11-23-2023

MEMRISTOR-BASED DIGITAL CIRCUIT DESIGN

Khalid Mohammed Alammari
University of Windsor

MEMRISTOR-BASED DIGITAL CIRCUIT DESIGN

By

Khalid Alammari

A Dissertation
Submitted to the Faculty of Graduate Studies through the
Department of Electrical and Computer Engineering in Partial Fulfillment
of the Requirements for the Degree of Doctor of Philosophy at the University of
Windsor

Windsor, Ontario, Canada
2023

Memristor-Based Digital Circuit Design

by

Khalid Alammari

APPROVED BY:

_____

N. Dimopoulos, External Examiner
University of Victoria

_____

R. Riahi
Department of Mechanical, Automotive and Materials Engineering

_____

R. Rashidzadeh
Department of Electrical and Computer Engineering

_____

E. Abdel-Raheem
Department of Electrical and Computer Engineering

_____

A. Ahmadi, Co-Advisor
Department of Electrical and Computer Engineering

_____

M. Ahmadi, Co-Advisor
Department of Electrical and Computer Engineering

September 29, 2023

Declaration of Co-Authorship/Previous Publication

I. Co-Authorship

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

Chapter 2 of this thesis was co-authored with Dr. Abubaker Sasi, Prof. Majid Ahmadi and Dr. Arash Ahmadi, and Prof. Mehrdad Saif. The design describes using Verilog HDL, analysis of the simulation results at behavioral level in terms of performance using a designed test bench and writing the manuscript was performed by the author. Dr. Abubaker Sasi Created a memristor-based standard logic cell library and synthesized the case study and analysis of the results. Prof. Majid Ahmadi, Dr. Arash Ahmadi, and Prof. Mehrdad Saif supervised and edited the manuscript.

Chapter 3 of this thesis was co-authored with Dr. Arash Ahmadi, and Prof. Majid Ahmadi. The design, simulation, implementation, analysis of results and writing were performed by the author. The contribution of Dr. Arash Ahmadi, and Prof. Majid was to oversee the research, provide feedback and give comments to improve the manuscript.

Chapter 4 of this thesis was co-authored with Prof. Majid Ahmadi, and Dr. Arash Ahmadi. The design, simulation, implementation, analysis of results and writing were performed by the author. The contribution of Prof. Majid Ahmadi and Dr. Arash Ahmadi was to oversee the research, provide feedback and give comments to improve the manuscript.

Chapter 5 of this thesis was co-authored with Prof. Majid Ahmadi and Dr. Arash Ahmadi. In all cases, the design, simulation, implementation, analysis of results and writing of the manuscript were performed by the author, while Prof. Majid Ahmadi and Dr. Arash Ahmadi provide feedback and give comments to improve the manuscript.

Chapter 6 of this thesis was co-authored with Dr. Moslem Heidarpur, Prof. Majid Ahmadi, and Dr. Arash Ahmadi. The design, simulation, implementation, analysis of results and writing were performed by the author. Dr. Moslem Heidarpur analyzed the results using Python to verify the proposed design. The contribution of Prof. Majid Ahmadi and Dr. Arash Ahmadi were to oversee the research, provide feedback and give comments to improve the manuscript.

I am aware of the University of Windsor Senate Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my thesis and have obtained written permission from each of the co-authors to include the above materials in my thesis. I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis includes 5 original papers, 3 of them have been previously published/ accepted in peer reviewed journals and conferences and 2 have been submitted, as follows:

Thesis Chapter	Publication Title	Publication status
Chapter 2	K. Alammari, A. Sasi, M. Ahmadi, A. Ahmadi, and M. Saif. "Hybrid Memristor-CMOS Based FIR Filter Design." *In Chaos and Complex Systems, pp. 91-99. Springer, 2020*.	Published
Chapter 3	K. Alammari, A. Ahmadi, and M. Ahmadi, "Hybrid Memristor-CMOS Based Up-Down Counter Design" *2020* 27th IEEE International Conference on Electronics, *Circuits and Systems (ICECS)*, Glasgow, UK, 2020, pp.1-4	Published
Chapter 4	K. Alammari, M. Ahmadi, and A. Ahmadi. "A Memristive Based Design of a Core Digital Circuit for Elliptic Curve Cryptography." *in the journal of Circuits, Systems and Computers ,October 2023 Issue"*	Published
Chapter 5	K. Alammari, M. Ahmadi, and A. Ahmadi. "Memristive-Based Full Adder" .IEEE ISCAS 2024.	Submitted
Chapter 6	K. Alammari, M. Heidarpur, M. Ahmadi, and A. Ahmadi. "LIF Neuron -A Memristive Realization". *IEEE Transaction on CAS II*	To be Submitted

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Further miniaturization in CMOS technology has faced severe challenges, such as increased leakage power and reduced circuit reliability, which has caused fundamental restrictions on advancing efficient computing architecture. These problems can be addressed by the new emerging devices known as memristors owing to their nanoscale size and the ability to integrate with the exciting CMOS technology. Memristors are passive devices with variable resistance. The resistance value will remain constant when no electrical field is applied, giving this device a unique behavior by saving its last state. This unique behavior makes this device very appealing to a wide variety of applications such as memory, neuromorphic systems, analog circuits, digital circuits, and logic design. The primary focus of this thesis is the utilization of memristors in based logic circuit design aiming to serve CMOS transistor scaling limitation problems by exploiting the integration of memory with the computational capabilities within memristor devices in various designs. Memristors have the potential to attract modern CMOS design since they can be fabricated in high density on the top of the silicon layer, which adds new capabilities to CMOS technology to address the challenges it faces with respect to any further scaling-down.

In this thesis, an area efficient Memristor-based digital circuits were implemented utilizing the memristive gates in the style of the integral form of the hybrid Memristor-CMOS and the memristor-only covering a number of conventional arithmetic and computational building blocks. As an example, a 4-bit Memristive Up-Down counter has been implemented using the integral form of the hybrid Memristor-CMOS. The proposed counter is realized using four T Flip-Flops (TFF) connected by AND and OR gates. There are two digital signals (UP and DOWN) that control the counting direction. The verification was performed in the Cadence environment and NC-Verilog. The simulation results have illustrated that the Up-Down counter was successfully able to start or stop counting at any logical state and resume its operation from any other desired logic state. The design demonstrates that the hybrid Memristor-CMOS based Up-Down counter requires fewer number of transistors than in a traditional CMOS based Up-Down counter with 34 CMOS devices and 110 memristors. Therefore, the design has a small layout area and shows a reasonable reduction in power consumption and delay. While other Up-Down counter designs that relied on the pure Memristive method "IMPLY" reported less area, this design overcomes the issues of delay and complexity produced by the lengthy operational steps associated with IMPLY-based design. Moreover, this work was extended in Chapter 6 to provide a comprehensive memristive design for a leaky integrate-and-fire (LIF) neuron circuit. The unique about this proposal is the involvement of the memristor devices in every aspect of the design.

The newly discovered Memristor device is finding its way into today's circuit design. These memristor-based structures hold promise to provide continued growth and will significantly enhance the speed and power of digital computing beyond Moore scaling while maintaining compatibility with standard CMOS.

I dedicated this work in appreciation to my parents as an expression of my gratitude for all their guidance, support, and encouragement throughout my life

Acknowledgements

I would like to express my gratitude and appreciation to my advisor Prof. Majid Ahmadi. Not only has he been an exceptional advisor throughout my entire academic career, but he is a truly great person who cares profoundly for his students. Without his guidance, I can unequivocally say that I would not academically be where I am today.

I would also like to express my gratitude to my co-advisor Dr. Arash Ahmadi. He is a wealth of technical knowledge, and his input is always invaluable and significant throughout the course of this work.

In addition to my advisors, I would like to thank the rest of my thesis committee, Dr. Riahi, Dr. Rashidzadeh, and Dr. Abdel-Raheem, for their participation in my seminars and their constructive comments and advice.

A very special thanks is required for Dr. Roberto Muscedere for his valuable help in RCIM Lap.

I would also like to thank all my friends and colleagues in the Research Centre for Integrated Microsystems (RCIM) at the University of Windsor.

Last but not least, my endless appreciation with sincere thanks goes to my family, who have given me endless support, and provided me with the opportunity to reach this far with my studies.

Table of Contents

List of Figure

List of Tables

List of Appendices

List of Abbreviations

CBRAM	Conductive Bridge Memory
CMOS	Complementary Metal-Oxide Semiconductor
DRAM	Dynamic Random Access Memory
EDA	Electronic Design Automation
FeRAM	Ferroelectric Memories
FIR	Finite Impulse Response
HDL	Hardware Description Language
HRS	High-Resistance State
IF	Integrate-and-Fire
IMPLY	Material Implication Logic
LIF	Leaky integrate-and-Fire
LRS	Low-Resistance State
MAGIC	Memristor Aided Logic
MRL	Memristor Ratio Logic

NVM	Non-Volatile Memory
ReRAM	Redox-Based Resistive Random-Access Memory
RRAM	Resistive Random Access Memory
SRAM	Static Random Access Memory
VLSI	Very Large-Scale Integration
VTEAM	Voltage Threshold Adaptive Memristor
PCM	Phase-Change-Memory
XNOR	Exclusive-NOR.
XOR	Exclusive-OR.

Chapter 1

Introduction

The Memristor derived from memory-resistor is a two-terminal passive device whose nonlinear behavior is described by the unique I-V curve. The curve shows that the Memristor is substantially different from the other fundamental circuit elements since it can retain the last resistance state of its variable resistance. The resistance value remains constant when no electrical field is applied, which makes this device an interesting element for future computing and memory systems. This introductory chapter reviews the memristor device definition, functionality, applications, and the thesis outline.

1.1 The Memristor Device- A Background

In his preliminary research in nonlinear circuit theory, Leon Chua predicted that a new element should exist to join the other three passive circuit elements in order to achieve the symmetrical relationship between voltage (v), current (I), charge (q), and flux (φ), as summarized in Fig. 1.1 [1] The new device would complete the circle by realizing the missing relationship between the charge and the flux. through what Chua called a memristance (M) which denotes as.

$$d\varphi = M.dq$$

At the same time, the rest of the relations have already existed: The resistor (R)links voltage and current which denotes as.

$$dv = R.di$$

The capacitor(C) links voltage and charge which denotes as.

$$dq = C.dv$$

The inductor (L)links current and flux which denotes as.

$$d\varphi = L.di$$

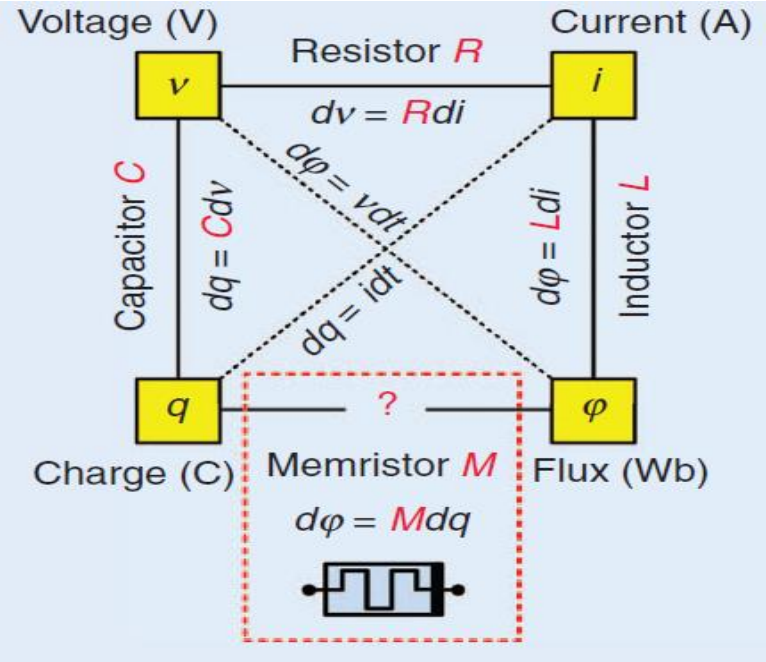


Figure 1.1: The fourth missing fundamental circuit elements [1]

Chua mathematically explored the properties of this new device and observed that the device could be described as a resistor that exhibits a form of memory. Hence, the term Memristor[2] appeared for the first time in 1971, or more broadly defined as a memristive system[3]. However, memristors attracted much attention only after 2008, when (HP Labs) [4] announced the fabrication of the first working memristor. Earlier memristor studies to capture the interest of researchers were memory switching, exploiting the device's unique characteristics of memory behavior and variable resistance state presented on the memristor I-V curve, which can be very appealing to future computing architecture[5]. Moreover, other remarkable properties of memristors, such as non-volatile, low power, and high switch speed, make this device a promising substitute for conventional high-speed memories. However, later memristor works have shown that the emerging devices are not only for memory design but also can perform traditional logical functions[6] and provide an architectural solution for future computing systems.

1.2 Memristor-Based Applications

The new emerging devices, with their exceptional properties and capabilities, have the potential to transform various fields and find applications in a wide range of areas. Such as memory storage, logic design, filter design, and neuromorphic computing. Here is a review of some of the key applications of memristors:

1.2.1 Memory

Memristors have the unique property of being non-volatile, meaning they can retain their state even without power consumption. This characteristic makes them ideal for memory applications. Additionally, low power and compatibility with CMOS are the memristor's main features to attract these devices to be a suitable alternative for traditional memory technologies such as SRAM and DRAM memories. There are several memories which have been fabricated based on memristive behavior with different switching mechanisms, such as Phase-Change-Memory (PCM) [7], Conductive Bridge Memory (CBRAM) [8], Ferroelectric Memories (FeRAM) [9], Resistive Random Access Memory (RRAM) [10] and Redox-based resistive random access memory ReRAM [11]. Each presents different characteristics regarding reliability, endurance, memory retention term, programmable states, and energy efficiency.

However, Both RRAM and ReRAM devices are considered promising candidates for the next generation of non-volatile memory applications, and they are competing to become mainstream memory technology. RRAM has a compact structure with an oxide layer sandwiched between two electrodes that send and receive electrical signals. The switching characteristics of RRAM devices are strongly associated with the distribution of oxygen vacancies in the oxide layer. As a result, the device provides a fast-switching speed, excellent scalability, and compatibility with CMOS fabrication at low programming voltage[12-13]. While the resistance switching behavior of the ReRAM device between two resistance states, the low-resistance state (LRS) and the high-resistance state (HRS) highly depends on the oxide materials, a resistance-changeable material.

The change in the resistance is achieved by a voltage pulse above a threshold value. As a result, ReRAM devices offer high-density storage, high switching speed, and relatively good endurance at low manufacturing costs.

1.2.2 Neuromorphic Computing

Brain-inspired computing is considered an alternative approach[14-17] to conventional computing systems Von-Neumann, which faces severe challenges [18] regarding computing speed and power consumption. Thus, hardware implementation for neuromorphic systems has been an ambitious research field and very appealing to computing architecture. Although CMOS offers the platform to realize neuromorphic systems, the physical implementation of the massive synaptic interconnections among neurons and the synaptic adaptability requires a large silicon area with considerable power consumption. Hence, the interest to develop a compact adaptable device obeying biological learning rules to implement synaptic connections has motivated the investigation on alternative nanotechnologies. Memristors can attract the area of neuromorphic computing as they can adapt and change their behavior over time in response to different stimulation patterns similar to the human brain[19]. Moreover, memristors can reproduce Spike Time Dependent Plasticity Mechanism observed in biological neurons (STDP) [20].

1.2.3 Memristive Logic Design

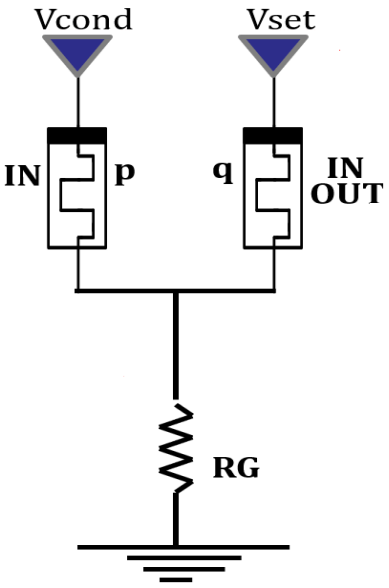
Memristor-based circuits open new pathways for exploring future memory and computing systems. This is because memristors can perform logic besides their unique memory behavior. Hence, memristor-based circuits offer the implementation of logic and memory in one platform, which can be applied to in-memory computing systems. One significant advantage of memristors' involvement in the architecture of in-memory computing is the ability to compute logic and save the data in one platform. This will satisfy this architecture's need for small hardware with powerful data processing

capability. Thus, there has been intensive research covering the field of circuit design involving memristor devices[21]. The Memristor-based logic design can be classified into analog-based and digital-based logic. In the environment of analog circuit design, memristive analog arithmetic circuits can be implemented utilizing the memristor's memory behavior[22]. Memristance values can be configured by the memristor's internal resistance states. Memristor's variable resistance will change upon an applied electrical field. While in the digital environment, memristors can perform conventional logical functions in two different approaches based on the computation of the logical states. The first method is resistance- dependent, the logical states are evaluated based on the memristance of the Memristor. Memristance values can swing between two resistance states. High Resistance State R_{HRS} represents logic ‘ 0’, and Low Resistance State R_{LRS} represents logic ‘ 1’. The second method is voltage- dependent [23]. Here the logical state of this method is determined according to the output voltage level. Thus, low and high voltage represent the logical states (‘ 0’and ‘1’), respectively.

1.3 Digital Architecture with Memristor

A rapid interest in Memristive logic design has paved the road for a broader range of memristor-based circuits implementation, covering a large number and wide scope of conventional arithmetic and computational building blocks. The Memristor-based logic design is an ambitious step toward future VLSI technology strives to serve CMOS further scaling down.

Only-Memristor based method is one of the Memristor-based logic designs utilizing only memristor devices to construct all logic gates. The value of the resistance of the output memristor defines the logical states (‘0’ and ‘1’). The resistance of the Memristor is bounded by a minimum resistance R_{ON} and a maximum resistance R_{OFF}. Material Implication Logic (IMPLY) [24] and Memristor Aided Logic (MAGIC) [25] are well-known logics in the pure memristive style. The IMPLY-based logic structure is shown in Fig. 1.2. It has two memristors connected to the ground through a resistor. The output is copied to one of the Memristors in the final computation step, exploiting the feature of the non-volatile offered by the Memristor device to use the Memristor as input or output as required.



p	q	P Imp q
HRS(0)	HRS(0)	LRS(1)
HRS(0)	LRS(1)	LRS(1)
LRS(1)	HRS(0)	HRS(0)
LRS(1)	LRS(1)	LRS (1)

Figure 1.2: IMPLY logic gate (a) Schematic of Memristor-based. (b)truth table.

This logic is sequential. Hence, a multi-level voltage is required to achieve the computational steps. IMPLY logic requires a lengthy sequence of stateful logic operations to synthesize a Boolean function. Thus, the number of computational steps required for n-input logic gate is equal to $2^{n-1}+1$ [26].This sequential nature causes the logic to suffer from extensive delay. The other drawback facing this logic is the complicated control circuitry required for READ / WRITE operation. While the logical structure of MAGIC requires two memristors for the inputs and another separate memristor for the output. MAGIC can construct the entire Boolean traditional gates such as AND, NAND, OR, and NOR as shown in Fig.1.3. ,with two sequential steps, initialization step and computational step. The initialization step is achieved by writing a known logic state for every Memristor separately [27]. The two sequential steps require only one voltage level to perform the logic. However, cascading two or more logic circuits with MAGIC is challenging. Also, in some input combinations, the voltage across the output memristor is insufficient to change the logical state. Hence, the output memristor remains at the initialized state during the performing of logic operation and causes the loss of the input for the next operation.

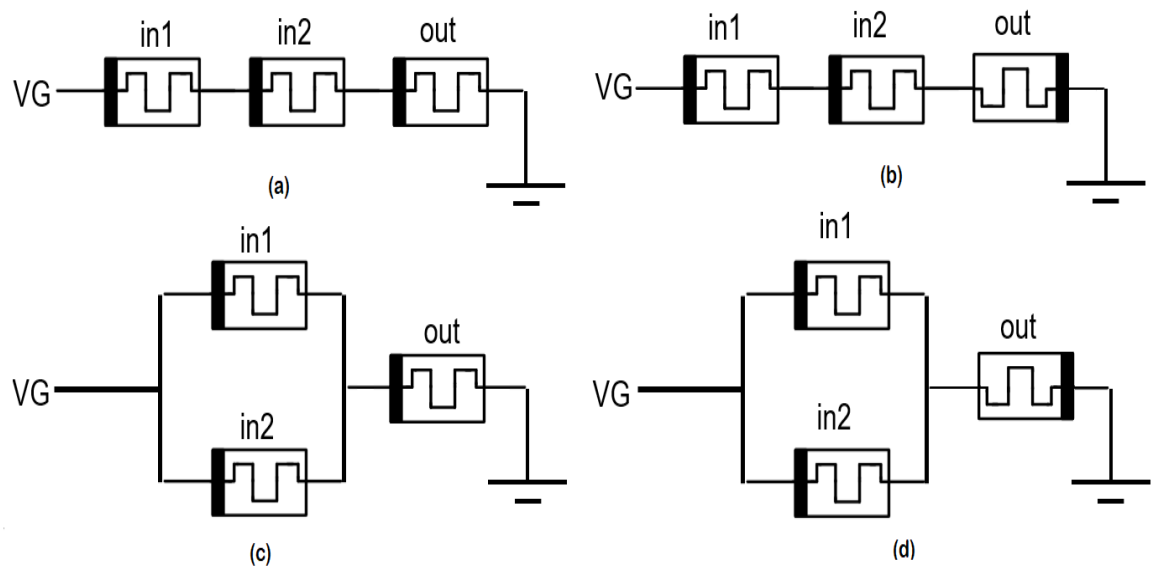


Figure 1.3: Schematic of Memristor-based MAGIC logic gate.
(a) AND. (b) NAND (c) OR. (d) NOR.

Memristor Ratioed Logic (MRL) [28] is another memristor-based logic design method. In this method, Memristor devices and CMOS transistors can be combined in an integrated platform. By fabricating memristors devices in the metal layers inside the CMOS die, since CMOS-based dies are composed of a single layer of CMOS transistors with multiple metal layers stacked on top. The metal layers are etched to compose the wires that route the connections between the transistors on the CMOS layers[29]. Fig.1.4. [29] shows the integrated platform of CMOS-Memristor, which provides MRL logic with the capability to construct all logical gates by utilizing Memristors and stacking them between the upper layers of the CMOS inverter [30]. MRL-based AND gate and OR gate implementation requires only two memristors. V_1 and V_2 denote the voltages applied to both memristors, which change their resistance state based on the voltage level. V_{OUT} denotes the output voltage of the logic circuit, which is determined by the voltage divider across both memristive devices. The implementation of MRL- based NAND and NOR gates are similar to MRL-based AND and OR gates, except both NAND and NOR need a CMOS inverter to facilitate the circuits with the interface and control operation [31], which eliminates the effect of signal degradation associated with MRL-based AND and OR gates.

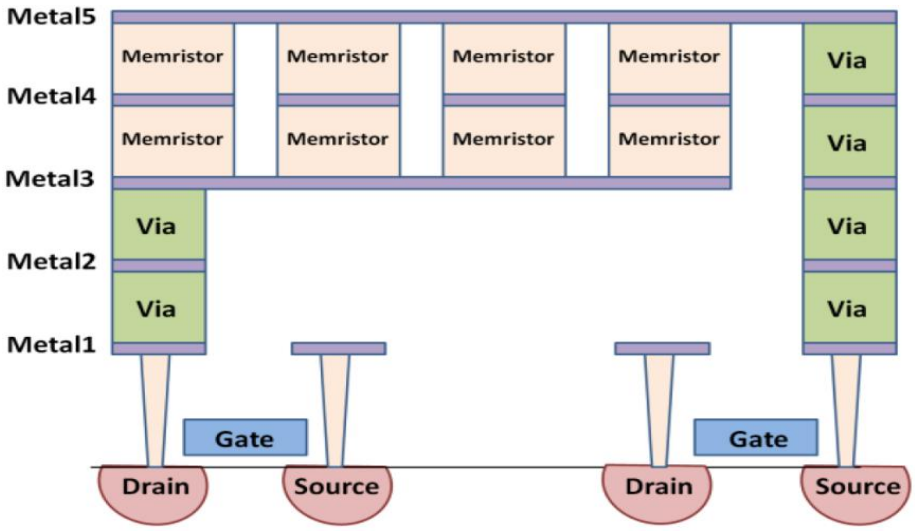


Figure 1.4: The integration of memristors on top of CMOS transistors. The memristors are located between metal layers 3 and 4, and 4 and 5[29]

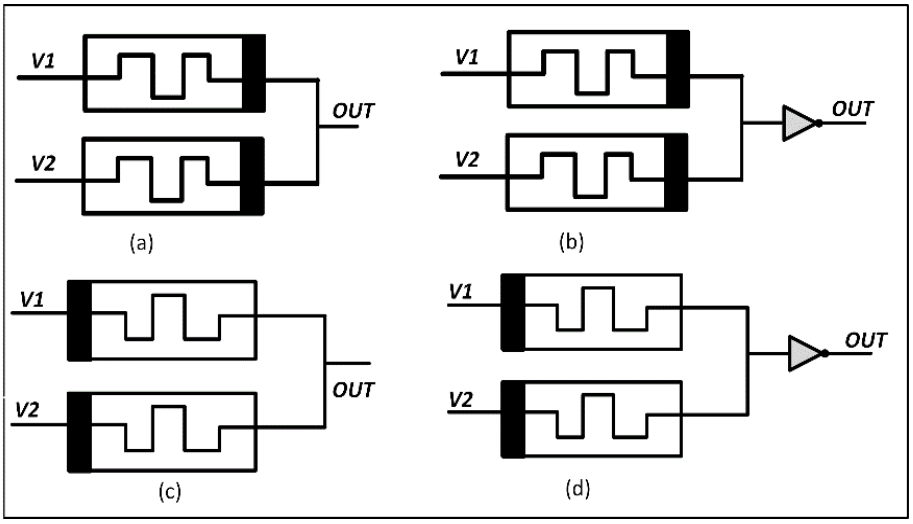


Figure 1.5: Schematic of MRL based logic gates
(a) AND, (b) NAND, (c) OR and (d) NOR

The schematic circuit for MRL-based basic gates AND, OR, NAND, and NOR are displayed in Fig.1.5. MRL logic does not require additional voltage supplies, with only one computational step required since memristors function as computational elements and do not store data.

Moreover ,all circuit testing and simulations of memristor-based digital circuits covered in this dissertation were simulated in the Cadence Virtuoso environment utilizing different memristor Verilog-A models due to the lack of real physical memristor device layout tools. Hence, choosing an accurate memristor model that simplifies the implementation of memristor-based digital circuits is crucial. Verilog-A models for memristor devices presented in appendix A and appendix B were utilized to provide the desired memristive behavior for the proposed digital circuits.

1.4 Memristive logic design challenges

Implementing logic with Memristor is a new vision for digital logic architecture. However, several challenges associated with Memristor-based circuit design can impose restrictions on the performance and the reliability of memristive digital architecture. One of the first obstacles hindering the logical design with memristors is the non-ideal characteristics exhibited by Memristors. These non-idealities characteristics, such as non-linear resistance switching and limited endurance over time, can impact the reliability and accuracy of any Memristor-based logic circuit design. Although memristive behavior has been observed in various materials, including TiO_2 [35], HfO_2 [36], Al_2O_3 [37], Nb_2O_5 [38], $SrTiO_3$ [39], identifying and developing suitable materials that exhibit the desired memristive behavior is crucial. The material should demonstrate stable resistance switching characteristics. Also, the issue of sneak path current [40] severely affects information storage, computation, and data processing in the crossbar structure [41]. The structure attracts IMPLY and MAGIC-based logic, which allows the integration of a large number of memristors in a small area and enables high storage capacity and computational capabilities.

A sneak path current is the undesired current flowing through the path, which is typically the path of least resistance due to the non-linear resistance behavior of the individual memristors within the crossbar array.

The resistance of a memristor can change dynamically based on the amount and direction of charge that has passed through it. This change of resistance influences the path that the current takes as it flows through the crossbar array.

The sneak path current leads to an error in reading / writing the resistance state of memristors. Also, it induces a high amount of energy consumption inside the crossbar array. Integration with existing CMOS technology is another problem that arises in the crossbar structure. This integration requires addressing compatibility issues, developing suitable fabrication processes, and designing circuit architectures. Also, Memristor based circuit designs are subject to signal degradation mechanisms that can impact their performance over time due to various factors related to memristors, such as electromigration, thermal stress, or chemical reactions.

In fact, the degradation processes can result in changes to the device's resistance characteristics, leading to performance degradation. The degradation effect is pronounced in the voltage-based logic design MRL when Memristor is unable to drive the following gate stage. Adding a CMOS inverter can remedy this issue by amplifying the output to restore the signal. However, this remedy comes at the price of power consumption [42].

1.5 The Thesis Motivation

The unique characteristics of the new emerging devices make them a compelling area of research and development for future memory and computing technologies. Also, memristors can solve CMOS's fundamental limitations of further scaling down by integrating the memristor devices with CMOS transistors. The integration of memristor devices with conventional CMOS is an auspicious tool for implementing digital architecture since Memristors can be loaded and stacked between the upper layers of the CMOS layers in high circuit density to construct all logical gates.

1.6 The Thesis Objective

In this research, memristors' capabilities and limitations are studied and analyzed. Hence, several models of memristors were utilized to implement various memristor-based circuits for different applications. The integration of memristors with CMOS as well as only-memristors methods, are explored for logic circuits design beyond Moore's approach.

1.7 The Thesis Outline and summary of contributions

This thesis is organized as a collection of published or submitted papers. All of the papers describe research results obtained during this PhD study.

Chapter 2 presents a compact, low power design of a hybrid Memristor-CMOS based Finite Impulse Response (FIR) Filter with reasonable performance compared to the CMOS-based design. The proposed design has been described using Verilog High Description Language (HDL) and tested with Cadence design systems, NC-Verilog, and MATLAB. The significant aspect of this proposed architecture was the standard cells memristive library design, which contains all characterized cells employed in the FIR filter. This library is utilized by Synopsys compiler to produce memristive netlist gates. The simulation results have shown that the behavioral model of the design can distinguish between all input signals and passes only signals with the desired frequency. The proposed hybrid Memristor- CMOS based FIR is shown to be more efficient in terms of area, consumed power, and delay.

Chapter 3 proposes a fast, low area, and low power for the Up-Down counter. Hence, a hybrid Memristor-CMOS platform MRL was utilized to implement a 4-bit Up-Down counter with 34 MOSFETs and 110 memristors. The proposed counter employs fewer numbers of transistors than the conventional CMOS counter and shows a reasonable reduction in power consumption and delay. Although the IMPLY- based Up-Down counter reported less area. The MRL-based Up-Down counter overcomes the issues of delay and complexity produced by the lengthy operational steps associated with the IMPLY-based counter. The simulation results have illustrated that the Up-Down counter was successfully able to start or stop counting at any logical state and resume its operation from any other desired logic state, which proven the design and functionality.

Chapter 4 presents a design for the building blocks of XAX- module, a core element used as digital circuit for elliptic curve cryptography based on the MRL scheme. The memristor device of $Pt\ /TaOx/Ta$ has been used to provide the design with the Memristive behavior. The design of MRL- based XAX-module comprises simple blocks of the DFF circuit together with the XOR-AND-XOR function. The MRL- based DFF is the main component in the module in terms of device numbers. The MRL- based DFF utilizes 16 Memristor devices and 14 MOSFETs, while the XOR-AND-XOR function is another block that participates in the XAX-module with 12 memristor devices and 4 MOSFETs. In comparison, the same CMOS-based design employs 24 MOSFET devices. This work proves that utilizing MRL- based gates in the blocks that made up the XAX structure implies a significant reduction in the number of transistors. The Memristors can be fabricated on the upper layers of CMOS inverter. This Combination has led to almost 15% in area saving compared to the same CMOS-based-XAX module. This is mainly because the layout area estimate of the design was carried out based on the size of CMOS inverter occupied by each cell. From the performance point of view, the design power consumption and delay were also examined and compared against other CMOS-based designs. The design was implemented and verified in Cadence Virtuoso at each stage to confirm its functionality.

Chapter 5 proposes a fast and efficient area Memristor-only-based full adder and a hybrid CMOS/Memristor-based full adder. In the Memristor-only style, the MAGIC logic is favorited over IMPLY logic in the implementation of the full adder due to the reduced area layout offered by the MAGIC approach thanks to the simplicity of mapping MAGIC gates to the crossbar array, whereas mapping IMPLY gates require a resistor before any row which increases the design area. Also, IMPLY requires more voltage sources than MAGIC to perform the logic computations. MAGIC-based full adder design utilizes only 25 memristors, while the hybrid CMOS /Memristor full adder requires 18 Memristor devices and 4 MOSFETs. Consequently, both designs occupy a small area as compared to the CMOS- based design. However, the MAGIC-based full adder is more area efficient than the MRL-based full adder. While the MRL- based full adder is a considerably low-cost design since it is CMOS compatible and requires only one voltage level. Moreover,

the proposed MAGIC-based full adder has fewer computational steps in comparison to the same IMPLY-based design, another pure memristor method. The Memristor device of VTEMA and Pt / TaOx / Ta have been successfully used to provide the design with the Memristive behavior while simulations confirm the proposed full adder functionality.

Chapter 6 provides a design for a leaky integrate-and-fire (LIF) neuron circuit with memristive behavior. The proposed architecture is based on the hybrid CMOS/Memristor gates, which enable integration with existing CMOS-based neurons to support large-scale neuromorphic systems. The exception of this proposal is the involvement of the memristor devices in every aspect of the design. The design was divided into three parts: A synaptic block with a memristive up-counter, a core neuron circuit with an MRL-based up-down counter previously explained in Chapter 3, and a comparator to generate a spike when the membrane potential exceeds the threshold value. The LIF neuron circuit is area-efficient due to the MRL-based gates employed in the proposed. The design was implemented and verified in Cadence Virtuoso at each stage to confirm its functionality.

Chapter 7 presents thesis conclusion and future Work.

References

[1] I. Vourkas and G. Sirakoulis, "Emerging Memristor-Based Logic Circuit Design Approaches: A Review," *in IEEE Circuits and Systems Magazine*, third quarter, vol. 16, no. 3, pp. 15-30, 2016.

[2] L. Chua, “Memristor-The missing circuit element,” *IEEE Transactions on Circuit Theory*, vol.18, no.5, pp. 507-519, Sep. 1971.

[3] L. Chua and S. Kang, “Memristive Devices and Systems,” *Proceedings of the IEEE*, Vol. 64, No. 2, pp. 209-223, February 1976.

[4] D. Strukov, G. Snider, D. Steward, and R. Williams. The missing memristor found. *Nature*, 453(7191):80–83, May 2008.

[5] J. Borghetti, Z. Straznicky, X. Li, D. Ohlberg, W. Wu, D. Stewart, and R. Williams, “A hybrid nano memristor/transistor logic circuit capable of self-programming,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 6, pp. 1699–1703, 2009.

[6] J. H. Poikonen, E. Lehtonen and M. Laiho, "On Synthesis of Boolean Expressions for Memristive Devices Using Sequential Implication Logic," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 7, pp. 1129-1134, July 2012.

[7] H. Wong, S. Raoux, S. Kim, J. Liang, J. Reifenberg, B. Rajendran, M. Asheghi, K. Goodson, “Phase change memory”, *Proceedings of the IEEE* 98 (12) ,2201–2227,2010.

[8] Y. Wang, C. Zhang, H. Yu, W. Zhang, “Design of low power 3d hybrid memory by nonvolatile CBRAM-crossbar with block-level data-retention”, *in Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design, ACM*, pp. 197–202,2012.

[9] S. Ali, and P. Glenn Gulak. "A survey of circuit innovations in ferroelectric random-access memories." *Proceedings of the IEEE* 88, no. 5 ,pp.667-689, 2000.

[10] W. Philip, H. Lee, S.Yu, Y. Chen, Y.Chen, B. Lee, F.Chen, and M.Tsai. "Metal–oxide RRAM." *Proceedings of the IEEE* 100, no. 6. pp 1951-1970, Jun.2012.

[11] R. Waser, R. Dittmann, G. Staikov, K. Szot, Redox-based resistive switching memories–nano ionic mechanisms, prospects, and challenges, *Advanced Materials*, vol. 21 ,pp. 2632–2663,2009

[12] H. Akinaga and H. Shima, “Resistive Random Access Memory (ReRAM) based on metal oxides,” *Proc. IEEE*, vol. 98, no. 12, pp. 2237–2251, Dec. 2010.

[13] H.Wong, H.Lee, Y.Chen and M. Tsaj, "Metal–Oxide RRAM" *in Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951-1970, June 2012.

[14] G. Indiveri, E. Chicca and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *in IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211-221, Jan. 2006.

[15] B. Varkey, P. Gao, E. McQuinn, S. Choudhary, A. Chandrasekaran, J. Bussat, R. Icaza, J. Arthur, P. Merolla, and K. Boahen., "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *in Proceedings of the IEEE*, vol. 102, no. 5, pp. 699-716, May 2014.

[16] C. Mead, "Neuromorphic electronic systems," *in Proceedings of the IEEE,* vol. 78, no. 10, pp. 1629-1636, Oct. 1990.

[17] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum and H. -. P. Wong, "An Electronic Synapse Device Based on Metal Oxide Resistive Switching Memory for Neuromorphic Computation*," in IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2729-2737, Aug. 2011

[18] A. Stefano, P. Narayanan, , H. Tsai, , R. Shelby, I. Boybat, C. Nolfo, S. Sidler, M .Giordano, M. Bodini, , N. Farinha and B. Killeen, "Equivalent-accuracy accelerated neural-network training using analogue memory". *Nature* 558, pp. 60–67, 2018.

[19] S. H. Jo, T. Chang, I. Ebong, B. Bhadviya, P. Mazumder, W. Lu, Nanoscale memristor device as synapse in neuromorphic systems, *Nano Letters* 10 (4) pp.1297–1301,2010.

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

[21] C. Wang, Z. Jiang, X. Malik, A. Pan, Y. Stathopoulos, S. Serb, A. Wang, S. Prodromakis, and C. Papavassiliou, " Multi-State Memristors and Their Applications: An Overview*," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 12, no. 4, pp. 723-734, Dec. 2022.

[22] Y. Pershin, M. Ventra, "Practical approach to programmable analog circuits with memristors", *IEEE Transactions on Circuits and Systems I*: Regular Papers 57 (8) pp.1857–1864, 2010.

[23] T. Singh, "Hybrid Memristor-CMOS (MeMOS) based Logic Gates and Adder Circuits," CoRR, arXiv:1506.06735 [cs.ET], pp. 1-11, June 2015.

[24] 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*," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2054-2066, Oct. 2014.

[25] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G.Friedman, A.Kolodny, and U.C. Weiser, "Magic-memristor aided logic," *IEEE Transactions on Circuits and Systems II:* Express Briefs, vol. 61, no. 11, pp. 895–899, Nov 2014.

[26] J. Borghetti, J. Philip , J. Yang, R. Duncan, and R. Stanley , "Memristive switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, pp. 873–876, Apr. 2010.

[27] P. Thangkhiew, R. Gharpinde, P. Chowdhary, K. Datta and I. Sengupta, "Area efficient implementation of ripple carry adder using memristor crossbar arrays," *11th International Design & Test Symposium (IDT)*, pp. 142-147. 2016.

[28] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C.Weiser and E. G. Friedman, "MRL-Memristor Ratioed Logic," 13th *International Workshop on Cellular Nanoscale Networks and their Applications*, Turin, Italy, pp. 1-6, 2012.

[29] S. Kvatinsky, Y. Nacson, Y. Etsion, A. Kolodny, U. Weiser, R. Patel, and E. Friedman, "Memristive multistate pipeline register". *In 2014 IEEE. 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)* ,pp. 1-2, July 2014.

[30] I. Vourkas and G. C. Sirakoulis, "Emerging memristor-based logic circuit design approaches: A review," *IEEE Circuits and Systems Magazine*, vol. 16, no. 3, pp. 15–30, 2016.

[31] K. Cho, S. Jin Lee, and K. Eshraghian,"Memristor-CMOS logic and digital computational components" *Microelectronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.

[31] Choi, B.J., Jeong, D.S., Kim, S.K., Rohde, C., Choi, S., Oh, J.H., Kim, H.J., Hwang, C.S., Szot, K., Waser, R. and Reichenberg, B., Resistive switching mechanism of TiO_2 thin films grown by atomic-layer deposition. Journal of applied physics, 98(3), p.033715. 2005.

[32] P. Chen and S. Yu, "Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design," *IEEE Transactions on Electron Devices*, vol. 62, no. 12, pp. 4022–4028, Dec 2015.

[33] A. Siemon, S. Menzel, A. Marchewka, Y. Nishi, R. Waser, and E. Linn, "Simulation of taox-based complementary resistive switches by a physics-based memristive model," *in 2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1420–1423, June 2014.

[34] S. Kvatinsky, M. Ramadan, E. Friedman and A. Kolodny, "VTEAM: A General Model for Voltage-Controlled Memristors," *in IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786-790, Aug. 2015.

[35] Kim, Y.M. and Lee, J.S., Reproducible resistance switching characteristics of hafnium oxide-based nonvolatile memory devices. *Journal of Applied Physics*, 104(11), p.114115. 2008.

[36] Wu, Y., Yu, S., Lee, B. and Wong, P., Low-power TiN/Al2O3/Pt resistive switching device with sub-20μ, A switching current and gradual resistance modulation. *Journal of Applied Physics*, 110(9), p.094104. 2011.

[37] Chen, L., Sun, Q.Q., Gu, J.J., Xu, Y., Ding, S.J. and Zhang, D.W., 2011. Bipolar resistive switching characteristics of atomic layer deposited Nb2O5 thin films for nonvolatile memory application. *Current Applied Physics*, 11(3), pp.849-852. 2011.

[38] Szot, K., Speier, W., Bihlmayer, G. and Waser, R., Switching the electrical resistance of individual dislocations in single-crystalline SrTiO3. *Nature materials*, 5(4), pp.312-320. 2006.

[39] M. Zidan, A. Eltawil, F. Kurdahi, H. Fahmy, and K. Salama, "Memristor multiport readout, a closed-form solution for sneak paths", *IEEE Transactions on Nanotechnology*, vol. 13, no. 2, pp. 274-282, 2014.

[40] L. Xie, H. Nguyen, M. Taouil, S. Hamdioui and K. Bertels, "Fast Boolean logic mapped on memristor crossbar," 33rd *IEEE International Conference on Computer Design (ICCD)*, New York, NY, USA, pp. 335-342, 2015.

[41] A. James, \Memristor threshold logic: An overview to challenges and applications," *arXiv preprint arXiv*:1612.01711, 2016.

Chapter 2

Hybrid Memristor-CMOS Based Finite Impulse Response Filter Design

2.1 Introduction

The development of CMOS transistors shows major concerns, such as increased leakage power, reduced reliability, and high fabrication cost [1]. These factors have affected chip manufacturing procedure and functionality severely. Therefore, the demand for new devices is increasing. Memristor is considered as one of the key elements in memory and information processing design [2], [3] due to its small size, long-term data storage, low power, and CMOS compatibility. Recently, several methods in a logic design employing Memristors have been reported [4]-[7]. Material Implication Logic (IMPLY) [4], [5] is aimed to design logic gates with Memristors only. However, the involvement of a high number of computational steps and the need for READ/WRITE circuit are obstacles facing this logic. Memristor Aided Logic (MAGIC) [6] is another pure Memristive logic method similar to IMPLY logic. This logic requires two steps "initialization procedure prior to a computational step". However, cascading two or more logic circuits with MAGIC is difficult to implement. Memristor Ratioed Logic (MRL) [7] method is hybrid CMOS-Memristor circuit that has logical states ('0' and '1') represented based on the level of the output voltage. This method is suitable for most logic designs because of

CMOS compatibility, which offers less area and power consumption compared to conventional CMOS. There are several memristor based arithmetic and computational Circuits described in [8] and [10] using MRL; however, the area of filter design using Memristors has received less attention. In [8] Memristors have been used in the LC filter as a damping component. In [9] Memristor devices were used to provide weights for 6-tap FIR filter. In this chapter, Memristor based RRAM device has been employed to implement hybrid Memristor-CMOS based FIR filter. Unlike earlier memristor design [9], the proposed approach utilizes the designed cell library. The cell library proposed characterizes basic Memristive logic gates based on MRL method. The rest of this chapter is organized as follows. An introduction to the Memristor model is described in Section 2.1. In section 2.2. The design approach and device modeling are explained. In section 2.3. FIR filter design is presented, Section 2.4. A case study of the DFF circuit was presented to show the utility of cell library designed. In Section 2.5. The verification of the proposed filter is explained. In section 2.6. Simulation results and comparison with CMOS based FIR is provided. Finally, remarks and conclusions are proposed in section 2.7.

2.2 Design approach and device modelling

2.2.1 Memristor ratioed logic design

Memristor Ratioed Logic MRL design is based on the integration of CMOS technology with Memristor to implement varieties of logic gates. Pure Memristive design can be used to design AND/OR gates, while NAND and NOR can be designed by connecting the output of AND/OR Memristive gates to CMOS inverter. This arrangement is not only to obtain NAND and NOR gates but also overcomes the signal degradation problem. MRL method is a voltage-based model, hence the logical state of MRL is defined by the output voltage level, where, voltage level represents the high state “1” and the low state “0” by the high voltage and the low voltage respectively.

Table 2.1: RRAM Memristor parameters for simulation

Parameter	**I_e (A)**	**G_0 (m)**	**L (nm)**	**G_{max} (nm)**	**G_{min} (nm)**
Value	$6.14e^{-5}$	$2.75e^{-10}$	*5*	$6e^{-12}$	$3.14e^{-14}$

2.2.2 Device modeling

Metal oxide-based resistive switching memories RRAMs are intended for use in wide range application of nonvolatile memory. In this chapter, the Memristor based RRAM model [11], has been employed to implement the Memristive behavior of the devices in which resistance varies depending on the value, direction, and duration of the applied voltage. The device resistance is altering between Low Resistance State (LRS) and the High Resistance State (HRS). The current is dependent on the oxide layer state while the rate of vacancy generation (E_{ag}) /recombination (E_{ar}) has a direct impact on the oxide layer state. A Memristor based RRAM is simulated in Cadence Virtuoso. The utilized parameters for the model are specified in Table 2.1. Where I_e, g, g_0, are hopping current density, gap length, which can either be minimum value g_{min} or maximum value g_{max} based on the applied voltage and window resistance coefficient.

2.3 Finite impulse response filter design

Finite Impulse Response FIR filter is a core element in most applications of signal and image processing [12]. In these applications, low power, an area efficient, and high speed are required when designing these filters. Several stages are involved in the design and implementation of FIR filters based on the number of required coefficients. In this chapter, a low pass FIR filter with 15 coefficients is considered. Therefore, the designed filter has 15 stages. The characteristic of this filter is shown in eq 2.1

$$Y[n] = b_0X[n] + b_1X[n-1] + \cdots b_kX[n-k] \qquad (2.1)$$

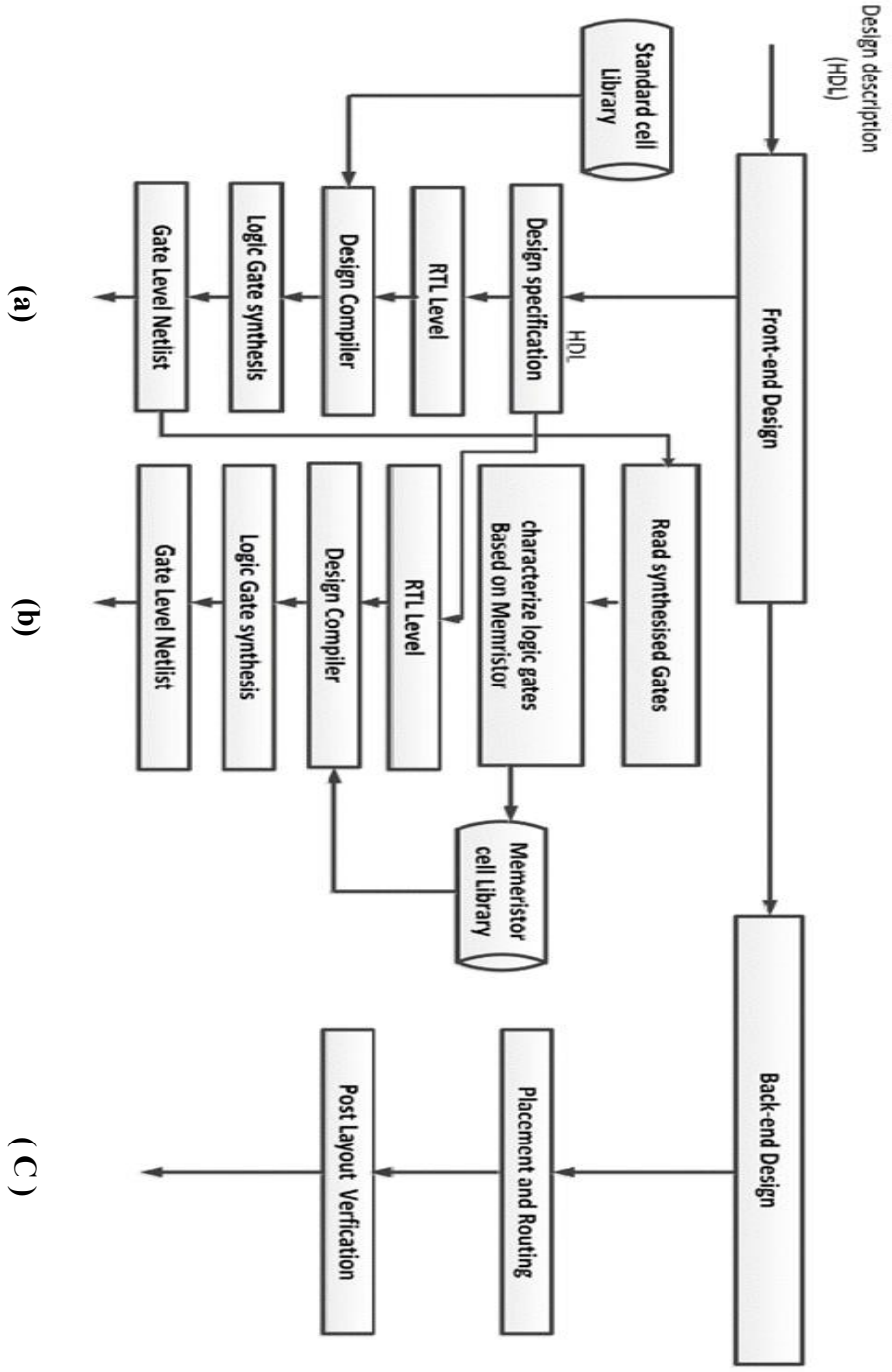


Figure 2.1: Flow chart displaying design flow based on Synopsys EDA tool for proposed FIR filter. (a) netlist CMOS Base (b) netlist memristive Base, (C) design layout.

Several procedures are involved in the implementation of the proposed filter as shown in Fig. 2.1. The design starts by describing the specification of the filter using Verilog High-Definition Language (HDL). The hardware description should be successfully verified and tested by NC-Verilog Cadence simulator using a well-designed test bench. Next, Synopsys design compiler goes through the descriptive Register - Transfer Level (RTL) in order to convert it to CMOS based gate netlist. This conversion is achieved from the characterized gates data provided by the CMOS standard cell library. Consequently, the gate netlist is verified and tested with the same test bench. Besides, the generated CMOS gate netlist is extracted to obtain all gates involved in the design. Finally, the Memristive gates which have been implemented and tested using MRL method were characterized to build standard Memristive cell library to be utilized by Synopsys compiler to generate the Memristive gates netlist . The synthesized FIR memristor based netlist gates should be verified and tested again with the test bench.

2.4 Case Study

The core of this work is to create a standard cell library for Memristive gates based MRL design. This procedure is very sensitive since it requires implementing and validating all gates in the design. As an example, a sequential D-Flip Flop (DFF) circuit shown in Fig.2.2 has been chosen to describe all steps involved in the design procedure. DFF has been implemented in the Cadence schematic level, verified in the behavioral level with NC-Verilog, and characterized its Memristive logic gates by Cadence tools. All characterized gates data is fed to the standard library.

This library contains information about the input capacitance of each pin of DFF, output net capacitance, transition time, values of rise /fall delay and power consumption. The synthesis tools rely on capacitance values to compute delay and dynamic power. Therefore, it is vital to calculate the capacitance values for each pin " input /output " where C is calculated based on eq (2.2)

$$C = \frac{i}{VDD} \int_{t1}^{t2} i(t)dt \qquad (2.2)$$

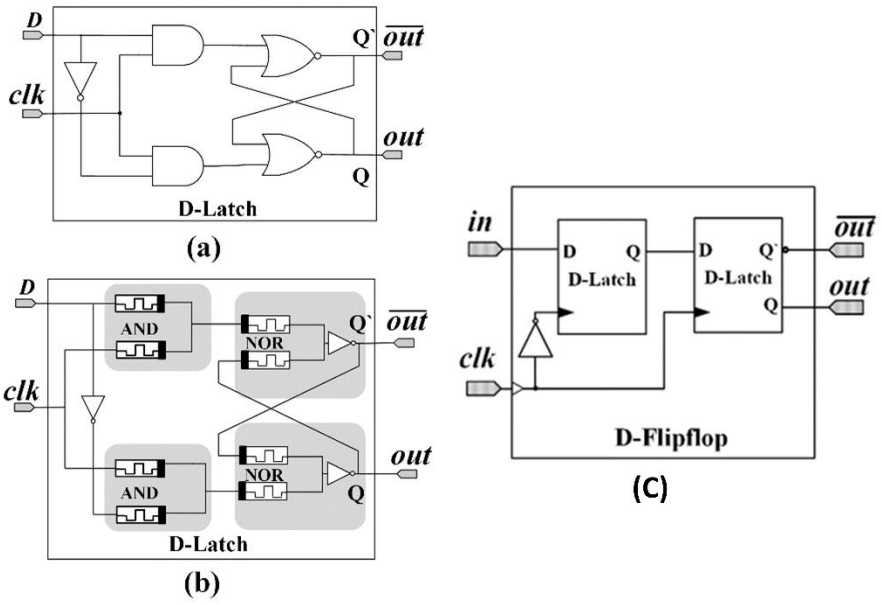
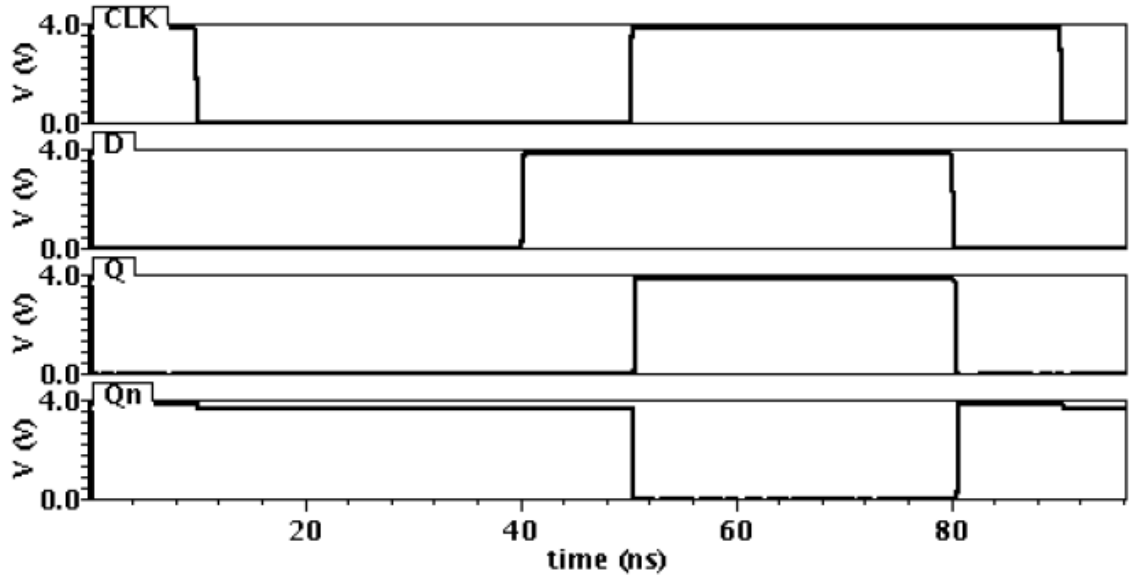


Figure 2.2: (a) D-latch schematic circuit. (b) MRL based D-latch.
(c) Implemented Memristor based DFF

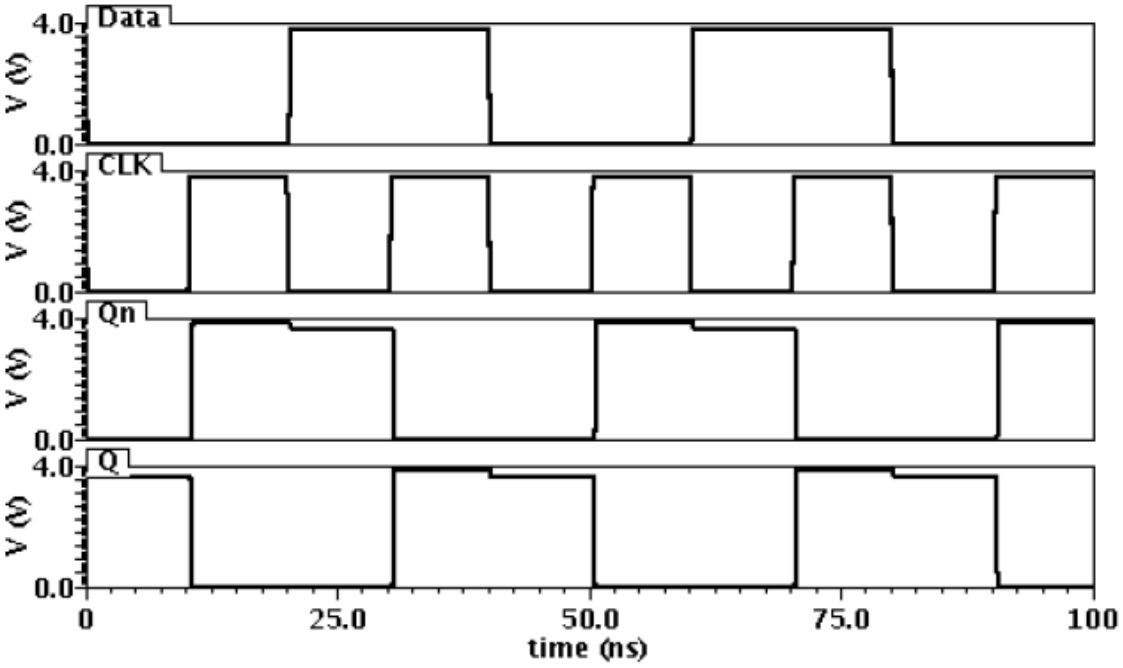
Here i(t) is the current passing through DFF. Since capacitance values are determined, the synthesis compiler can carry out the power consumption based on the energy values provided by the library. All energy values are kept in the library table after measuring DFF consumed energy. Eq (2.3) is used to calculate dynamic power consumption of the designed circuit.

$$PD = \alpha C f V_{DD} \tag{2.3}$$

Here α, C, and f, are switching activity factor, capacitance, and design frequency respectively. In addition, the DFF propagation delay is computed by measuring the time interval between the input slew and output slew. The slew for DFF alteration times is defined as the time when signal rises from 30% to 70% and falls from 70% to 30% of its V_{DD}. While DFF area approximation is linked directly to the size of CMOS inverter in every cell. Designed Memristive DFF based MRL has 16 memristors and 14 MOSFET.



(a)



(b)

Figure 2.3: Simulation results in schematic level.(a) MRL-based D-Latch.(b) MRL-based DFF.

Therefore, memristors are loaded on inverters between the upper layer of CMOS metal. Fig.2.3 shows both MRL based D-latch and DFF simulation results in the schematic level.

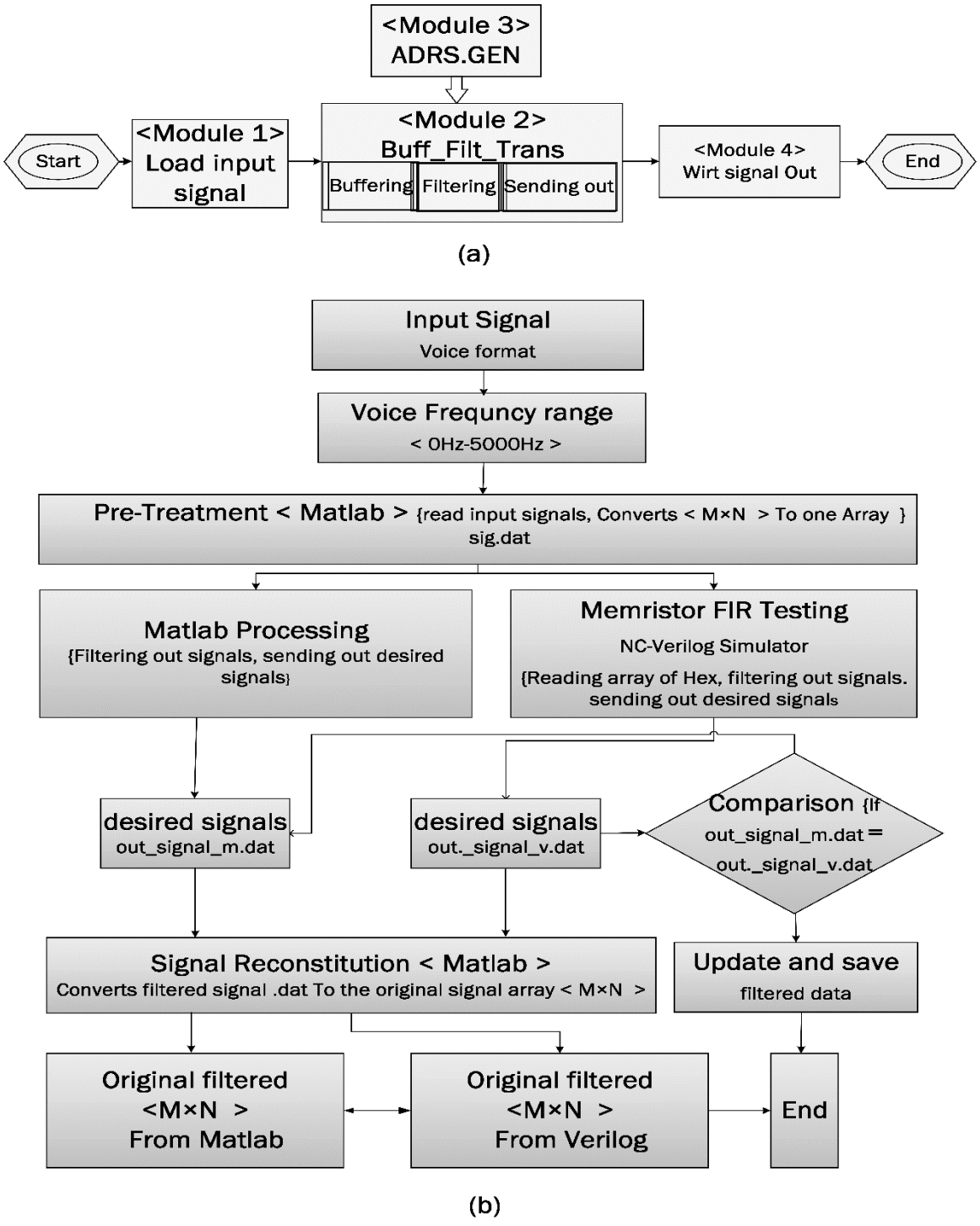


Figure 2.4: (a) Organization of the proposed FIR.

(b) Verification flowchart for proposed FIR

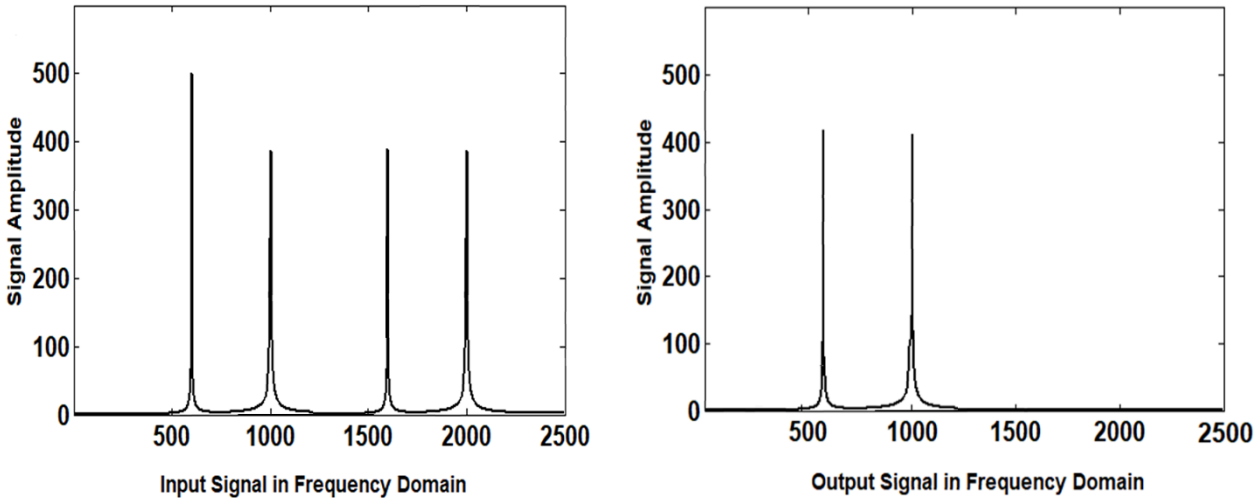
2.5 Design Verification

The verification of the proposed filter is achieved by making a comparison between the Memristive based filter output signals and the expected output signal which has been generated in MATLAB by the designed Verilog test bench. This mechanism of verification can be illustrated as shown in fig.2.4. It involves primarily several modules which are: <Module1>, <Module 2>, <Module 3> and <Module 4>. <Module 1> and <Module 4> are responsible for loading input signals “8 bits every cycle” and write the filtered signals in the output file “hex decimal format” respectively. Whereas, <Module 2> is responsible for storing data temporarily and sending out undesired frequencies. And <Module 3> is an address generator uses flags to make all the design modules synchronized under one main clock “CLK” which is lead to a great impact on speed enhancement and power consumption. Thus, signals flow throughout the modules are explained as flow. <Module 1> input signals received and stored temporarily in register, and then signals are sent serially every clock cycle to <Module 2> where signals are stored again in another register, also <Module 2> communicates with <Module 3> in order to use the generated addresses to find the desired signals and pass it serially to <Module 4> as a final stage. At the end, the desired signals are written in the form of the array corresponding to the filtered signals.

2.6 Experimental Results

Designing the hybrid Memristor-CMOS based FIR has been executed through different stages, from implementing the filter in the schematic level to test and verify the filter behavior in Cadence “TSMC 180 nm cell library”. The design utilized a Memristor based RRAM device. All related parameters for the device provided in Table 2.1, were employed to implement the Memristive behavior of the device. The filtering process of the proposed filter was verified by using a well-designed test bench, the implemented FIR filter was simulated and tested in the NC-Verilog Cadence simulator. The Verilog language can read/write files from a storage environment, making it possible to design a test bench to read data from a storage device, generate stimulus signals for the Verilog test module, and write the results to a storage device. Thus, the designed test bench for

the FIR filter is based on the aid provided by MATLAB. The test bench reads the input signals generated in MATLAB with different frequencies from the file "sig.dat", and then the output signal from the filter is stored on the file "out_signal_v.dat" to be later read in MTLAB. The filter has been designed with a cutoff frequency of 0.5 Π, where Π equals Fs/2 and Fs is the sampling frequency which is equal to 5000Hz. Therefore, the resulting signal from the file "out_signal_v.dat" shown in Fig.2.5. confirms that the FIR filter can distinguish between all input signals and filter out the two signals of 600Hz,1000Hz which lay within the pass band rejoin.



(a) **(b)**

Figure 2.5:(a) Input signals in frequency domain at 600Hz,1000Hz,1600Hz and 2000Hz.
(b) Filter output signal in frequency domain at 600Hz and 1000Hz.

Table 2.2: The characterized cells employed in the design

Gate	*Description*	*Equation*
AND2X1	Provides the logic AND for two inputs	$Y = (A.B)$
OR2X1	Provides the logic OR for two inputs	$Y = A+B$
XOR2X1	Provides the logic Exclusive OR for three inputs	$Y= A \oplus B \oplus C$
INVX	Provide logical inversion of single input	$Y = \overline{A}$
ADDFX2	Provide arithmetic sum S and carry out C	$S = (A \oplus B \oplus C)$, $C = (A \oplus B).C+ (A.B)$
DFFHQX1	positive edge triggered static D type flip flop	Case study
NOR2X1	Provide logical NOR	$Y = \overline{(A + B)}$
NAND2X1	provides the logical NAND of two inputs	$Y = \overline{(A . B)}$

Table 2.3: Synthesis results of memristor based DFF

Design	Area(μm^{-2})	Power(µW)	Delay (ns)
CMOS	459.04	83.60	*0.37*
Memristor	379.20	41.29	*0.16*

Table 2.4: Synthesis results of memristor based FIR filter

Design	Area(μm^{-2})	Power(*m* W)	Delay (ns)
CMOS	95813.62787	3.3099	0.36
Memristor	10973.69952	0.8776	0.31

One significant aspect of this proposed architecture was the design of the standard cells Memristive library which contains all characterized cells employed in the design are displayed in Table 2.2, such cells are OR, NOR, AND, NAND, DFF and variety of other cells with specified Boolean function.

In this chapter, a case study of the DFF circuit was presented to show the accomplishment of cells characterization process. Synopses synthesis tool utilizes the built library based memristor to deliver the DFF netlist and provide measurements such as power consumption, area, and delay as shown in Table 2.3. The results have proven that Memristive DFF with 16 memristors and 14 MOSFET has less area compared to DFF based CMOS. The design memristor base FIR appears to be compact compared to CMOS base FIR. Consumed power and delay are comparably reduced as well, as shown in Table. 2.4.

2.7 Conclusion

In this chapter, a hybrid memristor-CMOS based FIR filter design has been proposed with Memristive RRAM devices. The proposed filter has been verified and tested at different stages of the design in Cadence “TSMC 180 nm cell library”, NC-Verilog, and MATLAB. Although other modern technologies of the “TSMC cell library” were available, we selected the Cadence “TSMC 180 nm cell library” for research purposes and for academia to study the FIR filter design and prove the concept of the memristor base- digital design . The simulation results have indicated that the filter was effectively able to distinguish between all input signals, allowing only the desired signals to pass through the passband region and rejecting any signal beyond the stopband frequency.

References

[1] H. A.D. Nguyen, L. Xie, M. Taouil, S. Hamdioui, and K. Bertels, "Synthesizing HDL to Memristor Technology: A generic framework," *2016 IEEE/ACM International Symposium on Nano scale Architectures (NANOARCH)*, pp. 43–48,July 2016.

[2] S. Shin, K. Kim, and S. Kang, "Memristor applications for programmable analog ICS," *IEEE Transactions on Nanotechnology*, vol. 10, no. 2, pp. 266–274, March 2011.

[3] Y. V. Pershin and M. Di. Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 8, pp. 1857–1864, Aug 2010.

[4] M. Teimoory, A. Amirsoleimani, J. Shamsi, A. Ahmadi, S. Alirezaee, and M. Ahmadi, "Optimized implementation of memristor-based full adder by material implication logic," *in 2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 562–565,Dec 2014.

[5] F. S. Marranghello, V. Callegaro, A. I. Reis, and R. P. Ribas, "Four-level forms for memristive material implication logic," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, pp. 1–5, 2019.

[6] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magic memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov 2014.

[7] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, "MRL memristor ratioed logic," *in 2012 13th International Workshop on Cellular Nanoscale Net-works and their applications*, pp. 1–6 , Aug 2012.

[8] V. Ntinas, I. Vourkas, and G. Sirakoulis, " LC filters with enhanced memristive damping," *in 2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2664–2667, May 2015.

[9] F. M. Bayat, F. Alibart, L. Gao, and D. Strukov, "A reconfigurable fir filter with memristor-based weights," *arXiv preprint arXiv:1608.05445*, 2016.

[10] H. A. D. Nguyen, L. Xie, M. Taouil, S. Hamdioui, and K. Bertels, "Synthesizing HDL to Memristor Technology: A generic framework," *in 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp.43–48,July 2016.

[11] P. Chen and S. Yu, “Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design,” *IEEE Transactions on Electron Devices*, vol. 62, no. 12, pp. 4022–4028, Dec 2015.

[12] R. Gonzalez and R. E. Woods, “Digital image processing, prentice hall, New Jersey”, 2002.

Chapter 3

Hybrid Memristor-CMOS Based Up-Down Counter Design

3.1 Introduction

Further miniaturization in CMOS technology has faced serious challenges such as increase in leakage power and reduction of circuit reliability [1]. This has caused fundamental restrictions on advancement of efficient computing systems architecture [2]. Memristor is a new emerging resistive device [3] that can considerably enhance the performance of future computing systems [4] due to its high-density and on-chip fabrication capability as well as non-volatile memories [5]. Therefore, a rapid increase in the Memristor-based designs using various logic design approaches have been presented in recent years [6], [7–9]. The Memristor Ratioed Logic (MRL) [6] is a hybrid Memristor-CMOS circuit which involves the CMOS circuits to provide the interface and control operation [10]. The logical states of this logic (’0’ and ’1’) are defined based on the output voltage level. Material Implication Logic (IMPLY) [7], [8] and Memristor Aided Logic (MAGIC) [9] are both pure Memristive logic methods aimed for crossbar design. The logical states for both designs are dependent on the Memristance of the output Memristor. That requires more than one voltage level in IMPLY logic and several additional devices for input and output of the MAGIC logic which makes changing the logical states for both designs difficult to implement [11].

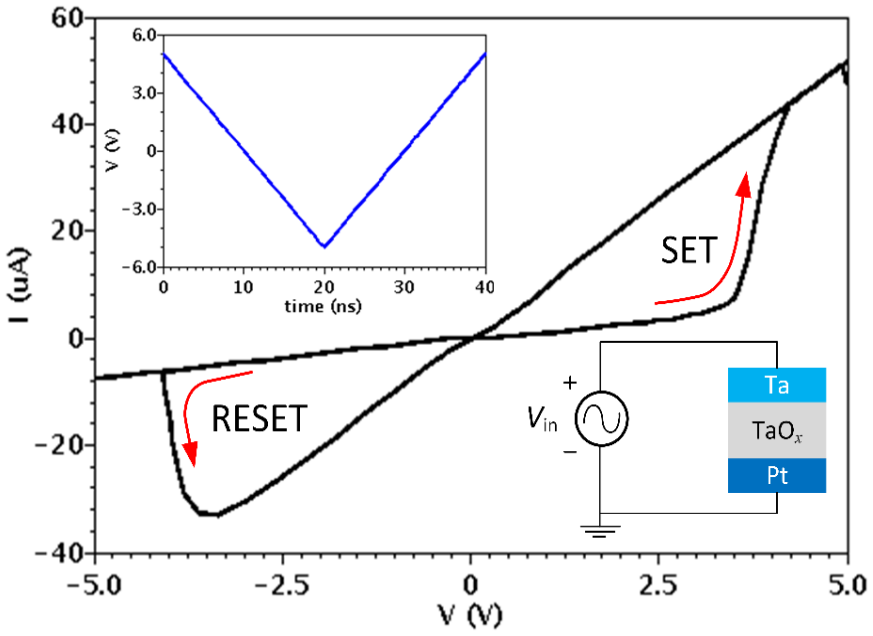


Figure 3.1:The memristive behavior confirm by Pt/TaOx/Ta device I-V curve, when current flows through the device from the bar side, the resistance of the device decreases “SET process”. Current enters from the non-bar side the device resistance increases “RESET process”.

In this chapter, a 4-bit Up-Down counter is designed using MRL method and simulated in Cadence. A comparison between our design and the CMOS-based Up-Down counter shows less occupied area and a considerable improvement in consumed power and delay. The rest of this chapter is structured as follows: Section 3.2. Provides a brief introduction to the Redox-based Memristor device and its modeling. As well as explained the Memristor Ratioed Logic design (MRL) method in detail. Section 3.3. Describes the implementation of the proposed 4-bit Memristive Up-Down counter. In Section 3.4. A simulation results and comparison with CMOS based Up-Down counter is provided. Finally, remarks and conclusion are presented in section 3.5.

Table 3.1: $Pt/TaOx/Ta$ Memristor parameters for simulation

Parameter	L_{disc} **(nm)**	L_{TaOx} **(nm)**	N_{min} **(m^{-2})**	N_{max} **(m^{-2})**	C_{31} **(Am/V)**	**A (nm^2)**
Value	04	11	05	05	$6e^{-12}$	$3.14e^{4}$

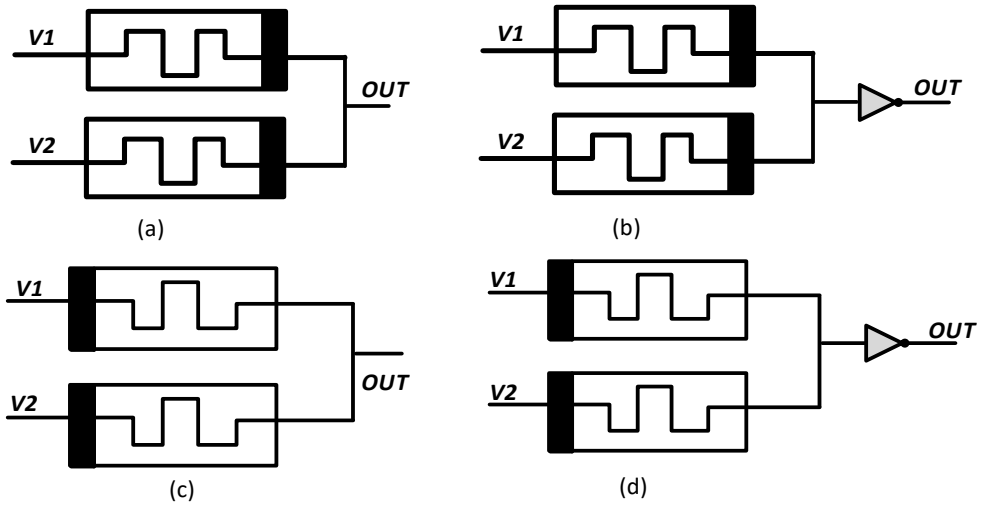


Figure 3.2:The hybrid memristor-CMOS logic gates

(a) AND, (b) NAND, (c) OR and (d) NOR

3.2 Device Modelling and Design Method

3.2.1 Device Modeling

The device Pt/TaOx/Ta is a Redox-based Resistive Random- Access Memory (ReRAM), provides the Memristive behavior for the proposed design based on the oxide materials (TaOx), which exhibits an alteration in the resistance state due to the changes in the oxygen vacancy density as a consequence of oxygen atoms redistribution when the voltage applies across the device. The Verilog-A model for the ReRAM device used in the design is presented in [12]. The parameters shown in Table 3.1 are utilized to simulate the device in Cadence . The I-V curve shown in Fig.3.1 indicates the realistic Memristive behavior of the device whilst the device resistance state is altered between low resistance state (LRS) and high resistance state (HRS) corresponding to the logic states (’0’ and ’1’) respectively.

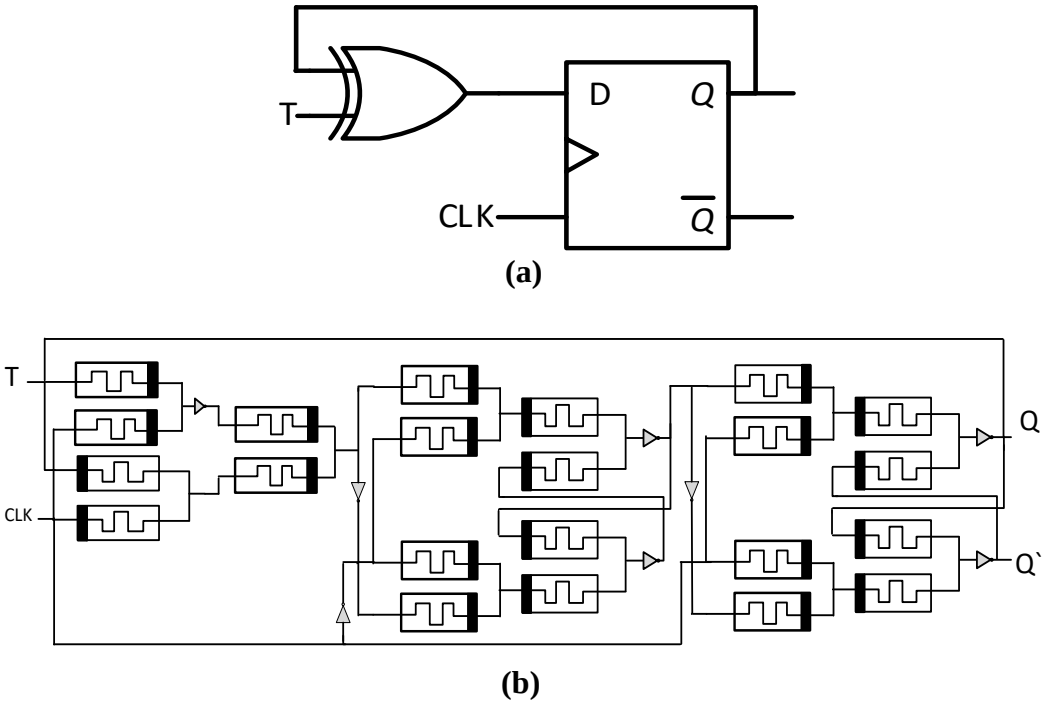


Figure 3.3: The circuit schematic of the memristor based TFF

(a) TFF logic circuit (b) MRL-based TFF circuit

The device resistance is dependent on the density of the oxygen vacancy known also as the device state variable N

$$\frac{dN}{dt} = \frac{1}{[EZ_{vo}AL_{disc}]} . I_{ionic}(t) \qquad (2-1)$$

A is the device cross-sectional area, E is the elementary charge, Z_{VO} is the charge number of the oxygen Vacancies and I_{ionic} is the ionic current.

3.2.2 Memristor Ratioed Logic Method

The MRL logic design is CMOS compatible [13]. Memristor device and CMOS technology can work together side by side to form all logical gates, giving the hybrid design the advantage over the conventional CMOS design to meet ever-growing demands for fast computing and additional storage.

Table 3.2 : Number of devices utilized in different TFF design

TFF Design	TGB [20]	Modified CMOS [20]	GDI [20]	MRL_ based TFF
MOSFET	26	24	22	16
Memristor	-	-	-	22
Delay (Ps)	79.1	60.6	50.2	34.1
Power (µW)	55.3	42.7	17.5	12.3

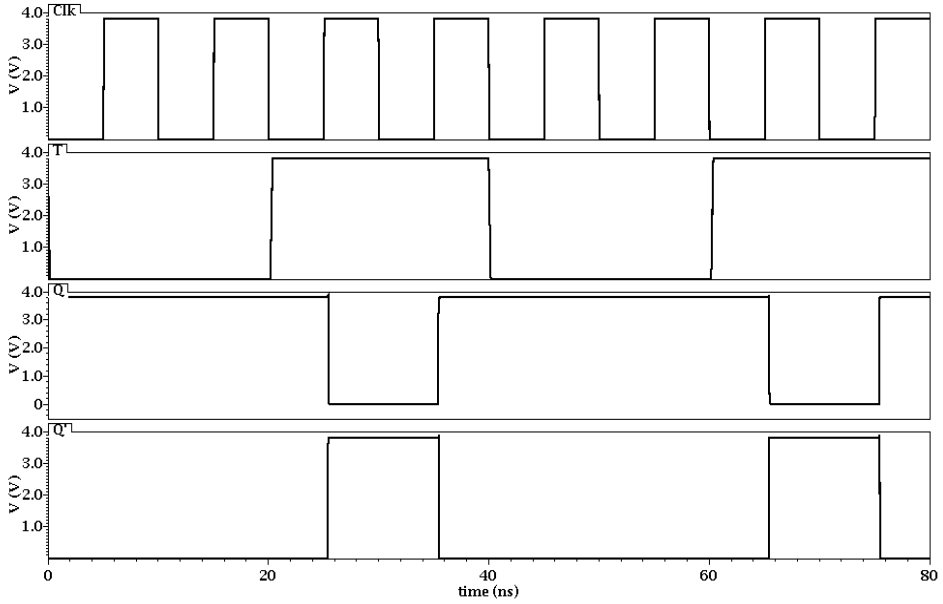


Figure 3.4: The simulations result for the MRL-based TFF

The logical state for the MRL method and CMOS technology are identified in the same way based on the level of the output voltage. Thus, low voltage and high voltage denote the logical states (’0’ and ’1’) respectively. In other words, this method is more appealing to current CMOS technology than other pure Memristive logic such as IMPLY and MAGIC methods, since the logical state in pure Memristive logic design is identified based on the output Memristance. However, the hybrid Memristor-CMOS method (MRL) suffers from signal degradation when implementing the AND and OR gates [14]. Fig.3.2. displays the implementation of AND, NAND, OR and NOR gates using the MRL method.

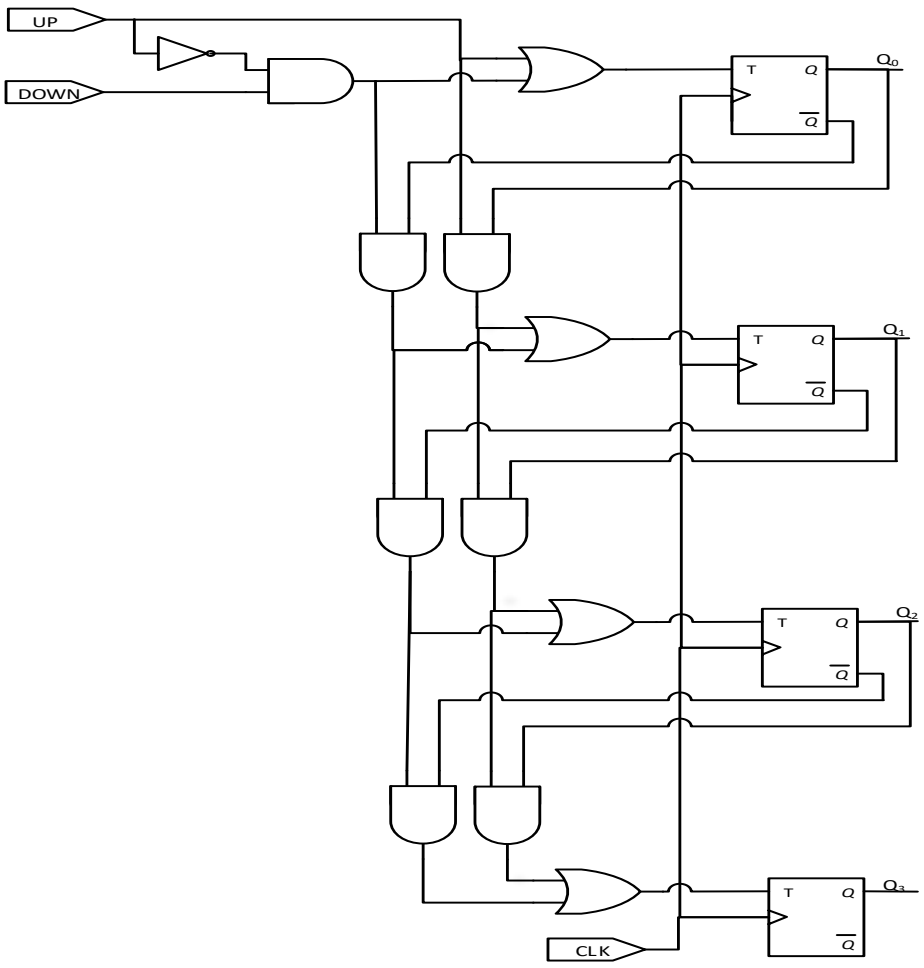


Figure 3.5: Circuit schematic of the implemented 4-bit memristor-based Up/Down Counter

3.3 Memristive Up-Down Counter

Counters are sequential circuits involved in most digital applications from basic digital clocks to various arithmetic operations. Several Up-Down counter circuits based on CMOS technology have been published [15,16,17]. However, fewer designs employ the Memristors devices as reported in [18]. In this chapter, a Memristive 4-bit Up-Down counter is designed based on the hybrid Memristor-CMOS method. The proposed design has been implemented by the use of four T flip-flops (TFF) connected with digital AND, OR gates. T_0 T_1 T_2 T_3 and Q_0 Q_1 Q_2 Q_3 are the input data and states output for all four TFF consecutively. A clock signal CLK is used to trigger all TFF simultaneously. A mode control consists of two input signals UP, DOWN to perform the required

computations for up or down counting. The Memristive TFF circuit employs D flip-flop (DFF) and 2-input XOR gate [19] as shown in Fig .3.3 (a). The DFF is an edge-triggered circuit consisting of two D-latches connected in series with master-slave configuration to prevent any invalid input states possibility. The hybrid Memristor-CMOS logic gates AND, OR and NOR are utilized and connected based on the proposed schematic as shown in Fig .3.3 (b). The DFF involves 16 Memristors and 14 MOSFETs while the XOR gate requires 6 Memristors and 2 MOSFETs. Hence, the TFF consists of 22 memristors and 16 MOSFETs. The validation of the Memristive TFF has been confirmed by the simulation results in Fig.3.4.

3.4 Experimental Results

The design of hybrid Memristor-CMOS based Up-Down Counter has been achieved by implementing the counter shown in Fig 3.5 in the schematic level and verified the design in the behavior level in Cadence. The ReRAM device is used in this design to implement the Memristive behavior of the device based on the parameters presented in Table 3.1. The TFF is an essential element in the design since it assesses the counter in regard to area, delay, and power consumption. Table 3.2 presents the number of devices utilized in different TFF designs. Clearly the Memristive TFF consumes less area in comparison with different CMOS-based designs [20]. The simulation results for the Memristive counter in Fig. 3.6. illustrate that the output states of the four memristive TFF are capable of producing 4-bit sequence of $Q_0Q_1Q_2Q_3$ once TFFs are triggered. This binary sequence repeats itself every 16 clock pulses. Thus, Memristive TFFs are counting in sequence. The MRL based counter is able to make changes to its count direction whether up or down at any point within their counting sequence. This is achieved by using an additional input signal (Control Mode) which specifies the direction of the counting, either Up or Down as follows:

Up= 0, DOWN= 0: No change mode.

UP= 0, DOWN= 1: Down counting mode.

UP= 1, DOWN= 0: Up counting mode.

UP= 1, DOWN= 1: Up counting mode

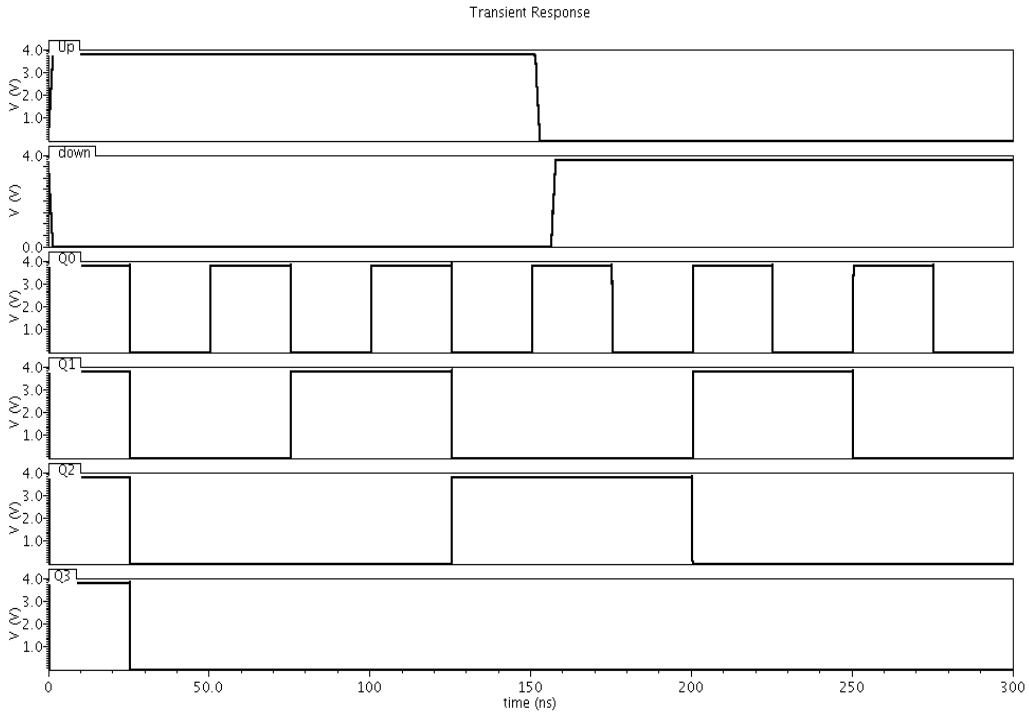


Figure 3.6: The simulations result for the MRL-based Up/Down Counter

The proposed counter considering the area in a number of transistors for 4-bit counter composes of 110 Memristors and 34 MOSFETs, which is less than half of the devices reported in [15,16,17]. The Memristors are loaded on the inverters between the upper layers of CMOS metal. This arrangement makes the hybrid Memristor-CMOS design more compact and occupies less area compared to CMOS based design. While this is not the case when our design is compared to the IMPLY-based Up-Down counter as the area reported in [18] is smaller. However, our design shows a decent speed, and it is considerably faster than IMPLY-based Up-Down counter as a result of the sequential nature associated with IMPLY method. The IMPLY-based Up-Down counter requires 52 computational steps [18]. The experimental results have shown that, the hybrid Memristor- CMOS based Up-Down counter has a delay of 779.8 ps. Moreover, the proposed Memristive counter consumes only 51.1 μw while CMOS-based up-down counter [17] consumed 3.85 mw.

3.5 Conclusion

In this chapter, a 4-bit Memristive Up-Down counter has been implemented using ReRAM devices based on the MRL design. The proposed counter has been verified and tested at different stages of the design in the Cadence environment and NC-Verilog. The simulation results have illustrated that the Up-Down counter was successfully able to start or stop counting at any logical state and resume its operation from any other desired logic state. The design demonstrates that a hybrid Memristor-CMOS based Up-Down counter requires fewer number of transistors than in traditional CMOS based Up-Down counter. Therefore, a less occupied layout area is resulted from the design which is led to a notable reduction in consumed power. Moreover, the design shows a reasonable computation time comparing not only to CMOS based design but also to IMLY based Up-Down counter.

References

[1] H. Anh, Du Nguyen, L. Xie, M. Taouil, S. Hamdioui, and K. Bertels, "Synthesizing HDL to memristor technology: A generic framework," *in 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. IEEE, pp. 43–48, 2016.

[2] S. Kaxiras, "Architecture at the end of Moore" *in Architecture and Design of Molecule Logic Gates and Atom Circuits*. Springer, pp. 1–10, 2013.

[3] L. Wang, C. Yang, J. Wen, S. Gai, and Y. Peng, "Overview of emerging memristor families from resistive memristor to spintronic memristor," *Journal of Materials Science: Materials in Electronics*, vol. 26, no. 7, pp. 4618–4628, 2015.

[4] J. Borghetti, Z. Li, J. Straznicky, X. Li, D. A. Ohlberg, W. Wu, D. R. Stewart, and R. S. Williams, "A hybrid nano memristor/transistor logic circuit capable of self-programming," *Proceedings of the National Academy of Sciences*, vol. 106, no. 6, pp. 1699–1703, 2009.

[5] A. Sasi, A. Amirsoleimani, A. Ahmadi, and M. Ahmadi, "Hybrid memristor-CMOS based linear feedback shift register design," *in 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, pp. 62–65, 2017.

[6] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, "MRL memristor ratioed logic," *in 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1–6, Aug 2012.

[7] M. Teimoory, A. Amirsoleimani, J. Shamsi, A. Ahmadi, S. Alirezaee, and M. Ahmadi, "Optimized implementation of memristor-based full adder by material implication logic," *in 2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, pp. 562–565, 2014.

[8] F. S. Marranghello, V. Callegaro, A. I. Reis, and R. P. Ribas, "Four-level forms for memristive material implication logic," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 27, no. 5, pp. 1228– 1232, 2019.

[9] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magic Memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov 2014.

[10] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nature Electronics*, vol. 1, no. 1, pp. 22–29, 2018.

[11] K. Cho, S. Jin Lee, and K. Eshraghian, " Memristor-CMOS logic and digital computational components," *Microelectronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.

[12] A. Siemon, S. Menzel, A. Marchewka, Y. Nishi, R. Waser, and E. Linn, "Simulation of taox-based complementary resistive switches by a physics-based memristive model," *in 2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1420–1423, June 2014.

[13] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, "Mrl-memristor ratioed logic," in 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications. IEEE, pp. 1–6, 2012.

[14] A. Amirsoleimani, M. Ahmadi, and A. Ahmadi, "Logic design on mirrored memristive crossbars," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 11, pp. 1688–1692, 2017.

[15] M. Alioto, R. Mita, and G. Palumbo, "Design of high-speed power efficient MOS current-mode logic frequency dividers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 11, pp. 1165–1169, 2006.

[16] S. Abdel-Hafeez and A. Gordon-Ross, "A digital CMOS parallel counter architecture based on state look-ahead logic," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 19, no. 6, pp. 1023–1033, 2010.

[17] T. Zhang and Q. Hu, "A high-speed and low-power up/down counter in 0.18-mm CMOS technology," *in 2012 International Conference on Wireless Communications and Signal Processing (WCSP). IEEE*, pp. 1–3, 2012.

[18] A. Chakraborty, A. Dhara, and H. Rahaman, "Design of memristor based up-down counter using material implication logic*," in 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI). IEEE*, pp. 269–274, 2016.

[19] M. M. Mano and M. Ciletti, Digital design: with an introduction to the Verilog HDL. Pearson, 2013.

[20] E. Abiri, M. R. Salehi, and A. Darabi, "Design and simulation of low power and high-speed t-flip flap with the modified gate diffusion input (gdi) technique in nano process," *in 2014 22nd Iranian Conference on Electrical Engineering (ICEE)*. IEEE, pp. 82–87,2014.

Chapter 4

A Memristive Based Design of a Core Digital Circuit for Elliptic Curve Cryptography

4.1 Introduction

Memristor devices have attracted much attention since Leon Chua first introduced them [1]. The new emerging device is two-terminal elements [2] with variable resistance. The resistance can change or save its last state based on the current flowing through the Memristor. This unique behavior makes this device very appealing, not only to memory design [3], but also to many other applications such as analog circuit design [4], signal processing [5], neural networks [6] and logic design [7], alongside all other Memristor-based applications. The Memristor-based logic design is an ambitious step towards future VLSI technology. Hence, there has been intensive research covering the field of logical design involving Memristor devices. Only-Memristor based method is one of the Memristor-based logic designs as the logical states (‘0’ and ‘1’) are defined by the value of the resistance of the output Memristor known as Memristance ‘M’. Memristance values can swing between two resistance states. High Resistance State R_{OFF} and Low Resistance State R_{ON} based on the magnitude, direction, and duration of the applied

voltage. Material Implication Logic (IMPLY) [8] and Memristor Aided Logic (MAGIC) [9] are well-known logics for only Memristor method. Memristor Ratioed Logic (MRL) [10] is another memristor-based logic design method. In this method, both Memristor devices and CMOS transistors can be combined in an integrated platform due to the similarity in their logical state. Although CMOS is a mature technology, the integrated structure of CMOS-Memristors will provide CMOS with an opportunity to address the challenges it faces with respect to any further scaling-down [11]. The MRL logical states (’0’ and ’1’) are defined by the level of the output voltage. In this chapter, we present a more efficient design for XAX- Module [12-14] using the Memristor devices.

The building block XAX- Module or a multiple copy of the module is involved in different multiplier architecture reported in [12-14]. The main improvement in this design is using the MRL- based gates instead of standard CMOS gates since current CMOS technology is reaching its limit in term of transistors size. The organization of this chapter is as follows: Section 4.2. Provides a brief introduction to the $Pt/TaOx/Ta$ Memristor device and their modelling along with explanation on the design of Memristor Ratioed Logic (MRL) method. Section 4.3. Includes information about the implementation of hybrid CMOS- Memristor XAX-Module. In Section 4.4. Simulation results and discussion on MRL-based XAX-Module. Finally, remarks and conclusions are presented in Section 4.5.

4.2 Memristor and Design Method

4.2.1 Memristor Modelling

Resistive random-access memory (ReRAM) is commonly used in a wide range of non-volatile memory applications. In this chapter, the *Pt / TaOx / Ta* device model is used in our simulations where the interface of the top electrode Pt and the oxide materials (TaOx) is considered as a Schottky barrier [15]. The device delivers the Memristive behavior to the proposed XAX- module based on the resistive switching mechanism. The electronic resistance R_{el} of the device changes to reflect the logical state of the Resistive RAM. High Resistance State R_{OFF} represents logic ‘0’ while the Low Resistance State R_{ON} represents logic '1’.

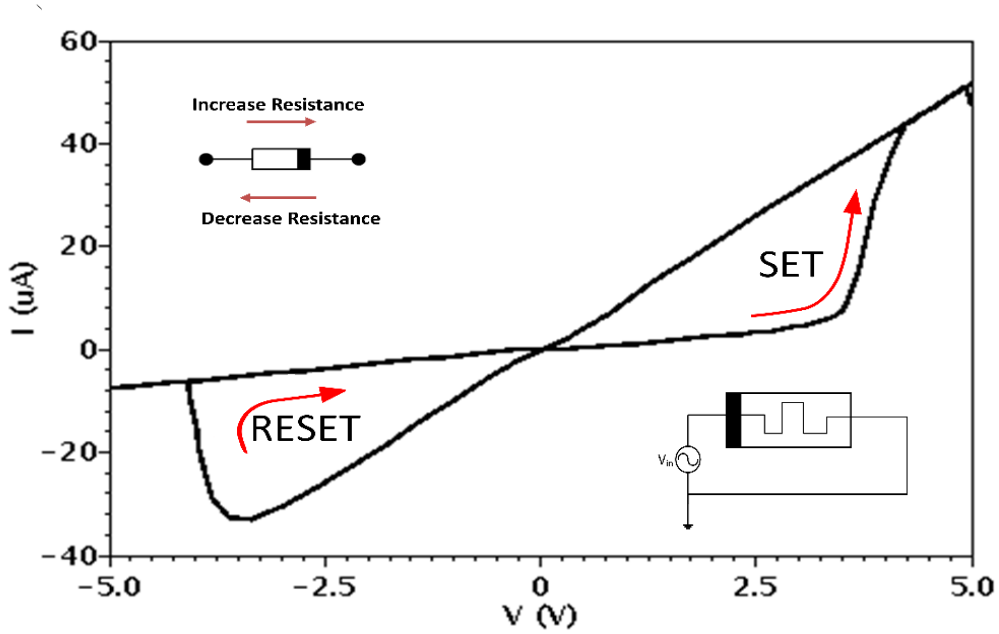


Figure 4.1: Memristor I–V curve for resistive switching memories (ReRAM)

$$R_{el} = \frac{L_{Disc}}{[N.e.\mu_n.A]} \qquad (4.1)$$

where, L_{Disc} is the length of the disc zone in which switching activity takes place, A is the device cross-sectional area, e is the elementary charge, μ_n is the mobility of the electrons and N is oxygen vacancy in the disc zone. The average oxygen vacancy in the disc zone is considered as the device state variable N, which can be mathematically formulated as follows :

$$\frac{dN}{dt} = \frac{1}{[e.Z_{vo}.A.L_{Disc}]}.I_{ion}(t) \qquad (4.2)$$

where, Z_{VO} is the charge number of the oxygen vacancies and I_{ion} is the ionic current which is responsible for the change of the state variable. The average current in the disc, which is sandwiched between the top electrode Pt and the plug zone is determined by equation 4.3 .

Table 4.1: $Pt/TaOx/Ta$ parameters for simulation

Parameter	**L_{disc} (nm)**	**L_{TaOx} (nm)**	**N_{min} (m^{-2})**	**N_{max} (m^{-2})**	**C_{31} (Am/V)**	**A (nm^2)**
Value	04	11	05	05	$6e^{-12}$	$3.14e^{4}$

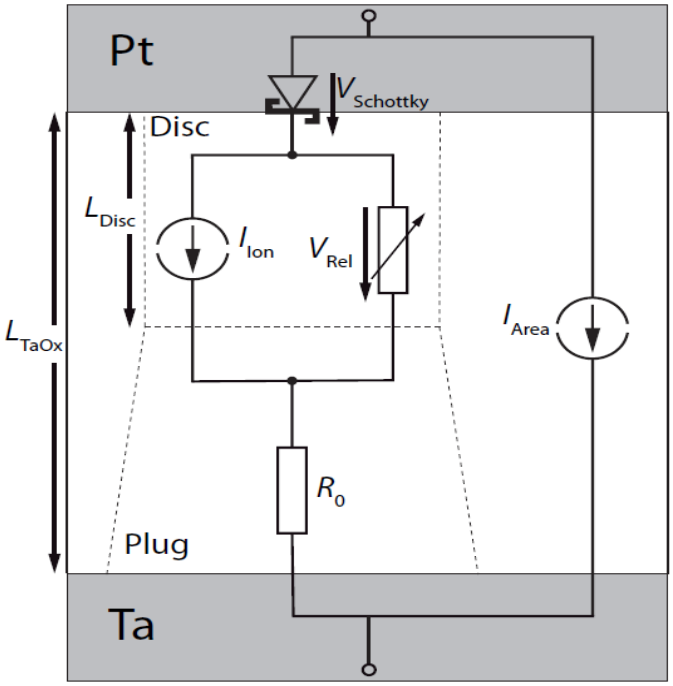


Figure 4.2: The equivalent circuit for $Pt/TaOx/Ta$ device[15]

$$I_{Schottky} = I_{ion} + I_{Rel} \qquad (4.3)$$

Where, I_{Rel} is the current in the electronic resistance of the disc. The total current as can be seen in equation 4.4 is presented by two parts. The Schottky current and the areal leakage current which is the dominant current during the High Resistance State R_{OFF}.

$$I_{tot} = I_{Schottky} + I_{areal} \qquad (4.4)$$

The ReRAM device Verilog-A model used in this design has been described in detail in [15]. The parameters found in Table 4.1 were applied to simulate the device in Cadence Virtuoso.

The I-V curve shown in Fig.4.1 represents the $Pt\ /TaOx\ /Ta$ device's true Memristive behavior, whilst Fig.4.2 [15] depicts the device equivalent circuit.

4.2.2 Memristor Design Method

Integrating memristor devices with conventional CMOS is a very promising method for the implementation of digital architecture. The integrated platform is CMOS-compatible, which provides MRL logic with the capability to construct all logical gates by utilizing Memristors and stacking them between the upper layers of the CMOS inverter [16]. Unlike other Memristive logic designs, such as IMPLY logic [8] and MAGIC logic [9]. Implementing gates in a pure Memristive style can be challenging because of the requirement of more than one voltage level with a read/ write circuit in IMPLY logic and the complication associated with joining over two circuits in MAGIC logic. The integrated platform MRL was exploited to implement the Memristive behavior for the proposed design. The logical state of this method is defined based on the level of the output voltage. Thus, low and high voltages represent the logical states ('0' and '1'), respectively. MRL-based AND gate and OR gate implementation requires only two memristors. V_1, and V_2 denote the voltages applied to both Memristors which change their resistance state based on the voltage level. V_{OUT} denotes the output voltage of the logic circuit which is determined by the voltage divider across both Memristive devices as follow:

In MRL- AND gate when V_1= 0, V_2= 0. V_{OUT}, in this case, is presented as:

$$V_{OUT,AND} = \frac{R_{ON}}{R_{ON} + R_{OFF}} \times V_{CC} \cong 0 \qquad (4.5)$$

While in MRL- OR gate when V_1=1, V_2= 0 .V_{OUT}, in this case, is given by:

$$V_{OUT,OR} = \frac{R_{OFF}}{R_{ON} + R_{OFF}} \times V_{CC} \cong 1 \qquad (4.6)$$

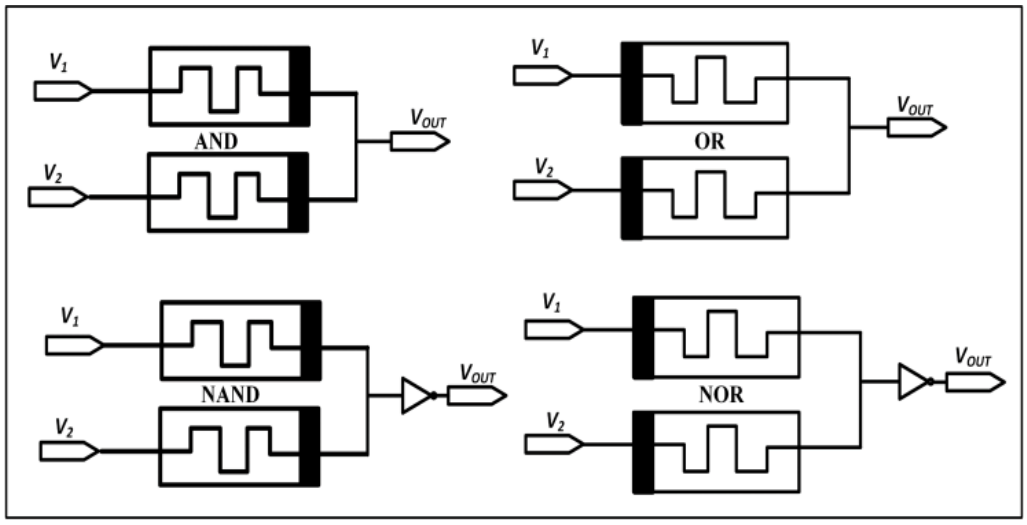


Figure 4.3: The schematic of MRL based basic gates

The schematic circuit for MRL-based basic gates AND, OR, NAND and NOR are displayed in Fig. 4.3. The MRL- based NAND and NOR gates are similar to MRL-based AND and OR gates except each one of them needs a CMOS inverter to facilitate the circuits with the interface and control operation [17] which eliminates the effect of signal degradation associated with MRL-based AND and OR gates.

4.3 Proposed Circuit

Applying different logical gates based on the MRL method shown in Fig. 4.3 to the existing CMOS architecture effectively reduces the number of devices and increases the logic density. In this chapter, a proposal for XAX- Module [12-14] has been presented based on the MRL-based logic gates. The XAX- module shown in Fig.4.4 consists of one AND gate, two XOR gates, and three flip-flops. The MRL-based AND requires two Memristors. The Memristors are combined in series and offered the appropriate polarities. Whereas, the MRL-based XOR gate requires six Memristors and two MOSFETs.

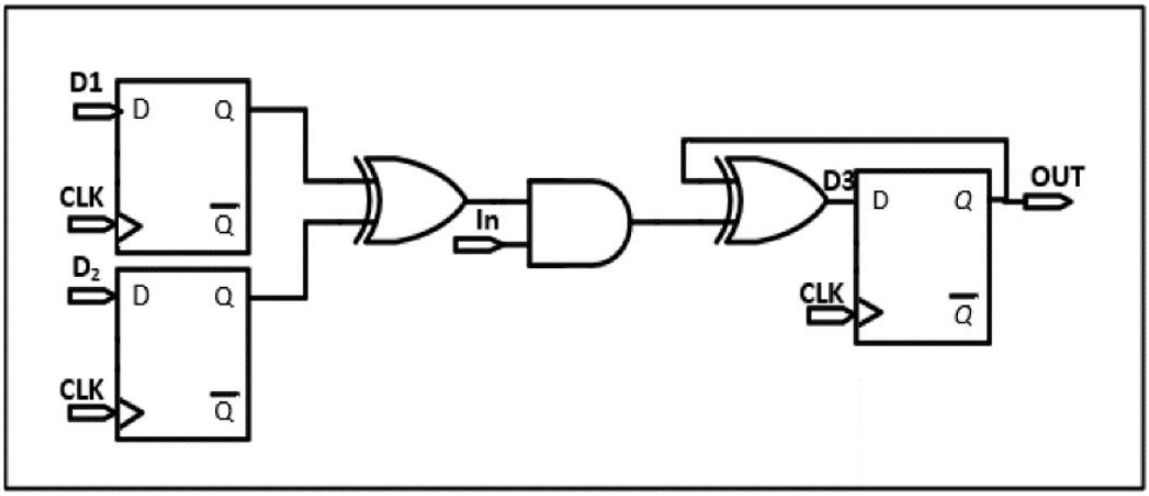


Figure 4.4: The XAX- module schematic

In addition, the D flip-flop (DFF) circuit used in the module comprises of two D- latch circuits serially interconnected. A clock signal CLK is connected directly to the first latch and to the second latch via inverter. This structure is known as master-slave. Fig 4.5 shows the gate level schematic of D-latch. The MRL based- D-latch is implemented using two MRL based- AND gates and two MRL based- OR gates. The D-latch based on the MRL method is depicted in Fig 4.6. uses 8 Memristor devices and 6 MOSFETs making the total number of devices involved in the MRL-based DFF, 16 Memristor devices and 14 MOSFETs.

4.4 Results and Comparison

The proposed XAX-module has been implemented based on the hybrid CMOS-Memristor method. The Cadence Spectre circuit-level simulation has been used to evaluate the proposed XAX-module. The parameters presented in Table 4.1. are considered for the *Pt / TaOx / Ta* device to provide the design with the Memristive behavior. The design of MRL- based XAX-module is made up of simple blocks of the DFF circuit together with the XOR-AND-XOR function based on the schematic shown in Fig. 4.4.

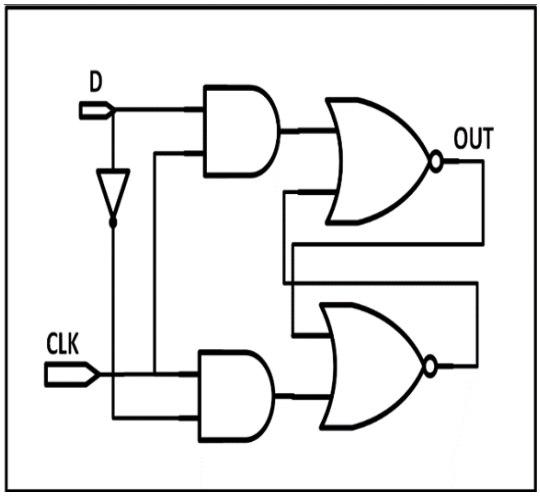


Figure 4.5: The gate level schematic of D latch

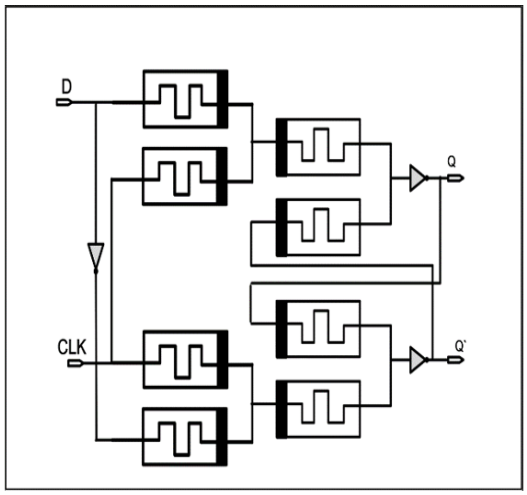


Figure 4.6: The MRL-based D-latch schematic

The MRL- based DFF is the main component in the module in terms of devices number. The MRL- based DFF has been realized based on the schematic diagram shown in Fig. 4.7. The MRL- based DFF is presented in Fig 4.8. utilizes 16 Memristor devices and 14 MOSFETs. Because of this, the MRL- based DFF consumes less area as compared to different DFF designs reported in [14] and [18]. The validation of the MRL based- D-latch and DFF has been confirmed by the simulation results in Fig 4.9. The XOR-AND-XOR function is another block that participates in the XAX-module. The XOR-AND-XOR function involves two MRL-based -XOR gates combined with one MRL-based AND gate. MRL- based AND gate requires only two Memristor devices while MRL-based XOR gate requires six Memristor devices and two MOSFETs. The MRL-based XOR gate used in this proposed design is utilizing a fewer number of MOSFET devices compared to other XOR designs reported in [14] and [19]. Therefore, the total number of devices in the XOR-AND-XOR function is 14 Memristor devices and 4 MOSFETs. Whereas the same CMOS-based design employs 24 MOSFET devices.

In this chapter, it is proven that utilizing MRL- based gates in the blocks that made up the XAX structure implies a significant reduction in the number of transistors. The Memristors can be fabricated on the upper layers of CMOS inverter as explained in [20-21].

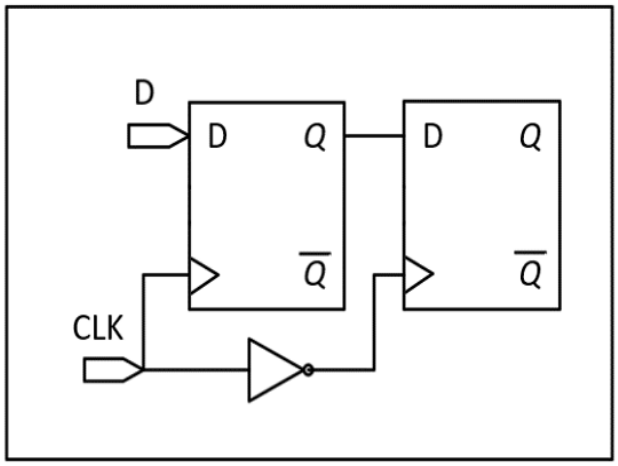


Figure 4.7: The structure of master-slave for DFF

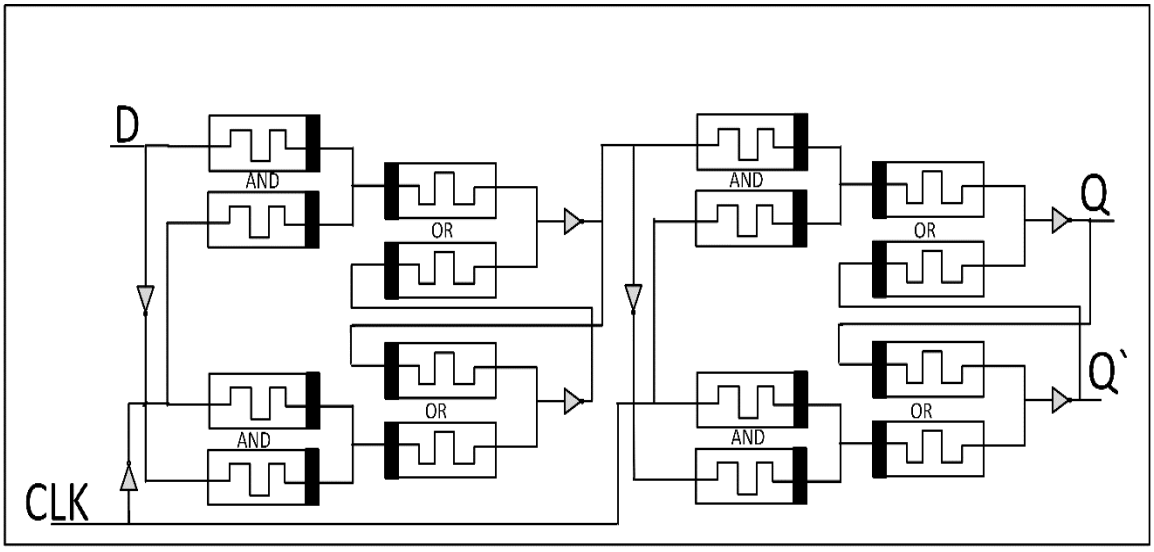
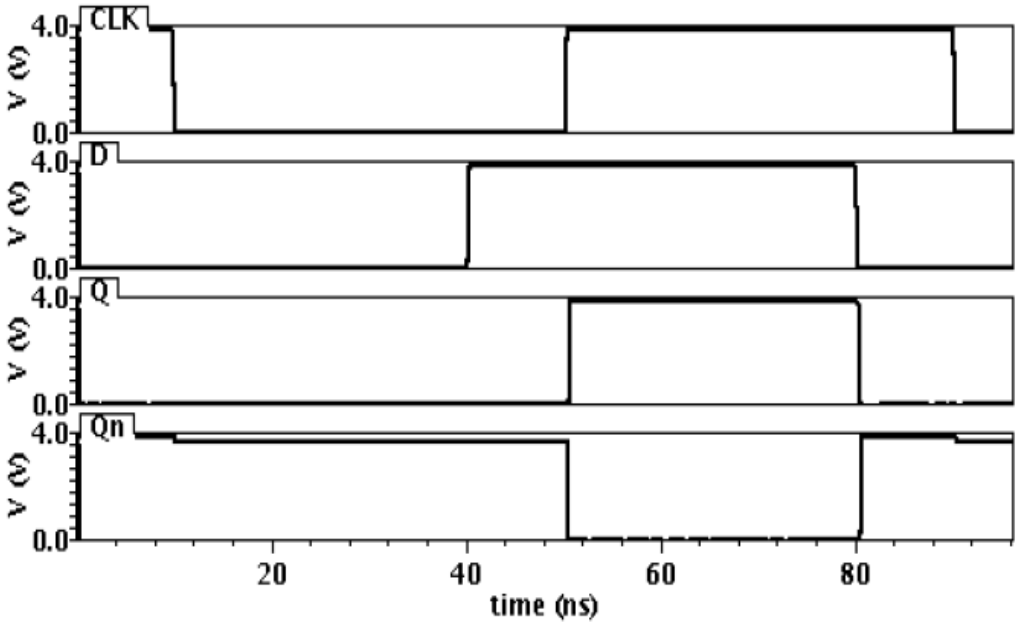
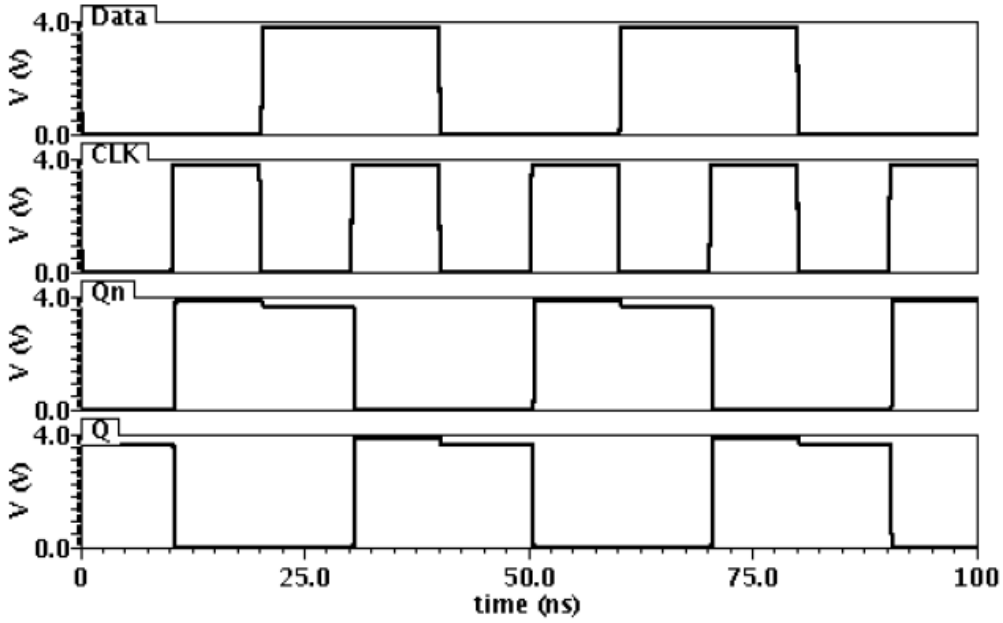


Figure 4.8: The MRL- based DFF schematic

This combination has led to almost 15% in area saving compared to the same CMOS-based-XAX module presented in [12-13]. This is mainly because of the layout area estimate of the design was carried out based on the size of CMOS inverter occupied each cell. From the performance point of view, the designed power consumption and delay were also examined and compared against other CMOS-based designs of the circuit.



(a)



(b)

Figure 4.9: The simulation results in schematic level

(a) MRL-based D- Latch (b) MRL - based DFF

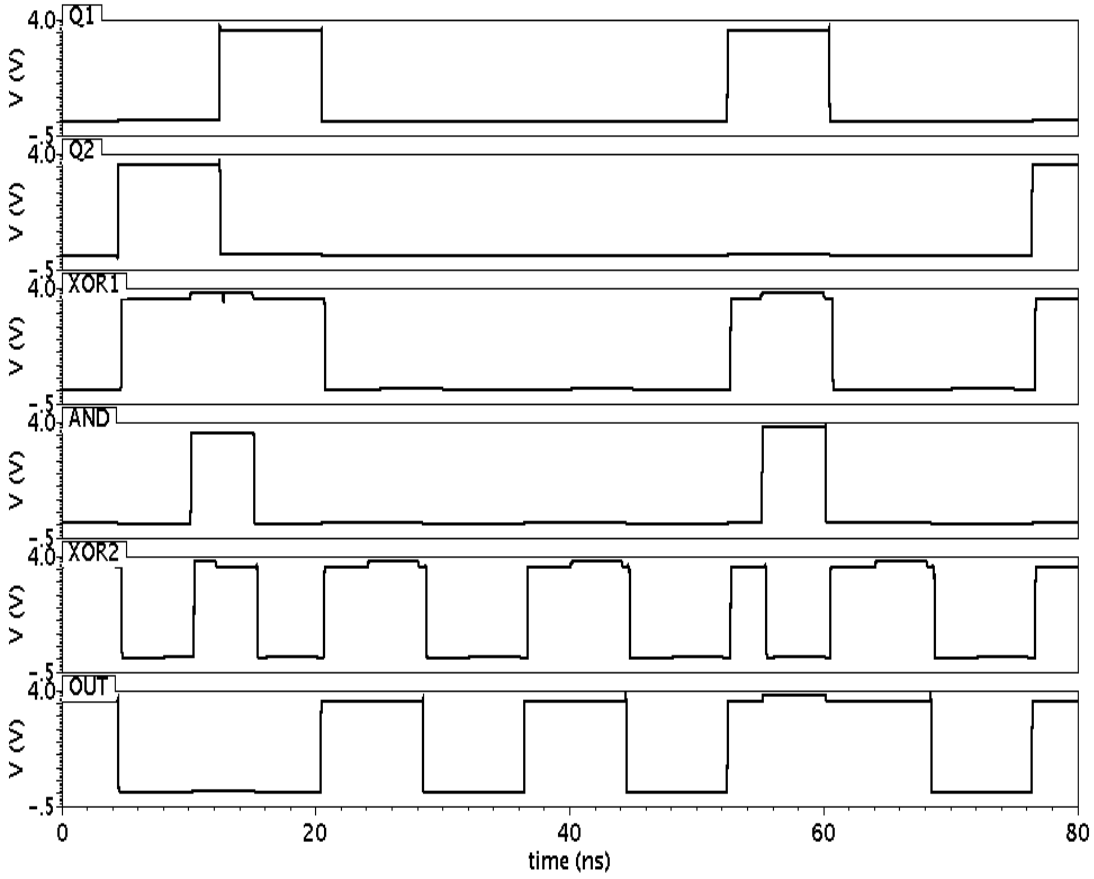


Figure 4.10: The behavior of the proposed CMOS-Memristor XAX module

The consumed power in the design was calculated by connecting two zero-DC supplies, one to V_{dd} and the other to the ground. The current flowing in this branch represents the amount of energy consumed by the design due to the logic transition inside the design. The current is measured and integrated over every alteration time by Cadence Virtuoso calculator. Equation 4.7 is used to calculate the design dynamic power consumption.

$$P_{Dyn} = \alpha C f V_{dd}^2 \qquad (4.7)$$

Here, α is the switching activity factor, C is the capacitance value, f is the working frequency and V_{dd} is the source voltage.

Table 4.2: A comparison between numbers of different CMOS based XAX-Module and proposed design

Design	XAX-module [12]	XAX-module [13]	MRL_ based XAX
CMOS	27	27	23
Memristor	-	-	62
Delay (Ps)	57	83	54
Power (μW)	49.45	41.00	40.32

In addition, the measurement of the propagation delay relies on the transition time at each cell in the design. Therefore, the delay was calculated by measuring the time interval between the input slew and output slew. The slew is defined as the time when the signal rises from 30% to 70% and falls from 70% to 30% of its V_{dd}. Table 4.2 presents a comparison between the proposed design and several different CMOS-based XAX-modules. The proposed design has shown that it performs much faster and consumes less power because of the fewer numbers of deployed transistors. The validation of the module has been confirmed by the simulation results in Fig.4.10.

4.5 Conclusions

The XAX-module using logic-based design suffers from power and area overheads due to the number of involved MOSFET devices. One way used to reduce these effects is by considering MRL-based gate in the design. The MRL design indicates that it has the potential to attract modern CMOS design and could be a practical solution to ever-growing concerns regarding CMOS transistor downsize. In this chapter, the building blocks of XAX- module have been implemented based on the hybrid CMOS-Memristor method. The Memristor device of *Pt / TaOx / Ta* has been used to provide the design with the Memristive behavior.

The proposed MRL- based XAX-module employs fewer numbers of transistors. Thus, the design has a small layout area and consumes considerable low power which makes this design more efficient compared to other CMOS designs. The design was implemented and verified in Cadence Virtuoso at each stage to confirm its functionality.

References

[1] L. O. Chua, “Memristor—The missing circuit element,” *IEEE Trans. Circuit Theory*, vol. CT-18, no.5, pp. 507–519, Sep. 1971.

[2] L. Guckert and E. E. Swartzlander, "Optimized Memristor-Based Multipliers," *in IEEE Transactions on Circuits and Systems I*: Regular Papers, vol.64, no.2, pp. 373-385, Feb. 2017.

[3] L. Wang, C. Yang, J. Wen, S. Gai, and Y. Peng, “Overview of emerging memristor families from resistive memristor to spintronic memristor,” *Journal of Materials Science: Materials in Electronics*, vol.26, no.7, pp.4618–4628, 2015.

[4] S. Shin, K. Kim, and S. Kang, “Memristor applications for programmable analog ics,” *IEEE Transactions on Nanotechnology*, vol. 10, no. 2, pp. 266–274, March 2011.

[5] K. Alammari, A. Sasi, M. Ahmadi, A. Ahmadi, M. Saif “Hybrid Memristor-CMOS Based FIR Filter Design”. In: Stavrinides S., Ozer M. (eds) *Chaos and Complex Systems. Springer Proceedings in Complexity. Springer*, Cham, 2020.

[6] W. Zhang, C. Li, T. Huang, and X. He, “Synchronization of memristor based coupling recurrent neural networks with time-varying delays and impulses,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3308–3313, Dec 2015.

[7] L. Guckert and E. E. Swartzlander, “MAD Gates Memristor Logic Design Using Driver Circuitry," *in IEEE Transactions on Circuits and Systems II*: Express Briefs, vol. 64, no. 2, pp. 171-175, Feb. 2017.

[8] F. S. Marranghello, V. Callegaro, A. I. Reis, and R. P. Ribas, “Four-level forms for memristive material implication logic,” *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 27, no. 5, pp. 1228– 1232, 2019.

[9] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Magic-memristor-aided logic,” *IEEE Transactions on Circuits and Systems II*: Express Briefs, vol. 61, no. 11, pp. 895–899, Nov 2014.

[10] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C.Weiser and E. G. Friedman, "MRL -Memristor Ratioed Logic," *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications,* Turin, Italy, pp. 1-6, 2012.

[11] S. E. Thompson and S. Parthasarathy, “Moore's Law: The Future of Si microelectronics,” *Materials Today*, Vol. 9, No. 6, pp. 20-25, June 2006.

[12] P. H. Namin, R. Muscedere and M. Ahmadi, "Low power design of a word-level finite field multiplier using Reordered Normal Basis," *49th Asilomar Conference on Signals, Systems and Computers*, pp. 437-440, 2015.

[13] P. H. Namin, C. Roma, R. Muscedere and M. Ahmadi, "Efficient VLSI Implementation of a Sequential Finite Field Multiplier Using Reordered Normal Basis in Domino Logic," *in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems,* vol. 26, no. 11, pp. 2542-2552, Nov. 2018.

[14] K. Leboeuf, A. H. Namin, Wu. Huapeng, R. Muscedere and M. Ahmadi, "Efficient VLSI implementation of a finite field multiplier using reordered normal basis," *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*, pp.1218-1221,2010.

[15] A. Siemon, S. Menzel, A. Marchewka, Y. Nishi, R. Waser, E. Linn, "Simulation of TaOx-based Complementary Resistive Switches by a Physics-based Memristive Model", *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1420-1423, 2014.

[16] K. Cho, S. Jin Lee, and K. Eshraghian, “Memristor-CMOS logic and digital computational components,” *Microelectronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.

[17] M. A. Zidan, J. P. Strachan, and W. D. Lu, “ The future of electronics based on memristive systems,” Nature Electronics, vol. 1, no. 1, pp. 22–29, 2018.

[18] A. Sasi, A. Amirsoleimani, A. Ahmadi, and M. Ahmadi, “Hybrid memristor-CMOS based linear feedback shift register design,” *in 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, pp. 62–65. 2017.

[19] Tejinder Singh, “Hybrid Memristor-CMOS (MeMOS) based Logic Gates and Adder Circuits,” *CoRR, arXiv:1506.06735* [cs.ET], pp. 1-11, June 2015.

[20] J. Cong and Bingjun Xiao, "mrFPGA: A novel FPGA architecture with memristor-based reconfiguration," *IEEE/ACM International Symposium on Nanoscale Architectures*, pp. 1-8, 2011.

[21] R. Huang. L. Zhang. D. Gao, Y. Pan, S. Qin, P. Tang, Y., Cai, and Y. Wang, Y, "Resistive switching of silicon-rich-oxide featuring high compatibility with CMOS technology for 3D stackable and embedded applications". *Applied Physics* A, 102, pp.927-931. 2011.

Chapter 5

Memristive-Based Full Adder

5.1 Introduction

Since HP Labs announced the fabrication of the first working memristor in 2008 [1], researchers and scientists have primarily investigated the use of memristor devices. The remarkable properties of memristors are their small size, non-volatile, low power consumption and compatibility with CMOS devices making them a practical alternative to CMOS in wide variety of applications. Earlier memristor studies to capture the interest of researchers were the resistive memory switching, exploiting the unique property of the memristor's ability to retrieve the last state concerning the resistance value. However, later memristor works have shown that the emerging devices are not only for memory design but also can perform conventional logical functions in the style of Material Implication (IMPLY) [2], Memristor Aided Logic (MAGIC) [3] and Memristor Ratioed Logic (MRL) [4], which paves the road for a broader range of various memristor-based circuits implementation, covering a large number and wide scope of conventional arithmetic and computational building blocks such as multiplier [5-6], counter [7-8], and

linear feedback shift register [9] . This chapter presents memristive designs for a full adder, utilizing the based logic designs of pure memristor and MRL-based logic designs. In the pure memristive style, the MAGIC logic is favorited over IMPLY logic in the implementation of the full adder due to the reduced layout area offered by the MAGIC approach thanks to the simplicity of mapping MAGIC gates to the crossbar array, whereas mapping IMPLY gates requires a resistor *R* before any row [10] which increases the design area. In addition, IMPLY requires more voltage sources than MAGIC to perform the logic computations. Alternatively, the MRL- based full adder was implemented based on the Memristor-CMOS Platform. The design does not require additional voltage supplies and can successfully integrate with other existing CMOS designs at a low implementation cost. The reset of the chapter is organized as follows: Section 5.2 provides an overview on the memristor devices and methods of design including pure memristor based logic and Memristor Ratioed Logic (MRL) method. Section 5.3 presents the proposed full adder circuit implementation in both styles of MAGIC and MRL. In Section 5.4, simulation results and discussion revealed. Finally, the paper is concluded in section 5.5.

5.2 Memristor and Design Method

The term memristor appeared for the first time in 1971[11] to realize the missing relationship between the charge (q) and the flux (φ) through what Chua called a memristance (M) which denotes as.

$$d\varphi = M.dq \qquad (5.1)$$

Changing the flux or the charge will lead to a change in the resistance. The memristor can thus be identified as a two-terminal passive device with variable resistance. The resistance value will remain constant when no electrical field is applied, giving this device a unique behavior by saving its last state.

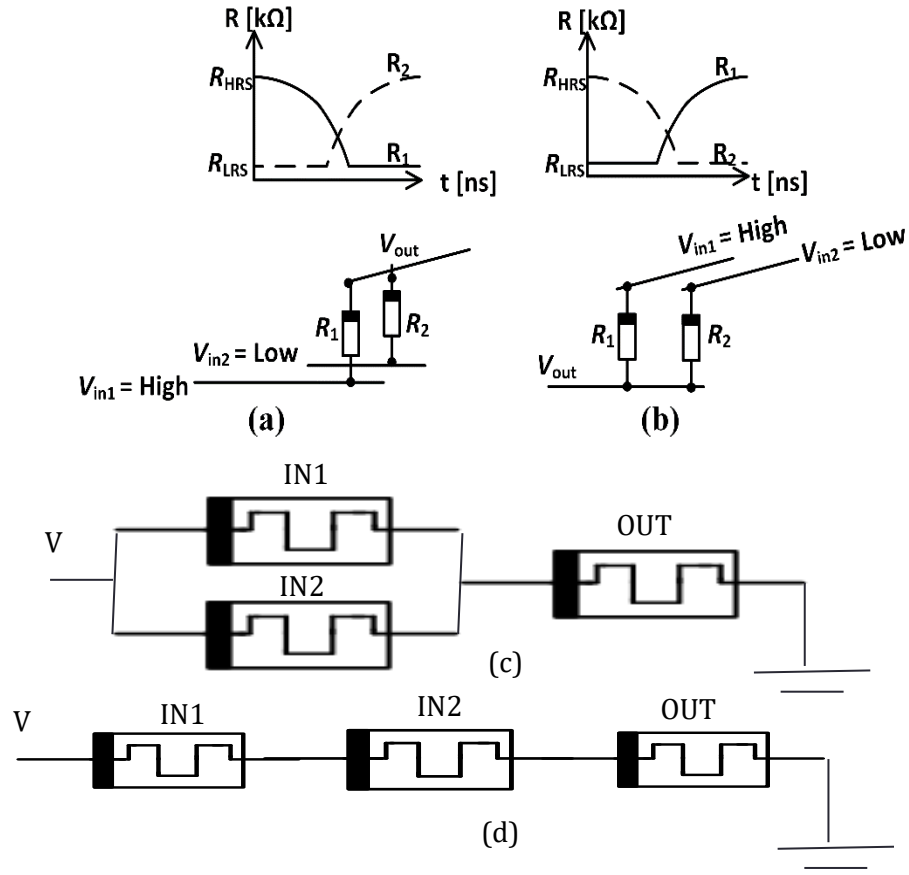


Figure 5.1: MRL- based logic gates (a) AND, (b) OR
MAGIC- based logic gates (C) OR, (d) AND

5.2.1 Pure Memristor Based Logic

Crossbar structure attracts pure memristor-based logic since data can be processed aside from any memory device that can be used to hold or send data. IMPLY and MAGIC are the most well-known logics for pure memristive design. The logical state in the pure memristive environment is evaluated based on the memristance of the memristor. Therefore, logic '1' is defined by the Low Resistance State (LRS) R_{ON}, while logic '0' is determined by the High Resistance State (HRS) R_{OFF}. Both IMPLY and MAGIC can construct the entire Boolean traditional gates such as AND, NAND, OR, and NOR.

However, the logical structure of MAGIC requires two memristors for the inputs and another memristor for the output, as shown in Fig. 5.1 (c)(d).

In contrast, the IMPLY-based logic structure has two memristors connected to the ground through a resistor. The output is copied to one of the memristors in the final computation step, exploiting the feature of the non-volatile offered by the memristor to use the memristor as input or output as required. In addition, both logics require different voltage levels to obtain the desired output logic.

5.2.2 Memristor Ratioed Logic Design

The MRL is another memristive logic design method that combines CMOS transistors and memristor devices in one CMOS-Memristor integrated platform to obtain various logic gates. The integrated platform has a high circuit density since the memristive gats can be fabricated on the inverter's top metal layers [12]. MRL-based gates require only two memristors connected in series to implement AND/OR gates as shown in Fig.5.1(a)(b). Similarly, the inverting gates NAND/NOR implementation can be structured with two memristors connected to an additional CMOS inverter, as memristor devices do not support NOT function [13]. The output voltage level of the integrated platform denotes the logical state as follows: low voltage and high voltage represent logical states ('0' and '1') respectively.

5.3 Full Adder Designs

5.3.1 Pure Memristor Based design

This section proposes a MAGIC design style for the full adder. The idea is to map the MAGIC-based gates into a crossbar structure since MAGIC-based gates employ only memristors, leading to a compact design. The full adder is mapped into a crossbar structure of seven rows and five columns, depicted in Fig. 5.2. The memristors are connected between every row and column to represent the full adder Boolean expression. The working principle of this design can be described as follows:

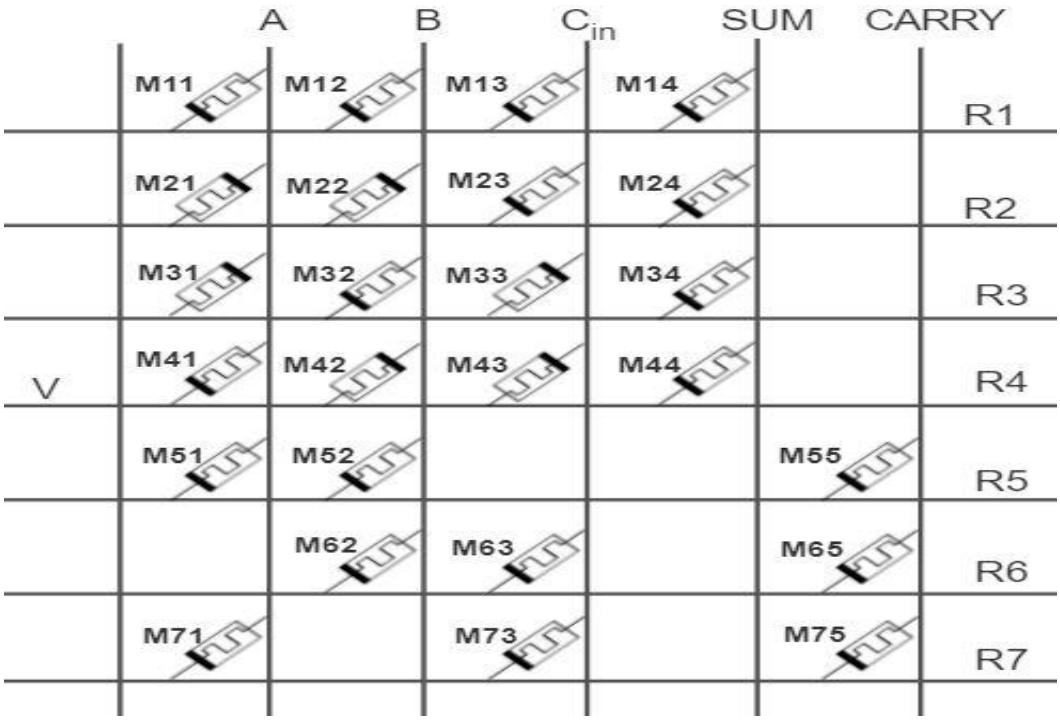


Figure 5.2: Schematic of proposed MAGIC- based Full adder

1. Each row represents a logic gate with input data stored in two or three memristors while output data is stored in one memristor in the form of resistance.
2. The memristor serves here as a switch that allows the row and column to be electrically connected, when the memristor is at low state resistance LRS.
3. To control the computation operations in the design, more than one voltage supplies V_{set}, V_{reset}, V_{read} and GND are required.
4. The logic gates in this architecture perform in parallel rows leading to a reduction in the design Computation time.

The design procedure consists of two sequential steps for the eight cases as follows: The first step is the initialization step, by writing a known logic state to all memristors. Hence, all output memristors in columns SUM and CARRY are set to HRS, while input memristors with the same polarity as the output memristors are set to HRS, and the rest of M21, M22, M31, M33, M42, and M43 are set to LRS, respectively. The second step is the computation step, is achieved as follows:

1. V_{set} is applied to all rows R1-R7 and both output columns SUM and CARRY to keep all output memristors at their HRS to compute the logical states of the input memristors based on the input data to the full adder for every case. In this case (A=1, B=1, C_{in}=0). All input memristors kept their current state except the memristors in columns C_{in}. M13 and M23 change to LRS, M33 and M43 change to HRS.
2. Prepare the output memristors for the reading process by connecting A, B, and C_{in} to GND and apply ½ V_{set} , -½ V_{set} to V, and output columns SUM and CARRY, respectively. All output memristors maintain their HRS because the volage across all of them is only ½ V_{set} while M55 changes to LRS due to the voltage difference is equal to V_{set}.
3. The reading process is accomplished by connecting A, B, C_{in} to GND and applying V_{read}, a voltage value below the memristor's threshold voltage V_{th}. Therefore, (SUM = 0) since all voltage on rows R1-R4 are equal to 0 V, while (CARRY=1) is because the voltage on R5 is equal to V_{read}.Hence, V_{CARRY} in this case can be calculated as:

$$v_{CARRY} = v_{read} \frac{RM65//RM75}{RM55 + (RM65//RM75)} \cong 1 \qquad (5.2)$$

This satisfies the logical operation accomplished by the MAGIC based- full adder for the case (1,1,0). The same concept is applied to all other seven cases.

5.3.2 CMOS-Memristor platform Based design

This section implements an MRL-based full adder based on integrating CMOS and memristors in one platform. The design employs two half-adder circuits connected through two memristors with the appropriate polarities to represent the OR gate. The half-adder circuit is made by adding the AND gate to the output of the XOR gate. All gates involved in the design are MRL based.

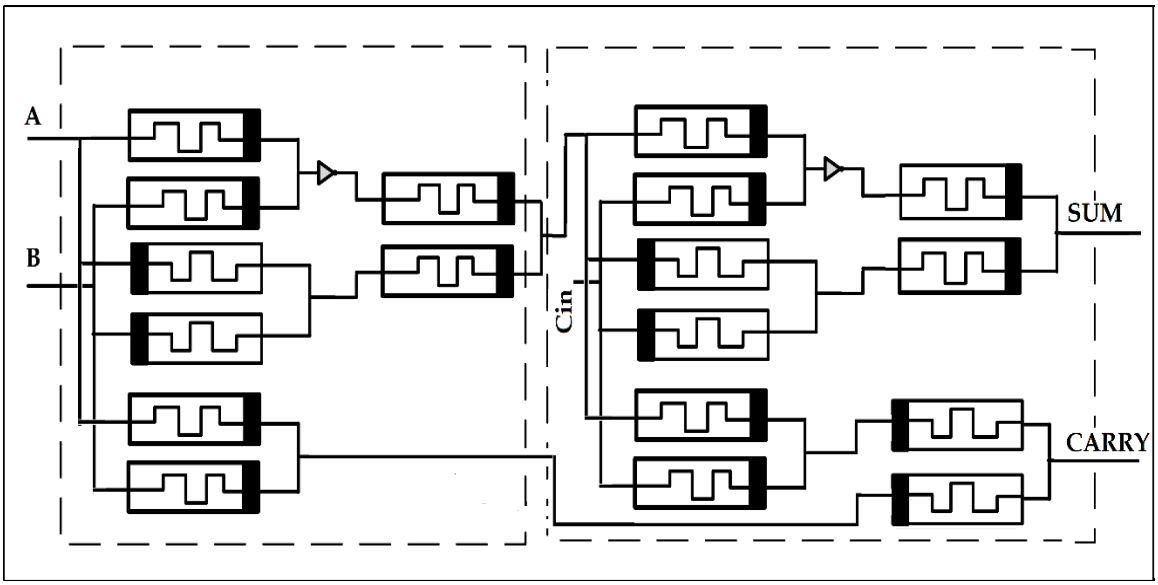


Figure 5.3: Schematic of proposed MRL- based Full adder

Hence, the memristive half adder has 2 MOSFETs and 8 memristors, while the MRL-based full adder includes 4 MOSFETs and 18 memristors. The schematic circuit for an MRL-based basic full adder is displayed in Fig. 5.3.

5.4 Results and Comparison

Implementation of the proposed full adder has been evaluated in the Cadence Spectrum Environment; Two memristor devices were used to provide the design with memristive behavior. The Voltage Threshold Adaptive Memristor (VTEAM) model [14] is chosen for the simulation of the MAGIC-based full adder due to VTEMA threshold voltage's property, which serves the MAGIC-based gates demand for different voltage levels, and the Pt / TaOx / Ta [15] is a stable model preferred for the integration of Memristor-CMOS in the MRL-based full adder.

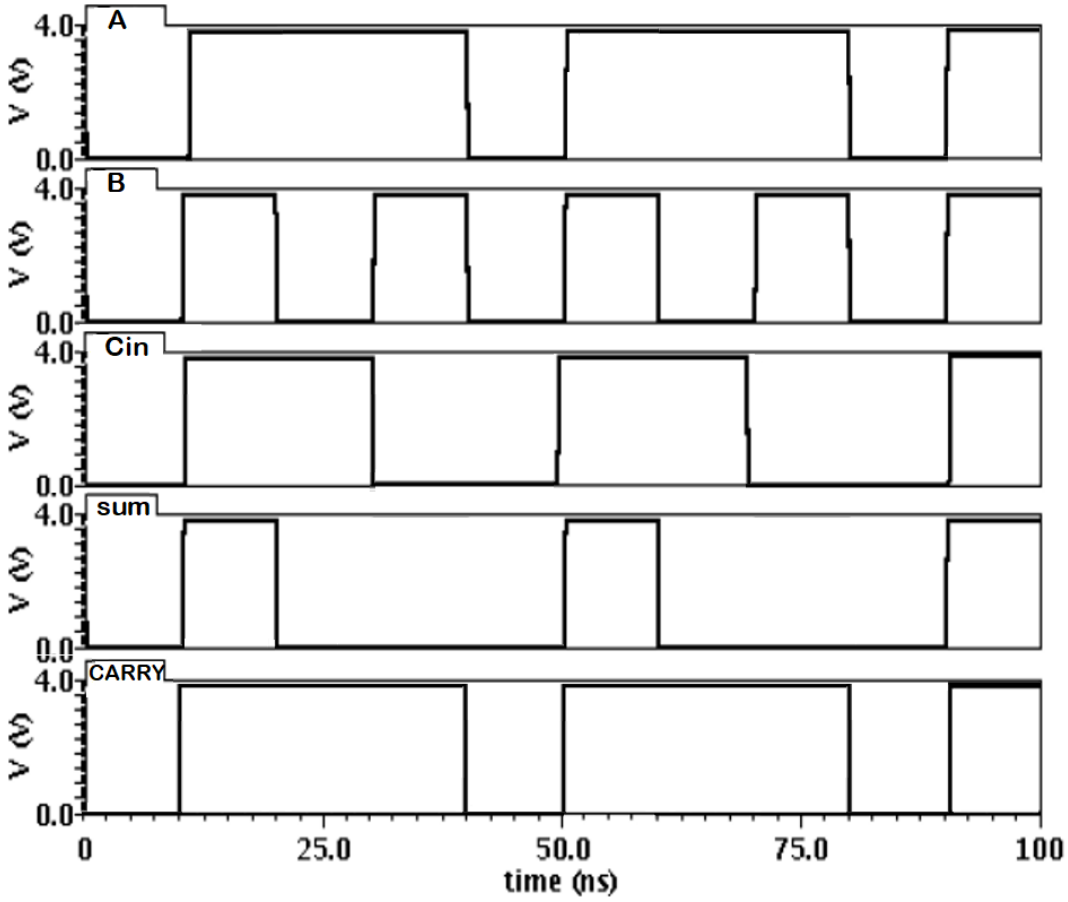


Figure 5.4: The behavior of the proposed MRL- based Full adder

In the pure memristive style, the full adder was accomplished by mapping four gates of 3-input MAGIC-based AND, three gates of 2-input MAGIC-based AND, and five gates of MAGIC-based OR in crossbar structure to realize the output results by the memristors in the columns of SUM and CARRY. The memristors involved in this proposal are 25 memristors, making this structure a very compact design. Moreover, the design achieved a decent speed compared to the other MAGIC-based full adder presented in [16] since the proposal requires only 4 computational steps. In contrast, the full adder presented in [17] requires 10 computational steps.

Table 5.1. A comparison between a number of
different full adders in the pure memristive style and proposed design

Based-Adder Design		MAGIC [16]	RSI IMPLY [17]	CRS IMPLY [18]	Proposed
Devices	MOSFET	-	-	-	-
	Memristor	28	9	16	25
Voltage Supplies		3	3	6	4
Added Hardware		-	1R	1R	-
computational steps		16	10	14	4

A comparative analysis of different full adders in the pure memristive style [16-18] is presented in Table 5.1. While, the design of the full adder in the integrated platform of CMOS-Memristor includes two half adder circuits connected with an OR gate to compute the CARRY. The half-adder circuit involves an MRL-based XOR gate and an MRL-based AND gate. The MRL- based AND requires only 2 memristors, while the MRL-based XOR requires 6 memristors and 2 MOSFETs. Hence, the device's total number in the proposal is 18 memristors and 4 MOSFETs. This leads to an area reduction of approximately 50% compared to the design in [19] and 25% compared to the method presented in [20] due to the fewer MOSFET devices involved in the half-adder's main element, the MRL-based XOR gate. Also, the MRL- based full adder is considerably faster than the same design in the pure memristive style since it requires only one computational step. The simulation results in Fig. 5.4 have confirmed the validation of the full adder.

5.5 Conclusion

This paper presents a memristive based-full adder circuit. The adder is implemented on the memristors-based logic design utilizing only 25 memristors in the MAGIC style and 18 memristors, and 4 MOSFETs in the MRL style.

Consequently, both designs occupy a small area as compared to the CMOS- based design. However, the MAGIC-based full adder is more area efficient than the MRL-based full adder. While the MRL- based full adder is a considerably low-cost design since it is CMOS compatible and requires only one voltage level. Moreover, the proposed MAGIC-based full adder has fewer computational steps in comparison to the same IMPLY-based design, another pure memristor method. The Memristor device of VTEMA and Pt / TaOx / Ta have been successfully used to provide the design with the Memristive behavior while simulations confirm the proposed full adder functionality.

References

[1] D.B.Strukov, G. S. Snider, D. R. Steward, and R. S. Williams. The missing memristor found. *Nature*, 453(7191):80–83, May 2008.

[2] 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," *in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2054-2066, Oct. 2014

[3] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G.Friedman, A.Kolodny, and U.C. Weiser, "Magic-memristor aided logic," *IEEE Transactions on Circuits and Systems II*: Express Briefs, vol. 61, no. 11, pp. 895–899, Nov 2014.

[4] K. Cho, S.-J. Lee, and K. Eshraghian,"Memristor-CMOS logic and digital computational components" *Micro electronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.

[5] L. Guckert and E. E. Swartzlander, "Optimized Memristor-Based Multipliers," *in IEEE Transactions on Circuits and Systems I*: Regular Papers, vol.64, no.2, pp. 373-385, Feb. 2017.

[6] S. Haghiri, A. Nemati, S. Feizi, A. Amirsoleimani, A. Ahmadi, and M. Ahmadi, "A memristor based binary multiplier," *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON, Canada, pp. 1-4,2017.

[7] A. Chakraborty, A. Dhara, and H. Rahaman, "Design of memristor-based up-down counter using material implication logic," *in 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*. IEEE, pp. 269–274, 2016.

[8] K. Alammari, A. Ahmadi, and M. Ahmadi, "Hybrid Memristor-CMOS Based Up-Down Counter Design," *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Glasgow, UK, pp. 1-4,2020.

[9] A. Sasi, A. Amirsoleimani, A. Ahmadi, and M. Ahmadi, "Hybrid memristor-cmos based linear feedback shift register design," *24th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, pp. 62–65. 2017.

[10] P. L. Thangkhiew, R. Gharpinde and K. Datta, "Efficient Mapping of Boolean Functions to Memristor Crossbar Using MAGIC NOR Gates," *in IEEE Transactions on Circuits and Systems I*: Regular Papers, vol. 65, no. 8, pp. 2466-2476, Aug. 2018.

[11] L. O. Chua, “Memristor-The missing circuit element,” *IEEE Trans. Circuit Theory*, vol. CT-18, no.5, pp. 507-519, Sep. 1971.

[12] I. Vourkas and G. C. Sirakoulis, “Emerging memristor-based logic circuit design approaches: A review*,” IEEE Circuits and Systems Magazine*, vol. 16, no. 3, pp. 15–30, third quarter 2016.

[13] M. A. Zidan, J. P. Strachan, and W. D. Lu, “The future of electronics based on memristive systems,” *Nature Electronics*, vol. 1, no. 1, pp. 22–29, 2018.

[14] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, “Vteam: A general model for voltage-controlled memristors,” *IEEE Transactions on Circuits and Systems II*: Express Briefs, vol. 62, no. 8, pp. 786–790, Aug 2015.

[15] A. Siemon, S. Menzel, A. Marchewka, Y. Nishi, R. Waser, E. Linn, "Simulation of TaOx-based Complementary Resistive Switches by a Physics-based Memristive Model”, *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1420-1423, 2014.

[16] N. Talati, S. Gupta, P. Mane and S. Kvatinsky, "Logic Design Within Memristive Memories Using Memristor-Aided logic (MAGIC)," *in IEEE Transactions on Nanotechnology*, vol. 15, no. 4, pp. 635-650, July 2016.

[17] P. Huang, J. Kang, Y. Zhao, S, Chen, R.Han, & Z. Zhou, et al. “Reconfigurable non volatile logic operations in resistance switching crossbar array for large-scale circuits,” *Advanced Materials*, 28(44), 9758. 2016.

[18] Y. Yang, J. Mathew, S. Pontarelli, M. Ottavi and D. K. Pradhan, "Complementary Resistive Switch-Based Arithmetic Logic Implementations Using Material Implication," *in IEEE Transactions on Nanotechnology*, vol. 15, no. 1, pp. 94-108, Jan. 2016.

[19] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C.Weiser and E. G. Friedman, "MRL-Memristor Ratioed Logic," *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, Turin, Italy, pp. 1-6, 2012.

[20] T. Singh, “Hybrid Memristor-CMOS (MeMOS) based Logic Gates and Adder Circuits,” *CoRR, arXiv:1506.06735* [cs.ET], pp.1-11, June 2015.

Chapter 6

LIF Neuron -A Memristive Realization

6.1 Introduction

Inspired by the human brain, neuromorphic systems have drawn much attention in recent years due to their massive parallelism, low power consumption, fault tolerance and capacity for adaptive learning [2-5]. Besides conventional computing systems, Von-Neumann is facing severe challenges [6], which made brain-inspired computing an alternative approach to replace conventional computing systems. Thus, hardware implementation for neuromorphic systems has been an ambitious research field and very appealing to computing architecture. The Complementary Metal Oxide Semiconductor (CMOS) technology offers the platform to realize the neuromorphic systems to emulate the computations inside the human brain, which is required for the implementation of neuron circuits and electronic synapses. Therefore, several works have been proposed on many neuron circuits, such as integrate-and-fire (I&F) neuron [7], leaky integrate-and-fire (LIF) neuron [8] and electronics synapses [9]. Although these CMOS-based circuits have shown maturity towards emulating and understanding the neuromorphic systems, it comes at the price of occupying large silicon area with considerable power consumption due to the high number of transistors involved. With the aim to solve this problem, the new emerging non-volatile memory (NVM) devices known as memristors could be the promising candidate for future computing architecture, owing to their nanoscale size and the ability to integrate with the exciting CMOS technology.

Moreover, memristors can behave similarly to the human brain in adjusting or changing their responses to stimulation patterns [10], making them a perfect choice to imitate biological behavior. There has been intensive research [11-13] covering the area of neuromorphic systems engaging memristors, whereas memristor devices are used to implement synapses with crossbar structure. However, it is very difficult to integrate the pure memristive crossbar structure with CMOS neurons due to the differences in their logical states. In this chapter, a LIF neuron model has been implemented based on the hybrid CMOS/Memristor, taking the advantage of the ability to integrate standard CMOS with memristor devices and the capability of fabricating memristor devices on top of CMOS substrate, which can provide additional storage and fast computing. The rest of this chapter is arranged as follows: Section 6.2 briefly introduces the Memristor Ratioed Logic (MRL), which is exploited as a design method for the (LIF) neuron. Section 6.3 includes information about implementing hybrid CMOS/ Memristor leaky integrate and fire (LIF) neuron. In Section 6.4, simulation results and discussion on LIF neuron-based MRL. Finally, remarks and conclusions are presented in section 6.5.

6.2 Memristive design method

Memristor is a new emerging resistive device that adds new capabilities to CMOS technology, which can enhance the performance of neuromorphic architectures. The integrated platform of CMOS/Memristors, also known as Memristor Ratioed Logic MRL [14], offers the ability to perform all logic gates utilizing memristors and stack them in between CMOS upper layers [15], which will allow faster computing and additional storage at reasonable chip area for neuromorphic hardware implementation. MRL logic is CMOS compatible. Hence, the logical state of this method is based on the output voltage level. Thus, low and high voltage represent logical states (’0’ and ’1’), respectively. MRL-based AND and OR gates implementation requires only two memristors. While NAND and NOR gates require a CMOS inverter to be connected to the output of AND and OR gates to facilitate the circuits with the interface and control operation [16]. Fig.6.1 displays MRL-based AND, NAND, OR and NOR.

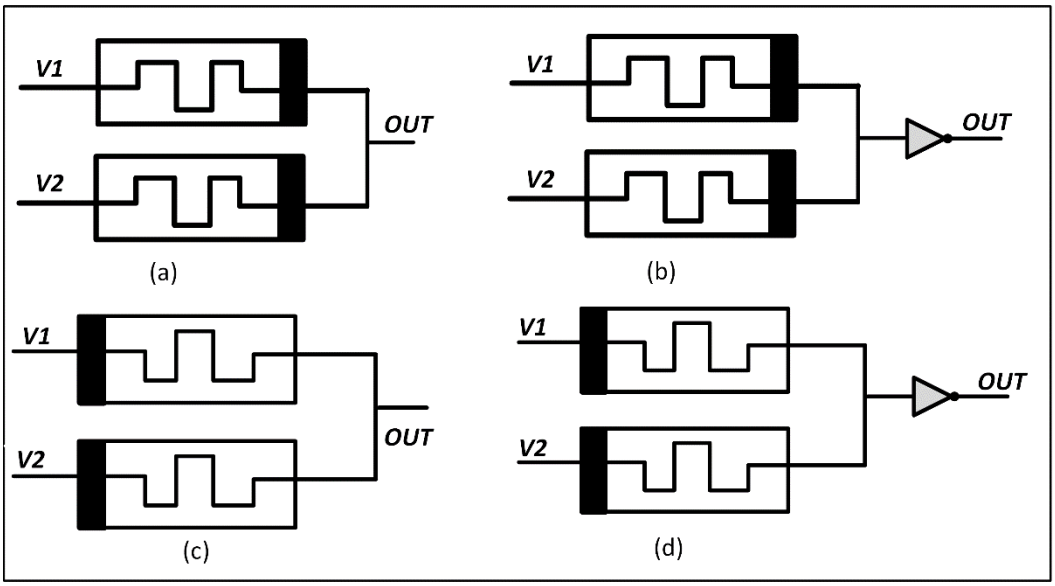


Figure 6.1: The schematic of MRL based logic gates
(a) AND, (b) NAND, (c) OR and (d) NOR

6.3 Hybrid CMOS/ Memristor LIF Neuron

The LIF model is the most popular structure used to emulate the behavior of biological neurons. The LIF neuron releases a spike when the membrane potential V_{mem} reaches a certain threshold V_{th}. Fig.6.2 depicted the simplest form of LIF model, V_{mem} is the membrane potential, R_{Leak} is leak path resistance and C_{mem} is capacitive path. , I_{input} is used to model the input spikes.

$$C_{mem}\frac{dV_{mem}}{dt} = -\frac{1}{R_{Leak}}(V_{mem} - V_{reset}) + I_{input} \qquad (6.1)$$

$$I_{Leak} = -\frac{1}{R_{Leak}}(V_{mem} - V_{reset}) \qquad (6.2)$$

If $V_{mem} > V_{threshold}$ than release spike and set $V_{mem} = V_{reset}$

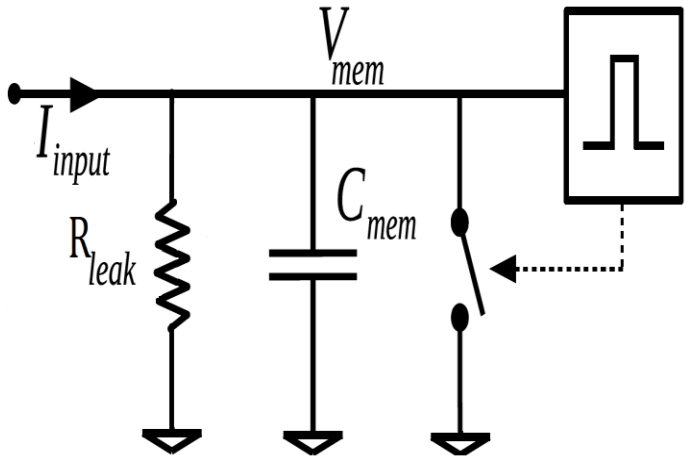


Figure 6.2: The. schematic diagram of the leaky integrate-and-fire model

A proposal for a memristive LIF neuron model has been presented in Fig. 6.3 based on the MRL method. The design involves several building blocks as follows:

6-3-1 Synapse Block

The neuron receives several input spikes and realizes the synaptic weight of the inputs respectively. The main element in this block is the 4-bit memristor-based counter. The up-counter has been implemented using the MRL-based up-down counter [17]. The MRL-based counter can change its count direction, whether up or down, at any point within the counting sequence. Hence, up-counting is chosen from the (Control Mode). The counter starts counting from the initial value until it reaches the synaptic weight, which is defined by the pulse width. Synaptic with larger values have wider pulses, thus having more effect on the neuron core. Synaptic weights are sent to the neuron core block. At the same time, a feedback signal is sent to reset the counter. We assume that spikes are not arriving to the block at the same time.

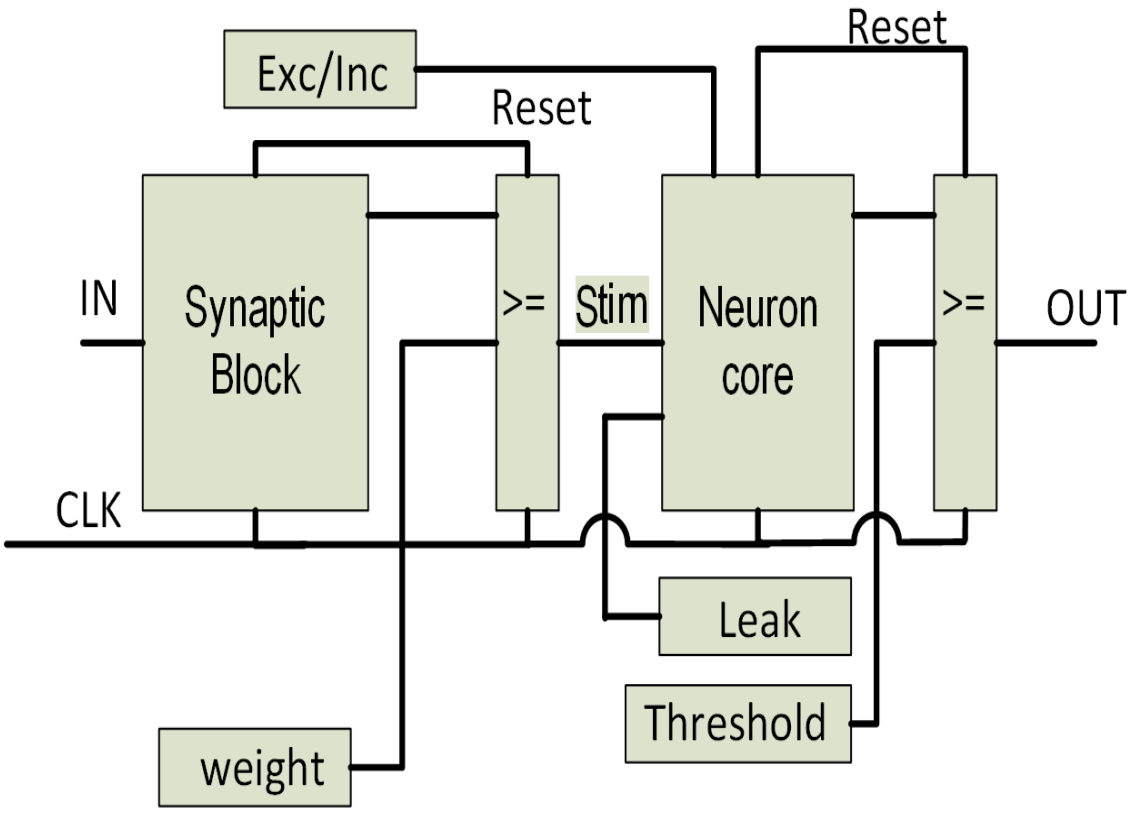


Figure 6.3: The Schematic of the proposed MRL- based LIF neuron

6-3-2 Neuron Core

The integration and leak function are performed in this block. The integration process commences when a pulse is received from the synapse block. The membrane potential is realized by a 4-bit memristor-based up-down counter, which was implemented and tested in Cadence Virtuoso schematic level and previously published in [17].The counter updates its logical states every time receives a synapse pulse. The counter is incremented or decremented based on the pulse type received from the synapse block. The parameter Exc/Inh indicates the type of the input spike, whether excitatory or inhibitory. However, when no synapse pulse is applied to the neuron core, the leak function takes place by continuously decrementing the counter.

6-3-3 Neuron Output

The final stage in the design is to compare the output of the up-down counter “membrane potential” with a threshold value via a 4-bit memristor-based comparator. A spike is generated when the membrane potential exceeds the threshold value. Simultaneously, a feedback signal is sent to the neuron core to reset the membrane potential to its initial value.

6.4 Experimental Results

The model for LIF neurons has been implemented based on the MRL method. Hence the memristor devices are employed in the structure depicted in Fig.6.3. The LIF neuron is made up of a few simple blocks: counters, comparators, and logical gates. The main element in the design is the memristive counter which was implemented and tested in Cadence Virtuoso schematic level and previously published in [17].The mechanism of the LIF neural model can be explained as follows: The up-down counter begins counting as the input “Stim” from the synapse block entering the counter. The counter starts the up counting when the parameter Exc/Inh is at logic 1. Otherwise, down counting is performed. The width of the pulse, which is received from the synapse block, provides information about the synaptic weights. However, when no synapse pulse is applied to the neuron core, the counter has no counting activity and the counter’s value starts to decrease emulating the leak behavior observed in the neuron model definition. Finally, a comparator is used to detect the threshold level. Fig.6.4. Demonstrates the behavior of the memristive LIF model in Cadence Virtuoso. In the first period, the synaptic weights are accumulated which contributes to the membrane potential until the threshold value (V_{th}=7) is reached when the neuron fires. In the second period of the simulation, the membrane potential starts to decrease its value since there is no input spikes to the up-down counter “stim= 0” ,“leak behavior”. Fig 6.5 confirms the action optional of the LIF neuron model.

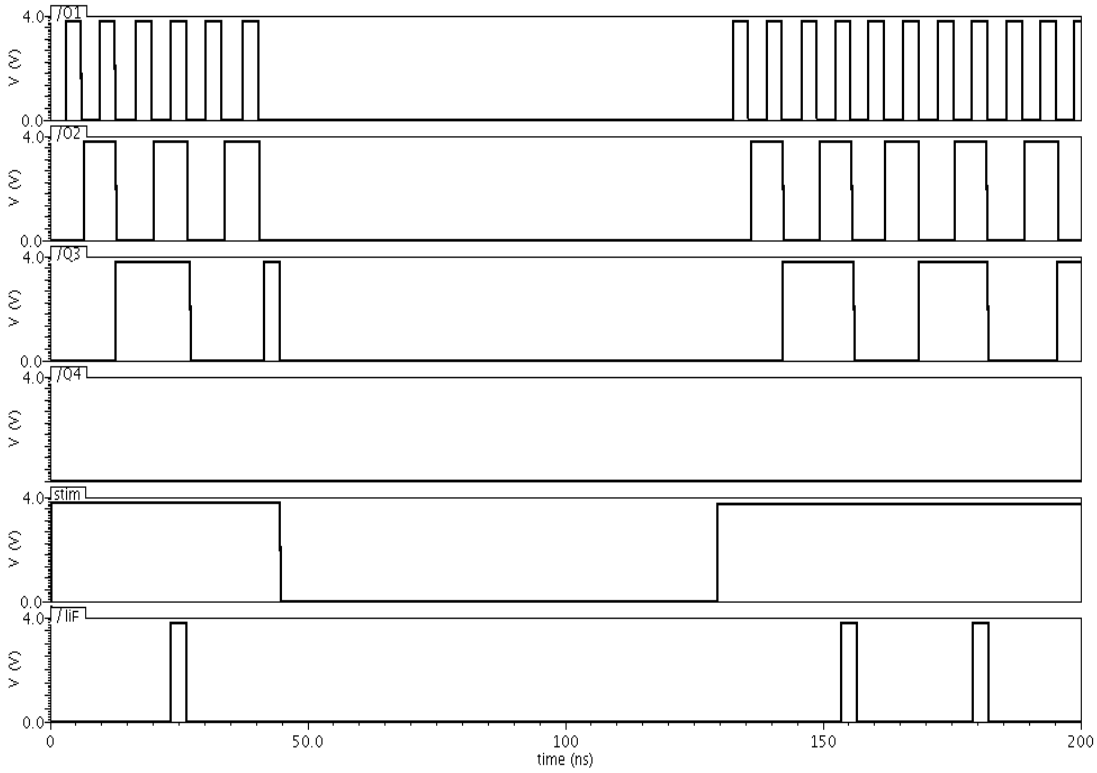


Figure 6.4: The behavior of the proposed MRL- based LIF neuron

The exceptional about this proposed memristive LIF neuron circuit compared to other works utilizing memristors [18-19] is the comprehensive involvement of the memristor devices in every aspect of the design. In [18] the LIF neuron is implemented using an integrator, comparator with eight switches, and control logic. While in [19] the neuron circuit comprises an analog demultiplexer, summing amplifier, comparator, and digital part includes AND, OR gates with DFF. However, both designs have employed memristor devices only to implement synapses.

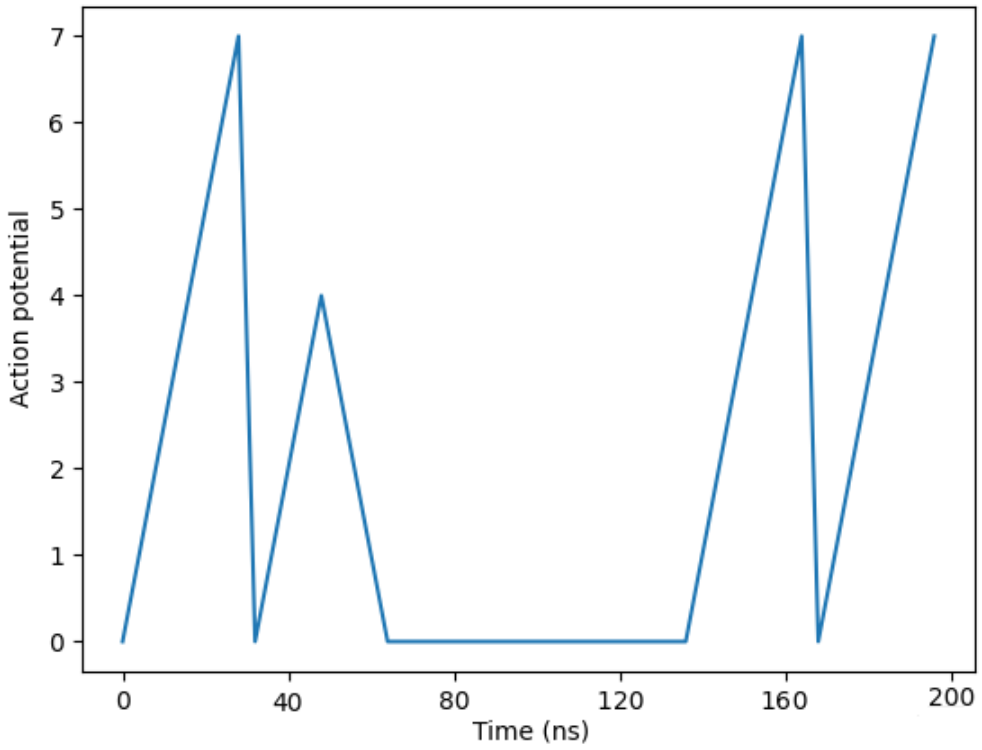


Figure 6.5: The action potential of the memristive LIF neuron model shows three fire events taking place as the threshold value (Vth=7) is reached and one short spike due to short current because of no input spikes to inter the up-down counter “leak behavior commences”.

At the same time, our design shows that both memristive counters in the synapse block and the neuron core block employ four MRL-based TFF each, connected with other MRL-based Boolean gates shown in Fig.6.1. The circuit of MRL-based TFF consists of MRL-based DFF and MRL-based 2-input XOR gate. The memristive TFF shown in Fig.6.6. utilizes 22 memristor devices and 16 MOSFETs [17].

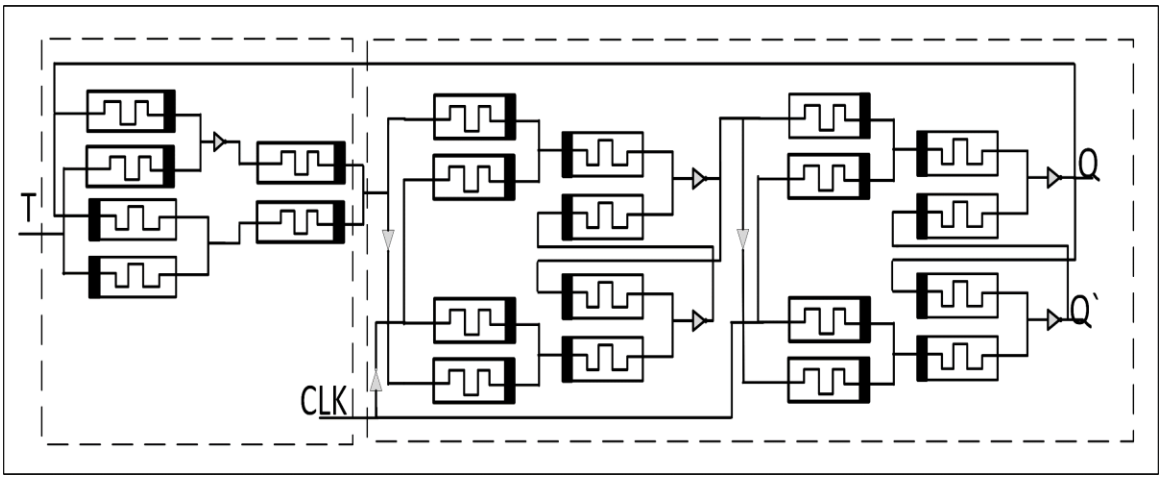


Figure:6.6: The circuit schematic of the memristor based TFF

Table 6.1. Number of all devices in the proposed design of the Memristive LIF neuron

Design Element		Up-down counter	Up-counter	Comparator/ Synaptic	Comparator/ Core
Device	MOSFET	34	32	16	16
	Memristor	110	94	66	66

While the comparator was implemented based on the memristor-based MRL logic gates depicted in Fig.6.7. The number of all devices of the memristive neuron circuit are listed in Table 6.1. Indicates that the MRL-based design is efficient in terms of layout area since memristor devices can be fabricated on the upper layers of CMOS transistors [20-21]. The integral platform of CMOS transistors and Memristor devices has led to almost 50% in area saving [22] compared to the same CMOS-based design; This is because of the limited utilized number of CMOS transistors in the proposed LIF neuron model.

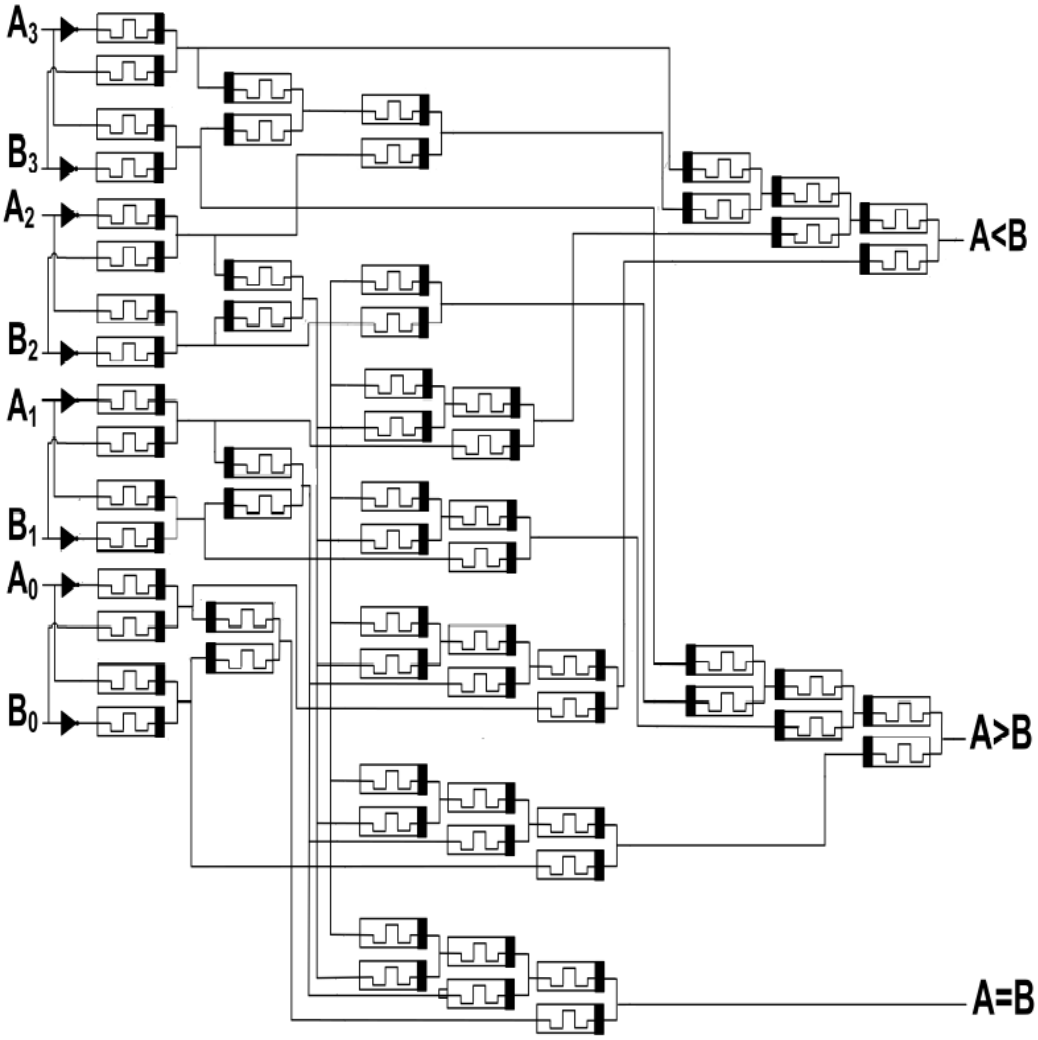


Figure:6.7: The circuit schematic of the memristor based 4-bit comparator

6.5 Conclusion

This work utilizes the memristor model presented in Verilog-A [23] to provide the proposed architecture of leaky integrate-and-fire (LIF) neuron with the memristive behavior. The proposed architecture is based on the hybrid CMOS/Memristor gates, which enable integration with existing CMOS-based neurons to support large-scale neuromorphic systems. The exception of this proposal is the involvement of the

memristor devices in every aspect of the design since we divided our design into three parts: A synaptic block with a memristive up-counter , a core neuron circuit with an MRL-based up-down counter previously published, and a comparator to generate a spike when the membrane potential exceeds the threshold value. The LIF neuron circuit is area-efficient due to the MRL based gates employed in the proposed. The design was implemented and verified in Cadence Virtuoso at each stage to confirm its functionality.

References

[1] S. E. Thompson and S. Parthasarathy, “ Moore's Law: The Future of Si Microelectronics,” *Materials Today*, Vol. 9, No. 6, pp. 20-25, June 2006.

[2] G. Indiveri, E. Chicca and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity, "*IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211-221, Jan. 2006.

[3] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. Chandrasekaran, J. Bussat, R. Alvarez, J. Arthur, P. Merolla, and K. Boahen. "Neuro grid: A mixed-analog-digital multichip system for large-scale neural simulations." *Proceedings of the IEEE 102, no. 5*, pp.699-716,2014.

[4] C. Mead, "Neuromorphic electronic systems," *in Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629-1636, Oct. 1990.

[5] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum and H. -. P. Wong, "An Electronic Synapse Device Based on Metal Oxide Resistive Switching Memory for Neuromorphic Computation," *IEEE Transactions on Electron Devices*, vol. 58, no. 8, pp. 2729-2737, Aug. 2011.

[6] A. Stefano, P. Narayanan, H. Tsai, R. Shelby, I. Boybat, C. Nolfo, S. Sidler, "Equivalent-accuracy accelerated neural-network training using analogue memory." *Nature 558*, pp.60-67,2018.

[7] S. Brink et al., "A Learning-Enabled Neuron Array IC Based Upon Transistor Channel Models of Biological Phenomena," *in IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, no. 1, pp. 71-81, Feb. 2013.

[8] D. Sangya, V. Kumar, A. Shukla, N. Mohapatra, and U. Ganguly. "Leaky integrate and fire neuron by charge-discharge dynamics in floating-body MOSFET." *Scientific reports 7*, no. 1, 2017.

[9] Seo, Jae-sun, B. Brezzo, Y. Liu, B. Parker, S. Esser, R. Montoye, B. Rajendran "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons." *In 2011 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1-4. ,2011.

[10] S. Gotarredona, T. Masquelier, T. Prodromakis, T. Indiveri, G. Linares-Barranco, B. “STDP and STDP variations with memristors for spiking neuromorphic learning systems. Frontiers in neuroscience, 7, 2. 2013

[11] A. Stefano, S. Balatti, V. Milo, R. Carboni, Z. Wang, A. Calderoni, N. Ramaswamy, and D. Ielmini. "Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM." *IEEE Transactions on Electron Devices 63, no. 4* , pp. 1508-1515, 2016.

[12] P. Sangsu, J. Noh, M. Choo, A. Sheri, M. Chang, Y. Kim, C. Kim "Nanoscale RRAM-based synaptic electronics toward a neuromorphic computing device, "Nano tec no " Nanotechnology, vol. 24, no. 38, pp. 1–6, Sep. 2013.

[13]. R. Bipin, Y. Liu, J. Seo, K. Gopalakrishnan, L. Chang, D. Friedman, and M. Ritter. "Specifications of Nanoscale Devices and Circuits for Neuromorphic Computational Systems*," in IEEE Transactions on Electron Devices*, vol. 60, no. 1, pp. 246-253, Jan. 2013.

[14] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser and E. G. Friedman, "MRL-Memristor Ratioed Logic," *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, pp. 1-6 ,Turin, Italy, 2012,.

[15] K. Cho, S. Lee, and K. Eshraghian, "Memristor CMOS logic and digital computational components, "*Microelectronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.

[16] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," Nature Electronics, vol. 1, no. 1, pp. 22–29, 2018.

[17] K. Alammari, A. Ahmadi, and M. Ahmadi, "Hybrid Memristor-CMOS Based Up-Down Counter Design," *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 1-4Glasgow, UK, 2020.

[18] C. Myonglae, B. Kim, S. Park, H. Hwang, M. Jeon, B. Lee, and B. Lee. "Neuromorphic Hardware System for Visual Pattern Recognition with Memristor Array and CMOS Neuron," *in IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2410-2419, April 2015.

[19] J. Yuning, P. Huang, D. Zhu, Z. Zhou, R. Han, L. Liu, X. Liu, and J. Kang, "Design and Hardware Implementation of Neuromorphic Systems with RRAM Synapses and Threshold-Controlled Neurons for Pattern Recognition," *in IEEE Transactions on Circuits and Systems I*: Regular Papers, vol. 65, no. 9, pp. 2726-2738, Sept. 2018.

[20] J. Cong and B. Xiao, "mrFPGA: A novel FPGA architecture with memristor-based reconfiguration," *2011 IEEE/ACM International Symposium on Nanoscale Architectures*, pp. 1-8, 2011.

[21] H. Ru, L. Zhang, D. Gao, Y. Pan, S. Qin, P. Tang, Y. Cai, and Y. Wang, "Resistive switching of silicon-rich-oxide featuring high compatibility with CMOS technology for 3D stackable and embedded applications", *Appl. Phys*. A 102, pp.927–931, 2011.

[22] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser and E. G. Friedman, "MRL-Memristor Ratioed Logic,"*13th International Workshop on Cellular Nanoscale Networks and their Applications*, Turin, Italy, pp. 1-6, 2012.

[23] A. Siemon, S. Menzel, A. Marchewka, Y. Nishi, R. Waser, E. Linn, "Simulation of TaOx-based Complementary Resistive Switches by a Physics-based Memristive Model", *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1420-1423, 2014.

Conclusion

7.1 A Summary of Conclusions

Overall, the dissertation contributes to the area of memristive logic design by implementing several conventional arithmetic and computational building block circuits using memristive-only logic "MAGIC gates" and the integrated platform of Memristors-CMOS "MRL gates" . Moreover, it provides less area and high-speed memristor-based digital design with a solution for circuits degradation and fanout.

7.2 Memristor-Based Logic Design

This work reported in this dissertation mainly focused on memristor-based logic design aiming to address CMOS's challenges concerning further scaling down. Throughout this course, several comparisons were carried out between the two technologies of Memristors and CMOS to explore the possibility of adding new capabilities to CMOS technology to address its challenges and not to compete with CMOS in any way. Though, CMOS is a mature technology. The Memristors-based logic design indicates that it has the potential to attract modern CMOS design, and the integral platform of the CMOS Memristor could be a practical solution to ever-growing concerns regarding CMOS transistor downsize. Several approaches of memristor-based logic gate styles are investigated in this research, utilizing, and developing different memristor models to implement an efficient memristive digital design , given the variety of the work proposed between chapters and sections. Conclusions have been summarized for each section.

In Chapter 1. A brief overview of the memristor device definition and historical background was presented. After that, Memristor device potential applications were discussed in the next section, in this section; some of the key applications of memristors, such as memory storage, neuromorphic computing, and logic design, was presented, followed by an insight into the digital architecture with Memristor and the challenges it faces its development.

In Chapter 2. A compact, low power design of a hybrid Memristor-CMOS based Finite Impulse Response (FIR) Filter was presented. The proposed design has been described using Verilog High Description Language (HDL) and tested with Cadence design systems, NC-Verilog, and MATLAB. The significant aspect of this proposed architecture was the standard memristor-based cell library, which contains all characterized cells employed in the FIR filter. This library is utilized by Synopsys compiler to produce memristive netlist gates. The simulation results have shown that the behavioral model of the design can distinguish between all input signals and passes only signals with the desired frequency. The proposed hybrid Memristor-CMOS based FIR shows a significant enhancement in terms of area, consumed power, and delay compared to equivalent CMOS architecture.

In Chapter 3. A fast, low area, and low power for the Up-Down counter was implemented based on the hybrid Memristor-CMOS platform MRL. The memristive counter employs fewer numbers of transistors than the conventional CMOS counter with 34 MOSFETs and 110 memristors. Although the IMPLY-based Up-Down counter reported less area, the MRL-based Up-Down counter overcomes the issues of delay and complexity produced by the lengthy operational steps associated with the IMPLY-based counter. The simulation results have illustrated that the Up-Down counter can successfully start or stop counting at any logical state and resume its operation from any other desired logic state, which proves the design and functionality.

In Chapter 4. A compact design for the building blocks of the XAX- module was presented based on the MRL scheme utilizing the device $Pt\ /TaOx/Ta$. The MRL-based XAX-module design comprises simple blocks of the DFF circuit together with the XOR-AND-XOR function. The MRL- based DFF is the main component in the module in terms of device numbers. The MRL- based DFF utilizes 16 Memristor devices and 14

MOSFETs, while the XOR-AND-XOR function is another block that participates in the XAX module with 12 memristor devices and 4 MOSFETs. At the same time, the same CMOS-based design employs 24 MOSFET devices.

This work proves that utilizing MRL- based gates in the blocks that made up the XAX structure implies a significant reduction in the number of transistors. The Memristors can be fabricated on the upper layers of CMOS inverter. This Combination has led to almost 15% in area saving compared to the same CMOS-based-XAX module. This is mainly because the layout area estimate of the design was carried out based on the size of CMOS inverter occupied by each cell. From the performance point of view, the design power consumption and delay were also examined and compared against other CMOS-based designs. The design was implemented and verified in Cadence Virtuoso at each stage to confirm its functionality.

In Chapter 5, a fast and efficient area Memristor-only-based full adder and a hybrid CMOS/Memristor-based full adder are proposed. In the Memristor-only style, the MAGIC logic is favorited over IMPLY logic in the implementation of the full adder due to the reduced area layout offered by the MAGIC approach thanks to the simplicity of mapping MAGIC gates to the crossbar array, whereas mapping IMPLY gates require a resistor before any row which increases the design area. Also, IMPLY requires more voltage sources than MAGIC to perform the logic computations. MAGIC-based full adder design utilizes only 25 memristors, while the hybrid CMOS /Memristor full adder requires 18 Memristor devices and 4 MOSFETs. Consequently, both designs occupy a small area as compared to the CMOS- based design. However, the MAGIC-based full adder is more area efficient than the MRL-based full adder. While the MRL- based full adder is a considerably low-cost design since it is CMOS compatible and requires only one voltage level. Moreover, the proposed MAGIC-based full adder has fewer computational steps in comparison to the same IMPLY-based design, another pure memristor method. The Memristor device of VTEMA and Pt / TaOx / Ta have been successfully used to provide the design with the Memristive behavior while simulations confirm the proposed full adder functionality.

In Chapter 6, memristor devices and CMOS transistors work together to form a hybrid CMOS/memristor neuron circuit for leaky integrate-and-fire (LIF) neurons. The proposed design was implemented using Pt / TaOx / Ta memristor device and simulated in Cadence Virtuoso. The simulation results demonstrate the design functionality.

The proposed model is small in terms of the area since it utilizes the hybrid CMOS/memristor gates. This will allow current neuromorphic systems to increase their density to meet the ever-growing demand.

In Chapter 7, a summary about the dissertation, conclusion and future work is presented.

7.3 Possible Future Works and Extensions

While the proposed works of memristor-based logic design hold great promises for future VLSI, it is important to note that memristor devices are still in the early stages of development. Further researchers are needed to continue working on improving the memristors' reliability, scalability, and integration into existing electronic systems. The memristive structures proposed through this thesis have many possible extensions in various applications that can have many positive implications in digital architecture. The work of XAX- Module in Chapter 4 can be utilized to introduce the newly discovered device to implement a memristive design for finite field multiplier, while the work of the Memristor based-full adder circuit in Chapter 5 can be engaged in different configurations. Also, the memristive circuit for the leaky integrate-and-fire (LIF) neuron in Chapter 6, has a potential possibility to be extended to other neuromorphic systems with the opportunities to increase their density.

Appendices

Appendix A Verilog A model for VCM- Pt/TaOx/Ta device

```
// VerilogA for VCM- Pt/TaOx/Ta device, vcm_Pt/TaOx/Ta_unit,
veriloga

`include "constants.vams"
`include "disciplines.vams"

module vcm_Pt/TaOx/Ta_unit(te,be);

inout te,be;
electrical
te,be,gnd,N,T,schottky,schottkytunnel,trackRLZw,trackw,tunnel,
resistorRel,resistorR0,ion,trackphib ;ground gnd;

     parameter real A      = 3.14159e-14;
     parameter real e      = 1.602e-19;
     parameter real kb     = 8.617e-5;      // eV/K
     parameter real kbsi   = 1.3807e-23;    // VAs/K
     parameter real Ninit  = 5;             // concentration/1e26
     parameter real T0     = 300;
     parameter real Arich  = 11e6;          // A/m²/K Hur
     parameter real LDisc  = 4;             // nm
     parameter real phin0  = 0.5;           //
     parameter real phin   = 0.1;
     parameter real es     = 2.3e-10;       // 26*e0
     parameter real esopt  = 4.8698e-11;    // 5.5*e0
     parameter real pi     = 3.1416;
     parameter real un     = 1.6e-5;         //electrons mobility
     parameter real Nmax   = 5;             // Nmax=5e26/1e26
     parameter real Nmin   = 0.0003;        // Nmin=1e24/1e26
     parameter real mdiel  = 9.10938e-31;        // me
     parameter real h      = 6.626e-34;
     parameter real hq     = 1.05457e-34;
     parameter real LTaOx  = 11n;       //oxide materialthickness
     parameter real phit   = 0.2;      // eV
     parameter real C31    = 6e-12; // fit for area current
     parameter real zvo    = 2;     // charge of ions in e
     parameter real Nion   = 1e26; // ionic currentconcentration
     parameter real a      = 0.5e-9;  // hopping distance in m
     parameter real nyo    = 1e12;    // attemp frequenzy
     parameter real dWa    = 0.93; //  oxygen vacancies energy
     parameter real kTaO   = 1;        //W/mK
     parameter real R0     = 570;      //Ohm


     real
Iion,Emax,phib,psi,Ediel,w,RLZw,E00,E0,epsstrich,Ischottkytunnel
;
```

```
      analog
            begin
      ////////////////////////////////////////////////////////////////

                  V(schottkytunnel,ion)<+0;
                  V(schottkytunnel,resistorRel)<+0;
                  V(te,schottky)<+0;
                  V(te,tunnel)<+0;
      ////////////////////////////////////////////////////////////////
                  V(N,gnd)
<+idt(-1/A/L/1e-9/e/zvo*I(ion,resistorR0)/1e26,Ninit);
      ////////////////////////////////////////////////////////////////
                  V(T,gnd)
<+(V(resistorRel,resistorR0)*I(resistorRel,resistorR0)/kTaO*L*
1e-9/A+T0)/1000;
      ////////////////////////////////////////////////////////////////

                  if (V(schottky,schottkytunnel)<phin0-phin)
                        psi=phin0-phin-V(schottky,schottkytunnel);
                  else
                        psi=0;

                  phib=phin0-sqrt(sqrt(pow(e,3)*V(N,gnd)*1e26
*psi/8/pow(pi,2)/(es*esopt*esopt)));

                  if (V(tunnel,schottkytunnel)<0)
                        begin
                              RLZw=sqrt(2*es*(phib-
V(tunnel,schottkytunnel))/e/Nmax/1e26);
                              w=-phib*RLZw/(-
abs(V(tunnel,schottkytunnel))+phin-phib);
                        Ediel=V(tunnel,schottkytunnel)/w;
                        I(tunnel,schottkytunnel)<+-
A*pow(e,3)/8/pi/h/phib/e*pow(Ediel,2)*exp(-4*sqrt(2
*mdiel*pow(abs(phib*e),3))/3/hq/abs(Ediel)/e);
                        end
                  else
                        begin
                        I(tunnel,schottkytunnel)<+0;
                        w=0;
                              RLZw=0;
                              Ediel=0;
                        end
      ////////////////////////////////////////////////////////////////
                  I(schottky,schottkytunnel)<+A*Arich*pow(1000
*V(T,gnd),2)*exp(-phib/kb/1000/V(T,gnd))*(exp(1/kb/V(T,gnd)/1000
*V(schottky,schottkytunnel))-1);
```

```
      ////////////////////////////////////////////////////////////
                  if (((V(N,gnd)<Nmin)&(V(te,be)>0))|((V(N,gnd)>
Nmax)&(V(te,be)<0)))
                        begin
                              Iion=0;
                        end
                  else
                        begin
                              I(ion,resistorR0)
<+zvo*e*Nion*a*nyo*A*exp(-
dWa/kb/V(T,gnd)/1000)*sinh(zvo*a/2/kb*V(ion,resistorR0)/L/1e-9/1
000/V(T,gnd));
                        end
      ////////////////////////////////////////////////////////////
                           V(resistorRel,resistorR0)<+1/(V(N,gnd)*
1e26*e*un)*L*1e-9/A*I(resistorRel,resistorR0);
      ////////////////////////////////////////////////////////////
                  if (V(te,be)>0)
                        begin
                              I(te,be)<+C31*V(te,be)/Ldevice*exp(-
phit/kb/T0)*exp(sqrt(pow(e,3)/pi/es*abs(V(te,be))/Ldevice)/1.5/k
bsi/T0);
                        end
                  else
                        begin
                              I(te,be)<+C31*V(te,be)/Ldevice*exp(-
phit/kb/T0)*exp(sqrt(pow(e,3)/pi/es*abs(V(te,be))/Ldevice)/1.5/k
bsi/T0);
                        end
      ////////////////////////////////////////////////////////////
                  V(resistorR0,be)<+(1/(Nmax*1e26*e*un)*(Ldevice-
L*1e-9)/A+R0)*I(resistorR0,be);

            end


endmodule
```

Appendix B Verilog A model for VTEAM Memristor

```
// VerilogA model for Vteam memristor

`include "disciplines.vams"
`include "constants.h"

module Memristor(p, n,w_position);
  input p;//positive pin
  output n;//negative pin
  output w_position;// w-width pin
  electrical p, n,gnd;
  Distance w_position;
  ground gnd;

    parameter real model = 4; //  Vteam model
    parameter real window_type=5;// Vteam model
    parameter real dt= 1e-10 ;
    parameter real Roff = 1000;
    parameter real Ron = 50;
    parameter real D = 3e-09;
    parameter real uv = 1e-15;
    parameter real w_multiplied = 1e08;
    parameter real p_coeff = 2;// Windowing function coefficient
    parameter real J = 1;
    parameter real p_window_noise=1e-18;
    parameter real threshhold_voltage=0;
    parameter real c_off = 3.5e-6;
    parameter real c_on = 4e-5;
    parameter real i_off = 115e-6;
    parameter real i_on = -8.9e-6;
    parameter real x_c = 107e-12;
    parameter real b = 500e-6;
    parameter real a_on = 2e-9;
    parameter real a_off = 1.2e-9;
    parameter real K_on=-10;
    parameter real K_off=5e-4;
    parameter real Alpha_on=3;
    parameter real Alpha_off=1;
    parameter real v_on=-0.2;
    parameter real v_off=0.02;
    parameter real IV_relation=0;
    parameter real x_on=0;
    parameter real x_off=3e-09;
    parameter real alpha = 2;
    parameter real beta  = 9;
    parameter real c = 0.01;
    parameter real g = 4;
    parameter real N  = 14;
    parameter real q  = 13;
    parameter real a  = 4;
```

```
analog
      begin
     ////////////////////

    if(first_iteration==0) begin
        w_last=init_state*D;
// if this is the first iteration,
//start with w_init
        x_last=init_state*D;
// if this is the first iteration,
// start with x_init
    end

if (model==4) begin // VTEAM model

    if (V(p,n) >= v_off) begin
        dxdt =K_off*pow((V(p,n)/v_off-1),Alpha_off);
    end

    if (V(p,n) <= v_on) begin
        dxdt =K_on*pow((V(p,n)/v_on-1),Alpha_on);
    end

        if ((v_on<V(p,n)) && (V(p,n)<v_off)) begin
    dxdt=0;
    end

//Kvatinsky window2 VTEAM only
    if (window_type==5) begin

            if (V(p,n) >= 0) begin
            x=x_last+dt*dxdt*exp(-exp((x_last-a_off)/x_c));
            end

       if (V(p,n) < 0) begin
           x=x_last+dt*dxdt*exp(-exp((a_on-x_last)/x_c));
       end

        end // Kvatinsky window

    if (x>=D) begin
    dxdt=0;
    x=D;
    end
```

```
    if (x<=0) begin
    dxdt=0;
    x=0;
    end

       lambda = ln(Roff/Ron);

     //update the output ports(pins)
        x_last=x;
        Metr(w_position) <+ x/D;

     if (IV_relation==1) begin

        V(p,n) <+ Ron*I(p,n)*exp(lambda*(x-x_on)/(x_off-x_on));

     end

     else if (IV_relation==0) begin

        V(p,n) <+ (Roff*x/D+Ron*(1-x/D))*I(p,n);

     end

    first_iteration=1;

end // end VTEAM model


  end   // end analog

endmodule
```

Vita Auctoris

Khalid Alammari received the B.Sc. degree in Electrical Engineering from Sirte University, Sirte, Libya, in 1996, and M.Sc. diploma in communications and signal processing and M.Sc. degree with merit in Electrical Power Engineering from Newcastle University, Newcastle upon Tyne, United Kingdom, in2002 and 2003 respectively. He is now a Ph.D. student at the University of Windsor, Canada. His current research interests include logic design, in-memory computing, RRAM, memristive circuits, and VLSI circuit design.