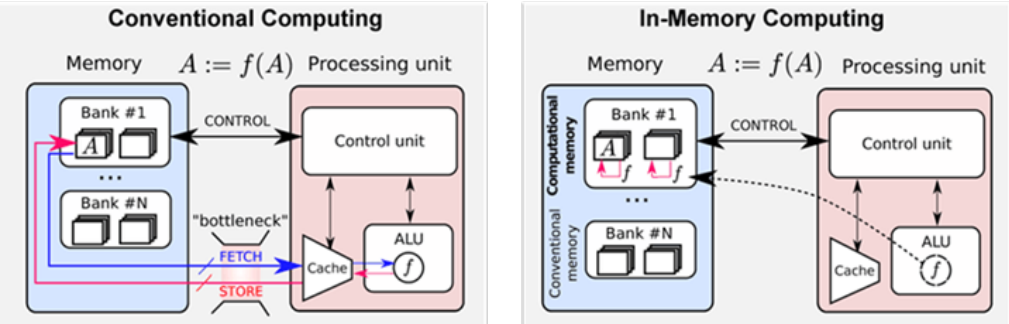

Figure 1.1: Processor architecture of conventional and In-Memory computing.

1.2 History Of Memristor

The memristor is a relatively new addition to the world of electronics, first conceptualized by Leon Chua in 1971. The term "memristor" is a combination of "memory" and "resistor," indicating its ability to 'remember' the quantity of charge that has flown through it. Chua theorized that a 4^{th} fundamental circuit element, in addition to the capacitor, resistor and inductor, was needed to complete the theoretical framework of electronics [4].

Since then, memristors have been extensively studied for their potential applications in various fields, including artificial intelligence, neuromorphic computing, and non-volatile memory storage. While memristor-based technologies are still in the research and development phase, they hold promise for revolutionizing computing and electronics in the future.

However, it wasn't until 2008, that innovators at HP Labs, led by R. Stanley Williams, announced the creation of working memristor. Their device was capable of remembering its resistance state even when the power was turned off, akin to non-volatile memory. This breakthrough opened up possibilities for a new generation of computer memory and processing systems, promising faster speeds and reduced energy consumption compared to conventional

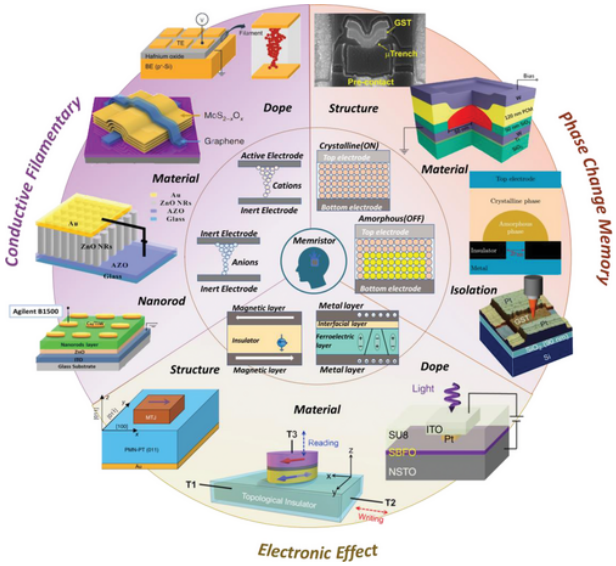

Figure 1.2: An overview of resistive switching mechanisms and associated techniques for device optimization.

technologies [5]. Memristor offers unprecedented opportunities for advancing both theoretical understanding and practical applications [6]. The journey of memristor technology from a theoretical concept to experimental realization has been marked by significant milestones and breakthroughs. Since then, memristors have garnered widespread attention for their potential to address critical challenges in diverse fields, ranging from computing and memory to neuromorphic engineering and beyond as illustrated in Figure 1.2 [7].

1.3 Memristor I-V Characteristics and Properties

The memristor has captivated researchers and engineers due its unique property of remembering, positioning it as the fourth vital element of a circuit, following capacitors, resistors, and inductors as shown in Figure 1.3 [8]. A crucial characteristic of memristors is their constricted hysteresis loop. When the I-V

characteristics go through the origin, this happens. A memristor that is operated by a periodic signal exhibits an origin-pinched hysteresis loop. This loop's non-zero region shows memory behavior [9] (as shown in Figure 1.4 [10]). The great reliability capabilities of RS devices is one of their most important characteristics for any application. This refers to the maximum number of RS programming cycles that a device may withstand before its electrical properties start to deviate from the permitted bounds. One operating cycle in bistable RS devices consists of one set transition and one reset transition. Retention times are the lengths of resistance states (t_{HRS} and t_{LRS}) in the absence of electrical stress. Important characteristics of RS devices for additional applications include linearity in multiple state electronic synapses for artificial neural networks (ANNs), switching curve in selectors and electronic neurons, and LRS and HRS capacitance in radio frequency switches [11].

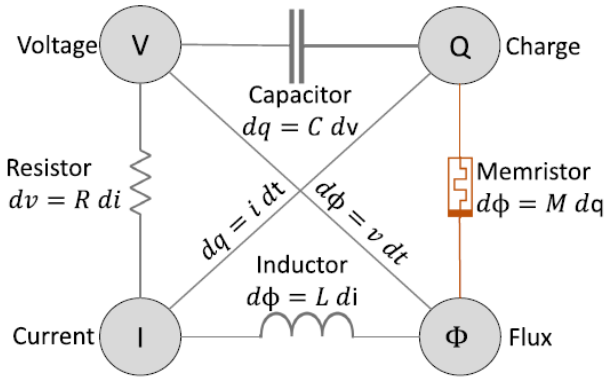

Figure 1.3: Fundamentals of electrical components in networks.

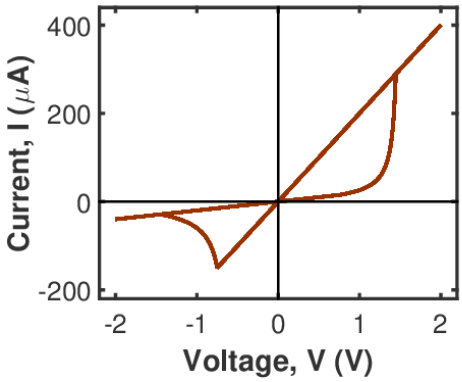

Figure 1.4: I-V characteristics of memristor model

1.4 Switching Mechanism and Materials For Neuromorphic Devices

Neuromorphic devices can be broadly categorized into two basic groups based on operational principles. The behaviors of neuromorphic devices are determined by these many functioning mechanisms, which also influence the design of the devices' materials and structural architecture. This section will include a thorough explanation of these mechanisms and a list of the suitable material systems for each category [12].

1.4.1 Filamentary type

Figure 1.5 [12] depicts the basic metal/insulator/metal sandwich construction of the anion migration-based filamentary device. The insulating layer's resistance may alter dramatically when the anode is given the proper voltage. Even after the voltage pulse has stopped, this kind of resistance shift can continue. Thus, since the 1960s, this type of device—known as RRAM—has been employed as memory. For this reason, the resistive switching layer is another name

for the insulating layer. RRAMs are scalable down to less than 10nm, have strong retention if greater than 10 years at 85° C, rapid read/write times of less than 1ns, and a high endurance of greater than 10^{12} cycles. A conductive filament made up of an area with a higher concentration of oxygen vacancies forms in these RRAMs between the two metallic electrodes. Filamentary RRAM is the first type of memristor discovered in 2008, and since then, it has been used as a typical device in neuromorphic computing due to its memory effect and correlation to the memristor equation. In filamentary type devices, two typical switching characteristics—bipolar switching and unipolar switching—are exhibited. In unipolar switching, same polarity voltage pulses are used to perform the set and reset operations, but in bipolar switching, the voltage pulses used to induce these operations must be of distinct polarities. This illustrates the distinct dynamics employed for these two forms of switching, the thermochemical mechanism (TCM) and the valence change mechanism (VCM), respectively.

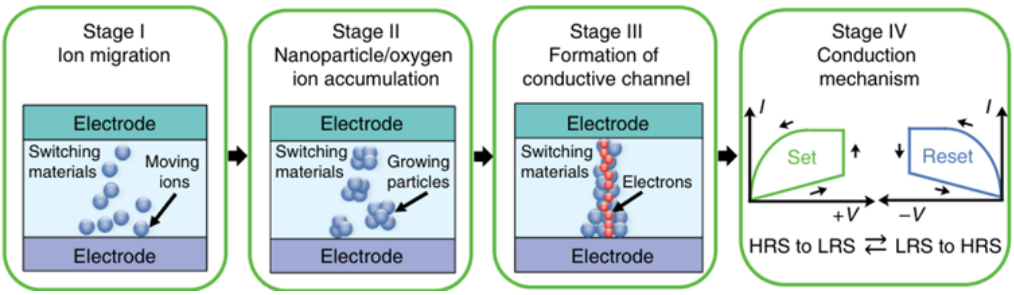

Figure 1.5: Filamentary switching mechanism.

1.4.2 Non-filamentary/ Interfacial type

Further scaling down of filamentary type devices has proven to be extremely challenging due to the stochastic nature of filament creation and rupture. As a result, non-filamentary devices have gained popularity recently, particularly

those based on interfacial reactions, as they have shown good *D2D* consistency. They have received a lot of attention lately. There are no vacancies in the zero-bias scenario, as Figure 1.6 [8] illustrates. The creation of oxygen vacancies occurs close to the interface between LSMO and ZnO because a positive voltage is provided to the top electrode while the bottom electrode is grounded. When the voltage is raised, the oxygen ions in the ZnO gain enough energy to begin migrating from their lattice locations to the top electrode, where they fill the oxygen vacancies. The oxygen vacancies produce the conducting filament type structure at V_{set} when they align in a specific orientation. The device now switches from an HRS to an LRS state of resistance. The negatively charged oxygen ions then begin to migrate toward the bottom electrode when the negative voltage is delivered to the top electrode. The conducting filaments created by the oxygen vacancies are ruptured and the sample's state of resistance changes from LRS to HRS at V_{reset} because all of the oxygen vacancies are filled by the oxygen ions.Therefore, the primary cause of the RS process in the Ag/LSMO/ZnO/ITO device is the development or rupture of conducting filament type structure by the application of positive or negative voltage.

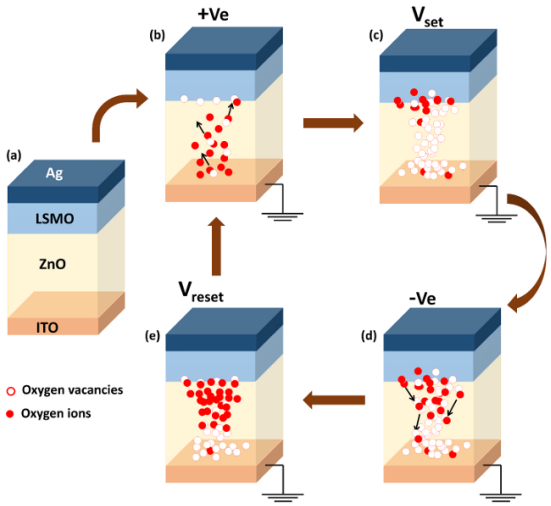

Figure 1.6: Interfacial switching mechanism

1.5 Applications of Memristor

Since May 2008, when HP Labs unveiled their seminal paper in Nature, heralding the discovery of the memristor, a significant surge of interest rippled through research and engineering communities globally. This momentum catalysed a fervent exploration of applications for this groundbreaking device. Efforts were directed towards integrating the memristor into circuit architectures to leverage its unique features, as well as adapting existing configurations by incorporating memristors or harnessing their novel properties to develop innovative architectures. Consequently, a plethora of articles and papers have emerged over the past three to four years. Within this section, we delve into descriptions of some of the more prevalent or promising applications of the memristor which can be seen in Figure 1.7 [12].

Applications of memristors span various fields, with notable emphasis on non-volatile memory:

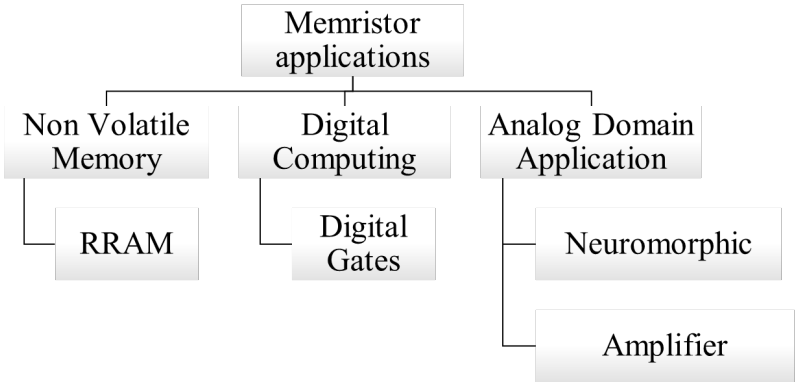

Figure 1.7: Applications of memristor.

1. **Non-volatile memory**: Non-volatile memristor memory is one of the most practical uses of memristors, and it is expected to be widely used in the near future. Memristors possess the unique ability to retain their previous state even when in the OFF mode, akin to a form of memory. Due to this feature, memristors are attractive options for non-volatile random access memory (NVRAMs). Research literature has extensively explored the integration of memristors into memory architectures, particularly dense crossbar arrays Figure 1.8 [13]. Notably, fabrication technology for 3 nm memristors is already accessible. Crossbar latch memory has been successfully created by HP Labs, however it is currently 10 times slower than DRAMs.

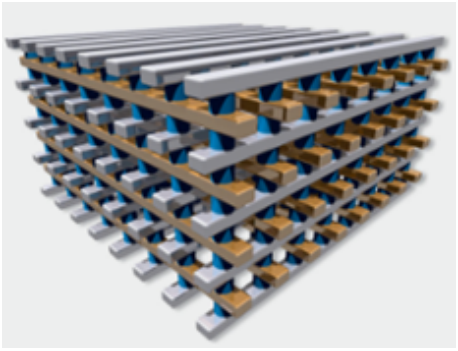

Figure 1.8: Non-volatile memory application of memristor

2. **Resistive RAMs (ReRAMs)**: ReRAMs, also known as Resistance Switching Memories or memristor memories, are the focus of research endeavours in numerous companies worldwide. Organizations such as IMEC, Sharp, Fujitsu, HP, and Unity Semiconductor have dedicated funds to the development of memristive materials. The two logical values "1" and "0," which stand for ON mode and OFF mode, respectively, are the center of these memory' operation. Transition metal oxides serve as pivotal materials in these structures, with notable examples including NiO, (TiO_2), (ZrO_2), ($SrTiO_3$), and (HfO_2). Two distinct switching mechanisms—unipolar and bipolar—are observed in these materials. Potential can be applied throughout the device to change its state. Unipolar switching devices need a voltage greater than two threshold voltages to begin changing states, but bipolar switching devices require a negative voltage to return to their initial state. Figure 1.9 shows application of memristor in biological synapse [14].

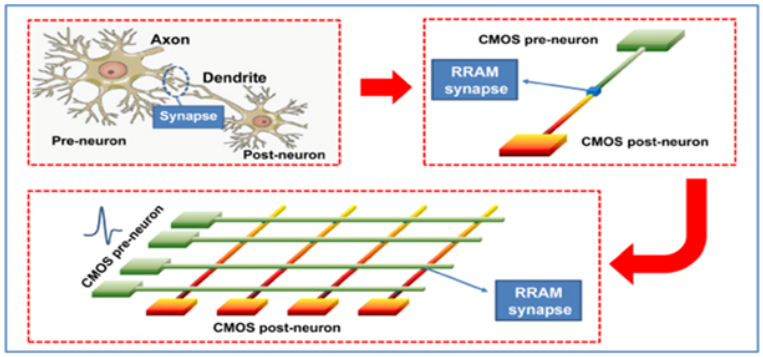

Figure 1.9: Biological synapse and its replication in RRAM Synapse.

1.6 Challenges and Perspectives

The challenges facing memristors in the realm of neuromorphic devices have been prominent for over a decade, despite significant research and notable progress. Even while some devices, such as STT-MRAM, phase change devices, and filamentary anion/cation migration-based devices, have made their way into the market, enduring problems still exist. Further use of filamentary devices is hindered by the intrinsic stochasticity of filament growth, and complexity is increased by the laborious procedure of using multi-buffer layers to nurture superior MTJ on substrates [15].

Addressing these challenges necessitates concerted efforts. New neuromorphic devices that meet requirements like stability, high on/off resistance ratio, and scalability—such as ferroelectric, MIT-based, nonfilamentary anion migration-based, and electrolyte-gated devices need to be developed further. Furthermore, enhancing the linearity and symmetry of synaptic devices, crucial for accurate computations, remains a priority. Certain emerging neuromorphic devices exhibit promising linearity improvements, yet refining resistance states and available state ranges demands further exploration.

Additionally, simulating internal ion dynamics within synapses offers a promising avenue for advancing artificial synaptic devices and emulating the complexity of human brains. Novel circuits or arrays tailored to complex neuron models further contribute to the evolution of artificial neuromorphic networks.

Navigating the future trajectory involves strategic contemplation regarding device development, fabrication methodologies, algorithms, and integration approaches for neuromorphic computing. Despite the flourishing research at the network level, bridging gaps in neuroscience knowledge remains imperative. Anticipating breakthroughs in materials science and physics holds potential for significantly enhancing the performance of neuromorphic devices.

1.7 Organization of Thesis

The sections of the thesis are structured as follows:

Chapter 1: This chapter covers outline the specific goals and objectives of the study, define what a memristor is and its fundamental properties, trace the historical development and significant milestones in memristor research. It discusses the resistive switching mechanism in memristors and various applications of memristors in neuromorphic computing, memory storage, and other fields.

Chapter 2: It introduces the importance of analytical models in understanding and designing memristors. Described the various memristive analytical models, its characteristics, and specifications. Compared the other models with the new research model and discussed parameter optimization and the performance of the new model compared to existing models. Lastly, summarizing the key findings and their implications for future research and applications.

Chapter 3: In this chapter the significance of implementing logic gates using

memristors is discussed and introduced MATLAB Simulink as a simulation tool.This chapter presents Design, implementation, and simulation results of basic logic circuits and some combinational logic circuits.

Chapter 4: Experimental implementations are detailed in this chapter. The importance of sequential logic circuits in digital systems and the role of memristors in their implementation is showcased in this chapter. the implementation of sequential and some complex logic circuits are detailed.

Chapter 5: This section draws conclusions from the proposed logic design by discussing the comparative analysis of power, performance, and area parameters of the designed circuits and outlines the future scope of the work.

Chapter 2

Memristor Analytical Model and Design Methodology

Interest in researching the emerging memristive devices has surged due to their demonstrated non-volatile memristive behavior, which holds promise for applications in non-volatile memory technology. These devices are being viewed as potential substitutes for flash memory technology, especially in the era of data-centric computing where there is a growing demand for memory technologies that align with present and future needs. Among the various emerging devices, RRAM devices stand out for their scalability, high density, low power consumption, speed, endurance, retention, and compatibility with CMOS technology [16]. They have emerged as one of the most popular non-volatile memory technologies, inspiring a great deal of study to understand their workings and provide models for precise device functioning and streamlined architectures. When an RRAM device is in the *SET* or ON state, it is indicated by a *(LRS)*, and when it is in the *RESET* or OFF state, by a *(HRS)*. RRAM devices normally have a simple two-terminal metal-insulator-metal (MIM) configuration. The device stores data bits by switching between these

resistance levels. Based on their polarity of switching, RRAM devices can be classified as bipolar or unipolar; unipolar devices switch on a single polarity bias, whereas bipolar devices require bias from both polarities.

Memristors are currently used in a variety of applications, including memory, synapses, neuromorphic computing, deep neural networks, and logic gates. In logic gate applications, the logic states "0" and "1" are represented by the *(HRS)* and *(LRS)* of the memristor, respectively. This study utilizes an analytical model of a memristor in MATLAB Simulink and Cadence Virtuoso to verify its current–voltage *(I–V)* characteristics. After that, the model is combined with additional circuit elements to carry out logic operations using a variety of combinational logic gates, such as two- and three-input decoders, half adders, full adders, and half subtractors. A detailed comparison of these logic gates with existing CMOS technology is also provided.

2.1 Memristor Modeling: Motivation

In the realm of new semiconductor technologies, the significance of modeling cannot be overstated. Developing accurate and comprehensive models is crucial for understanding device operation, optimizing performance, and ensuring compliance with required specifications. Several models have been proposed, each with its own level of accuracy, features, and outcomes. Therefore, developers seeking to create robust and adaptable models for RRAM devices must familiarize themselves with past methodologies and encountered limitations.

Previous studies have offered insights into RRAM device mechanisms, fabrication technologies, material stacks, and some existing models. Recently, RRAM modeling theories are consolidated and proposed an optimized model.

Our focus here lies on exploring various modeling techniques and addressing associated challenges. We delve into boundary condition models, categorized as pseudo-compact models, and examine critical modeling techniques that can significantly aid developers. Additionally, we discuss simulation techniques and platforms, such as SPICE, essential for RRAM model development.

2.2 Linear Ion Drift Model

HP provided a model based on the device's physical architecture to help others understand their creation [17]. It is conjectured that the device, whose width is represented by the symbol D, is composed of two discrete regions: one that is doped with positive oxygen ions, or oxygen vacancies, and the other that is not. A resistor is used to symbolize each zone. The zone that is doped, with width w acting as the state variable, is more conductive due to its reduced resistance, while the undoped region has a high resistance. The model also assumes equal average ion mobility μv, linear ion drift, uniform electric field distribution, and ohmic conductance. Equation 2.1 shows the memristor model developed by HP in 2008. The state equation can be expressed with the assumption that the electric field is homogeneous throughout the device. Moreover, the equation for the current-voltage relationship is given. It is clear from the aforementioned relationship that memristance properties are much improved by smaller values of D, which indicate thinner devices. This finding clarifies why memristance is more common in electronics at the nanoscale.

$$\mathbf{i}(t) = \frac{v(t)}{R_{OFF}(1 - \sqrt{1 - \frac{2\mu_D}{rD^2}\varphi(t)})} \tag{2.1}$$

2.3 Non-Linear Ion Drift Model

A voltage-controlled memristor with a nonlinear relationship between voltage and the derivative of its internal state is proposed and modeled within the context of the nonlinear ion drift model. Additionally, this model explains asymmetric switching behavior. Although the linear drift model describes the hysteresis features of the memristor, it has problems with basic electrodynamics. Empirical evidence and experiments have demonstrated the intrinsic nonlinearity of implemented memristors, rendering the linear ion drift model inaccurate. Particularly for applications such as logic circuits, nonlinear characteristics are imperative, leading to the development of more suitable models.

First introduced by HP, the linear ion drift model focuses on linear drift effects in the bulk of the memristor device. Although nonlinear effects at the boundaries were observed, they were not comprehensively defined. The nonlinear dependence of dopant drift on applied voltage was identified and formulated by Yang *et al.* in 2008 [16], who provided an accurate current-voltage relationship that took these effects into account. This knowledge was further improved by later work by Mika Laiho and Eero Lehtonen.

Additionally, Yang *et al.*[18] reported that a spatially heterogeneous metal/oxide electronic barrier controls conduction in memristive devices. Positively charged oxygen vacancies drift across this barrier and serve as native dopants, creating or dissolving conductive channels that cause switching. At borders or the metal/oxide interface, the concentration of vacancies is higher [19]. Interestingly, only at the top interface can ON/OFF switching take place, indicating that the top electrode functions as the active electrode.

Fitting constants $\beta, \gamma, \eta,$, and χ are represented in Equation 2.2. The memristor's ON state, in which electrons tunnel through the thin residual

electronic barrier, is roughly represented by the first term β sinh (αv). The state variable of the device, denoted as w, ranges from 0 (OFF) to 1 (ON).

Based on the aforementioned relationship, it can be inferred that smaller values of D (indicating thinner devices) result in significantly improved memristance characteristics. This observation underscores why memristance is more prominent in nanoscale devices.

$$I = W^n \beta sinh(\alpha v) + X(e^{\gamma v} - 1) \tag{2.2}$$

2.4 Yakopic Model

The Yakopcic model closely resembled several RRAM devices, while not having been verified for RRAM devices explicitly throughout its development. Based on the Pickett-Abdalla model, which used a similar state variable but was modified to include neuromorphic systems, this model was first evaluated for TiO_2 systems, which are among the most common alongside HfO_2-based RRAM devices [20]. This model was among the first to incorporate synaptic functioning into its equations, and it was approved for use with the tool the HP lab team uses to explain memristive system functions. The device's dynamic resistance and current are directly influenced by the state variable *w(t)*, which is restricted between zero and one. The current within device is described as Equation 2.3:

$$\mathbf{I}(t) = \begin{cases} a_1 W(t) sinh(bv(t)), & v(t) \geq 0 \\ a_2 W(t) sinh(bv(t)), & v(t) < 0; \end{cases} \tag{2.3}$$

This function essentially acts as a window, constraining the state change

variable within specific boundaries, as given by Equation 2.4, 2.5:

$$\mathbf{f}(W) = \begin{cases} e^{-\alpha_p(W-W_p)} f_p(W, W_p), & W \geq W_p \\ 1, & W < W_p \end{cases} \tag{2.4}$$

$$\mathbf{f}(W) = \begin{cases} e^{-\alpha_n(W+W_n-1)} f_n(W, W_n), & W \leq 1 - W_n \\ 1, & W > 1 - W_n \end{cases} \tag{2.5}$$

$$\frac{dW}{dt} = g(v(t)) \times f(W(t)) \tag{2.6}$$

Here, *fp(w, w_p)* and *fn(w, w_n)* represent window functions that limit the value of *f(w)* under certain conditions. The analytical nature of the coupled equations allows for their solution using mathematical solvers like MATLAB. These equations can also be solved in MATLAB using built-in solvers such as *idt()* and *ddt(),* employing time step integration methods. The model was simulated using characterization data from HP Labs for TiO_2 memristors, yielding satisfactory fitting when calibration of fitting parameters was properly executed [21].

2.5 Our Research Model

A semi-empirical model was recently reported by Kumar et al. [22] to address discrepancies pertaining to the absence of a non-linear profile in the drift current at device borders. Nevertheless, the physics of a memristive system based on the interfacial switching mechanism is not entirely captured by this model. The stated model [22] has been validated with experimental results on Y_2O_3 and WO_3 based memristive systems, showing maximum error deviation MED of approximately 4.44%.

The suggested non-linear analytical model, which is represented as a parallel connection between a memristor and a rectifier, offers a more accurate

non-linear profile and takes into consideration non-ideal effects like resistive switching characteristics' rectifying nature and asymmetrical and non-zero crossing switching characteristics. The newly proposed piecewise window function satisfies all necessary conditions.

The non-linear analytical model, synaptic plasticity functionality, and synaptic learning behavior of WO_3 and Y_2O_3 -based memristive systems are described in depth in this study. Equation 2.7 governs the *I–V* relationship for the suggested model.

While previously described models are limited to bipolar systems, the suggested analytical model can be used to both unipolar and bipolar memristive systems. Under positive bias conditions, memristive devices generally show a higher hysteresis loop area than under negative bias conditions. This is because, in Y_2O_3-based memristive systems under positive bias, oxygen vacancies migrate in the direction of the inte rface between the Al top electrode and the Y_2O_3 switching oxide layer. The resistive switching phenomena at the Al/Y_2O_3 contact is largely dependent on these movements. Similar resistive switching phenomena have been experimentally reported by Chang et al. for WO_3 based memristive systems at the Pt/WO_3 interface.

The proposed nonlinear model [22] is applicable to both bipolar and unipolar memristive systems. Equation 2.7 below illustrates the I–V relationship of the memristor.

$$\mathbf{I}(t) = \begin{cases} b_1 w^{a_1}(e^{\alpha_1 Vi(t)} - 1) + \chi(e^{\gamma V_i(t)} - 1) & V_i(t) \geq 0 \\ b_2 w^{a_2}(e^{\alpha_2 V_i(t)} - 1) + \chi(e^{\gamma V_i(t)} - 1) & V_i(t) \leq 0 \end{cases} \tag{2.7}$$

Here, Equation 2.8. defines the piecewise window function *f (w)* which ensures that w [0,1]. Equation 2.9. illustrates the state variable derivative in the time domain, with A and m denoting the effect of the input voltage on the state

variable.

$$\mathbf{f}(w) = log \begin{Bmatrix} (1+W)^p, & 0 \leq W \leq 0.1 \\ (1.1)^p, & 0.1 \leq W \leq 0.9 \\ (2-W)^p, & 0 \leq W \leq 1 \end{Bmatrix} \tag{2.8}$$

$$\frac{dW}{dt} = A \times V_i^m(t) \times f(W) \tag{2.9}$$

The time derivative of the state variable w(t) is determined by equation (2.9), and it depends on the window function and input voltage properties. where A and m stand for parameters that determine how the state variable and input voltage are related, and m is usually an odd number to guarantee that changes in the applied voltage's polarity cause corresponding changes in the state variable's rate.

Table 2.1 presents a comparison and highlights the fundamental differences between our proposed model and other reported models. Notably, the piecewise window function used in our model offers superior controllability over the nonlinear analytical model in terms of bipolar resistive switching. It is effectively applicable within a lower input voltage window, requires minimal parameters as degrees of freedom, and provides better tunability in device conductance. This enhances the proposed analytical model's capability to perform both analog and digital logic operations. Furthermore, experimental validation confirms its real-time application with lower variability, offering significant advantages in its novelty compared to the data reported in the literature [22].

Table 2.1: Comparison of resistive swiching model

Model	Device type	State variable	Control mechanism	Simulation compatible
Linear ion drift	Bipolar	$0 \leq w \leq D$ Doped region physical width	Current	SPICE
Non-linear ion drift	Bipolar	$0 \leq w \leq 1$ Doped region normalized width	Voltage	No
Yakopcic	Bipolar	$0 \leq w \leq 1$ Not explained physically	Voltage	SPICE/ Verilog/MAPP
Our Research Model	Bipolar	$0 \leq w \leq 1$ Not explained physically	Voltage	MATLAB

For neuromorphic computing applications, the suggested general non-linear analytical model can be used to any material-based memristive system displaying resistive switching, synaptic learning, and synaptic plasticity properties. This model takes into consideration rectifying nature and non-ideal effects that other published models in the literature do not take into account, and together with its piecewise window function, it displays enhanced non-linear behavior in device current at device boundaries. In comparison to equivalent experimental results, the synaptic plasticity characteristic of the modeled data shows roughly 4.44% MED for Y_2O_3, which are the lowest values published to date. This approach can also be used to develop and model memristive systems for applications including artificial neural networks and neuromorphic computing.

These properties of memristive systems are expected to aid in device

development for implementing hardware for neural network systems.Therefore, the implementation of digital circuits will be discussed and has been carried out using this analytical model.

Chapter 3

Memristor-based Combinational Logic Implementation on MATLAB Simulink

3.1 Motivation

The area and power efficiency constraints of CMOS technology have prompted cutting-edge studies on nanodevices. Memristors, with their unique switching properties, show promise in implementing combinational logic and neural networks, potentially replacing existing CMOS technology for edge computing devices. This study presents the design, implementation, and performance evaluation of memristor-based combinational logic circuits, including adders, subtractors, and decoders, using MATLAB Simulink and Cadence Virtuoso. We propose an optimized design of these circuits and conduct a comparative study with conventional methods. Our memristor model, experimentally validated for a high-density Y_2O_3-based memristive crossbar array, demonstrates ultralow device-to-device and cycle-to-cycle variability. Power consumption

in these circuits is reduced by more than 90% compared to traditional CMOS technology. Additionally, the number of components used in memristor-based logic circuits is significantly reduced, enhancing area efficiency and paving the way for complex logic circuitry design at the micrometer scale.

Many contemporary computing tasks, including artificial intelligence (AI) and scientific computing, require parallel processing of large data volumes. The separation of memory and computation units in the von Neumann architecture results in significant time and energy consumption for data transfer [23], making it inefficient for data-intensive computing [24]. To address this, large-scale parallel designs like graphic processing units (GPUs) and specialized systems like tensor processing units (TPUs) have been developed [25]. Currently, MOSFET and CMOS technologies are foundational in digital electronics and support the mainstream computational paradigm [26]. Calculators, digital measuring devices, computers, digital processing, control automation, industrial processing, and digital communications all make extensive use of combinational logic circuits [27]. However, CMOS technology is approaching its limits as predicted by Moore's law, necessitating new technologies with lower power consumption and smaller area.

A memristor is a two-terminal device renowned for its electrical programmability [28], nanoscale size [17], precise tunability [29], and ability to function as a resistive element across various scales [30]. It is capable of efficiently performing both logic and memory operations [31] with low energy and power consumption and multibyte storage capability [32]. Memristors operate using pulse-based methods and adjustable resistance, making them ideal for regulating synaptic weights in neuromorphic computing processes. Pure memristor-based logic circuits include Memristor-Aided Logic (MAGIC) and Material Implication (IMPLY) memristor logic [33]. However, MAGIC cannot

perform cascade connections between multiple logic gates due to structural complications, and IMPLY increases operational time as it requires multiple steps in the circuit [34]. Recently, buffers have been used to improve logical states and reduce power consumption, although they increase chip area [35]. By incorporating memristors into circuits, one can overcome current technology limitations in speed, energy, power, and area, enabling efficient combinational logic gate design.

Several studies [24, 36, 37] have demonstrated the implementation of AND and OR logic gates using memristor circuits or CMOS-memristor hybrids. This work implements various combinational logic gates using a memristor model proposed by Kumar *et al.* [22], inspired by experimental results of Y_2O_3-based memristors and memristive crossbar arrays (MCAs) exhibiting ultralow device-to-device (D2D) and cycle-to-cycle (C2C) variability [22]. The proposed analytical model effectively demonstrates digital transitions between the HRS and LRS.

3.2 Design Procedure

To design and implement various combinational circuits, the previously mentioned nonlinear analytical model is employed to create a memristor element in MATLAB Simulink. A sinusoidal input voltage waveform is used to achieve a pinched hysteresis resistive switching response with a perfect zero crossing at the origin. It is important to note that some parameter values have been validly modified to achieve perfect digital behavior compared to the previously reported analog response [22], as outlined in Table 3.1 To further explore the memristive behavior with digital pulses, a rectangular voltage input pulse with an amplitude of +2 V and 0 V is applied, where +2 V corresponds

Table 3.1: Analytical modeling: comparing parameter values and understanding the physical interpretation

Parameters	Modified Values	Values in [22]	Physical Significance
b_1	1.59×10^{-7}	6.7×10^{-7}	Experimental fitting parameter
a_1	1.5	1.5	Degrees of influence of the state variable
α_1	1.6	0.8	Controlling parameters of Hysteresis loop area under positive bias
χ	1×10^{-11}	1×10^{-11}	Amplitude of ideal diode conduct
γ	1	1	Characteristics like thermal voltage and ideality factor of diode
A	5×10^{-2}	3×10^{-4}	Effect control of window function
m	5	5	Effect control of input of the state variable
p	5	1.5	Window function bounding parameter between 0 and 1

to logic 1 and 0 V corresponds to logic 0. Figure 1 displays the simulated output waveform for the digital logic design, and the pinched hysteresis loop Figure 3.1 in the *I–V* characteristics clearly demonstrates the abrupt transitions from HRS to LRS and vice versa. This confirms the digital behavior of the memristor, making it more suitable for digital logic design. In bipolar memristive behavior, the SET (ON) state occurs at positive voltage, while the RESET (OFF) state occurs at negative voltage. The resistance values for HRS and LRS are 0.526MΩ and 0.0527MΩ, respectively.

To design and simulate inverter and other combinational circuits, 180-nm CMOS technology is used in conjunction with a memristor. Dong *et al.* [38] reported that employing memristor-CMOS hybrid logic circuit efficiently

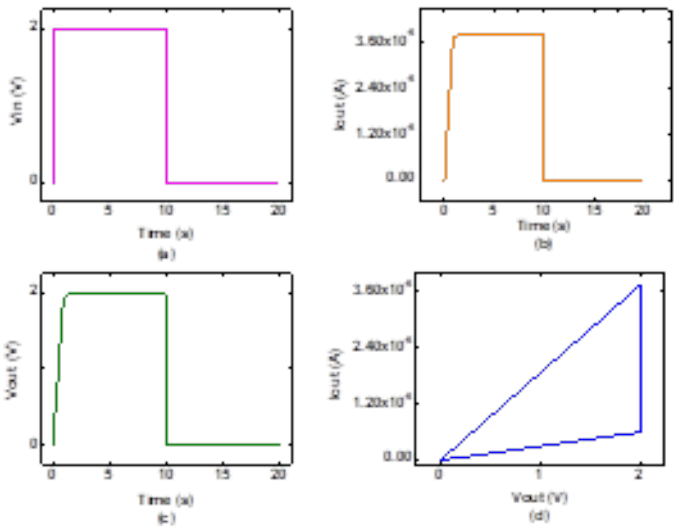

Figure 3.1: (a) Input voltage, (b) output current, (c) output voltage, (d) resistive switching response of memristor (positive half cycle).

reduced circuit delay.

3.3 Circuit Design and Simulation Outcomes

As previously mentioned, MATLAB Simulink is used to design and implement all combinational logic circuits, with the memristor functioning as a switching device, turning "ON" and "OFF" based on the input voltage amplitude. The output current of the memristor is represented by I_{out}. Equation 3.1 illustrates the calculation method used to determine logic "1" and "0".

$$\begin{aligned} For\ Logic\ 1 : V_{out} &= R_{on} \times I_{out} \\ For\ Logic\ 0 V_{out} &= R_{off} \times I_{out} \end{aligned} \tag{3.1}$$

$$\begin{aligned} Sum &= X \oplus Y \\ Carry &= X.Y \end{aligned} \tag{3.2}$$

$$\begin{aligned} Difference &= X \oplus Y \\ Borrow &= \bar{X}.Y \end{aligned} \tag{3.3}$$

3.3.1 Half Adder Half Subtractor

Equation 3.2 and Equation 3.3 show the boolean expressions, while Table 3.2 depicts the combined truth table for a half adder and half subtractor, respectively. Figure 3.2 shows the designed circuit layouts in MATLAB Simulink for the half adder Figure 3.2(a) and half subtractor Figure 3.2(b). Here, to perform the logic operations, two diferent input voltages, +2 and 0 V, are applied with diferent duty cycles of 0.5 and 0.25 to create the input logic (0,0), (0,1), (1,0), (1,1). The output logic combinations are presented in Figure 3.3. Table 3.2 shows the design process flow during logic computation in the case of half adder. Figure 3.3(a) shows both input voltage pulses Figure 3.3(b) gives sum and carry outputs while Figure 3.2(c) gives diference and borrow for a half adder and half subtractor, respectively.

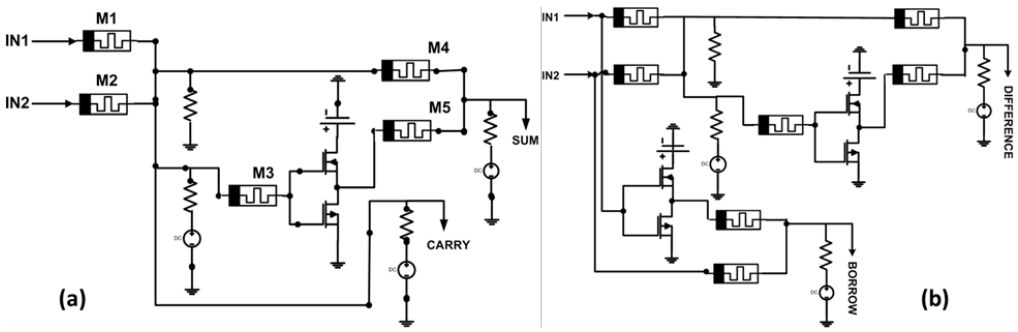

Figure 3.2: Memristor-based designs for (a) half adder and (b) half subtractor

Table 3.2: Computation design of half adder using memristor

IN1/ IN2 /Components	0,0	1,1	1,0	0,1
M1/ M2	HRS/HRS	LRS/LRS	LRS/HRS	HRS/LRS
M4	HRS	LRS	LRS	LRS
M3	HRS	LRS	HRS	HRS
CMOS logic output	1	0	1	1
M4	LRS	HRS	LRS	LRS
Sum logic output	0	0	1	1
Carry logic output	0	1	0	0

Table 3.3: Half adder and half subtractor logic table

Input 1	Input 2	Sum	Carry	Difference	Borrow
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	0	1	0
1	1	0	1	0	0

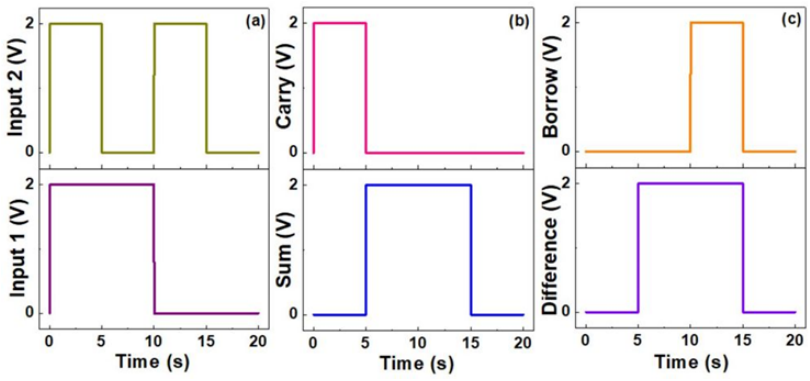

Figure 3.3: Waveforms of the (a) input (b) output for half adder circuit, and (c) output for half subtractor circuit using memristors

3.3.2 Full Adder and Full Subtractor

To design and implement a full adder circuit, two half adders and an additional "OR" gate are used, where the "OR" gate is responsible for the carry output, as illustrated in Figure 3.4(a). Similarly, for the full subtractor, two half subtractors and an additional "OR" gate are employed, as shown in Figure 3.4(b). The designed circuits are tested with three different duty cycles of 0.5, 0.25, and 0.125 to generate the following logic inputs: (000), (001), (010), (011), (100), (101), (110), and (111). The output logic combinations are detailed in Table 3.4. Figure 3.5(a) displays the input voltage pulses for the three inputs, the sum and carry outputs for the full adder Figure 3.5(b), and the difference and borrow outputs for the full subtractor Figure 3.5(c).

$$\begin{aligned} Sum &= X \oplus Y \oplus Z \\ Carry &= X.Y + (X \oplus Y).Z \end{aligned} \tag{3.4}$$

$$\begin{aligned} Difference &= X \oplus Y \oplus Z \\ Borrow &= \bar{X}.Y + X \bar{\oplus} Y.Z \end{aligned} \tag{3.5}$$

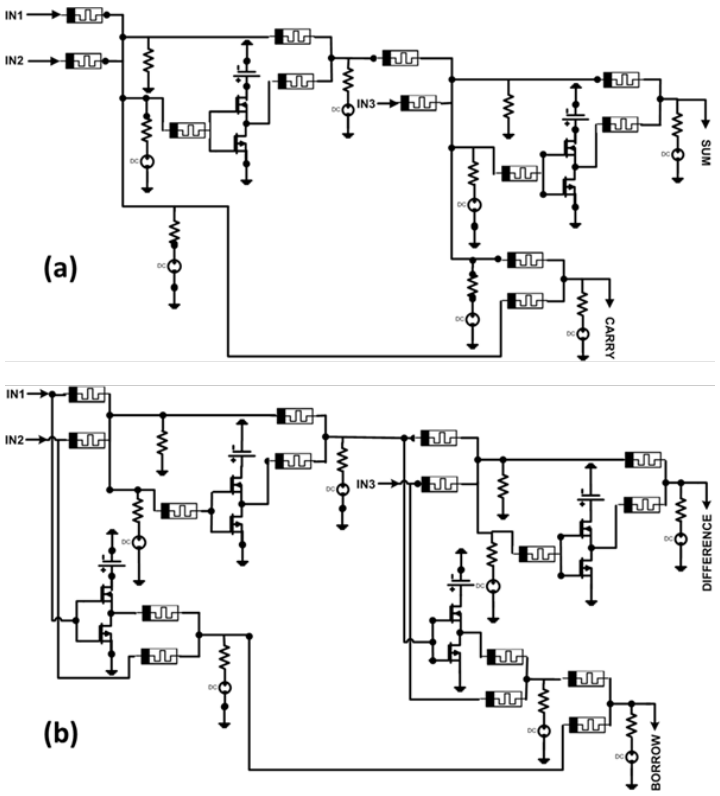

Figure 3.4: Memristor-based combinational circuits designs for (a) full adder and (b) full subtractor.

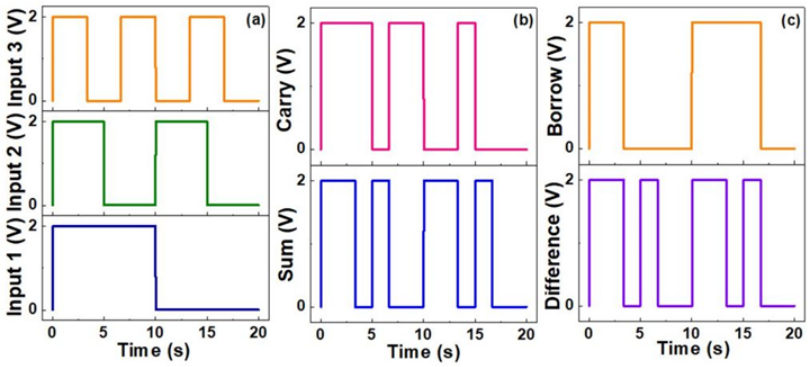

Figure 3.5: (a) 3 Input waveforms, (b) output waveform of full adder circuit, and (c) output waveform of full subtractor circuit using memristors

Table 3.4: Full adder and full subtractor logic table

Input 1	Input 2	Input 3	Sum	Carry	Difference	Borrow
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	1
1	1	0	0	1	0	1
1	1	1	1	1	1	1

3.3.3 2:4 Decoder and 3:8 Decoder

For the 2:4 decoder circuit implementation, similar input voltage scheme is applied as utilized in half adder and half subtractor. Here, two 1-bit inputs are applied to the 4 memristors-based AND gate to perform the decoder operation according to Equation 3.6. Figure 3.6 shows the implemented circuit layout of 2:4 decoder while Figure 3.7 depicts the results for 2:4 decoder. Table 3.5 shows the truth table of the 2:4 decoder circuit. Further, to extend the functionality, 3:8 decoder has also been implemented, as shown in Figure 3.8 and the results are displayed in Figure 3.9.

$$\begin{aligned} Output\ 1 &= A.B \\ Output\ 2 &= A.\bar{B} \\ Output\ 3 &= \bar{A}.B \\ Output\ 4 &= \bar{A}.B \end{aligned} \tag{3.6}$$

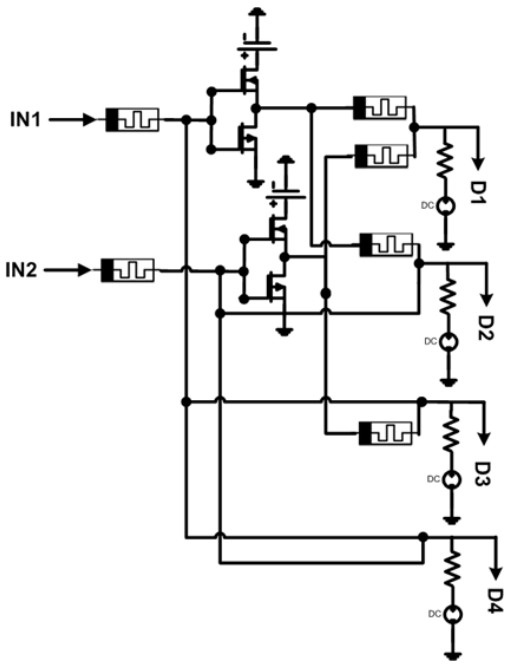

Figure 3.6: Implemented circuit for 2:4 decoder using memristors

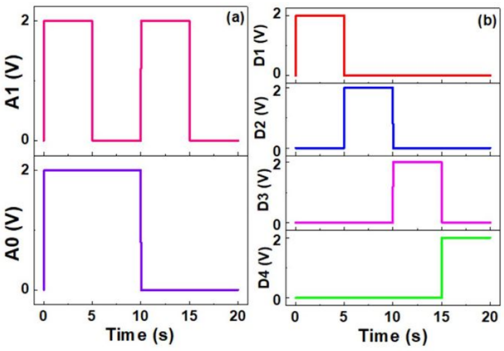

Figure 3.7: (a) 2 input waveforms and (b) output waveform of 2:4 decoder circuit using memristors

Table 3.5: 2:4 decoder logic table

Inputs		Outputs			
A0	A1	D1	D2	D3	D4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

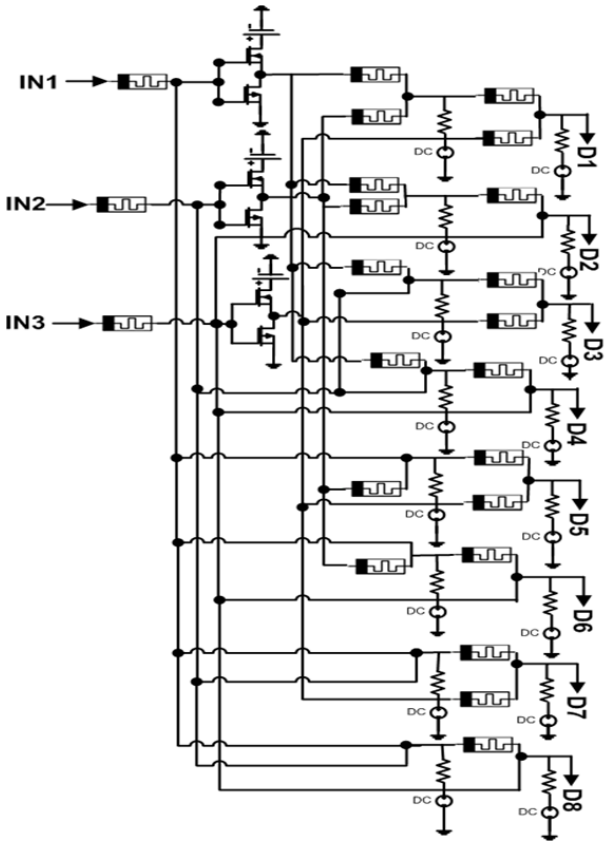

Figure 3.8: Implemented circuit for 3:8 decoder using memristors.

Table 3.6: 3:8 decoder logic table

Inputs			Outputs							
A1	A2	A3	D1	D2	D3	D4	D5	D6	D7	D8
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

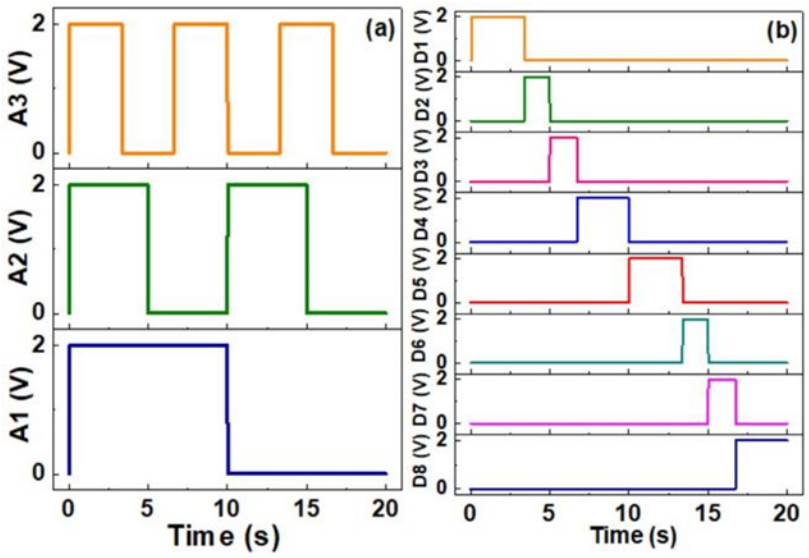

Figure 3.9: (a) 3 input waveforms, (b) Output waveform of 3:8 decoder circuit using memristor.

3.4 Performance Evaluation

In this section, performance evaluation for the implemented combinational circuits is presented. The performance evaluation is based on the various critical circuit parameters such as utilized power, circuit area, and the total number of components used in the respective circuit design. By considering the aforementioned parameters, here, we have discussed all of this one by one in detail.

3.4.1 Component Comparison

Here, Table 3.7 shows the comparison between the number of memristors and CMOS inverters used in memristor-based combinational logic gates with conventional CMOS technology. It is observed that the memristor-based combinational logic circuits require a significantly lesser number of components than necessitated for only transistor-based conventional circuits.

Table 3.7: Component comparison with area

Implemented Logic Circuits	Memristor-based Logic Gates		Transistor-based Logic Gates		Improvement (%)
	No. of Memristors	No. of CMOS Inverters	No. of MOSFET	No. of CMOS Inverters	
Half Adder	5	1	10	2	50
Full Adder	12	2	22	3	44
Half Subtractor	7	2	10	3	30.7
Full Subtractor	16	4	22	4	23.07
2:4 Decoder	6	2	32	2	82.35
3:8 Decoder	22	3	64	3	74.62

As observed from the comparison in Table 3.7 the memristor-based logic shows a significant improvement in the utilized components to implement the combinational circuits. Importantly, the utilized circuit power and area are directly related to the number of utilized components in the logic circuits.

3.4.2 Area Calculation

The area of the memristor-based combinational logic circuits has been calculated by using layout calculation rules, as reported by Kang *et al.* [39]. Here, the area of the memristor is considered as 9 nm^2 [40] while the area of MOSFET is calculated to be 1.06 nm^2. The total area of memristor is much smaller than the total area covered by the MOSFET and memristor can be implemented on polysilicon layer of MOSFET. Therefore, a thousand memristors can be fabricated on the same chip-level area as consumed by a single CMOS [37]. Table 3.8 shows the comparison of area of the memristor-based circuits and CMOS-based circuits in which memristor-based circuits consume significantly less area due to their nanometer scale as compared to CMOS.

Table 3.8: Area comparison of memristor-based and CMOS-based logic circuits

Combinational Logic Gates	Area of Memristors based circuits (μm^2)	Area of CMOS based circuits(μm^2)
Half Adder	1.06	16.03
Full Adder	2.31	35.99
Half Subtractor	2.31	17.28
Full Subtractor	4.81	40.98
2 input Decoder	2.31	17.28
3 input Decoder	3.56	63.43

3.4.3 Power Comparison

The power consumption of the circuits is another important parameter while designing the circuit. For the memristor-based combinational circuits, power is calculated by integrating the product of the output voltage and the summation of the input currents of the circuit. Figure 3.9 shows the worst case utilized power comparison between memristor-based combinational logic and CMOS-based combinational logic circuits. Here, it should be noted that the CMOS-based combinational logic circuits are implemented in Cadence Virtuoso by adopting 180 nm CMOS technology. As observed from the calculation, the memristor-based logic circuits consume much lesser power during operation as compared to CMOS-based logics which further strengthened to use the memristor technology for various logic implementations.

Table 3.9: Power consumption in memristor-based and CMOS-based combinational logic circuits

Combinational Logic Gates	Power of Memristors based Circuits (μW)	Average Power (μW)	Power of CMOS based Circuits(μW)
Half Adder	15	7.5	358.66
Full Adder	45	14.06	2160
Half Subtractor	15	5.5	154.08
Full Subtractor	45	15	2240
2 input Decoder	15	7.5	150.56
3 input Decoder	25	11.56	701.92

In this work, we have utilized our proposed non-linear memristor analytical model to design and implement various combinational logic circuits via MATLAB Simulink and Cadence Virtuoso. The model is validated based on experimental demonstration using low-variance in-house fabricated memristors

and MCA. The obtained resistive switching response shows clear digital behavior, which makes it a suitable candidate for digital logic design. Moreover, the memristor-based combinational logic circuits show significantly better performance in terms of the number of components, total circuit chip area, and utilized power as compared to those for the existing CMOS-based combinational logic circuits wherein 180 nm CMOS technology has been used. Therefore, the designed circuits are highly reliable for their use in future complex circuits and integrated circuits.

Chapter 4

Memristor-based Sequential and Advanced Digital Logic Implementation on Cadence Virtuoso

Neurons' dendritic architecture include computational capabilities that help to explain how information can be processed by single-celled creatures and how biological neural networks can do complicated tasks with minimal energy usage [15]. Dendrites are capable of non-linear programming of many inputs and simple logical processes. IMPLY (Material Implication) [41], MAGIC (Memristor Aided Logic) [42], and MRL (Memristive Logic Proportional Circuit) [43] are examples of current memristor-based logic circuits.

Different memristor resistance states are used in IMPLY to represent logic states, and the inferred logic is achieved by applying different control voltages to two memristors. However, several connections and operations are needed for multi-input IMPLY, which results in lengthy computation durations and

difficult processes. For instance, MAGIC requires initializing input and output memristors and requires a certain design to provide the desired logic function. IMPLY and MAGIC cannot be used with traditional CMOS technologies and require consecutive signal inputs. Furthermore, because the findings are saved in particular memristors, further circuitry is needed to read, write, and convert memristances to voltages. As a result, providing the logic signal as a voltage directly offers several benefits and makes integrating it with later CMOS circuits easier.

By joining memristors with varied polarity in series, the suggested circuits combine memristors and CMOS to provide a variety of logic gate circuits. However, in order to improve functionality and lessen the voltage loss brought on by cascading, modern MRL circuits need CMOS NOT gates [44].

Developing cutting-edge systems requires small, low-power, ultra-fast processing devices, and memristors significantly meet these requirements while being important components of digital circuit design. In this study, we use SPECTRE in Cadence Virtuoso to design, develop, and evaluate the performance of advanced digital logic circuits including memristor-based sequential logic.

For these memristor-based logic gates and combinational circuits, we suggest an improved design and conduct a comparative analysis with traditional 180-nm CMOS technology. Experimental results from a high-density Y_2O_3-based memristive crossbar array (MCA) provide a full validation of the used memristor model, demonstrating significantly low values for both cycle-to-cycle (C2C) and device-to-device (D2D) variability coefficients.

The performance of different logic architectures is greatly improved by the used memristor-based technique, increasing their area and power efficiency. This is a significant advancement in the design of ultra-fast, small, low-cost,

low-power circuits.

Over the past several decades, CMOS technology has driven digital electronics design and the implementation of various digital circuits for logic operations. However, over time, CMOS technology has faced numerous challenges, including the need for large chip areas, high power consumption, and low data processing speeds, leading to saturation in scaling and device performance. Additionally, CMOS-based logic circuits suffer from high leakage power consumption, further reducing device reliability.

Nanoscale memristors offer significant potential to overcome these limitations due to their high scalability, fast operating speeds, high density in crossbar array architecture, and non-volatile nature. Memristive devices can pave a new path for future computer technology, enabling instant ON and OFF switching without data loss and reducing system boot times. While physical limitations of current CMOS transistors hinder chip size and density due to scaling effects, the memristive technology is poised to effectively overcome these barriers.

Memristor technology supports computing via vector-matrix multiplication in resource-intensive applications with low power consumption, high storage capability due to remarkable scalability, fast processing speeds, and the ability to compute new logic patterns. In this work, several logic gates have been designed and implemented using industry-standard Verilog-A coding in the Cadence Virtuoso platform. A nonlinear analytical memristor model is used to implement these logic circuits. The memristor model is thoroughly validated by experimental results from Y_2O_3-based single memristors and memristive crossbar arrays (MCAs), which display stable switching responses with ultra-low device-to-device (D2D) and cycle-to-cycle (C2C) variability parameters.

4.1 Design Methodology

The memristor model discussed in Chapter 2 was used to create the memristor symbol, whose Verilog-A code was initially written in Cadence Virtuoso. The parameters are optimised as per the limitations given in [22] are given in Table 4.1. Using memristor symbol, logic circuits were implemented in 180 nm and 90 nm GPDK technologies.

It is suggested by MRL to integrate memristors with CMOS in order to establish operational compatibility with CMOS devices. This reduces chip size and enhances device density through a compatible process. Two memristors of the opposite polarity are connected in series, with the logic state represented as a voltage, in order to fully utilize the polarity of the memristors. The voltage at the node can be changed by division by adjusting the inputs to both ports, which will change the states of the memristors.

Now, we move to the design implementation part of this work. This section presents the basic gates, which serve as the building blocks for designing more advanced circuits.

4.1.1 OR and AND Gate

To implement an OR gate, two memristors are connected in parallel as shown in figure 4.1. If the inputs are (0,0) or (1,1), the output will be 0 and 1, respectively. For the (0,1) input combination, M1 offers high resistance, and M2 offers low resistance, allowing current to flow from IN2 through M2 to M1. The voltage drops across M1, resulting in output voltage of 1. For symmetric input pairs, the output matches input, while for asymmetric inputs, the output is 1. The obtained results, as shown in the Figure 4.2, demonstrate the logic 0 and logic 1 states.

Table 4.1: Memristor analytical model parameter optimization

Parameters	Literature Values [1]	Optimised Values for this work	Physical Interpretation
a_1	1.5	1	Degree of influence of state variable on current
b_1	6.7×10^{-7}	1.59×10^{-7}	Controlling parameter of conductivity slope for positive voltage polarity
α	0.8	0.5	Control parameters for the rate of change of state variable
χ	1×10^{-10}	1×10^{-11}	Constant for determining the boundedness of state variable
γ	1	1	Constant for determining the boundedness of state variable
A	3×10^{-4}	5×10^{-2}	Control the effect of the window function
m	5	5	Control the effect of input on the state variable
p	1.5	2	Bounding parameter for window function between 0 and 1

Table 4.2: OR gate logic table

V_{IN1}	V_{IN2}	V_{OUT}
0	0	0
0	1	1
1	0	1
1	1	1

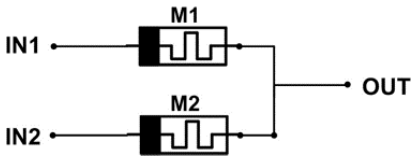

Figure 4.1: Memristor-based design for OR gate

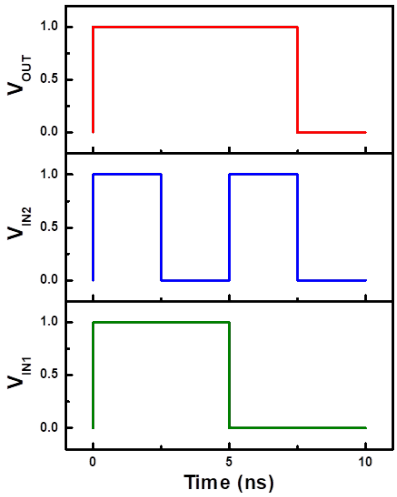

Figure 4.2: Waveforms of the input and output for OR gate circuit using memristors

In AND circuit, Figure 4.3, the memristors are inverted. For symmetric inputs, the output matches the input. For asymmetric inputs, let's take the (0,1) case. Since M1 is logic 0 and the memristor is inverted, M1 offers low resistance and M2 offers high resistance. The voltage drops across M2, resulting in 0V at the output node. Hence, for asymmetric inputs, the output is 0. This behavior is clearly demonstrated in the results shown in Figure 4.4.

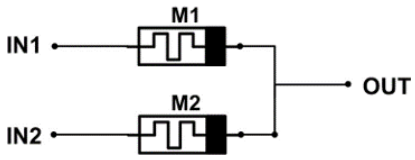

Figure 4.3: Memristor-based designs for AND gate

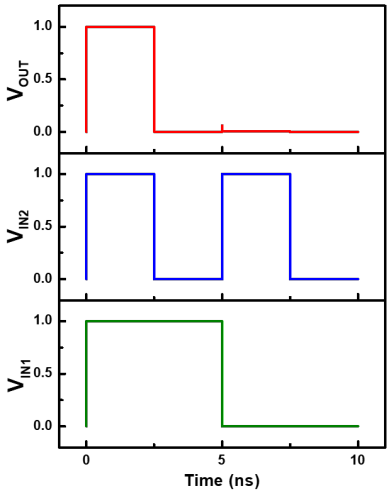

Figure 4.4: Waveforms of the input and output for AND gate circuit using memristors

Table 4.3: AND Gate logic table

V_{IN1}	V_{IN2}	V_{OUT}
0	0	0
0	1	0
1	0	0
1	1	1

4.1.2 NOT Gate

In this circuit design, as shown in Figure 4.5, the memristor is connected to a transistor, with the negative side connected to the drain of the transistor. The memristor is always supplied with a high voltage from a constant voltage source. When the input is 0, the transistor offers high resistance, causing the voltage to drop across the transistor. Consequently, the voltage at the output node is 1. Conversely, if the input is 1, the transistor offers low resistance, causing the voltage to drop across the memristor, resulting in 0 volts at the output node. The results are depicted from Figure 4.6.

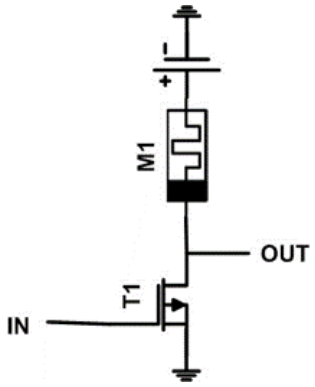

Figure 4.5: Memristor-based design for NOT gate

Table 4.4: NOT gate logic table

V_{IN}	V_{OUT}
0	1
1	0

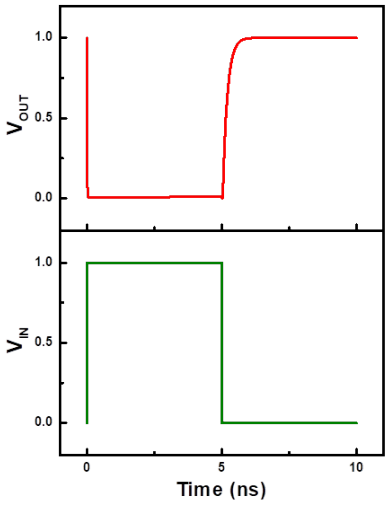

Figure 4.6: Waveforms of the input and output for NOT gate circuit using memristors

4.1.3 NOR and NAND Gate

The NOR gate circuit consists of an OR gate followed by a NOT gate. The circuit diagram is shown in Figure 4.7, and the output waveform is displayed in Figure 4.8. The NAND gate is designed using an AND gate followed by a NOT gate. The circuit diagram is shown in Figure 4.9, and the output waveform is displayed in Figure 4.10.

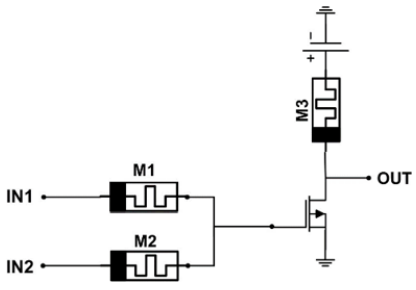

Figure 4.7: Memristor-based design for NOR gate

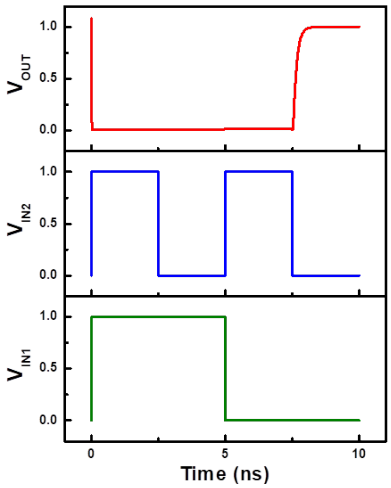

Figure 4.8: Waveforms of the input and output for NOR gate circuit using memristors

Table 4.5: NOR gate logic table

V_{IN1}	V_{IN2}	V_{OUT}
0	0	1
0	1	0
1	0	0
1	1	0

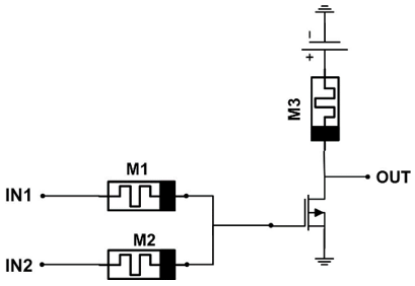

Figure 4.9: Memristor-based designs for NAND gate

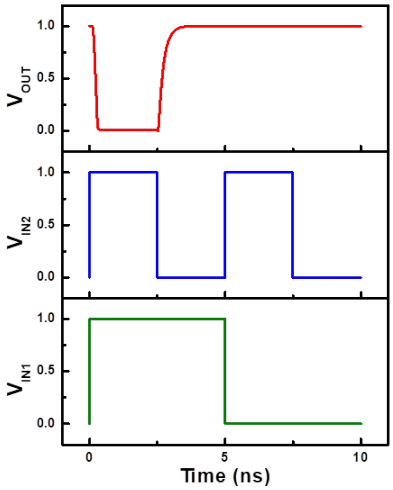

Figure 4.10: Waveforms of the input and output for NAND gate circuit using memristors

Table 4.6: NAND gate logic table

V_{IN1}	V_{IN2}	V_{OUT}
0	0	1
0	1	1
1	0	1
1	1	0

4.1.4 Multifunctional XOR Gate

The traditional schematic of an XOR gate composed of two AND gates, one OR gate, and one NOT gate. Here, we have optimized the XOR gate design to use only four memristors and one transistor as shown in Figure 4.11. Memristors M1 and M2 provide the AND logic functionality. Memristors M3 and M4 provide the OR logic functionality. The AND output is connected to the gate of the transistor, and the OR output is connected to the source of

the transistor [45]. The results can be observed from Figure 4.12. When the AND output is high, the output of the transistor is A + B; otherwise, it is 0, effectively implementing the XOR operation. This design reduces the number of components used compared to the traditional model.

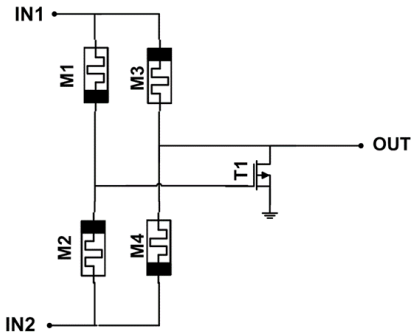

Figure 4.11: Memristor-based designs for multifunctional XOR

Table 4.7: Multifunctional XOR gate logic table

V_{IN1}	V_{IN2}	V_{OUT}
0	0	0
0	1	1
1	0	1
1	1	0

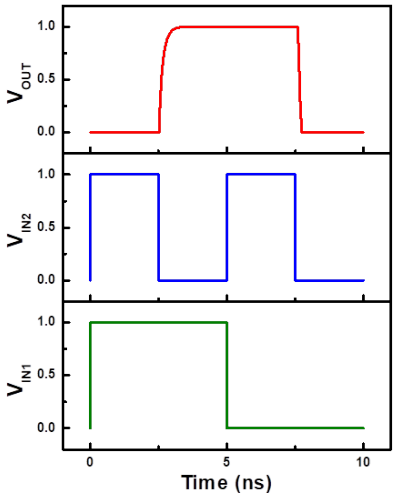

Figure 4.12: Waveforms of the input and output for multifunctional XOR gate circuit using memristors

4.2 Sequential Logic Circuit Implementation

4.2.1 SR Latch

The first sequential logic circuit, we implement is the SR latch. The SR latch consists of two cross-coupled NAND gates and has two triggering inputs, S and R. Two memristors form an AND gate. This AND gate is followed by a NOT operation using one transistor and one memristor, which together form a NAND gate. The output of one NAND gate is cross-coupled to the input of the other NAND gate, creating the SR latch structure, as shown in Figure 4.13. Set (S=0, R=1): The output Q is set to 1. Reset (S=1, R=0): The output Q is reset to 0. Hold (S=1, R=1): The latch holds the previous value of Q (Qn). Invalid State (S=0, R=0): This typically represents an invalid state (X). In this circuit, it results in Q being set to 1. This behavior can be verified by observing the output waveforms in Figure 4.14.

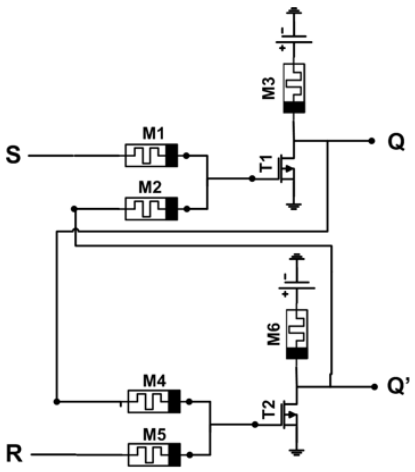

Figure 4.13: Memristor-based designs for SR latch

Table 4.8: SR Latch logic table

S	R	Q
0	1	1
1	0	0
1	1	Qn
0	0	X

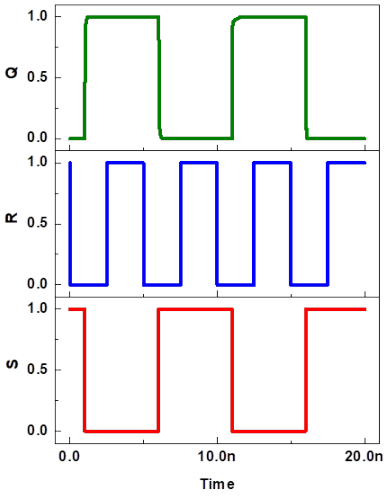

Figure 4.14: Waveforms of the input and output for SR Latch circuit using memristors

4.2.2 SR Flip Flop

The SR flip-flop is an extension of the SR latch and is controlled by three inputs: S, R, and an enable signal, typically provided by a clock signal, as given in Figure 4.15. It is an active-high circuit, meaning it will produce an output only when the enable (clock) signal is high, regardless of the S and R inputs. Set (*Clock=1, S=1, R=0*): The output Q is set to 1. Reset (*Clock=1, S=0, R=1*): The output Q is reset to 0. Hold (*Clock=1, S=0, R=0*): The flip-flop holds its previous value. Invalid State (*Clock=1, S=1, R=1*): This typically represents an invalid state. When the clock is low, the flip-flop retains its old value, irrespective of the S and R inputs. By observing the output waveforms in Figure 4.16, we can verify that the SR flip-flop responds correctly to the clock signal and the S and R inputs. The output changes state only when the clock signal is high, demonstrating the expected behavior of an active-high SR flip-flop.

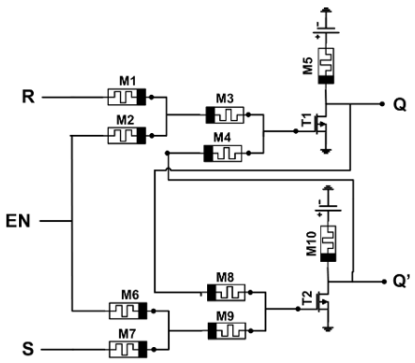

Figure 4.15: Memristor-based designs for SR FF

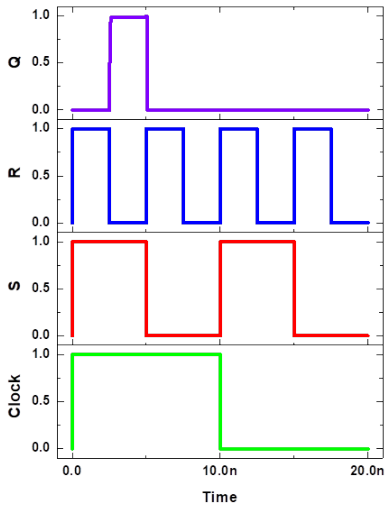

Figure 4.16: Waveforms of the input and output for SR FF circuit using memristors

Table 4.9: SR FF logic table

CLK	*S*	*R*	*Q*
1	1	1	X
1	1	0	1
1	0	1	0
1	0	0	Qn
0	X	X	Qn

4.2.3 D Flip Flop

The D flip-flop is derived from SR flip-flop by connecting the S input to the data (*D*) input and the R input to the inverted data input. This design, Figure 4.17 ensures that the D flip-flop only has one input, making it simpler and more predictable in operation compared to SR flip-flop. Set (*Clock=1, D=1*): The *S* input receives 1 and the *R* input receives 0, setting the output Q to 1. Reset (*Clock=1, D=0*): The *S* input receives 0 and the *R* input receives 1, resetting the output *Q* to 0. Hold (*Clock=0*): When the clock is low, the output *Q* retains its previous state, regardless of the *D* input. By observing the output waveforms, Figure 4.18, we confirm that *D* flip-flop accurately reflects the state of the *D* input when clock is enabled. This ensures that *Q* matches *D* when clock is high, and maintains its state when clock is low, demonstrating behavior of *D* flip-flop.

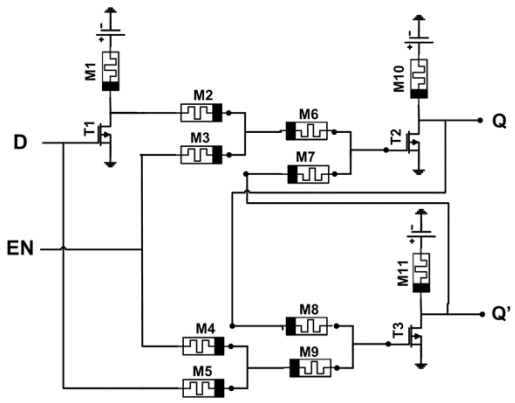

Figure 4.17: Memristor-based designs for D FF

Table 4.10: D FF logic table

Clock	*D*	*Q*
1	1	1
1	0	0
0	*X*	Q_n

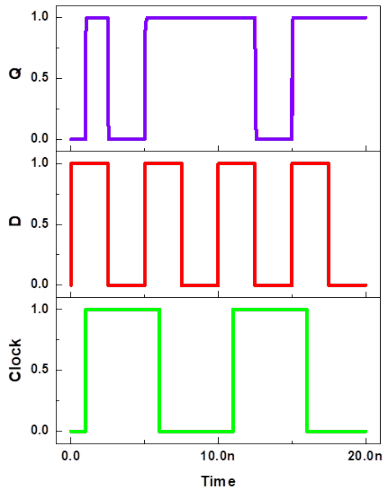

Figure 4.18: Waveforms of the input and output for D FF circuit using memristors

4.3 Combinational and Advanced Logic Circuit Implementation

4.3.1 Multifunctional Adder

This circuit utilizes the same XOR gate circuitry discussed previously and exhibits dual functionality as depicted in Figure 4.19. The transistor input provides the carry operation, which is the AND operation of both inputs. The transistor output provides the sum, which is the XOR operation of the inputs. The results can be verified with waveforms: For input (1, 1): The sum is 0, and the carry is 1; For input (1, 0) or (0, 1): The sum is 1, and the carry is 0; For input (0, 0): Both the sum and carry are 0. This configuration efficiently combines the operations to produce the desired outputs for a half adder which can be seen in Figure 4.20.

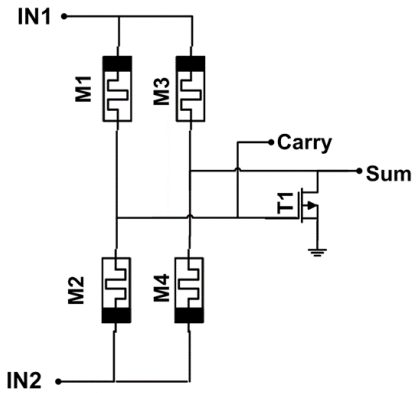

Figure 4.19: Memristor-based designs for multifunctional adder

Table 4.11: Multifunctional adder logic table

V_{IN1}	V_{IN2}	*Sum*	*Carry*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

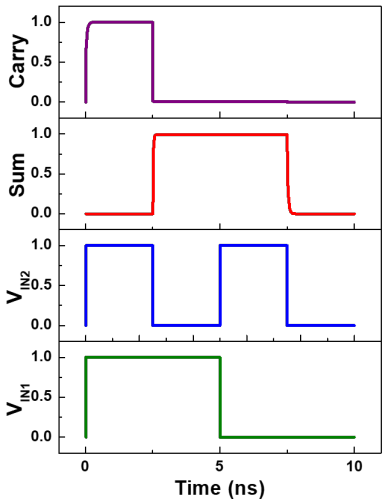

Figure 4.20: Waveforms of the input and output for multifunctional adder circuit using memristors

4.3.2 Multifunctional Subtractor

In this subtractor circuit, the difference is obtained from the output of transistor *T1*, which provides the XOR function of the two inputs. To implement the borrow operation, we use an additional transistor, *T2* as demonstrated in Figure 4.21. Difference (Diff): The output of *T1* gives the XOR of the two inputs. Borrow (B): Transistor *T2* takes the first input at the gate terminal and the OR function (A + B) at the source terminal. The output of *T2* gives the boolean

expression A'B (A AND NOT B). This setup ensures that the difference and borrow are correctly computed for the subtractor operation as indicated in Figure 4.22.

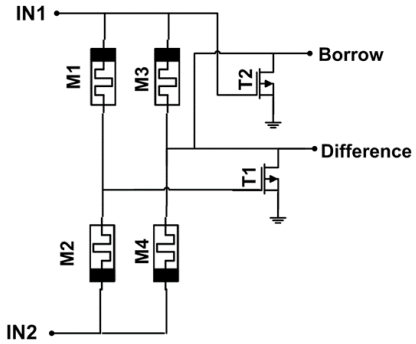

Figure 4.21: Memristor-based designs for multifunctional subtractor

Table 4.12: Multifunctional subtractor logic table

V_{IN1}	V_{IN2}	*Difference*	*Borrow*
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

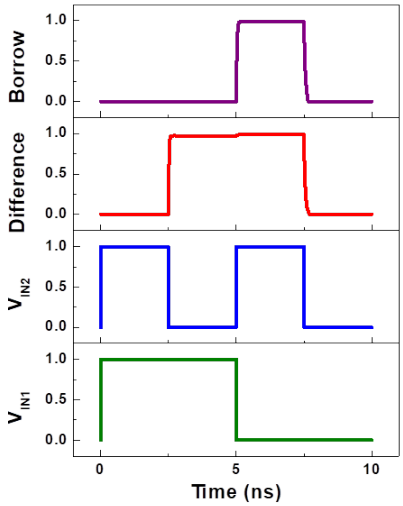

Figure 4.22: Waveforms of the input and output for multifunctional subtractor circuit using memristors

4.3.3 2:1 MUX

The 2:1 multiplexer (MUX) is designed using 2 AND gates and 1 OR gate. Additionally, a NOT gate is employed to invert the select input as represented in Figure 2.23 When *SEL=0*, *A0* is activated, and *OUT= IN0* When *SEL=1*, *A1* is activated, and *OUT=IN1* This 2:1 MUX design efficiently selects between two input lines based on the value of the select input, providing a versatile component for digital circuitry as illustrated in Figure 4.24.

Table 4.13: 2:1 MUX logic table

*V*IN1	*V*IN2	*Select*	*Vout*
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	1

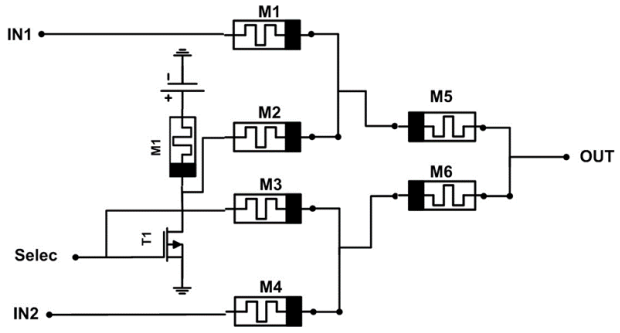

Figure 4.23: Memristor-based design for 2:1 MUX

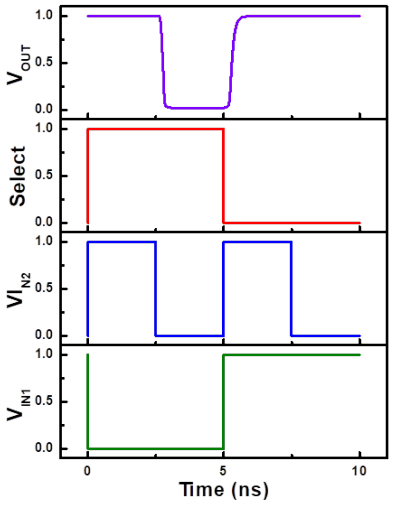

Figure 4.24: Waveforms of the input and output for 2:1 MUX circuit using memristors

4.3.4 2:4 decoder

The 2:4 decoder is designed using four AND gates and two NOT gates. This design provides optimal results, as each output is high for a specific combination of inputs as indicated in Figure 4.25. This ensures that the decoder operates correctly, providing a high signal at the appropriate output for each input combination. The design uses a minimal number of gates, making it an efficient

solution for decoding applications as displayed in Figure 4.26.

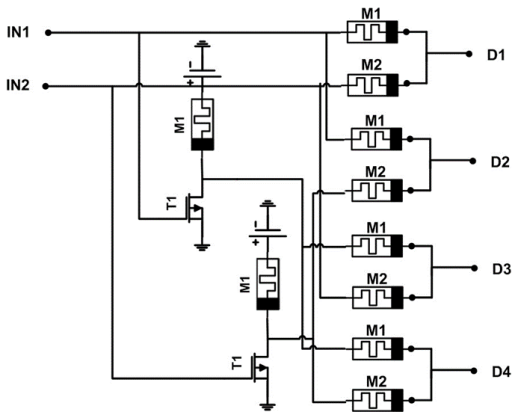

Figure 4.25: Memristor-based designs for 2:4 decoder

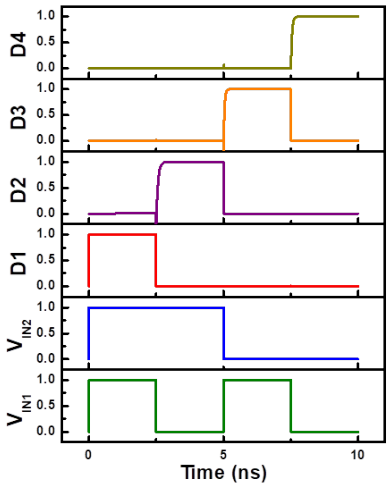

Figure 4.26: Waveforms of the input and output for 2:4 decoder circuit using memristors

Table 4.14: 2:4 decoder logic table

V_{IN1}	V_{IN2}	*D1*	*D2*	*D3*	*D4*
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

4.3.5 3:8 decoder

The 3:8 decoder utilizes 3-input AND gates, each constructed using 4 memristors, resulting in a total of 8 AND gates. Additionally, 3 NOT gates are employed to invert the inputs as portrayed in Figure 4.27. Output High: Each output is high only when the corresponding input combination satisfies the conditions defined by the AND gates. Output Low: The outputs are low for all other input combinations. The results demonstrate that each output is activated (high) only for a specific combination of inputs. This enables the decoder to accurately decode the input and select the appropriate output line. The utilization of memristors and gates optimizes the design for efficient decoding applications as evidenced by figure 4.28.

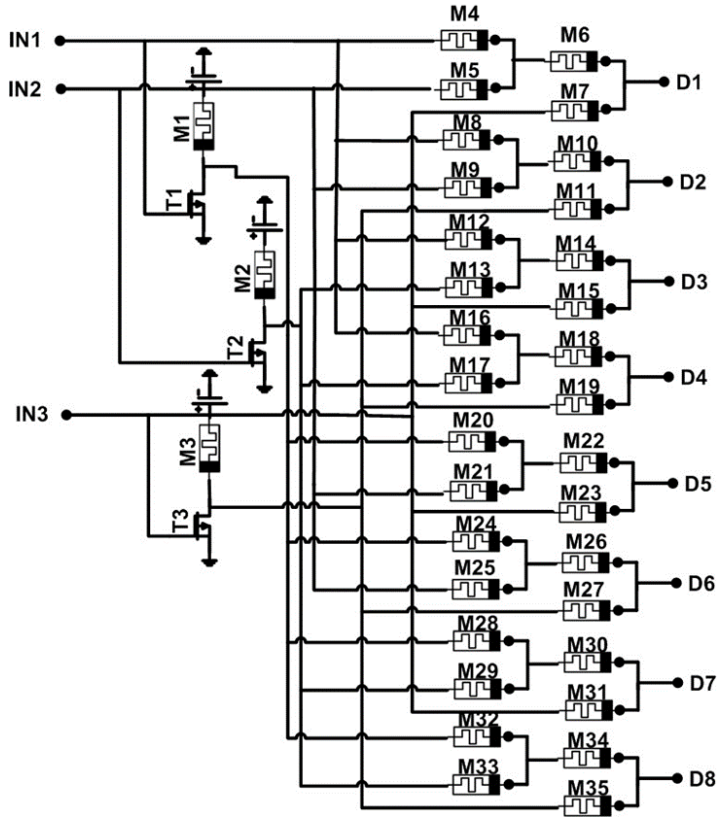

Figure 4.27: Memristor-based design for 3:8 decoder

Table 4.15: 3:8 decoder logic table

V_{IN1}	V_{IN2}	V_{IN3}	*D1*	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*	*D8*
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

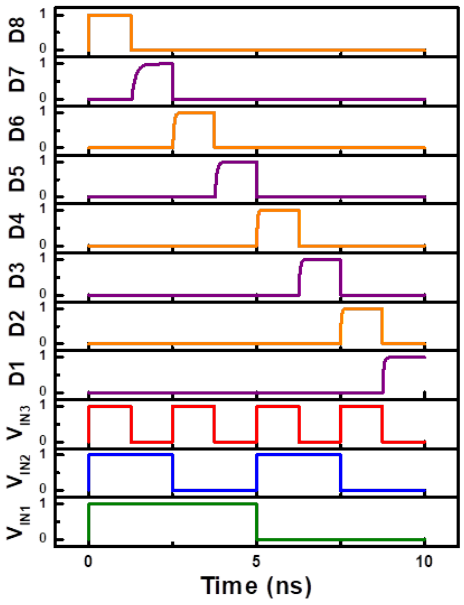

Figure 4.28: Waveforms of the input and output for 3:8 decoder circuit using memristors

4.3.6 1-bit Comparator

Next, we have designed advanced digital logic circuits using the memristive analytical model, starting with a 1-bit numeric comparator as displayed in Figure 4.29. This circuit compares two 2-bit inputs and outputs whether one is greater than, less than, or equal to the other. The design utilizes MRL. AND Gates: three AND gates with one input inverted (bubbled) each. (*M1T1, M2T2, M4T4*) NOT Gate: one NOT gate (*M3T3*) greater than Logic (*L1*):*L1* is obtained by the output of the AND gate *M1T1*, which takes V_{in1} and the inverted V_{in2}. Less Than Logic (*L2*): *L2* is obtained by the output of the AND gate *M2T2*, which takes the inverted V_{in1} and V_{in2}. equal to logic (*L3*): *L3* is obtained by the output of the NOT gate M3T3, which inverts the output of the XOR gate combining V_{in1} and V_{in2}. The waveform results can be verified against the

truth table for the 1-bit comparator to ensure the correct functionality of greater than, less than, and equal to operations. The output results can be verified with Figure 4.30. This 1-bit numeric comparator effectively utilizes memristor-based logic gates to perform basic comparison functions. The use of MRL enhances the design by saving chip area and increasing device density while maintaining operational compatibility with CMOS devices.

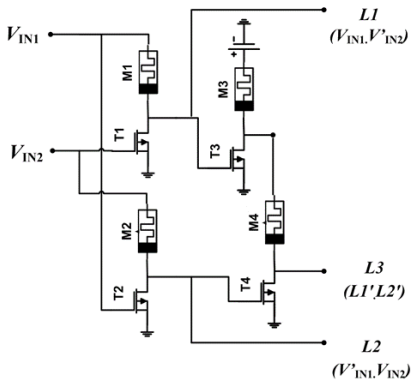

Figure 4.29: Memristor-based design for 1-bit comparator

Table 4.16: 1-bit comparator logic table

V_{in_1}	V_{in_1}	*L1* ($V_{in_1} > V_{in_2}$)	*L2* ($V_{in_1} < V_{in_2}$)	*L3* ($V_{in_1} = V_{in_2}$)
1	1	0	0	1
1	0	1	0	0
0	1	0	1	0
0	0	0	0	1

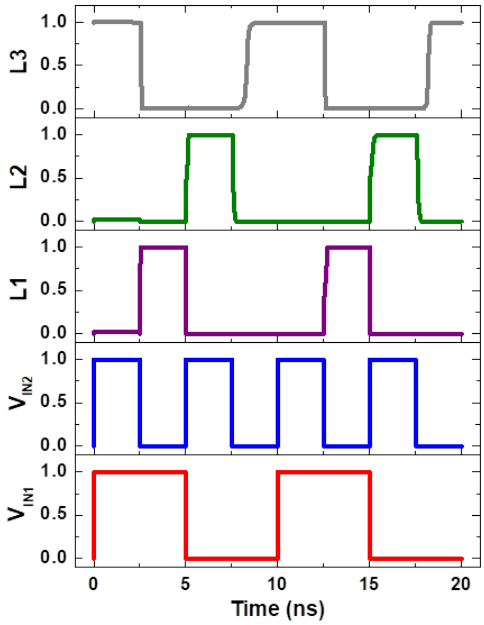

Figure 4.30: Waveforms of the input and output for 1-bit comparator circuit using memristors

4.3.7 2-bit Multiplier

The design of a 2-bit binary multiplier is the next task, as shown in Figure 4.31. This circuit represents the product of the two inputs by producing a 4-bit binary output from two 2-bit binary integers as inputs. *A1A0*: 2-bit input number and *B1B0*: 2-bit input number *S3*: The most significant bit (MSB) of the output. It is the AND of all four input bits. *S2*: The second MSB of the output. It is the AND of *A1* and *B0* , combined with the inversion of *A0B1*. *S1*: The second least significant bit (LSB) of the output. It is the XOR of the AND results of *A0B1* and *A1B0*. *S0*: The least significant bit (LSB) of the output. It is the AND of *A0* and *B0*. The output waveforms will show the binary product of the two 2-bit input numbers.The 2-bit binary multiplier multiplies two 2-bit input numbers and produces a 4-bit output. It uses six AND gates, one NOT gate, and one XOR

gate, implemented with memristor-based logic for optimal performance. The output waveforms verify the correct multiplication of the input binary numbers, demonstrating the circuit's functionality as illustrated in Figure 4.33.

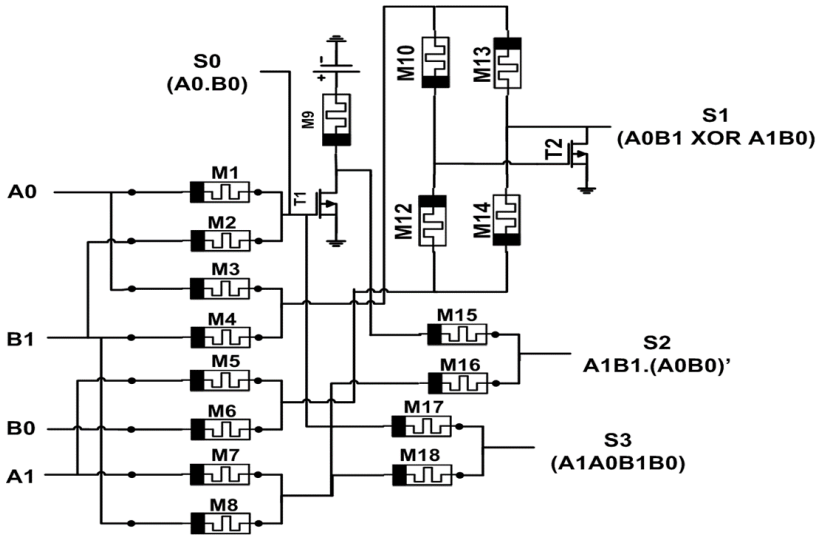

Figure 4.31: Memristor-based design for 2-bit multiplier

Table 4.17: 2-bit multiplier logic table

A1	*A0*	*B1*	*B0*	*S3*	*S2*	*S1*	*S0*
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

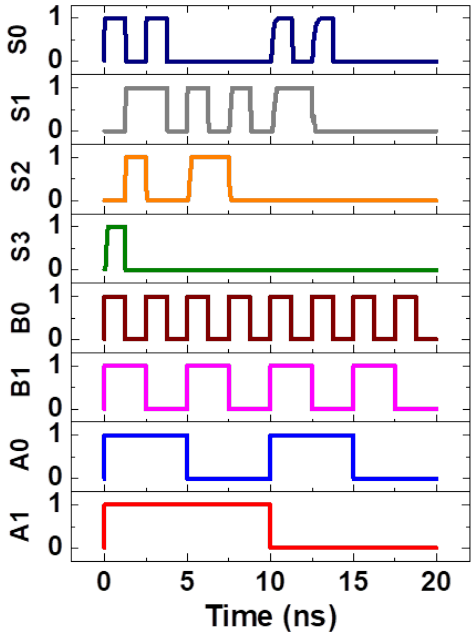

Figure 4.32: Waveforms of the input and output for 2-bit Multiplier circuit using memristors

4.3.8 3-bit Encoder

Next, we will design a 3-bit binary encoder. As shown in Figure 4.33, an encoder is a digital circuit that transforms an active input signal into a coded output. In this case, a 3-bit binary encoder has 8 input lines, and when any one of these inputs is high, it outputs a unique 3-bit binary code. The encoder logic is realized using 3 four-input OR gates. Each of these OR gates can be constructed using three 2-input OR gates.The OR gates are designed using memristor-based logic gates for optimal performance and efficiency. The output waveforms will demonstrate the encoder's functionality by showing unique 3-bit binary codes for each high input which can be seen in Table 4.18 The encoder converts any one of the 8 input lines into a unique 3-bit binary code. The design utilizes

memristor-based 2-input OR gates to construct 4-input OR gates, effectively reducing the complexity and increasing the efficiency of the circuit. The output waveforms in Figure 4.34 verify that the encoder correctly generates distinct output combinations for each active input signal.

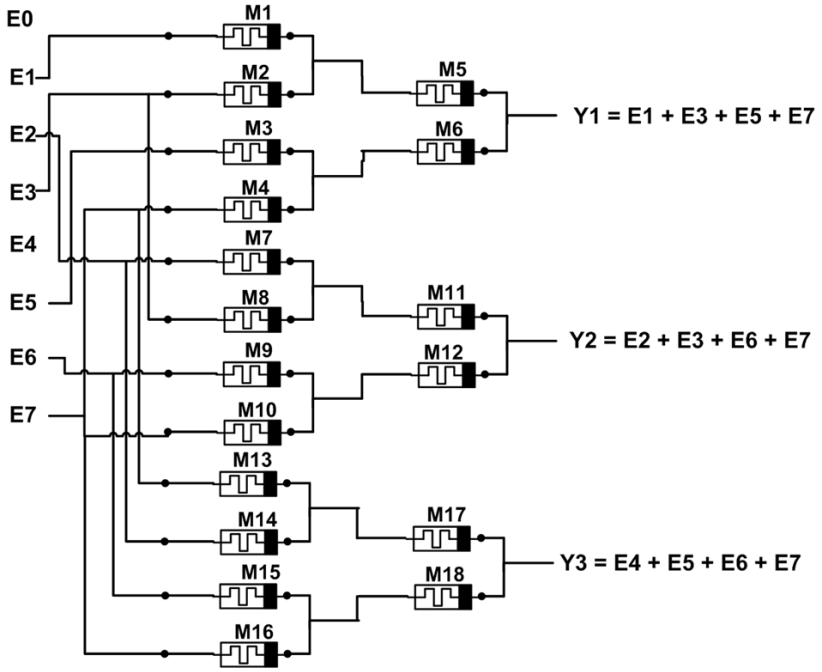

Figure 4.33: Memristor-based design for 3-bit encoder

Table 4.18: 3-bit binary encoder logic table

E0	*E1*	*E2*	*E3*	*E4*	*E5*	*E6*	*E7*	*Y0*	*Y1*	*Y2*
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

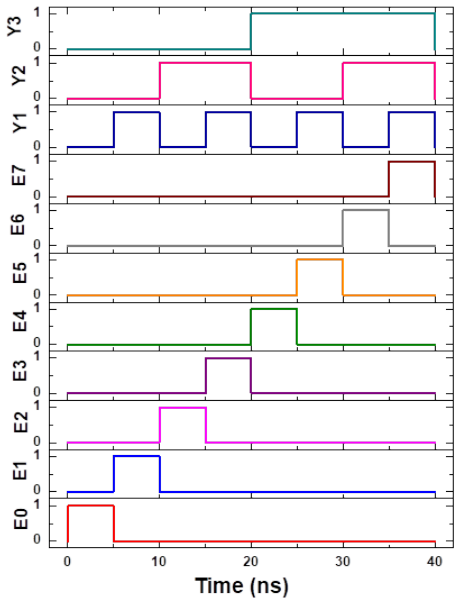

Figure 4.34: Waveforms of the input and output for 3-bit Encoder circuit using memristors

4.3.9 Multifunctional Module

Now, we introduce the multifunctional module [46], where all basic logic circuits are designed within a single module as shown in Figure 4.35. The module integrates multiple memristors and transistors to achieve various logic operations. Components and Functions: *M1*, *M2*: Perform OR logic operation. *M3*, *M4*: Perform AND logic operation. *M3, M4, M5, M6, T1*: Along with other components, assist in various logic functions. *M7, T2*: Invert the output of the AND operation to provide NAND logic. *M8, T3*: Invert the output of the OR operation to provide NOR logic. *M9, T4*: Invert the XOR output to provide XNOR logic. This multifunctional module is designed to provide all basic logic functions (OR, AND, NAND, NOR, XNOR) in one integrated setup. Each logic operation can be realized by configuring the appropriate memristors and transistors within the module.his configuration optimizes the design, reduces the need for multiple separate circuits, and demonstrates the versatility of memristor technology in achieving various digital logic functions. The output waveforms in Figure 4.36 validate the functionality of each logic operation, ensuring reliable performance.

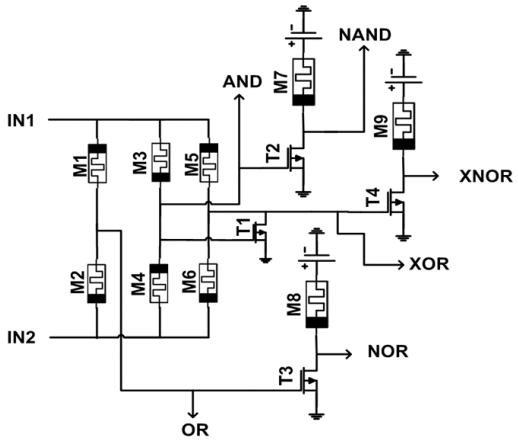

Figure 4.35: Memristor-based design for multifunctional module

Table 4.19: Multifunctional module logic table

V_{IN1}	V_{IN2}	*AND*	*OR*	*NAND*	*NOR*	*XOR*	*XNOR*
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

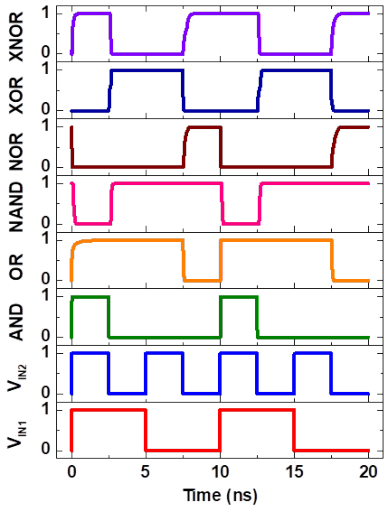

Figure 4.36: Waveforms of the input and output for Multifunctional Module circuit using memristors

4.4 Result and Discussion

4.4.1 Component Comparison

The Table 4.20. presents a comparative analysis of the number of components specifically, memristors and CMOS transistors used in the design of the aforementioned circuits. This analysis highlights our achievement of reducing the number of transistors in the design, thus optimizing the circuit area and

power efficiency. As observed from the table, the total number of devices in memristor-based logic circuits is significantly less than that in CMOS-based logic circuits. This is because memristors, with their compact size, occupy much less area compared to CMOS transistors.

Table 4.20: Comparison of number of components for logic circuits based on memristor and CMOS

Functions	Memristor based circuits		CMOS based circuits
	No. of Memristors	No. of Transistors	No. of Transistors
2-Input Adder	9	1	16
2-Input Subtractor	10	2	16
2:4 Decoder	10	2	28
3:8 Decoder	35	3	102
2:1 Mux	7	1	20
SR Latch	4	2	8
SR flip flop	8	2	20
D flip flop	9	3	22
1-Bit Comparator	4	4	20
3-Bit Binary Encoder	18	0	40
2-Bit Binary Multiplier	17	2	60
Multifunctional Module	9	4	44

4.4.2 Comparison of Power Consumption

The primary goal of this work is to design and implement various logic circuits that perform logic operations with better power efficiency compared to conventional CMOS technology. The power dissipation in these circuits depends on the output voltage and varies according to the type of logic gate. As a result, the output waveforms for different circuits will be distinct for the

same sets of inputs, leading to varying power dissipation across different logic gates and models.

$$Power = \ 1/T \int 2_0^T [\sum V_{Input} \times \sum I_{Input} \ dt] \tag{4.1}$$

Table 4.21: Comparison of power consumption for logic circuits based on memristor and CMOS

Functions	Memristor based circuits (nW)			CMOS based circuits (nW)		
	180 nm	90 nm	45 nm	180 nm	90 nm	45 nm
2-Input Adder	14.1	10.06	8.64	1060	958	550
2-Input Subtractor	15.13	11.54	9.6	1465	895	760
2:4 Decoder	13.4	12.6	6.3	3240	1125	875
3:8 Decoder	32.5	23.6	15.9	9825	4658	1254
2:1 Mux	10.7	8.3	8.1	3205	1920	617
SR Latch	11.52	5.15	0.16	1030	46	16
SR flip flop	22.84	14.52	12.4	2360	164	63
D flip flop (FF)	19.4	18.04	10.08	3036	190	120
1-Bit Comparator	5.25	2.62	0.9	1890	66	36
3-Bit Binary Encoder	23.8	18.4	12.9	5320	89	32
2-Bit Binary Multiplier	29.2	20.8	13.6	16320	246	2900
Multifunctional Module	14.06	13.28	9.2	9170	4589	360

4.4.3 Comparison of Area Calculation

The design area of any logic circuit is a crucial physical parameter, and optimizing it is critical for creating an area-efficient layout. Notably, if the number of devices in a circuit increases, the circuit consumes more area and power compared to a circuit with fewer devices [40].

Table 4.22: Comparison of area calculation for logic circuits based on memristor and CMOS

Functions	Memristor based circuits (μm^2)			CMOS based circuits (μm^2)		
	180 nm	90 nm	45 nm	180 nm	90 nm	45 nm
2-Input Adder	16.71	4.17	1.04	267.49	66.87	16.71
2-Input Subtractor	33.43	8.35	2.08	267.49	66.87	16.71
2:4 Decoder	33.43	8.35	2.08	468.11	117.02	29.25
3:8 Decoder	50.15	12.58	3.13	1705.26	426.31	106.57
2:1 Mux	16.71	4.17	1.04	334.36	83.59	20.89
SR Latch	33.43	8.35	2.08	133.74	33.43	8.35
SR flip flop	33.43	8.35	2.08	334.3	83.59	20.89
D flip flop	50.15	12.53	3.13	367.80	91.95	22.98
1-Bit Comparator	66.87	16.71	4.17	334.3	83.59	20.89
2-Bit Binary Multiplier	33.43	8.35	2.08	1003.1	250.77	62.69
Multifunctional Module	66.87	16.71	4.17	735.60	183.90	45.97

As illustrated in Table 4.20, the total number of devices in memristor-based logic circuits is significantly lower than in CMOS-based circuits. This is because a thousand memristors can be fabricated within the same chip-level area occupied by a single CMOS transistor. Specifically, a memristor occupies an area of 9 nm², whereas a CMOS transistor consumes 784 nm² . Consequently, a memristor requires 98.85% less space than a CMOS transistor for chip-level implementation. Some of the design rules are mentioned below and the area comparison for the designed circuits with CMOS circuits are shown in Table 4.22.

Minimum poly width= 3λ

Minimum poly spacing= 2λ

Minimum metal Spacing= 3λ

Minimum poly contact to poly edge spacing= 1λ

Minimum poly contact spacing= 2λ

Active contact size= 3λ

Where, $\lambda = \frac{channel\ length}{\mathbf{2}}$

4.4.4 Comparison of Delay

The delay in a circuit refers to the time required to obtain the output after applying the inputs. In this context, the average delay is considered, which depends on both the rise time and fall time delays. The average delay is calculated using Equation 4.2. For calculating the delay, the time difference between the input and output waveforms is considered, specifically when both the input and output reach 50% of their respective values. The delay is then obtained by subtracting these times.

$$\mathbf{A}verage\ delay\ = \frac{Rise\ time + Fall\ time}{2} \tag{4.2}$$

Table 4.23: Comparison of average delay for memristor based and CMOS based logic circuits

Circuit operation	Memristor based circuits (psec)			CMOS based circuits (psec)		
	180 nm	90 nm	45 nm	180 nm	90 nm	45 nm
2-Input Adder	152	257	115	352	145	77
2-Input Subtractor	81	4.81	81.975	371	126	54
2:4 Decoder	47	25.045	345.55	186	89	25
3:8 Decoder	4.28	0.95	345.55	456	112	44
2:1 Mux	5.43	6.25	6.75	350.5	84.22	51
SR Latch	643	105.5	106	197	37	30
SR flip flop	82.5	373	563.5	370	57	29
D flip flop	75	272	663	455	85	90
1-Bit Comparator	216	67	265	328.77	147	46
3-Bit Binary Encoder	45	26	456	397.28	280	72
2-Bit Binary Multiplier	248.7	42	345.55	323.9	516	64
Multifunctional Module	211	182	113.5	427.5	474	85

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

This thesis focuses on the design and implementation of various combinational and sequential logic circuits using memristors, providing a comprehensive analysis of their performance compared to conventional CMOS technology.

Key Findings in MATLAB Simulation of combinational logic circuits

- **Component Reduction:** Significant reduction in the number of components was achieved, with a maximum reduction of 77% for the 2:4 decoder and a minimum reduction of 23% for the 3-input full subtractor.
- **Power Consumption:** Memristor-based combinational logic circuits demonstrated a notable decrease in power consumption, with a maximum reduction of 98% for the 3-input full subtractor and a minimum of 90% for the 2:4 decoder compared to 180 nm CMOS technology.
- **Area Occupation:** The area required for these circuits was minimized, with the highest occupation being 4.81 μm^2 for the 3-input full subtractor

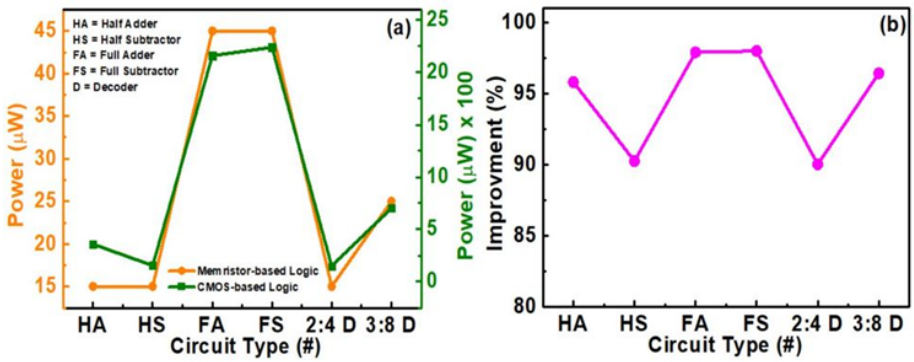

Figure 5.1: Power consumption in (a) memristor-based and CMOS-based combinational logic circuits. (b) Improvement in power consumption in memristor-based combinational logic circuits as compared to those in only CMOS-based logic circuits.

and the lowest being 1.06 μm^2 for the 2-input half adder, making these circuits highly suitable for chip design.

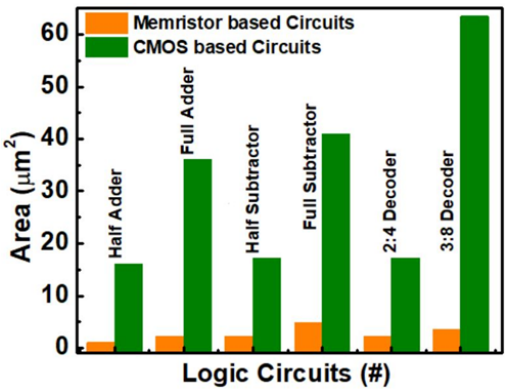

Figure 5.2: Area comparison of Memristor-based and CMOS-based logic circuits.

Implementation Success on Cadence Virtuoso

- **Hardware Descriptive Language:** The memristive analytical model was successfully implemented using Verilog, enabling the design of basic and combinational logic circuits on the Cadence Virtuoso platform.
- **Transistor Reduction:** The number of transistors in memristor-based logic circuits was significantly reduced compared to CMOS-based circuits, with a maximum reduction of 97.05%. When compared with existing literature, our designs achieved a maximum reduction of 85.71
- **Sequential Logic Circuits:** The design of sequential logic circuits using the memristive analytical model also showed a substantial reduction in transistor count, with a maximum reduction of 90% compared to CMOS circuits.

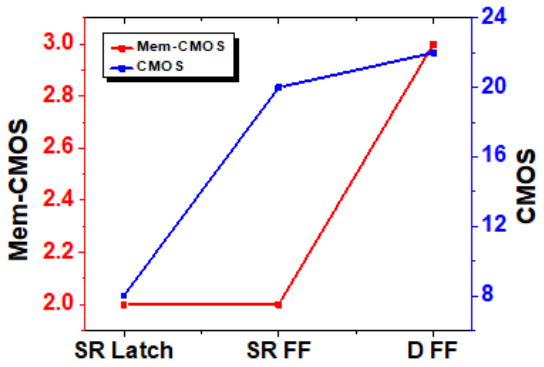

Figure 5.3: Component comparison for sequential logic circuits with CMOS circuits

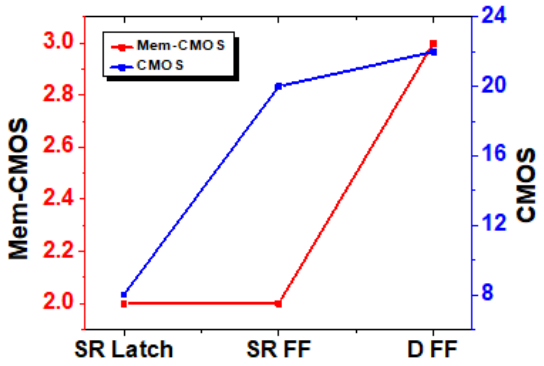

Figure 5.4: Component comparison for sequential logic circuits with CMOS circuits

Advanced Digital Logic Circuits

- **Optimization and Multifunctionality:** Advanced digital logic circuits were optimally designed using the memristive analytical model, achieving an average transistor reduction of 91.96% compared to CMOS circuits and 50% compared to available literature.

- **Power Consumption:** The power consumption was highest for 180 nm technology, followed by 90 nm and 45 nm nodes for both CMOS and memristor-based circuits.

- **Area Utilization:** Area utilization decreased with the reduction in technology node size for both memristor and CMOS-based circuits.

- **Operational Delay:** The delay was highest for 180 nm and lowest for 45 nm CMOS-based logic circuits, highlighting the performance benefits of scaling down technology nodes.

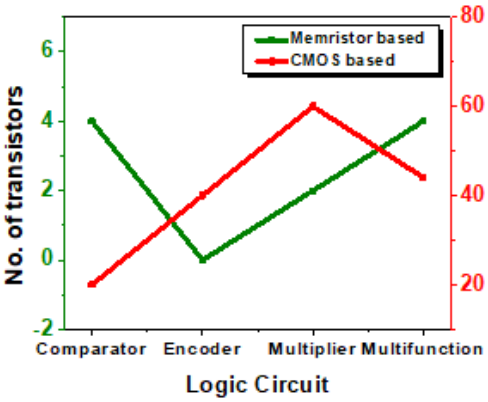

Figure 5.5: Component comparison for advanced digital logic circuits with CMOS circuits

In conclusion, this research demonstrates that memristor-based logic circuits offer significant advantages in terms of component reduction, power efficiency, and area optimization compared to traditional CMOS technology. The successful implementation of these circuits using Verilog and the Cadence Virtuoso platform underscores the potential of memristive technology in advancing digital logic design.

5.2 Future Scope

The proposed circuit, featuring a nonvolatile output with a feedback-based design utilizing memristors, opens up several promising avenues for future research and development.

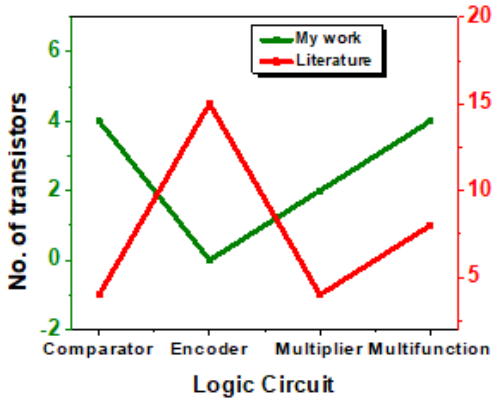

Figure 5.6: Component comparison for advanced digital logic circuits with memristor based circuits available in literature

Hybrid CMOS-Memristor Architectures

- **Integration of ALU and Memory Units:** The design of logic circuits as hybrid CMOS and memristor structures has the potential to redefine computer architecture by merging arithmetic logic units (ALU) with memory units [47]. This integration can lead to more compact and efficient computing systems.

- **Amplifiers and Oscillators:** The proposed methodology can be extended to design power- and area-efficient amplifiers and oscillators, which are fundamental components in various electronic devices.

- **Neuromorphic Networks:** Memristor-based designs can be crucial in developing neuromorphic networks, which mimic the neural structures of the human brain. These networks can benefit from the nonvolatile nature and high density of memristor-based circuits.

Biologically Inspired Circuit Designs

- **Artificial Neurons with Memristor-Based Synapses:** Designing artificial neurons in conjunction with memristor-based synapses can enhance the functionality and efficiency of neural network models, paving the way for advancements in artificial intelligence and machine learning [48].
- **Applications Favoring Data Density Over Speed:** The practical use of emerging logic families, which prioritize data density over speed, can be explored in applications such as data storage and retrieval, where high data density is critical.
- **Innovative Computing Solutions:** The combination of memristors with traditional CMOS technology can lead to innovative computing solutions that are more efficient in terms of area utilization, and power consumption addressing the growing demands for more compact and efficient electronic devices.

Further Miniaturization and Scalability

- **Scaling Down Technology Nodes:** As technology nodes continue to scale down, the role of memristors in achieving ultra-high-density memory and logic circuits will become increasingly significant. Research can focus on optimizing memristor designs for future technology nodes [49].
- **Integration with Emerging Technologies:** The integration of memristor-based circuits with other emerging technologies such as quantum computing and nanophotonics could unlock new possibilities in computing.

By pursuing these research directions, the field can significantly advance the development of more efficient, compact, and biologically inspired computing systems, leveraging the unique properties of memristors to meet future technological challenges.

Bibliography

[1] ERCIM News, Research and Innovation, In-Memory Computing: Towards Energy-Efficient Artificial Intelligence

[2] Kavehei, O. (2012). Memristive devices and circuits for computing, memory, and neuromorphic applications.

[3] Y. Xie, "Modeling, Architecture, and Applications for Emerging Memory Technologies," IEEE Design amp; Test of Computers, vol. 28, no. 1. Institute of Electrical and Electronics Engineers (IEEE), pp. 44–51, Jan. 2011

[4] L. O. Chua and Sung Mo Kang, "Memristive devices and systems," Proceedings of the IEEE, vol. 64, no. 2. Institute of Electrical and Electronics Engineers (IEEE), pp. 209–223, 1976.

[5] J. Zhu, T. Zhang, Y. Yang, and R. Huang, "A comprehensive review on emerging artificial neuromorphic devices," Applied Physics Reviews, vol. 7, no. 1. AIP Publishing, Feb. 24, 2020.

[6] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: ThrEshold Adaptive Memristor Model," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 1. Institute of Electrical and Electronics Engineers (IEEE), pp. 211–221, Jan. 2013

[7] W. Chen et al., "Essential Characteristics of Memristors for Neuromorphic Computing (Adv. Electron. Mater. 2/2023)," Advanced Electronic Materials, vol. 9, no. 2. Wiley, Feb. 2023.

[8] K. Kumari, S. Kar, A. D. Thakur, and S. J. Ray, “Role of an oxide interface in a resistive switch,” Current Applied Physics, vol. 35, pp. 16–23, Mar. 2022, doi: 10.1016/j.cap.2021.10.006.

[9] Maruf, M.H., Ali, S.I. (2020). Review and comparative study of I-V characteristics of different memristor models with sinusoidal input. International Journal of Electronics, 107, 349 - 375.

[10] G. Chakma, M. M. Adnan, A. R. Wyer, R. Weiss, C. D. Schuman, and G. S. Rose, “Memristive Mixed-Signal Neuromorphic Systems: Energy-Efficient Learning at the Circuit-Level,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 8, no. 1. Institute of Electrical and Electronics Engineers (IEEE), pp. 125–136, Mar. 2018.

[11] M. Lanza et al., “Standards for the Characterization of Endurance in Resistive Switching Devices,” ACS Nano, vol. 15, no. 11. American Chemical Society (ACS), pp. 17214–17231, Nov. 03, 2021.

[12] W. Sun et al., “Understanding memristive switching via in situ characterization and device modeling,” Nature Communications, vol. 10, no. 1. Springer Science and Business Media LLC, Aug. 01, 2019.

[13] http://www.labs.hpe.com/research/systems-research/themachine/Machinegallery-memristor.jpg

[14] Peng Gu et al., “Technological exploration of RRAM crossbar array for matrix-vector multiplication,” The 20th Asia and South Pacific Design Automation Conference. IEEE, Jan. 2015.

[15] J. Zhu, T. Zhang, Y. Yang, and R. Huang, “A comprehensive review on emerging artificial neuromorphic devices,” Applied Physics Reviews, vol. 7, no. 1. AIP Publishing, Feb. 24, 2020.

[16] D. Panda, P. P. Sahu, and T. Y. Tseng, “A Collective Study on Modeling and Simulation of Resistive Random Access Memory,” Nanoscale Research Letters, vol. 13, no. 1. Springer Science and Business Media LLC, Jan. 10, 2018.

[17] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” Nature, vol. 453, no. 7191. Springer Science and Business Media LLC, pp. 80–83, May 01, 2008.

[18] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams, “Memristive switching mechanism for metal/oxide/metal nanodevices,” Nature Nanotechnology, vol. 3, no. 7. Springer Science and Business Media LLC, pp. 429–433, Jun. 15, 2008.

[19] E. Lehtonen and M. Laiho, “CNN using memristors for neighborhood connections,” 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010). IEEE, Feb. 2010.

[20] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, “A Memristor Device Model,” IEEE Electron Device Letters, vol. 32, no. 10. Institute of Electrical and Electronics Engineers (IEEE), pp. 1436–1438, Oct. 2011.

[21] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, “Generalized Memristive Device SPICE Model and its Application in Circuit Design,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 8. Institute of Electrical and Electronics Engineers (IEEE), pp. 1201–1214, Aug. 2013.

[22] S. Kumar, R. Agrawal, M. Das, K. Jyoti, P. Kumar, and S. Mukherjee, “Analytical model for memristive systems for neuromorphic computation,” Journal of Physics D: Applied Physics, vol. 54, no. 35. IOP Publishing, p. 355101, Jun. 17, 2021.

[23] J. von Neumann, “First draft of a report on the EDVAC,” IEEE Annals of the History of Computing, vol. 15, no. 4. Institute of Electrical and Electronics Engineers (IEEE), pp. 27–75, 1993.

[24] M. A. Zidan, J. P. Strachan, and W. D. Lu, “The future of electronics based on memristive systems,” Nature Electronics, vol. 1, no. 1. Springer Science and Business Media LLC, pp. 22–29, Jan. 08, 2018.

[25] N. P. Jouppi et al., “In-Datacenter Performance Analysis of a Tensor Processing Unit,” Proceedings of the 44th Annual International Symposium on Computer Architecture. ACM, Jun. 24, 2017.

[26] Nikonov, D.E., Young, I.A. (2013). Overview of Beyond-CMOS Devices and a Uniform Methodology for Their Benchmarking. Proceedings of the IEEE, 101, 2498-2533.

[27] D. E. Nikonov and I. A. Young, "Overview of Beyond-CMOS Devices and a Uniform Methodology for Their Benchmarking," Proceedings of the IEEE, vol. 101, no. 12. Institute of Electrical and Electronics Engineers (IEEE), pp. 2498–2533, Dec. 2013.

[28] L. Chua, "Memristor-The missing circuit element," IEEE Transactions on Circuit Theory, vol. 18, no. 5. Institute of Electrical and Electronics Engineers (IEEE), pp. 507–519, 1971.

[29] S. Stathopoulos et al., "Multibit memory operation of metal-oxide bi-layer memristors," Scientific Reports, vol. 7, no. 1. Springer Science and Business Media LLC, Dec. 13, 2017.

[30] S. Duan, X. Hu, L. Wang, C. Li, and P. Mazumder, "Memristor-based RRAM with applications," Science China Information Sciences, vol. 55, no. 6. Springer Science and Business Media LLC, pp. 1446–1460, Apr. 12, 2012.

[31] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 10. Institute of Electrical and Electronics Engineers (IEEE), pp. 2054–2066, Oct. 2014.

[32] M. Khalid, "Review on Various Memristor Models, Characteristics, Potential Applications, and Future Works," Transactions on Electrical and Electronic Materials, vol. 20, no. 4. Springer Science and Business Media LLC, pp. 289–298, Jun. 29, 2019.

[33] S. Kvatinsky et al., "MAGIC—Memristor-Aided Logic," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 11. Institute of Electrical and Electronics Engineers (IEEE), pp. 895–899, Nov. 2014.

[34] G. Liu, S. Shen, P. Jin, G. Wang, and Y. Liang, "Design of Memristor-Based Combinational Logic Circuits," Circuits, Systems, and Signal Processing, vol. 40, no. 12. Springer Science and Business Media LLC, pp. 5825–5846, Jun. 28, 2021.

[35] S. Mandal, J. Sinha, and A. Chakraborty, "Design of Memristor – CMOS based logic gates and logic circuits," 2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC). IEEE, Mar. 2019.

[36] C. Gao, T. Li, T. Wang, and X. Cao, “Memristor-Based Logic Gate Circuit,” 2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET). IEEE, Aug. 2020.

[37] T. Singh, “Hybrid Memristor-CMOS (MeMOS) based Logic Gates and Adder Circuits.” arXiv, 2015.

[38] Z. Dong, D. Qi, Y. He, Z. Xu, X. Hu, and S. Duan, “Easily Cascaded Memristor-CMOS Hybrid Circuit for High-Efficiency Boolean Logic Implementation,” International Journal of Bifurcation and Chaos, vol. 28, no. 12. World Scientific Pub Co Pte Lt, p. 1850149, Nov. 2018.

[39] Kang, S.M., Leblebici, Y. (2002). CMOS Digital Integrated Circuits Analysis and Design.

[40] G. Kumar and K. Datta, “Design of digital functional blocks using hybrid memristor structures,” TENCON 2015 - 2015 IEEE Region 10 Conference. IEEE, Nov. 2015.

[41] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 10. Institute of Electrical and Electronics Engineers (IEEE), pp. 2054–2066, Oct. 2014.

[42] S. Kvatinsky et al., “MAGIC—Memristor-Aided Logic,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 11. Institute of Electrical and Electronics Engineers (IEEE), pp. 895–899, Nov. 2014.

[43] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “MRL amp;x2014; Memristor Ratioed Logic,” 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications. IEEE, Aug. 2012.

[44] B. Abdoli, A. Amirsoleimani, J. Shamsi, K. Mohammadi, and A. Ahmadi, “A novel CMOS-memristor based inverter circuit design,” 2014 22nd Iranian Conference on Electrical Engineering (ICEE). IEEE, May 2014.

[45] X. Yang, A. Adeyemo, A. Bala, and A. Jabir, “Novel techniques for memristive multifunction logic design,” Integration, vol. 65. Elsevier BV, pp. 219–230, Mar. 2019.

[46] C. Ji, T. Li, and X. Zou, “Multifunctional Module Design Based on Hybrid CMOS-Memristor Logic Circuit,” 2022 International Conference on Service Science (ICSS). IEEE, May 2022.

[47] X.-Y. Wang, Z.-R. Wu, P.-F. Zhou, H. H.-C. Iu, S.-M. Kang, and J. K. Eshraghian, “FPGA Synthesis of Ternary Memristor-CMOS Decoders for Active Matrix Microdisplays,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 9. Institute of Electrical and Electronics Engineers (IEEE), pp. 3501–3511, Sep. 2022.

[48] G. Liu, S. Shen, P. Jin, G. Wang, and Y. Liang, “Design of Memristor-Based Combinational Logic Circuits,” Circuits, Systems, and Signal Processing, vol. 40, no. 12. Springer Science and Business Media LLC, pp. 5825–5846, Jun. 28, 2021.

[49] Q. Xia et al., “MemristorCMOS Hybrid Integrated Circuits for Reconfigurable Logic,” Nano Letters, vol. 9, no. 10. American Chemical Society (ACS), pp. 3640–3645, Sep. 01, 2009.