

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

1-7-2021

Memristor-Based Digital Systems Design and Architectures

Abubaker Alarabi Sasi
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Sasi, Abubaker Alarabi, "Memristor-Based Digital Systems Design and Architectures" (2021). *Electronic Theses and Dissertations*. 8512.
<https://scholar.uwindsor.ca/etd/8512>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

MEMRISTOR-BASED DIGITAL SYSTEMS DESIGN AND ARCHITECTURES

by

Abubaker Sasi

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

2020

© 2020 Abubaker Sasi

All Rights Reserved. No Part of this document may be reproduced, stored or otherwise retained in a retrieval system or transmitted in any form, on any medium by any means without prior written permission of the author.

Memristor-Based Digital Systems Design and Architectures

by

Abubaker Sasi

APPROVED BY:

M. Bayoumi, External Examiner
University of Louisiana at Lafayette

A. Edrisy
Department of Mechanical, Automotive and Materials Engineering

R. Rashidzadeh
Department of Electrical and Computer Engineering

M. Khalid
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

23th of October, 2020

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. Amir soleimani Dr. Arash Ahamdi, and Prof. Majid Ahmadi. The design, simulation, implementation, analysis of results and writing the manuscript were performed by the author whereas Dr. Amir soleimani edited the manuscript, Dr. Arash Ahmadi and Prof. Majid Ahmadi supervised the research.

Chapter 3 of this thesis was co-authored with Dr. Amir soleimani, Dr. Arash Ahamdi, and Prof. Majid Ahmadi. In all cases, the design, simulation, implementation, analysis of results and writing the manuscript were performed by the author while Dr. Amir soleimani edited the manuscript, Dr. Arash Ahmadi and Prof. Majid Ahmadi supervised the research.

Chapter 4 of this thesis was co-authored with Dr. Arash Ahamdi, 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 Ahmadi was to oversee the research, provide feedback and give comments to improve the manuscript.

Chapter 5 of this thesis was co-authored with Dr. Arash Ahamdi, 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 Ahmadi was to oversee the research, provide feedback and give comments to improve the manuscript.

Chapter 6 of this thesis was co-authored with me (Abubaker Sasi), Dr. Arash Ahamdi, Prof. Majid Ahmadi and Prof. Mehrdad Saif. Creating a memristor-based standard logic cell library, synthesize the case study & proposed design and analysis of results was performed by me while the author Alammari described the design using Verilog HDL and edited the manuscript. Analysis the results at behavioural level, simulate and analyze in term of performance using test bench, and writing the manuscript performed by the author while Dr. Arash Ahamdi, Prof. Majid Ahmadi, and Prof. Mehrdad Saif provided supervision and edited 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 Publications

This thesis includes 5 original papers, 4 of them have been previously published/accepted in peer reviewed journals and conferences and 1 have been submitted, as follows:

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

Thesis Chapter	Publication Title	Publication status
Chapter 2	Sasi, Abubaker, Amirali Amirsoleimani, Arash Ahmadi, and Majid Ahmadi. "Hybrid memristor-CMOS based linear feedback shift register design." In 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 62-65. IEEE, 2017.	Published
Chapter 3	Sasi A, Amirsoleimani A, Ahmadi M, Ahmadi A. A Memristive TaOx-Based Median Filter Design for Image Processing Application. In 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD) 2018 Jul 2 (pp. 85-88). IEEE.	Published
Chapter 4	Sasi A, Ahmadi M, Ahmadi A, "Low Power Memristor-Based Shift Register Design," in ICECS 2020 The 27th IEEE International Conference on Electronics Circuits & Systems	Accepted
Chapter 5	Sasi A, Ahmadi M, Ahmadi A, "Characterizing a Standard Cell Library for Large Scale Design of Memristive based Signal Processing," in IET Circuits, Devices & Systems	Submitted
Chapter 6	Alammari, K., 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, Cham, 2020.	Published

III. General

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

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 owners to include such materials 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

Memristor is considered as a suitable alternative solution to resolve the scaling limitation of CMOS technology. In recent years, the use of memristors in circuits design has rapidly increased and attracted researchers interest. Advances have been made to both size and complexity of memristor designs. The development of CMOS transistors shows major concerns, such as, increased leakage power, reduced reliability, and high fabrication cost. These factors have affected chip manufacturing process and functionality severely. Therefore, the demand for new devices is increasing. Memristor, is considered as one of the key element in memory and information processing design due to its small size, long-term data storage, low power, and CMOS compatibility. The main objective in this research is to design memristor-based arithmetic circuits and to overcome some of the Memristor based logic design issues.

In this thesis , a fast, low area and low power hybrid CMOS memristor based digital circuit design were implemented . Small and large-scale memristor based digital circuits are implemented and provided a solutions for overcoming the memristor degradation and fanout challenges. As an example, a 4- bit LFSR has been implemented by using MRL scheme with 64 CMOS devices and 64 memristors. The proposed design is more efficient in terms of the area when compared with CMOS-based LFSR circuits. The simulation results proves the functionality of the design. This approach presents acceptable speed in comparison with CMOS-based design and it is faster than IMPLY-based memristive LFSR. The propped LFSR has 841 ps de-

lay. Furthermore, the proposed design has a significant power reduction of over 66% less than CMOS-based approach.

This thesis proposes implementation of memristive 2-D median filter and extends previously published works on memristive filter design to include this emerging technology characteristics in image processing. The proposed circuit was designed based on Pt/TaO_x/Ta redox-based device and Memristor Ratioed Logic (MRL). The proposed filter is designed in Cadence and the memristive median approved tested circuit is translated to Verilog-XL as a behavioral model. Different 512 × 512 pixels input images contain salt and pepper noise with various noise density ratios are applied to the proposed median filter and the design successfully has substantially removed the noise.

The implementation results in comparison with the conventional filters, it gives better Peak Signal to Noise Ratio (PSNR) and Mean Absolute Error (MAE) for different images with different noise density ratios while it saves more area as compared to CMOS-based design.

This dissertation proposes a comprehensive framework for design, mapping and synthesis of large-scale memristor-CMOS circuits. This framework provides a synthesis approach that can be applied to all memristor-based digital logic designs. In particular, it is a proposal for a characterization methodology of memristor-based logic cells to generate a standard cell library file for large-scale simulation. The proposed framework is implemented in the Cadence Virtuoso schematic-level environment and was verified with Verilog-XL, MATLAB, and the Electronic Design Automation (EDA) Synopses compiler after being translated to the behavioral level. The proposed method can be applied to implement any digital logic design. The framework is deployed for design of the memristor-based parallel 8-bit adder/subtractor and a 2-D memristive-based median filter.

I dedicate my dissertation work
to the soul of my father

Acknowledgment

First and foremost I express my sincere gratitude to my supervisor, Professor Majid Ahmadi, for his advice, guidance, patience and for the continuous support of my PhD research. His useful guidance, insightful comments, and considerable encouragements motivated me to complete this thesis. He always helped me in solving my research problems. His guidance helped me in all the time of research and writing of my thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. He has certainly shaped me as a researcher and has led me where I am now.

I record my deep sense of gratitude to my co-advisor, Doctor Arash Ahamdi, for his continuous suggestions in directing my research work.

In addition, I would like to thank the rest of my thesis committee: Doctor Rashid Rashidzadeh, Doctor Mohammed Khalid and Doctor Afsaneh Edrisy for their encouragement and insightful comments.

I would also like to thank all my friends and colleagues in the Research Centre for Integrated Microsystems (RCIM) in University of Windsor and friends in the University of Western Ontario whom have supported and believed in me.

Last but not the least, I would like to thank my family for their unconditional love, support and spiritual guidance.

Contents

Declaration of Co-Authorship / Previous Publication	v
Abstract	viii
Dedication	x
Acknowledgment	xi
List of Figures	xvii
List of Tables	xxi
List of Abbreviations	xxiii
Introduction	1
1.1 Background	1
1.1.1 Memristor	1
1.1.2 Why Memristor	2
1.2 Memristors Application and design challenges	4
1.2.1 Digital Logic Design using Memristor	4
1.2.2 Memristor Design Challenges	6
1.3 Proposed Solutions	7

1.4 Outline of dissertation and summary of contributions	8
References	11
Hybrid Memristor-CMOS Based Linear Feedback Shift Register Design	15
2.1 Introduction	15
2.2 Memristor device modelling	16
2.3 Memristor Ratioed Logic Design Technique	18
2.4 4-bit CMOS-Memristor based LFSR Design	18
2.5 Results and comparison	21
2.6 Conclusion	23
References	25
A Memristive TaO_x-based Median Filter Design for Image Processing Application.	27
3.1 Introduction	27
3.2 Memristor modeling and MRL logic	28
3.3 Memristive Median Filter Design	29
3.4 Experimental Results	33
3.5 Conclusion	35
References	37
Low Power Memristor-Based Shift Register Design	39
4.1 Introduction	39
4.2 Memristor Device and Logic Design Approach	40
4.2.1 Memristor Only Logic Design	40
4.2.2 Hybrid CMOS/Memristor Logic Design	41
4.3 Shift Register Designs	42
4.3.1 4-BIT memristor only-based shift register design	42
4.3.2 4-bit CMOS-memristor based shift registers design	45

4.4	Comparison and Evaluation	45
4.5	Conclusion	47
References		49
 Characterizing a Standard Cell Library for Large Scale Design of Mem-		
ristive based Signal Processing		51
5.1	Introduction	51
5.2	Memristor based-logic design	54
5.2.1	Memristor Modeling	54
5.2.2	Logic Design Approach	55
5.3	Synthesis methodology and implementation	56
5.4	Memristor based-logic design	61
5.4.1	Memristor Modeling	61
5.4.2	Case Study 1	62
5.4.3	Case Study 2	64
5.4.4	Simulation Results	71
5.5	Conclusion	78
References		80
 Hybrid Memristor-CMOS Based Finite Impulse Response Filter De-		
sign		84
6.1	Introduction	84
6.2	Design Approach and Device Modelling	85
6.2.1	Memristor Ratioed Logic Design	85
6.2.2	Device Modeling	86
6.3	Finite Impulse Response Filter Design	86
6.4	Case Study	87
6.5	Design Verification	90
6.6	Experimental Results	92
6.7	Conclusion	93

References	94
Conclusion and Future Work	96
7.1 Summary	96
7.2 Conclusion	98
7.3 Suggested Future Work	99
VITA AUCTORIS	100

List of Figures

1.1	The fourth missing fundamental circuit elements [1]	2
1.2	(a) IMPLY truth table. (b) Schematic of Memristor-based IMPLY logic gate.	4
1.3	(a) The MRL-based AND gate and the devices resistance evolution. (b) The MRL-based OR gate and the devices resistance evolution.	5
2.1	Memristor device I-V curve for Pt/TaOx/Ta VCM device with a bipolar triangular input voltage of 5 V. The circuit setup is displayed as the inset of the figure. Also, the Vin input voltage is illustrated as the top left inset in the figure	17
2.2	The hybrid CMOS-memristor based gates (a) OR gate. (b) AND gate.	18
2.3	(a) Transistor level schematic of MRL-based NOR gate. (b) Schematic of MRL-based NOR gate. (c) The simulations of the memristor NOR gate. A, B are inputs and OUT is logic output.	19
2.4	Proposed Four bits LFSR circuit.	20
2.5	Schematic of Memristor based D-Flipflop.	20
2.6	The circuit schematic of the memristor based D-Latch. (a) DLatch logic circuit. (b) D-Latch MRL-based circuit.	21
2.7	The simulations of the MRL-based D-Latch.	22
2.8	The simulations of the MRL-based D-Flipflop.	23

2.9	The simulations of the proposed LFSR.	24
3.1	The median filter operation and algorithm.	28
3.2	(a) The $I - V$ curve of Pt/TaO _x /Ta device by applying a triangular voltage in (-5 V, +5 V) domain and the redox-based memristor model [5] are displayed. (b) The MRL-based AND gate and the devices resistance evolution. (c) The MRL-based OR gate and the devices resistance evolution.	30
3.3	(a) 8-bit comparator circuit schematic. (b) 3-input 8-bit comparator circuit schematic. (c) The proposed median filter circuit schematic. .	31
3.4	The Cadence simulation result for the proposed memristive median filter.	31
3.5	(a) Modules organization of the proposed filter in Verilog-XL. (b) Design verification flowchart for proposed median filter.	33
3.6	Performance of designed filter for Lena, Bridge and Pepper 512×512 images contains salt and pepper noise with different noise density ratios. (a) Lena with 30 % noise. (a) Bridge with 20 % noise. (c) Pepper with 10 % noise. (d) Filtered Lena image. (e) Filtered Bridge image. (f) Filtered Pepper image.	35
3.7	Comparison of proposed design performance with other adaptive median filter designs. (a) PSNR comparison for Lena image. (b) PSNR comparison for Bridge image (c) MAE comparison for Lena image. (d) MAE comparison for Pepper image.	36
4.1	(a) Schematic of a two-input IMPLY gate. (b) Schematic of AND based MAGIC gate. (c) Schematic of an AND and OR logic gates. Polarity of memristor devices is the only difference between the two logic gates design.	40
4.2	Schematic of memristive Flip Flops.	41
4.3	(a) Schematic of proposed shift register. (b) Memristance switching of one memristor device	43

4.4	Hybrid CMOS-memristor based DFF design. Schematic of proposed D-Flipflop	44
4.5	The simulation results of the proposed MRL shift register	46
4.6	Analytical Monte Carlo Simulation of the power consumption for proposed designs (a) memristor based MRL shift register. (b)memristor only shift register	48
5.1	Memristor device I-V curve for ReRAM Pt/TaOx/Ta VCM device with a bipolar triangular input voltage of 5 V [14]	54
5.2	(a) MRL-based AND gate and its resistance progression. (b) MRL-based OR gate and its resistance progression.	55
5.3	Flow chart displaying design flow based on Synopsys EDA tool for proposed framework	58
5.4	The memristor-based adder/subtractor. (a) 8-bit Adder/subtractor schematic circuit. (b) Implemented Memristor-based 1-bit Adder/Subtractor	59
5.5	Simulation results of memristor-based 1bit adder and 1bit adder/subtractor. (a) MRL-based 1bit adder simulation results. (b) The simulations of the MRL-based 1bit adder/subtractor	60
5.6	Proposed memristor based median filter sorting circuit	65
5.7	Proposed memristive median simulation detection in 3×3 window	66
5.8	Schematic view of the implemented 4-bit memristor-based magnitude comparator.	67
5.9	Implemented memristor-based 4-bit magnitude comparator.	67
5.10	Flow chart of memristor based median filter design verification	69
5.11	Performance of designed filter for Boat, Camera man and Houses 512×512 images contains salt and pepper with different noise density ratios. (a)Filtered Boat image with 30 % noise. (b) Filtered Camera man image with 20 % noise. (c) Filtered Houses image with 10 % noise.	71

5.12	Hierarchical procedure for Process variation simulation . Simulation executes for 4-bit memristor-based magnitude comparator, 8-bit comparator, 3-input 8 bit comparator and finally median filter circuit. . .	75
5.13	Analytical Monte Carlo Simulation of the output voltage for proposal basic logic gates. (a) CMOS inverter. (b) Memristor based AND gate.(c) Memristor based NAND gate.(d) Memristor based OR gate. .	76
5.14	Monte Carlo output voltage variation of 1-bit adder/subtractor. (a) input A,B and Cin.(b)Optimized Monte Carlo simulation output voltage with buffer inserted. (c) Output voltage variation before buffers. . . .	77
6.1	(a) D-latch schematic circuit. (b) MRL based D-latch. (c) Implemented memristor based DFF.	88
6.2	Flow chart displaying design flow based on Synopsys EDA tool for proposed FIR filter.(a) netlist CMOS Base (b) netlist memristive Based, (C) design layout.	89
6.3	Simulation results in schematic level.(a) MRL-based D-Latch.(b) MRL-based DFF.	90
6.4	(a) Organization of the proposed FIR.(b) Verification flowchart for proposed.	91
6.5	(a) Input signals in frequency domain at 600 and 2000 Hz.(b) Filter output signal in frequency domain at 600 Hz.	92

List of Tables

2.1	PT/TAOX/TA memristor parameters for simulation	18
2.2	comparison between numbers of different cmos based lfsr implementation designs and proposed design	22
3.1	The simulation parameters for Pt/TaO _x /Ta device.	30
4.1	Voltage sequencing for computational steps of the Proposed shift Register. M_1 to M_8 indicate the memristance state of the design memristors, and Z means high impedance	45
4.2	Performance comparison of memristor only shift register and shift registers using cmos technology	48
4.3	The power consumption and delay of memristor based logic gates including the effects of process variation for 65nm technology	48
5.1	The simulation parameters for ReRAM and RRAM devices	53
5.2	list of logic cells involved in the design of adder/subtractor	63
5.3	Comparison of synthesis results of proposed adder/subtractor design	70
5.4	Comparison of experimental results of proposed memristive median filter	70
5.5	The power consumption and delay of memristor based logic gate including the effects of process variation for 180 nm and 65 nm technology	72

5.6	Performance parameters for different salt & pepper noise variance applied on boat image	73
5.7	Performance parameters for different salt & pepper noise variance applied on camera man image	73
5.8	Performance parameters for different salt & pepper noise variance applied on house image	74
6.1	RRAM memristor parameters for simulation	86
6.2	Characterized cells employed in the design	92
6.3	Synthesis results of memristor based DFF	93
6.4	Synthesis results of memristor based FIR filter.	93

List of Abbreviations

AFM	Amplitude Frequency Modulation Circuit
CAD	Computer Aided Design
CMOS	complementary metal-oxide semiconductor
DFF	D flip-flop
EDA	Electronic Design Automation
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GPF	Global Performance Factor
GPUs	Graphics Processing Devices
HDL	Hardware Description Language
HRS	High Resistance State
IMPLY	Material Implication Logic
LFSR	Linear Feedback Shift Register

LRS	Low Resistance State
MAE	Mean Absolute Error
MAGIC	Memristor Aided Logic
ME	Mean
MRL	Memristor Ratio Logic
PSNR	Peak Signal to Noise Ratio
SD	Standard Deviation
VTEAM	Voltage Threshold Adaptive Memristor
VLSI	Very Large Scale Integration

Introduction

In this Chapter first section presents a brief summary of Memristor-based digital systems design and their architectures and applications, introduces Memristor Ratioed Logic (MRL) Hybrid CMOS-Memristor Logic, one of the Memristor-based logic design. In the second Section discussion about the design challenges and issues in the field of Memristor-based logic design and a literature review for state-of-the-art solutions are presented. Finally, Section 1.3 presents the proposed solutions.

1.1 Background

1.1.1 Memristor

Memristor is an acronym for memory and resistor. In 1971, Leon Chua hypothesised that there should be the fourth fundamental passive device called Memristor based on the symmetrical relationship between current (I), voltage (V), charge (q) and flux (Φ) [2]. Memristor will complete the loop of the relationship between the three fundamental element as shown in seen in Fig. 1.1. The resistor maintains the relationship between the voltage and current ($f(v, i) = 0$), capacitor maintains the relation between charge and voltage ($f(v, q) = 0$), the current and the flux relationship is maintained by the inductor ($f(i, \Phi) = 0$ where $v = \frac{d\Phi}{dt}$ and $\Phi = Li$), the Memristor maintains a relationship between load and flux ($f(q, \Phi) = 0$). Memristor device is

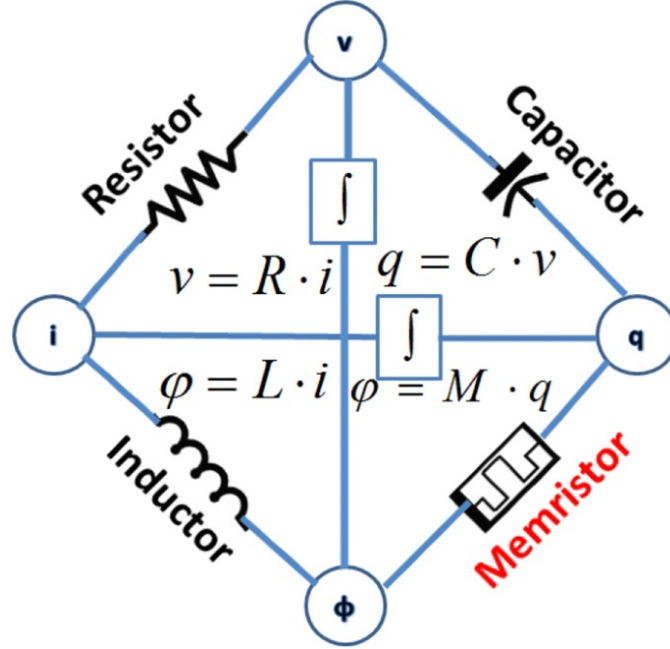


Figure 1.1: The fourth missing fundamental circuit elements [1]

passive component with two-terminals that recently attracted considerable interest from researchers because of its unique features. The device I/V curve exhibited a memory behavior and its variable resistance state makes it as a promising device for future memory and computing systems. The Memristor features such as low power, non-volatility, and high switching speed offer an alternative solution for traditional high speed memory. Because of its non-volatile nature, it has no leakage power which makes it an appropriate candidate for flash memories. In addition to its memory features, Memristors can also act as logic gate, and this will provide an architectural solution for future computing systems.

1.1.2 Why Memristor

The computational capabilities of digital computers-based on Metal oxide Semiconductors (CMOS) transistors, have improved exponentially in recent decades mainly by continuously shrinking the transistor dimension as Moore's Law predicts [3]. But since its invention in 1945 the assembly of hardware components into a computers,

called the Von Neumann architecture, has indeed changed. Mainly because of the considerable benefit of this type of architecture in terms of modularity of engineering design, which enables thousands of engineers to build a system without having to understand all components independently. However, the emergence of the internet of things placed crucial demands on energy consumption and processing speed for data-centered activities due to an exponential increase in data quantity. The disadvantages of conventional digital computers are therefore an growing issue [4]. Leakage currents on a system level become a problem as the channel length and the transistor gate dielectric thickness reach the limit of scaling. The continuous data transfer between data processing and memory units on the design level of Von Neumann dramatically reduces speed and energy performance. Therefore, the output gap between the two units contributes to significant delay in von Neumann architecture. Enhanced systems to boost programming skills and performance were introduced. For instance, multi-core Graphics Processing Devices (GPUs) and high-performance interconnections are quite successful attempts to increase parallelism in computers. [4, 5]. Therefore, in memory computing, the processing at the location where data is stored re-emerges as an alternative to traditional programming schemes. Memristor is an electronic device , also called resistance switch whose its internal resistance state conditions depend on the past of the voltage or current applied to it.

Although the conventional CMOS transistors scaling limitation was extended using FinFET architecture, FinFET is facing significant challenges due to different reasons such as doping damage, restriction in the logic chip design space, limitation of the electrostatics, and integration challenges [2, 3]. Therefore, substitutes to CMOS technology are in high demand. There are several alternative technologies, such as Double-Gate Tunnel FET [4], nanotube programmable devices [5], graphene transistors [6], and Memristor devices [7]. Among those technologies, Memristor devices are the most promising because of their great scaling ability, long-term data storage, low-power consumption, and CMOS compatibility [8, 9]. It is believed that these two terminal devices will play an essential role in the future fabrication of memory and information processing systems [10, 11].

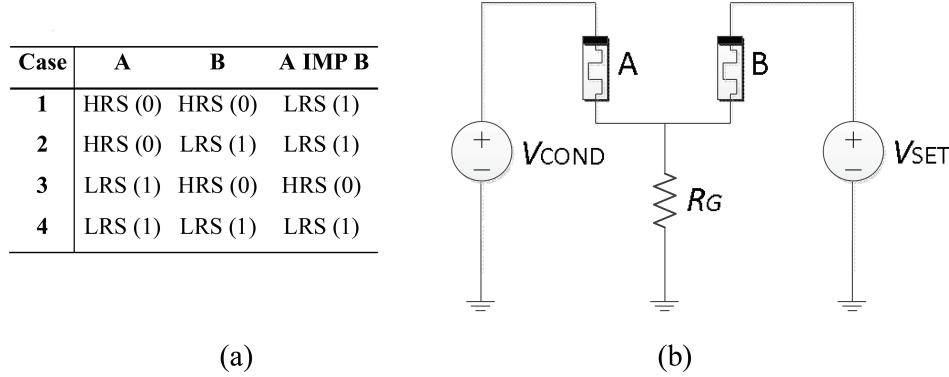


Figure 1.2: (a) IMPLY truth table. (b) Schematic of Memristor-based IMPLY logic gate.

1.2 Memristors Application and design challenges

Memristors attracted researchers interest in different circuit design fields because of the special features of nanoscale device. Memristor can be used in various applications, such as digital circuit application and analog circuit application domain. Memristors can implement logic which can be applied to in-memory computing schemes inside the memory. Unlike Von-Neumann architecture where the memory and processing units are separate and there is a constant need for vast quantities of interconnections between these units for constant communication, in-memory computing systems need not only one platform to implement logic and memory, but also no linkages are required. Here, we give a brief overview about digital circuits application to gain a better insight into the following thesis chapters.

1.2.1 Digital Logic Design using Memristor

Several Memristor-based logic methods [16, 17] have been presented in recent years and Material Implication (IMPLY) logic is one of them [17]. This logic is an only Memristor-based design method to implement logic gates. The IMPLY logic gate with Memristor device is illustrated in Fig. 1.2. This design approach suffers from significant latency due to the sequential process that requires a high number of computational steps to perform a one logic function. Another downside of IMPLY-based

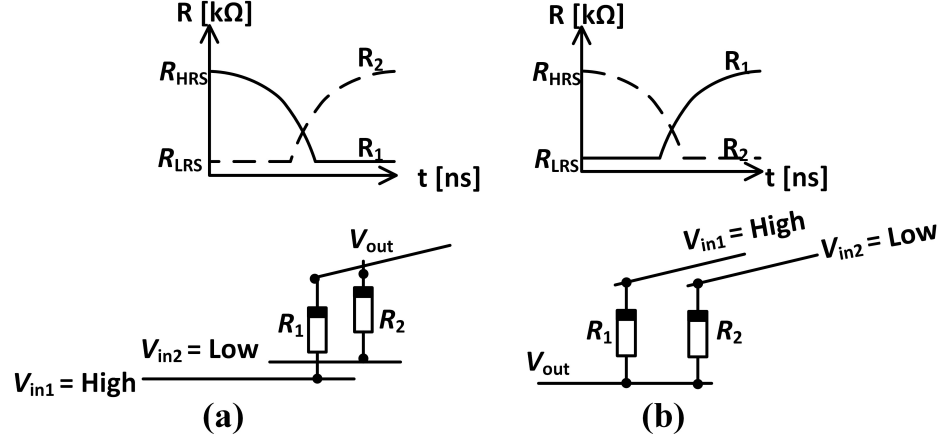


Figure 1.3: (a) The MRL-based AND gate and the devices resistance evolution. (b) The MRL-based OR gate and the devices resistance evolution.

logic is the circuitry necessary for READ / WRITE operation. In these design methods, computed operands are memorized and represented by the memristance states of the Memristors. All Boolean logic functions can be performed with a series of two Memristors utilizing the IMPLY method. Logic gates, such as OR, NOR, AND, and NAND gates can be implemented with IMPLY logic gates. An IMPLY Memristor-based logic gate consist of two Memristors and one resistor that is associated with the two Memristors from upper side and its bottom side associated with ground.

Memristive resistance based logic MAGIC logic [18] is another logic design technique. This design requires only one computational step after initialization to perform logic AND. This design method faces enormous challenges when connecting two or more logic gates. The MRL given in [19] is a hybrid Logic dependent CMOS-Memristor, as shown in Fig. 1.3. Unlike MAGIC based and IMPLY Logic, this logic is based on voltage, and the output is determined according to voltage level. MRL utilizes CMOS inverter with Memristor to obtain various logic gates. AND and OR gates can be implemented by pure memristive design. NOR and NAND gates are implemented by applying a CMOS-based inverter gate to the output of memristive OR and AND gates, respectively. In this logic, voltages are considered as logical states. Low voltage and high voltage denotes logic 0 and 1, respectively. Inputs as

voltages are applied to terminals V_{in1} and V_{in2} . This will change the resistance state of the two Memristors based on the voltage applied to them. The voltage on the common node of two Memristor device is determined by the voltage divider across both of the Memristors at the end of the logic operation. As shown in Fig. 1.3, the circuit schematics for hybrid CMOS Memristor based OR and AND are displayed.

1.2.2 Memristor Design Challenges

Using Memristor device for a variety of applications is promising, however, a wide range of issues also emerge. Some of these issues are unique. In fact, a robust Memristor based circuit construction is difficult because of the enhanced convergence between the analogue and digital signals at different voltages and power rates. In addition, signal degradation and fanout are the very challenging design issue that might face any Memristor-based logic circuits designer. Fanout takes effect when the current memorister output has to drive several Memristors. This issue degrades the performance of circuit as the load increases and leads to add more cycles which raises the over all energy consumption [20]. One of the key obstacles obstructing the discovery of Memristor-based crossbar circuit design is the occurrence of sneak-paths current [21]. A sneak path current is an undesired current which flows parallel to the selected memritor crossbar cells, thus affecting the reading considerably. These issue explained as an unexpected parallel resistance to a specific cell. Other challenge is the mapping of large scale circuits. The large scale simulations of the current Memristor models are complicated and in certain situations can not be performed in SPICE-based systems-level simulations. Nevertheless, the number of Memristor-based applications in today's circuit designs has been increasing exponentially. However, the design and mapping of large-scale Memristor-based applications is a challenging task due to the lack of comprehensive high-level design tools and simulation platforms. Currently, circuit design tools like SPICE, (H<) SPICE, ICAPS, and Cadence Virtuoso are not capable of providing designers with comprehensive design and simulation methodologies for Memristors [12].

1.3 Proposed Solutions

The most challenges that affect Memristor/CMOS based logic design circuits are fanout and degradation. The compatibility of Memristor device with CMOS increases logic density and provides an optimal solution to eliminate signal degradation in AND and OR memristive logic gates. The issue is resolved by adding a CMOS inverter to achieve the desired logic of NOR and NAND.

Regarding the challenges of mapping Memristor based design circuits, Xie et al. [13], presented a method for the automatic mapping of large-scale crossbar memristive-based Boolean logic circuits. This method involved the use of CMOS to control and drive the design. A programmable architecture for a large-scale neuromorphic-systems-based-memristive crossbar is proposed in [14]. The authors have proposed a framework for deep learning networks-based on the programming of spin electronics (spintronic devices). The framework mapping blocks consists of Memristors and transistors to mimic spindle behavior. In the paper presented in [15], the authors introduced a design methodology for Memristor crossbar architecture-based image compression. The author primary objective is to perform computational operations in a memristor crossbar and store the row-transformed image data in the same crossbar memory array. Therefore, the overall area, timing, and power of this architecture were reduced. The aforementioned methods were implemented based on memristive crossbars. Such design techniques presented real challenges, including those related to sneak path current and signal degradation. Moreover, memristive crossbar circuits require separate circuits to control input signals.

Material implication logic is also implemented to map Memristor-based Boolean logic [16, 17]. In these works, implication logic was employed to reduce the number of Memristor devices and operating cycles. However, the use of such methods is limited only to Boolean function implementation. Moreover, Memristor-based crossbar and implication logic design methods are not synthesizable using Computer-Aided Design (CAD) synthesis tools [18]. In addition, above mentioned design methods require sequential computational steps to achieve a logic gate operation. In such process, execution of one logic computation requires more than one clock cycle.

To solve the issue of sneak paths, the technique of gating Memristor crossbar with CMOS transistor is proposed, but has its own issues. The array width is limited by the scale of the gating transistors instead of the tiny Memristor cells. Therefore, the smallest practicable transistor will be used to avoid reducing the array density of the Memristor circuit considerably. In this case small transistors size will increase current leaking. Therefore the gating strategy decreases the severity of the issue of the sneak-paths, but does not totally eradicate it. In the other side, the usage of bigger transistors will reduce the leakage while design density decreases dramatically. In comparison, the manufacture of high density Memristor/transistor arrays does not yet look really promising. Another well-known technique for cell gating is the use of diodes but such a system is not ideal for bipolar Memristor devices. Many other methods have been suggested in an effort to fix the issue of sneak-paths without the use of gating technique. Nonetheless, these approaches minimise the intensity of the issue to varying levels, but do not have a complete healing approach [21].

1.4 Outline of dissertation and summary of contributions

The objectives of this dissertation are to examine Memristor-based digital systems design and architectures. Several Memristor device models were utilized to design different efficient Memristor based logic circuits and addressed different Memristor based digital design issues.

- Chapter 2 proposes a fast, low area and low power hybrid CMOS/Memristor based LFSR design. As an example a 4-bit Linear Feedback Shift Register (LFSR) has been implemented by using MRL scheme with 64 CMOS devices and 64 Memristors. The proposed design is more efficient in terms of the area when compared with CMOS-based LFSR circuits. In this chapter, a Pt/TaOx/Ta Memristor device model is utilized for simulations in which the Pt/TaOx interface has been considered as a Schottky diode barrier. This physical oriented model provides the realistic switching kinetics of the device for simulations. The

proposed LFSR design is evaluated using circuit level Cadence Spectre simulator. The proposed design consumes less area in comparison with CMOS-based designs. Although this design needs more area in comparison with IMPLY-based memristive LFSR, it is considerably faster.

- Chapter 3 proposes the implementation of memristive 2-D median filter and extends previously published works on memristive filter design to include this emerging technology characteristics in image processing. The proposed circuit was designed based on Pt/TaOx/Ta redox-based device and Memristor Ratioed Logic (MRL). The proposed filter is designed in Cadence and the memristive median approved tested circuit is translated to Verilog-XL as a behavioral model. Different $512 \times 512 pixels$ input images contain salt and pepper noise with various noise density ratios are applied to the proposed median filter and the design successfully has substantially removed the noise
- Chapter 4 proposes a fast and efficient area Memristor-only-based shift register, as well as a hybrid CMOS/Memristor-based shift register are proposed. Specifically, a 4-bit shift register with only 8 Memristor devices and a hybrid CMOS /Memristor with 64 Memristor devices and 64 CMOS transistors were implemented and simulated using Cadence Virtuoso. The simulation results demonstrate the design's efficient functionality. Compared to the implementation of a CMOS-Memristor based shift register, the implementation of the proposed design is more efficient when concerning area and speed with respect to the implementation of the Memristor IMPLY memristive shift register.
- Chapter 5 presents a comprehensive framework for the design and synthesis of large-scale Memristor-CMOS circuits. This framework provides a synthesis approach that can be applied to all Memristor-based digital logic designs. In particular, it is a proposal for a characterization methodology of Memristor-based logic cells to generate a standard cell library file for large-scale simulation. The proposed architecture is based on RRAM and ReRAM redox-based devices and the MRL design approach. The proposed framework is implemented in the Ca-

dence Virtuoso schematic-level environment and was verified with Verilog-XL, MATLAB, and the Electronic Design Automation (EDA) Synopses compiler after being translated to the behavioral level.

- Chapter 6 proposes a compact, low power design of 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 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 7 presents thesis conclusion and future Work.

References

- [1] M. Khalid, “Review on various memristor models, characteristics, potential applications, and future works,” *Transactions on Electrical and Electronic Materials*, pp. 1–10, 2019.
- [2] L. Chua, “Memristor-the missing circuit element,” *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.
- [3] G. E. Moore, “Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff.” *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, 2006.
- [4] Q. Xia and J. J. Yang, “Memristive crossbar arrays for brain-inspired computing,” *Nature materials*, vol. 18, no. 4, pp. 309–323, 2019.
- [5] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, “In-datacenter performance analysis of a tensor processing unit,” in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017, pp. 1–12.
- [6] S. Borkar, “Design perspectives on 22nm cmos and beyond,” in *2009 46th ACM/IEEE Design Automation Conference*, July 2009, pp. 93–94.
- [7] 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)*, July 2016, pp. 43–48.
- [8] K. Boucart and A. M. Ionescu, “Double-gate tunnel fet with high- κ gate dielectric,” *IEEE Transactions on Electron Devices*, vol. 54, no. 7, pp. 1725–1733, July 2007.
- [9] G. Agnus, W. Zhao, V. Derycke, A. Filoramo, Y. Lhuillier, S. Lenfant, D. Vuillaume, C. Gamrat, and J.-P. Bourgoin, “Two-terminal carbon nanotube pro-

- grammable devices for adaptive architectures,” *Advanced Materials*, vol. 22, no. 6, pp. 702–706, 2010.
- [10] J. Kedzierski, P. Hsu, P. Healey, P. W. Wyatt, C. L. Keast, M. Sprinkle, C. Berger, and W. A. de Heer, “Epitaxial graphene transistors on sic substrates,” *IEEE Transactions on Electron Devices*, vol. 55, no. 8, pp. 2078–2085, Aug 2008.
- [11] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *nature*, vol. 453, no. 7191, p. 80, 2008.
- [12] Y. Zhang, Y. Li, X. Wang, and E. G. Friedman, “Synaptic characteristics of ag/aginsbte/ta-based memristor for pattern recognition applications,” *IEEE Transactions on Electron Devices*, vol. 64, no. 4, pp. 1806–1811, April 2017.
- [13] N. Gergel-Hackett, B. Hamadani, B. Dunlap, J. Suehle, C. Richter, C. Hacker, and D. Gundlach, “A flexible solution-processed memristor,” *IEEE Electron Device Letters*, vol. 30, no. 7, pp. 706–708, July 2009.
- [14] K. Eshraghian, K. Cho, O. Kavehei, S. Kang, D. Abbott, and S. S. Kang, “Memristor mos content addressable memory (mcam): Hybrid architecture for future high performance search engines,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 8, pp. 1407–1417, Aug 2011.
- [15] H. Li and Y. Chen, “An overview of non-volatile memory technology and the implication for tools and architectures,” in *2009 Design, Automation Test in Europe Conference Exhibition*, April 2009, pp. 731–736.
- [16] K. Cho, S.-J. Lee, and K. Eshraghian, “Memristor-cmos logic and digital computational components,” *Microelectronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.
- [17] M. Teimoory, A. Amirsoleimani, A. Ahmadi, S. Alirezade, S. Salimpour, and M. Ahmadi, “Memristor-based linear feedback shift register based on material implication logic,” in *Circuit Theory and Design (ECCTD), 2015 European Conference on*. IEEE, 2015, pp. 1–4.
- [18] 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, 2014.
- [19] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “Mrl-memristor ratioed logic,” in *Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on*. IEEE, 2012, pp. 1–6.

-
- [20] A. P. James, “Memristor threshold logic: An overview to challenges and applications,” *arXiv preprint arXiv:1612.01711*, 2016.
- [21] M. A. Zidan, A. M. Eltawil, F. Kurdahi, H. A. H. Fahmy, and K. N. Salama, “Memristor multiport readout: A closed-form solution for sneak paths,” *IEEE Transactions on Nanotechnology*, vol. 13, no. 2, pp. 274–282, 2014.
- [22] A. Sarmiento-Reyes, J. J. Len, L. Hernandez-Martinez, and H. Vazquez-Leal, “A cad-oriented simulation methodology for memristive circuits,” in *2016 IEEE 7th Latin American Symposium on Circuits Systems (LASCAS)*, Feb 2016, pp. 103–106.
- [23] 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, 2018.
- [24] L. Guckert and E. E. Swartzlander, “Mad gates-memristor logic design using driver circuitry,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 2, pp. 171–175, 2017.
- [25] L. Xie, H. A. D. Nguyen, M. Taouil, S. Hamdioui, and K. Bertels, “A mapping methodology of boolean logic circuits on memristor crossbar,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 2, pp. 311–323, Feb 2018.
- [26] S. G. Ramasubramanian, R. Venkatesan, M. Sharad, K. Roy, and A. Raghunathan, “Spindle: Spintronic deep learning engine for large-scale neuromorphic computing,” in *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Aug 2014, pp. 15–20.
- [27] Y. Halawani, B. Mohammad, M. Al-Qutayri, and S. F. Al-Sarawi, “Memristor-based hardware accelerator for image compression,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 12, pp. 2749–2758, Dec 2018.
- [28] 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.
- [29] F. S. Marranghello, V. Callegaro, M. G. A. Martins, A. I. Reis, and R. P. Ribas, “Improved logic synthesis for memristive stateful logic using multi-memristor implication,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 181–184.
-

- [30] 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)*, July 2016, pp. 43–48.

Hybrid Memristor-CMOS Based Linear Feedback Shift Register Design

2.1 Introduction

The Complementary Metal-Oxide Semiconductor (CMOS) industry has become closer to its limitation as referred by Moore's law, and this is due to the slow growth of physical and technological scale of the CMOS design [1]. The newly developed memristor guarantees a decent option for replacing traditional CMOS design paradigm because of its great ability of scaling, long-term data storage, low power and CMOS compatibility property. Applying voltage over memristor results in altering the resistance of the device and it saves its last resistance state as the applied voltage is removed. Therefore, by applying a voltage, it can reproduce logic "1" or "0" which correspond to High Resistance State (HRS) and Low Resistance State (LRS) of the device, respectively.

Several memristor-based logic methods [1, 2, 3, 4, 5, 6] have been presented in recent years and Material Implication (IMPLY) logic is one of them [4, 5, 6]. This logic is a pure memristor-based method to implement logic gates. This approach suffers from sequential process that requires a high number of computational steps to

perform a logic function. Another drawback of the IMPLY-based logic is the required circuitry for READ/WRITE operation.

MAGIC logic [2] design technique is another memristive resistance-based logic. This design requires only one computational step to perform logic after initialization procedure. However, this design technique faces enormous challenges when connecting two or more logic designs together.

The Mmemristor Rratio Logic (MRL) presented in [3] is a hybrid CMOS-memristor based logic. Unlike MAGIC- and IMPLYbased logic, this logic is a voltage-based logic and output is determined based on the level of the voltage. This logic is not a sequential logic and the integration of memristor with CMOS results in less physical area consumption in comparison with conventional CMOS logic.

In this chapter, a 4-bit linear feedback shift register (LFSR) based on hybrid CMOS-memristor logic is designed and simulated. Comparing the designed 4-bit memristor-based LFSR with CMOS-based LFSR, the proposed design requires less area and provides acceptable speed. Furthermore, the proposed LFSR consumes less power in comparison with CMOS-based design. Rest of this chapter outlined as follows. Memristor device modeling and characterization are explained in section 2.2. The MRL logic fundamentals are presented in section 2.3. A 4-bit memristor-based LFSR is designed using MRL method in section 2.4. Finally, conclusion is presented in section 2.6.

2.2 Memristor device modelling

Redox-based resistive switches can be utilised in a wide application area starting from a single transistor to neuromorphic circuits. Profoundly, precise and prescient models [7, 8] of resistive switching devices are essential to validate the design in the future for memory and logic designs. In this chapter, a Pt/TaOx/Ta device model [8] is utilized for simulations in which the Pt/TaOx interface has been considered as a Schottky diode barrier. This physical oriented model provides the realistic switching kinetics of the device for simulations. The concentration of the oxygen vacancy is considered

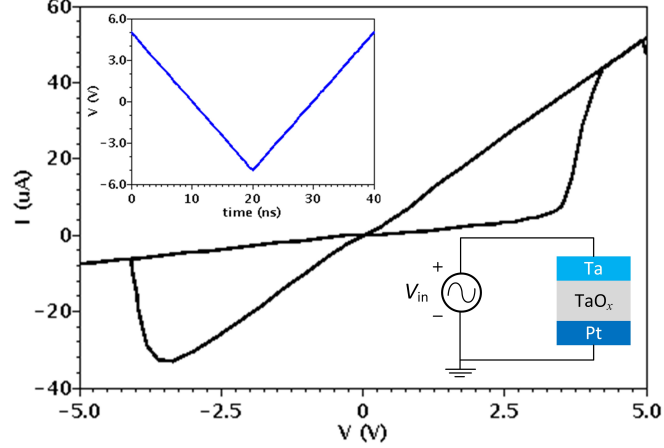


Figure 2.1: Memristor device I-V curve for Pt/TaOx/Ta VCM device with a bipolar triangular input voltage of 5 V. The circuit setup is displayed as the inset of the figure. Also, the V_{in} input voltage is illustrated as the top left inset in the figure

as a device state variable (N) which affects the electronic resistance of the device. In addition, ionic current and areal current are included in this model [8]. The average current passing through the device is represented as a sum of Schottky current and areal leakage current and it is given as follows,

$$I_{total} = I_{Schottky} + I_{Area} \quad (2.1)$$

The areal current is independent from device state variable while the ionic current of Pt/TaOx/Ta memristor device has a direct impact on the device state variable (N) which is determined by [8]. Here Z_{vo} , A , L_{disc} and E are the amount of charged oxygen vacancy, the device cross-sectional area of plug/disc, the length of the disc and charge, respectively. A single memristor device is simulated using Cadence Virtuoso. The resulting output of uniformity I-V curve for the applied signal is displayed in Fig.2.1. The applied parameters for the model are defined in Table 2.1. Here, polarity of the memristor is determined by a thick black bar. As the current flows through the device from the bar side, the resistance of the device decreases. In case, the current enters from the non-bar side the device resistance increases.

$$\frac{dN}{dt} = \frac{1}{E Z_{vo} A L_{disc}} \cdot I_{ionic(t)} \quad (2.2)$$

Table 2.1: PT/TAOX/TA memristor parameters for simulation

Parameters	N_{min} (m^{-2})	N_{max} (m^{-2})	N_{init} (m^{-2})	L_{Disc} (nm)	C_{31} (pAm/V)	A (nm^2)
Value	0.308	5	5	6	3.14	4

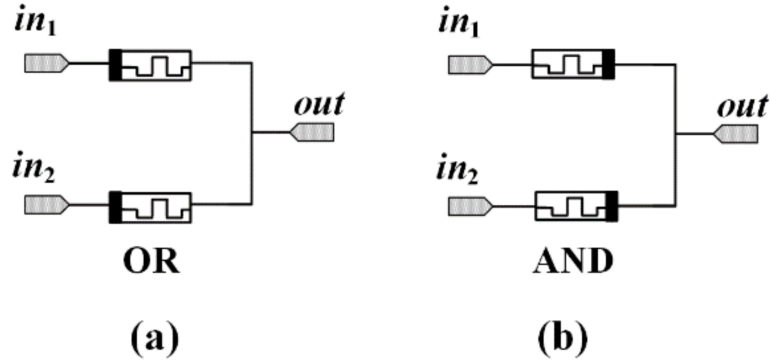


Figure 2.2: The hybrid CMOS-memristor based gates (a) OR gate. (b) AND gate.

2.3 Memristor Ratioed Logic Design Technique

MRL utilizes CMOS technology with memristor to obtain various logic gates. AND and OR gates can be implemented by pure memristive design. NOR and NAND gates are implemented by adding a CMOS-based inverter gate to the output of memristive OR and AND gates, respectively. In this logic method, voltages are considered as logical states. Low voltage and high voltage represented as logic 0 and 1, respectively. Voltages are applied to terminals in_1 and in_2 as illustrated in Fig2.2. Based on the voltage applied the resistance state of the two memristors will change. As it can be seen in Fig. 2.2, the circuit schematics for hybrid CMOS memristor based OR and AND are displayed. The NOR gate has been implemented in Fig. 2.3

2.4 4-bit CMOS-Memristor based LFSR Design

LFSR is considered as a very important unit that is used extensively in BIST designs, security designs, error correction designs and communication designs. It is a type of

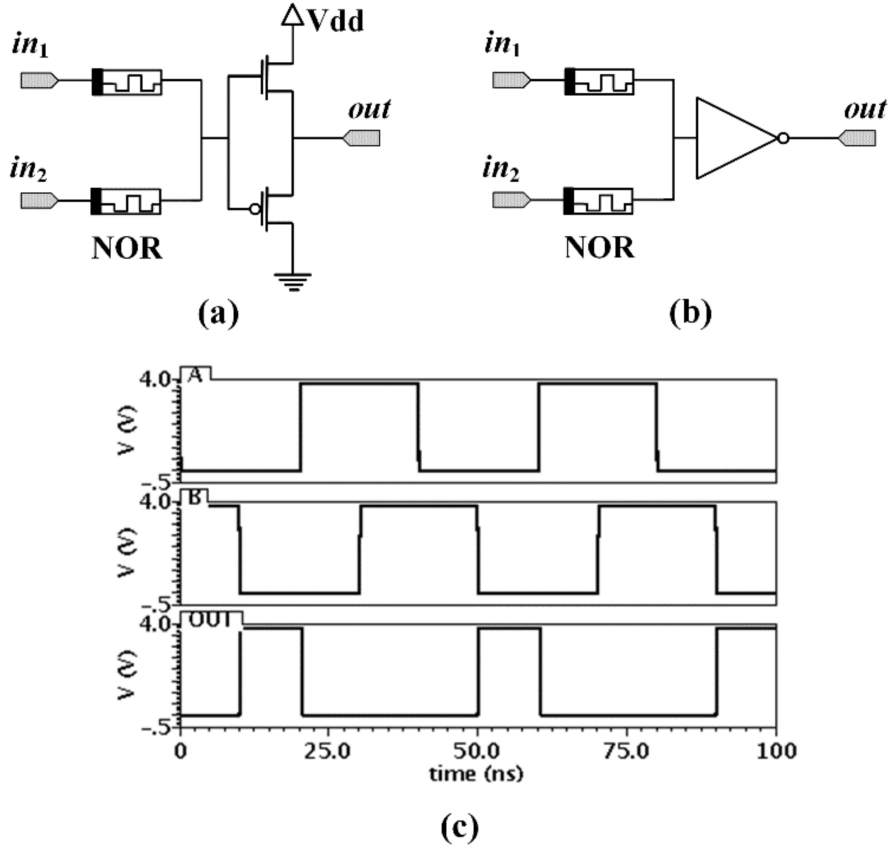


Figure 2.3: (a) Transistor level schematic of MRL-based NOR gate. (b) Schematic of MRL-based NOR gate. (c) The simulations of the memristor NOR gate. A, B are inputs and OUT is logic output.

shift register which is driven by the Exclusive-Or (XOR) of overall shifted register bits value. Several LFSR circuits was implemented based on the conventional CMOS design [9, 10, 11] while there are not many reported using a promising memristor technology [4]. The proposed LFSR is implemented based on four serially connected D-flip flops and a feedback XOR gate. The circuit schematic for the proposed LFSR is displayed in Fig. 2.4. Therefore, first step in designing hybrid CMOS-memristor based LFSR is to design a D-flip flop. In [2] IMPLY-based D-flip flop is designed with four memristors. In this method the logic operation requires 8 computational steps without considering initialization step. Here, a circuit schematic in Fig. 2.5 is

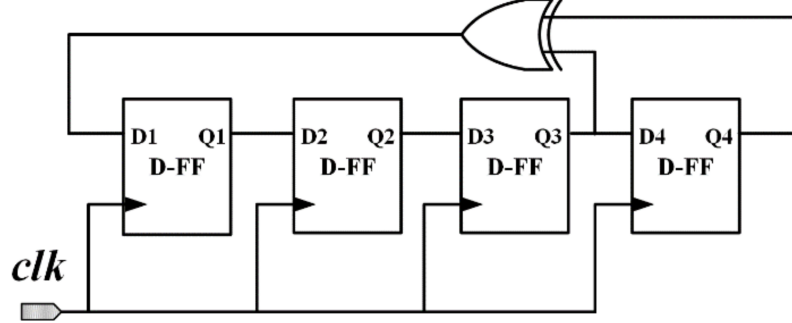


Figure 2.4: Proposed Four bits LFSR circuit.

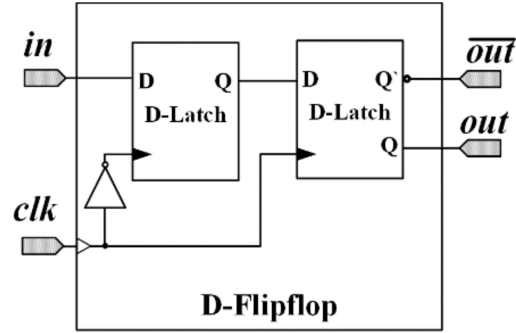


Figure 2.5: Schematic of Memristor based D-Flipflop.

considered for the implementation of the D-flipflop. The D-flipflop comprises of two serially connected D-latches with a master and slave structure. This structure is an edge-triggered which means by arriving a rising edge, the input data is stored in the master D-latch and when the falling edge arrives the data propagates to slave D-latch. The D-latch circuit schematic is displayed in Fig. 2.6 (a). It consists of two AND and two OR gates. The equivalent hybrid CMOS-memristor AND and OR logic gates are utilized and connected based on the proposed schematic as it can be seen in Fig. 2.6 (b). It has two input terminals "in" and "clk" as input. Also, it has two outputs "out" and $\overline{\text{out}}$ as output and inverted output, respectively. The functionality of the proposed D-latch has been proven by the simulation results in Fig. 2.7. This D-latch requires 8 memristors and 6 CMOS devices. Therefore, the D-flipflop comprises of 16 memristors and 16 CMOS devices. The simulation results for the proposed D-flipflop is displayed in Fig. 2.8 and it proves right functionality of the proposed design.

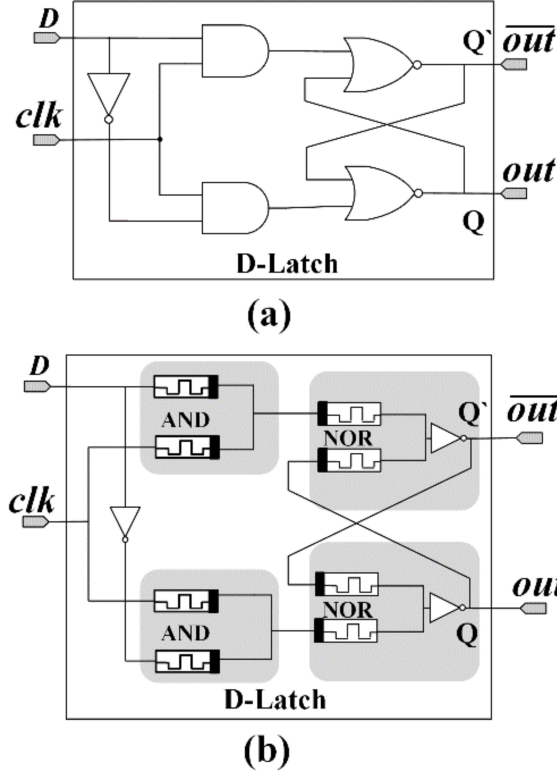


Figure 2.6: The circuit schematic of the memristor based D-Latch. (a) DLatch logic circuit. (b) D-Latch MRL-based circuit.

2.5 Results and comparison

The proposed LFSR design is evaluated using circuit level Cadence Spectre simulator. The parameters in Table 2.1 are considered for the memristor device Pt/TaOx/Ta and TSMC 65nm technology is utilized for CMOS devices. Simulation results of proposed LFSR are shown in Fig. 2.8. As illustrated in Fig. 2.8, all D-flipflops are shifted the logic outputs and the XOR adjusting the feedback to avoid the case of all Zeros. The hybrid CMOS-memristor based D-flipflop needs to have an initial pulse to start functioning. When DFFs initialized to logic 1 and the clock arrives, the LFSR generates pseudo-random values. The proposed LFSR provides different patterns when the outputs of the flip-flops are loaded with initial value. The seed value with all 0s should be avoided since it causes the LFSR makes all 0 patterns. As the LFSR

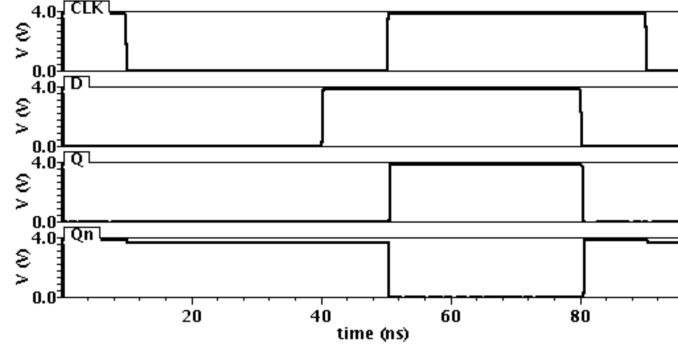


Figure 2.7: The simulations of the MRL-based D-Latch.

Table 2.2: comparison between numbers of different cmos based lfsr implementation designs and proposed design

Design	CMOS DLatch [10]	CMOS DLatch [11]	CMOS DLatch [11]	CMOS DLatch [11]	This Work
CMOS	318	154	81	121	64
Memristor	0	0	0	0	64
Power (μW)	9360	532.34	188	359	94.15

is clocked, it produces different patterns which act as a good pseudorandom pattern generator. As can be seen in Fig. 2.8, different numbers are generated by the proposed LFSR. The memristor occupies a small area in size of a via and it is compatible with CMOS technology. It can be fabricated over the CMOS layer. The proposed LFSR requires 64 memristors and 64 CMOS. The comparison in terms of number of devices utilized in the design is presented in Table 2.2. The proposed design consumes less area in comparison with CMOS-based designs. Although this design needs more area in comparison with IMPLY-based memristive LFSR [4], it is considerably faster. The IMPLY-based LFSR requires 55 computational steps due to the sequential nature of memristor-based IMPLY logic. The proposed hybrid CMOS-memristor based design

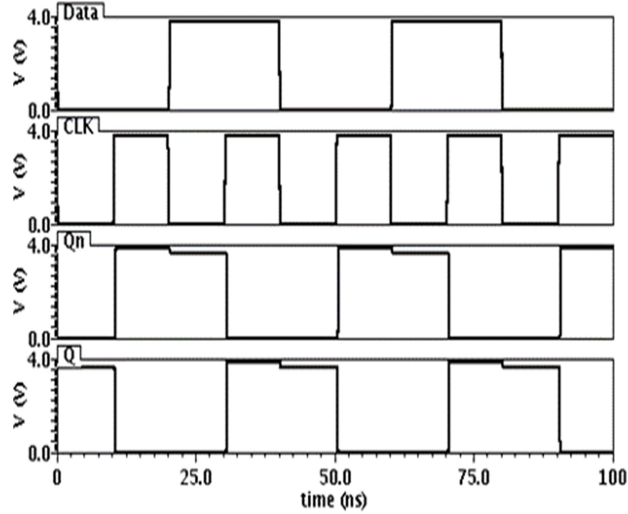


Figure 2.8: The simulations of the MRL-based D-Flipflop.

has a delay of 841 ps which is comparable with the conventional CMOSbased LFSR [10] with 813 ps delay. The delta cursor tool of Cadence Spectre has measured the delay. The delay is the average for rise time and fall time delay. The rise time delay is the time difference when the input reaches 10% of its swing and the output reaches to 90% of its swing. The fall time delay is the time difference between 90% of swing for input and 10% of swing for the output. Moreover, the proposed design has reduced the power consumption by 66.5% comparing with the lowest power consumed design as shown in Table 2.2. The proposed LFSR consumes only $94.15 \mu\text{W}$. The total power consumed by the proposed design measured by adding passive device to the design. The total power consumed calculated by measuring the currents passed through the DC sources and integrated them over time then divided it by clock period and multiplied it with the source voltage. This circuit consumes a considerable low power while provide acceptable speed and low area and implementation cost.

2.6 Conclusion

In this chapter, a hybrid CMOS-memristor based LFSR is implemented by the MRL design method. In the first step, a D-latch is designed using 8 memristors and 6 CMOS

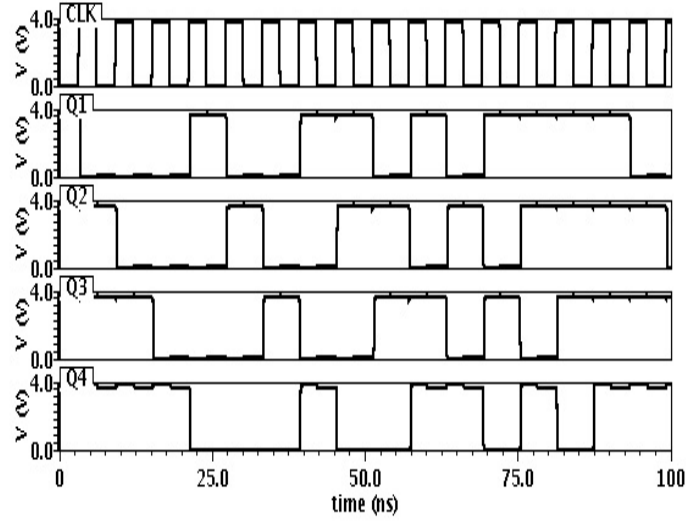


Figure 2.9: The simulations of the proposed LFSR.

devices. A D-flipflop is implemented with 16 CMOS and 16 memristors subsequently. The proposed LFSR requires 64 CMOS and 64 memristors. The functionality of the proposed LFSR was confirmed by simulation. This design shows how important the significant reduction of power consumption comparing with similar pure CMOS designs. Moreover, the proposed design uses small numbers of transistors comparing with conventional CMOS-based LFSR, which means it will occupy small layout area. In addition, proposed design has a decent computational timing, comparing with previous IMPLY-based memristive LFSR and conventional CMOS based one.

References

- [1] K. Cho, S.-J. Lee, and K. Eshraghian, “Memristor-cmos logic and digital computational components,” *Microelectronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.
- [2] 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, 2014.
- [3] 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, 2012, pp. 1–6.
- [4] M. Teimoory, A. Amirsoleimani, A. Ahmadi, S. Alirezaee, S. Salimpour, and M. Ahmadi, “Memristor-based linear feedback shift register based on material implication logic,” in *2015 European Conference on Circuit Theory and Design (ECCTD)*. IEEE, 2015, pp. 1–4.
- [5] A. Amirsoleimani, M. Ahmadi, M. Teimoory, and A. Ahmadi, “Memristor-based 4: 2 compressor cells design,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 1242–1245.
- [6] 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, 2014, pp. 562–565.
- [7] A. Amirsoleimani, J. Shamsi, M. Ahmadi, A. Ahmadi, S. Alirezaee, K. Mohammadi, M. A. Karami, C. Yakopcic, O. Kavehei, and S. Al-Sarawi, “Accurate charge transport model for nanoionic memristive devices,” *Microelectronics journal*, vol. 65, pp. 49–57, 2017.

- [8] A. Siemon, S. Menzel, A. Marchewka, Y. Nishi, R. Waser, and E. Linn, “Simulation of tao x-based complementary resistive switches by a physics-based memristive model,” in *2014 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 2014, pp. 1420–1423.
- [9] R. S. Katti, Xiaoyu Ruan, and H. Khattri, “Multiple-output low-power linear feedback shift register design,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 7, pp. 1487–1495, 2006.
- [10] A. Tyagi, N. Pandey, and K. Gupta, “Pfscl based linear feedback shift register,” in *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*. IEEE, 2016, pp. 580–585.
- [11] S. Agarwal, N. Pandey, B. Choudhary, and K. Gupta, “Design of mcml-based lfsr for low power and mixed signal applications,” in *2015 Annual IEEE India Conference (INDICON)*. IEEE, 2015, pp. 1–6.

A Memristive TaO_x -based Median Filter Design for Image Processing Application.

3.1 Introduction

Recently discovered memristor devices [1] are believed to be a game changer in memory and information processing industry. Within an ever-increasing variety of memristive systems, filtering operation is one of the potential applications. There are few works focused on utilizing memristor devices in filters [2, 3]. Median filters (MF) are one of the most well-known non-linear filters in image processing applications to remove salt and pepper noise in images. Building an efficient median filter with high performance and low cost is valuable for image processing applications.

In this chapter, we utilized Pt/ TaO_x /Ta memristor to implement a 2-D median filter with Memristor Ratioed Logic (MRL) [4]. Unlike the previous memristive filters [2, 3], the proposed filter is a two dimensional filter. It is a simple rank selection filter that removes impulse noise by changing the luminance value of the center pixel of the filtering window with the median of the luminance values of the pixels in its neighborhood. The proposed memristive median filter is tested for different images and it substantially removes salt and pepper noise. The performance of the proposed

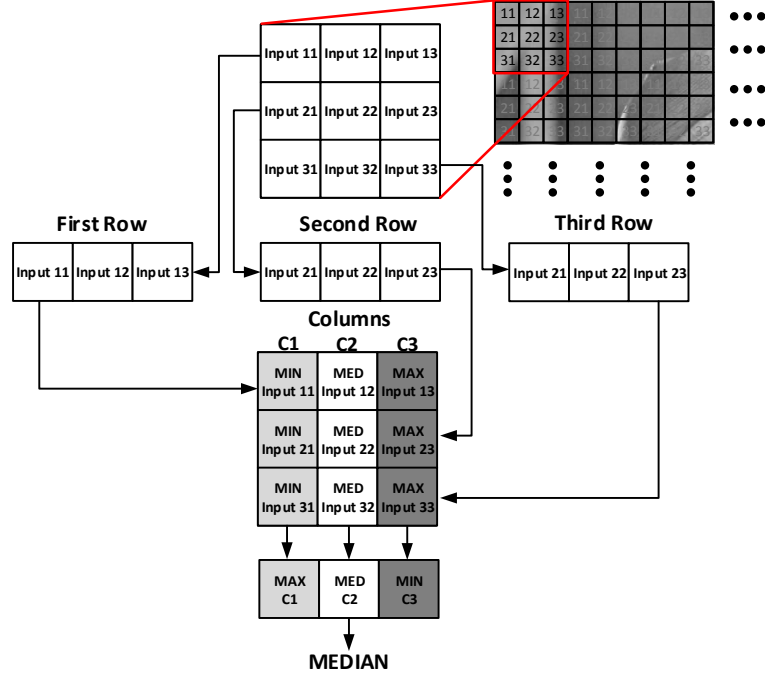


Figure 3.1: The median filter operation and algorithm.

filter promises an efficient median filter with lower hardware implementation cost by using a memristive emerging technology.

The rest of this chapter is structured as follows. A brief introduction about memristive modeling and MRL logic is given in Section II. In Section III, the memristive 2-D median filter design is discussed. The simulation results and comparison with previous designs are given in Section IV. Finally in section V, conclusions and remarks are provided.

3.2 Memristor modeling and MRL logic

For simulation, Pt/ TaO_x /Ta device Verilog-A model presented in [5] is utilized and it provides realistic switching behavior of the device. The $I-V$ curve of the Pt/ TaO_x /Ta device is displayed in Fig. 3.2(a) based on the parameters in Table 3.1. Several logic designs were proposed by utilizing memristor devices [4, 6, 7]. MRL is a hybrid CMOS/memristor-based logic like mirrored logic [6] while unlike the mentioned

memristive logic this logic is voltage-based. The compatibility of memristor device with CMOS increases logic density and provides an optimal solution to eliminates the signal degradation in AND and OR memristive logic implementation. The issue is resolved by adding a CMOS inverter to achieve the desired logic of NOR and NAND [4]. Here, the MRL logic is exploited to implement the proposed circuit design and two MRL basic AND and OR gates are displayed in Fig. 3.2(b) and Fig. 3.2(c), respectively.

3.3 Memristive Median Filter Design

Median filtering in two dimension corresponds to replacing the value of each element by the median of its neighborhood. The neighborhood (S_{xy}) is defined as eight surrounding elements. Thus, mathematically for an image $g(x, y)$ the median filtering operation is given as,

$$S_{xy} = \text{median}\{g(s, t)\} \quad (s, t) \in S_{xy} \quad (3.1)$$

The performance of the median filtering is considerably dependent on the sorting method, which is used to find the median value of the neighborhood pixels. As can be seen in Fig. 3.1, the proposed memristor-based median filter circuit finds the median pixel value in 3×3 window in the image. The basic block of the proposed filter is a memristor-based 4-bit magnitude comparator and it is implemented based on memristor-based MRL logic scheme. The outputs of the two 4-bits comparator are compared again with 2-bit comparator to find the largest pixel value between the two inputs. The output value of the comparator is connected to the selector of the first multiplexer to decide which pixel is maximum, and the inverted output value is connected to the second multiplexer to select the minimum pixel. The proposed median filter consists of seven comparator units and each unit is a 3-input 8-bit memristor-based comparator. Each of these units consists of three memristor-based 2-input 8-bit magnitude comparator circuit. As can be seen in Fig. 3.3, each of these 2 input 8-bit comparators implemented with two 4-bit memristor based magnitude comparator to compare between two pixels (eight bits for each input). The filter implements spatial

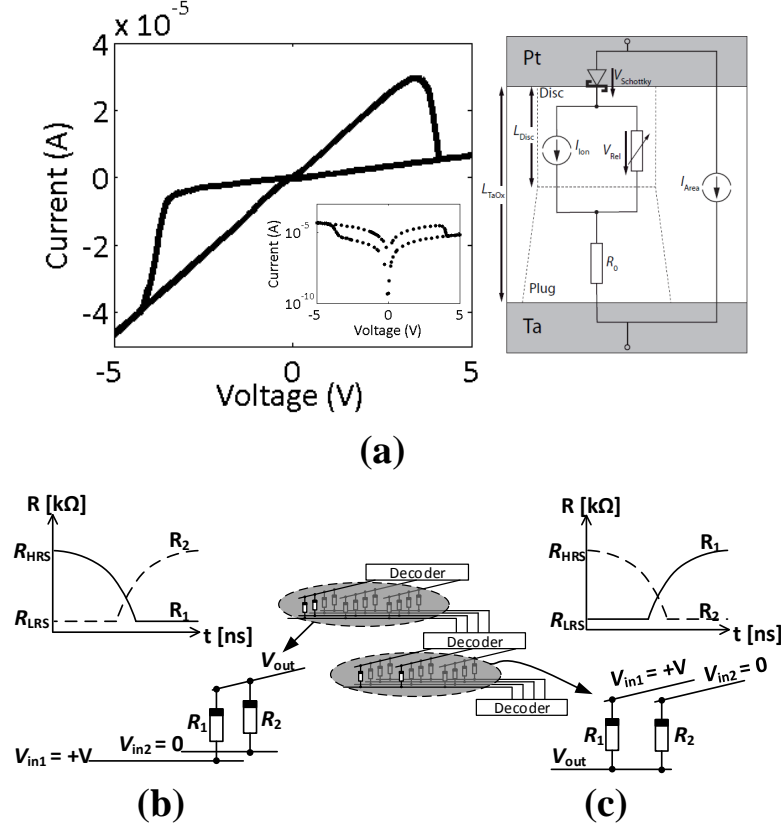


Figure 3.2: (a) The $I-V$ curve of $\text{Pt}/\text{TaO}_x/\text{Ta}$ device by applying a triangular voltage in (-5 V, +5 V) domain and the redox-based memristor model [5] are displayed. (b) The MRL-based AND gate and the devices resistance evolution. (c) The MRL-based OR gate and the devices resistance evolution.

Table 3.1: The simulation parameters for $\text{Pt}/\text{TaO}_x/\text{Ta}$ device.

Parameters	Value	Parameters	Value	Parameters	Value
$N_{\text{min}}(\text{m}^{-2})$	0.308	$N_{\text{max}}(\text{m}^{-2})$	5	$C_{31}(\text{pAm/V})$	6
$N_{\text{init}}(\text{m}^{-2})$	5	$L_{\text{Disc}}(\text{nm})$	4	$A(\text{nm}^2)$	3.14

processing to realize the influenced pixels by the noise. It arranges the noisy pixels by comparing each individual pixel in 3×3 image windows with its surrounding neighbor pixels, and it uses 21 comparisons to find the desired values (maximum, minimum

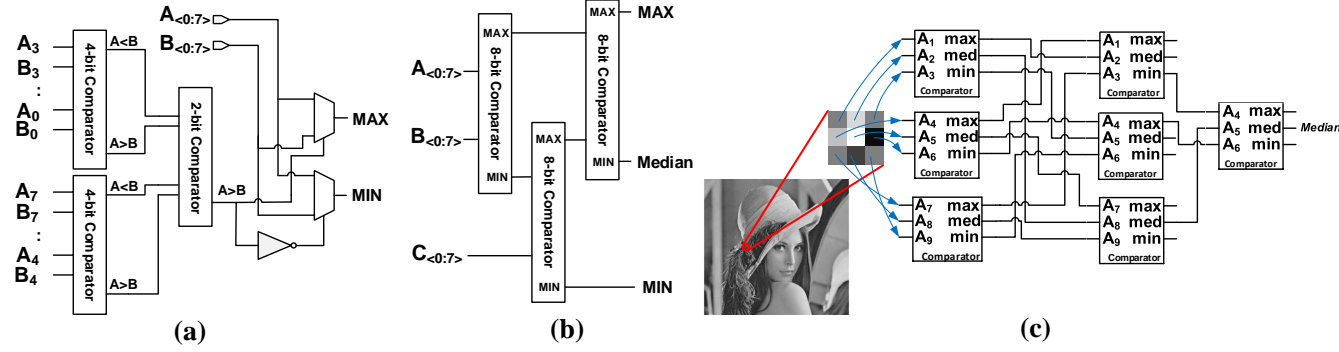


Figure 3.3: (a) 8-bit comparator circuit schematic. (b) 3-input 8-bit comparator circuit schematic. (c) The proposed median filter circuit schematic.

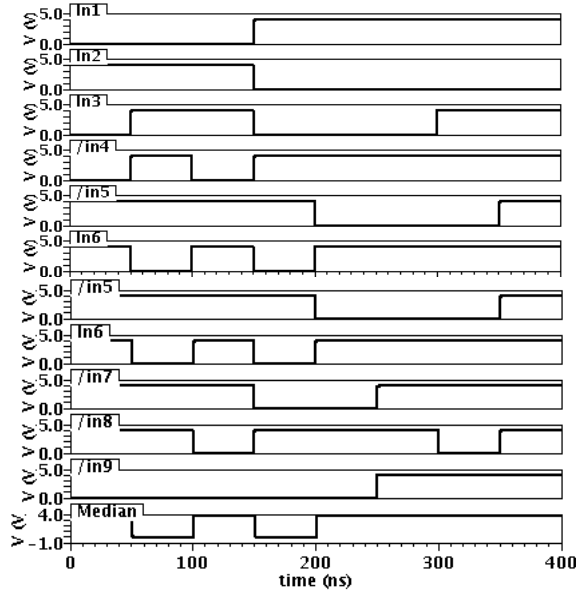


Figure 3.4: The Cadence simulation result for the proposed memristive median filter.

and median) among three 8-bits inputs. The proposed filter takes nine inputs and gives one output which is the median value. The memristor-based median design implemented and tested with Cadence design systems at schematic level before translated to Verilog-XL to perform the logic design and simulation test at the functional level. To confirm the functionality of the proposed memristive median filter a 3×3 pixel window is applied as nine input signals to the proposed circuit in Cadence to

illustrate the sorting performance of the design. The memristive filter output is displayed in Fig. 3.4 and it is successfully selected the median value for different cases. Fig.3.5 represents the overall organization of the proposed work. We intend to convert schematic of memristive based median filter into Verilog as a behavioral model which is capable of removing salt and pepper noise of an input image in a form of hex-array and sending out the filtered image as an array as well.

Two types of comparisons have been performed in this propose. First, the filtered pixels are compared directly (array form) with an expected pixels, that we have from Matlab, using a Verilog test bench which is able to detect any mismatch between the pixels. Second, the original image is reconstituted from filtered pixels using Matlab, and then a visual comparison will take place. The design consists mainly of four main Modules <Module A>, <Module B>, <Module C> and <Module D>. As it can be seen in Fig. 3.5, <Module A> and <Module D> are designed to load hex-pixels and write the filtered image pixels in a file, respectively. However, <Module B> is for buffering, reading addresses, filtering, serially sending out free-noise data. Also, <Module C> is for generating addresses. Besides that, all these tasks are pipelined under unique clock signal in order to get higher efficiency in terms of processing time and power consumption. Various flags are used to synchronize the design process between Modules. <Module A> loads the pixels, stacks the loaded pixels in a buffer A, and then sends the pixels serially. <Module B> receives pixels serially, one in each clock (CLK) cycle, and stores them in a buffer B. Whenever buffer B reaches the minimum number of pixels to store (number pixel in window 3×3), <Module C> interacts with <Module B> by sending addresses of the actual window for filtering operation. <Module B> uses these addresses to locate the pixels in buffer B and starts filtering and sending out the pixels. Finally, <Module D> receives the filtered pixels serially from <Module B> as well as their corresponding addresses in order to recreate a new array of free-noise pixels. The design modules organization and flowchart are displayed in Fig. 3.5.

The median filtering performance relies on the sorting process. There are many advanced algorithms such as bitonic sorting network, merge sort, heap sort, which

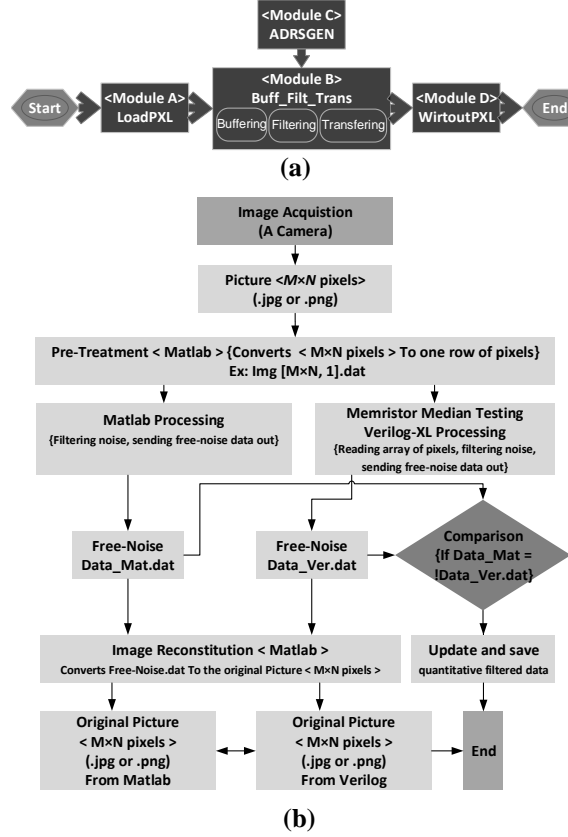


Figure 3.5: (a) Modules organization of the proposed filter in Verilog-XL. (b) Design verification flowchart for proposed median filter.

requires a total of $9 \times (9-1)/2 = 36$ comparisons to sort 9 pixels. In [30] as can be seen in Fig. 3.3, the number of comparisons is reduced to 21. First is the sorting of pixels in each row, then sorting the pixels in each column and finally is the diagonal sorting. This procedure needs 3 clock cycles to obtain the median value. This median filtering algorithm significantly improves the processing speed and considerably improves the efficiency of image processing.

3.4 Experimental Results

The simulations are done in Cadence with 65 nm CMOS TSMC technology and the redox-based Verilog-A model [5] for $\text{Pt}/\text{TaO}_x/\text{Ta}$ device with parameters in Table

3.1. The proposed median filter circuit is translated to Verilog-XL and it is tested for noise removal in images. In this test, three standard designated test 512×512 pixels images (Lena, Bridge and Pepper) with salt and pepper noise with different levels of noise density ratio starting from 10 % up to 50% are applied to the memristive median filter. The memristive filter is considerably removed noise from the images. Some samples of noisy and recovered images from filter can be seen in Fig. 3.6. The quality of the recovered images was determined using Peak signal to Noise Ratio (PSNR) and it can be determined by,

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right). \quad (3.2)$$

The parameter MSE is the Mean Square Error (MSE) and it is determined by,

$$\text{MSE} = \frac{\sum_{i=1}^M \sum_{j=1}^N (g_{i,j} - t_{i,j})^2}{M \times N}. \quad (3.3)$$

Where $g_{i,j}$ indicates the free noise image, $t_{i,j}$ is the filtered image, M and N are the image's column and row number, and the $M \times N$ is the number of pixels in the image. Also, another important parameter for image evaluation is Mean Absolute Error (MAE) and it can be determined by,

$$\text{MAE} = \frac{\sum_{i=1}^M \sum_{j=1}^N (g_{i,j} - t_{i,j})}{M \times N}. \quad (3.4)$$

The visual filtering results, which is obtained from memristor-based median filter after applying 10% up to 50% noise density on Lena, Bridge and Pepper images are represented in Fig. 3.7 and they are compared with the performance of six implemented adaptive median filter algorithms: Improved Adaptive Median Filters (CWM) [9], Nearest Neighborhood-based Adaptive center weighted Median Filtering technique (NNAM) [10], Adaptive length Median Filter (AMF) [11], Simple Adaptive Median Filter (SAMF) [12], Cluster-Based Adaptive Fuzzy Switching Median Filter (CAFS) [13] and Efficient Median Filter (EMF) [14]. From the comparisons, the proposed design demonstrates experimentally better PSNR results comparing with AMF. Moreover, it shows better denoising performance for high noise density in Lena image comparing with CWM. Furthermore, the performance of the proposed design

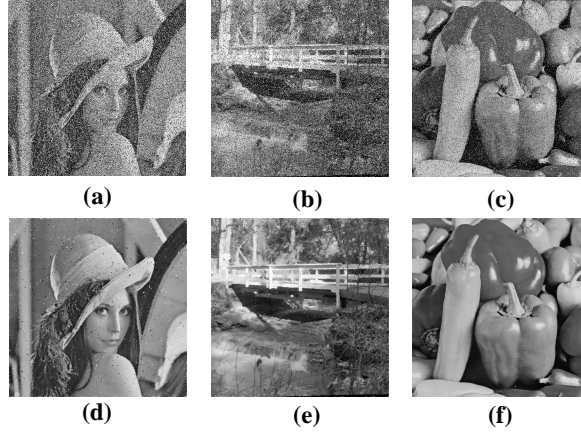


Figure 3.6: Performance of designed filter for Lena, Bridge and Pepper 512×512 images contains salt and pepper noise with different noise density ratios. (a) Lena with 30 % noise. (a) Bridge with 20 % noise. (c) Pepper with 10 % noise. (d) Filtered Lena image. (e) Filtered Bridge image. (f) Filtered Pepper image.

in denoising the bridge image has the highest PSNR value among the three design algorithms AMF, CWM, and NNAM. The proposed design shows acceptable MAE results comparing with SAMF, EMF and CAFS.

3.5 Conclusion

In this chapter, a memristive median filter was designed with $\text{Pt}/\text{TaO}_x/\text{Ta}$ device and it was verified and tested in Cadence environment, Verilog-XL and Matlab. The performance for restoring original images Lena, Bridge and Pepper from corrupted ones with 10% upto 50% random-valued impulse noise shows better PSNR results comparing with AMF and better denoising performance comparing with CWM algorithm for Lena image. Furthermore, it has comparable MAE results comparing with SAMF, EMF, and CAFS. Due to the high density offers by utilizing hybrid CMOS/Memristor-based design and promising results in restoration of the noisy images, the proposed memristive median filter design can be an alternative hardware to be applied in future image processing applications.

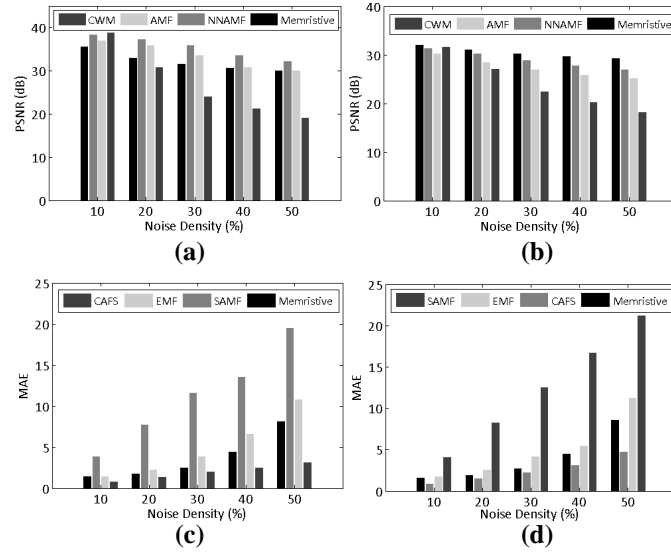


Figure 3.7: Comparison of proposed design performance with other adaptive median filter designs. (a) PSNR comparison for Lena image. (b) PSNR comparison for Bridge image (c) MAE comparison for Lena image. (d) MAE comparison for Pepper image.

References

- [1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *Nature*, vol. 453, no. 7191, pp. 80–3, May 01 2008.
- [2] F. M. Bayat, F. Alibart, L. Gao, and D. B. Strukov, “A reconfigurable fir filter with memristor-based weights,” *arXiv preprint arXiv:1608.05445*, 2016.
- [3] V. Ntinias, I. Vourkas, and G. C. Sirakoulis, “Lc filters with enhanced memristive damping,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 2664–2667.
- [4] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “MRLmemristor ratioed logic,” in *Cellular Nanoscale Networks and Their Applications (CNNA), 2012 13th International Workshop on*. IEEE, 2012, pp. 1–6.
- [5] A. Siemon, S. Menzel, A. Marchewka, Y. Nishi, R. Waser, and E. Linn, “Simulation of tao_x -based complementary resistive switches by a physics-based memristive model,” in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 1420–1423.
- [6] A. Amirsoleimani, M. Ahmadi, and A. Ahmadi, “Logic design on mirrored memristive crossbars,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2017.
- [7] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Memristor-based material implication (IMPLY) logic: Design principles and methodologies,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 10, pp. 2054–2066, 2014.
- [8] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. PremChand, “Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter,” *IEEE Signal processing letters*, vol. 18, no. 5, pp. 287–290, 2011.

-
- [9] S. S. Beagum, N. Hundewale, and M. M. Sathik, “Improved adaptive median filters using nearest 4-neighbors for restoration of images corrupted with fixed-valued impulse noise,” in *Computational Intelligence and Computing Research (ICCIC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–8.
 - [10] —, “Nearest neighborhood-based adaptive center weighted median filtering technique for restoration of images corrupted with impulse noise,” in *IEEE Int. Conf. on Computational Intell. and Computing Research*. IEEE, 2011, pp. 1492–1499.
 - [11] H.-M. Lin and A. N. Willson, “Median filters with adaptive length,” *IEEE Transactions on Circuits and Systems*, vol. 35, no. 6, pp. 675–690, 1988.
 - [12] H. Ibrahim, N. S. P. Kong, and T. F. Ng, “Simple adaptive median filter for the removal of impulse noise from highly corrupted images,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, 2008.
 - [13] K. K. V. Toh and N. A. M. Isa, “Cluster-based adaptive fuzzy switching median filter for universal impulse noise reduction,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, 2010.
 - [14] J. Zhang, “An efficient median filter based method for removing random-valued impulse noise,” *Digital Signal Processing*, vol. 20, no. 4, pp. 1010–1018, 2010.

Low Power Memristor-Based Shift Register Design

4.1 Introduction

the context of current system design, scientific researchers have widely examined the use of memristor devices since Leon Chua first presented them [1]. The outstanding features of memristors are their ability to save information, their high-density, and the fact that they consume less space than the alternatives [2]. Most memristor research has therefore been focused on memristor-based memory applications, such as memristor crossbar arrays [3]. However, recent studies have demonstrated that memristors can also be utilized to implement different Boolean functions. Moreover, memristors can be used in analog circuit design [4], neural networks [5], and hybrid memristor/CMOS circuits [6, 7]. In this chapter, the design of a memristor-only-based shift register and a hybrid memristor/CMOS-based shift register are proposed. In the implementation of a memristor-only-based shift register, few devices are utilized and the logic output is represented by the memristance state of the device. In the hybrid implementation, on the contrary, the logic output is represented by voltage alone. The rest of this chapter is arranged as follows. A brief about memristor devices and their applications in logic design is given in Section 4.2 . Section 4.3 includes information about the implementation of a 4-bit memristor-only shift register and

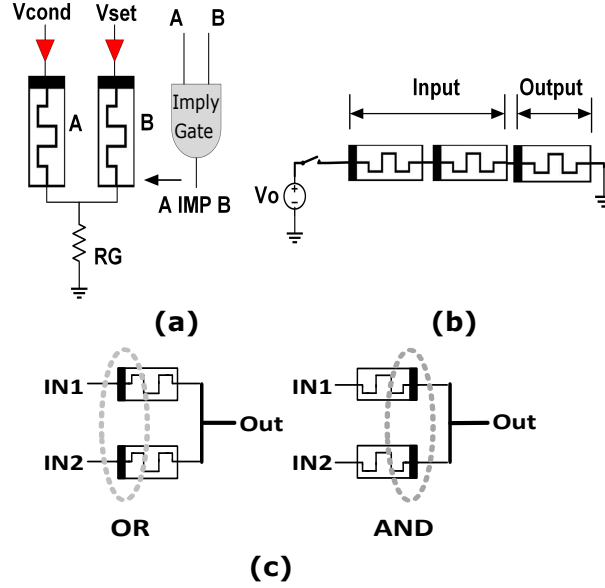


Figure 4.1: (a) Schematic of a two-input IMPLY gate. (b) Schematic of AND based MAGIC gate. (c) Schematic of an AND and OR logic gates. Polarity of memristor devices is the only difference between the two logic gates design.

implementation of a 4-bit CMOS-memristor based shift register. Section 4.4 contains the comparison and evaluation, and finally, Section 4.5 concludes this chapter.

4.2 Memristor Device and Logic Design Approach

A memristor can be defined as a passive device that has two-terminals with variable resistance that are serially connected. Changes in resistance due to the charge flow in a memristor are known as memristance. Changes in memristance depend on the last resistance state of the memristor, and that is why memristors are considered to be non-volatile devices.

4.2.1 Memristor Only Logic Design

IMPLY, memristor-based logic in crossbars, [8] and MAGIC [9] design are the most widely used methods in memristive-based logic design. In these design methods,

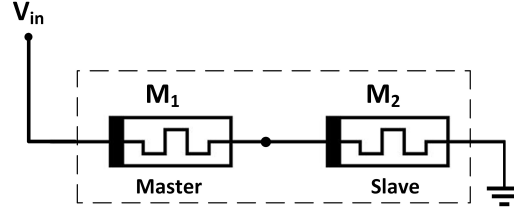


Figure 4.2: Schematic of memristive Flip Flops.

computed operands are memorized and represented by the memristance states of the memristors. All Boolean logic functions can be performed with a series of two memristors utilizing the IMPLY method [10]. Logic gates, such as OR, NOR, AND, and NAND gates can be implemented with IMPLY logic gates. An IMPLY memristor-based logic gate consist of two memristors and one resistor that is associated with the two memristors from upper side and the its bottom side associated with ground, as demonstrated in Fig. 4.1 (a). The gate produces desired results when different voltages levels of V_{COND} and V_{SET} are applied and a proper value for R_g resistor is chosen to be between $R_{ON} < R_g < R_{OFF}$. The main point of the implementation of this logic is to use the non-volatile feature of memristors. The computation logic in IMPLY is defined by the memristance of two memristors, A and B. Logic "0" is represented by the low resistance state(LRS) of the memristor (R_{ON}) and logic "1" is represented by a High Resistance State (HRS) R_{OFF} . In the MAGIC logic design approach, the structural features that distinguish this approach from IMPLY are that, it utilizes less hardware in its crossbar implementation (no resistors are used) than IMPLY logic and requires fewer voltage sources [11]. Moreover, the output results are copied to a separate memristor as illustrated in Fig. 4.1 (b), unlike with IMPLY logic, where one of two input memristors is used to store the output.

4.2.2 Hybrid CMOS/Memristor Logic Design

MRL is one of the most commonly used hybrid CMOS/ memristor-based logic design approaches [6]. In this design approach, the logic outputs are represented by voltages, unlike MAGIC and IMPLY approaches, which are memristance based. The ability

to connect memristors to CMOS inverters provides high circuit density due to the ability to build memristors in between different metal layers [12] to reduce the effect of signal degradation. MRL logic gates can be simply designed using a hybrid CMOS memristor approach. The schematic designs of AND and OR gates are shown in Fig. 4.1(c). Both AND and OR logic gates are comprised of two memristors with opposite polarity connected in series [6]. The only difference is the memristors' device polarity where the inputs are applied.

4.3 Shift Register Designs

4.3.1 4-BIT memristor only-based shift register design

This section proposes a memristor-based 4-bit shift register. The idea is to design a FF that is the core component of a shift register. A FF was implemented by serially connecting two memristors, as shown in Fig. 4.2. The first memristor M_1 saves data inputs and act as a master, where the second one M_2 saves the received data from the master unit and act as slave. The working principle of this architecture is the following:

1. M_1 is the data input memristor device and M_2 is the data output memristor.
2. V_{in} is the applied voltage to M_1 while M_2 is grounded. V_{in} is chosen to be twice of the threshold voltage (V_{th}) of the memristors. The general formula for the voltage drop at M_2 is shown in Equation 4.4

$$V_{M_2} = \frac{R_{M_2}}{R_{M_2} + R_{M_1}} \times V_{in} \quad (4.1)$$

where V_{M_2} is the voltage at M_2 , R_{M_2} is the resistance state of M_2 , which in our case, is always R_{OFF} , and R_{M_1} is the resistance state of M_1 . based on the memristance of R_{M_1} there are two cases.

- Case1 :if $R_{M_1} = R_{OFF}$ then:

$$V_{M_2} = \frac{V_{in}}{2} = \frac{2V_{th}}{2} = V_{th} \quad (4.2)$$

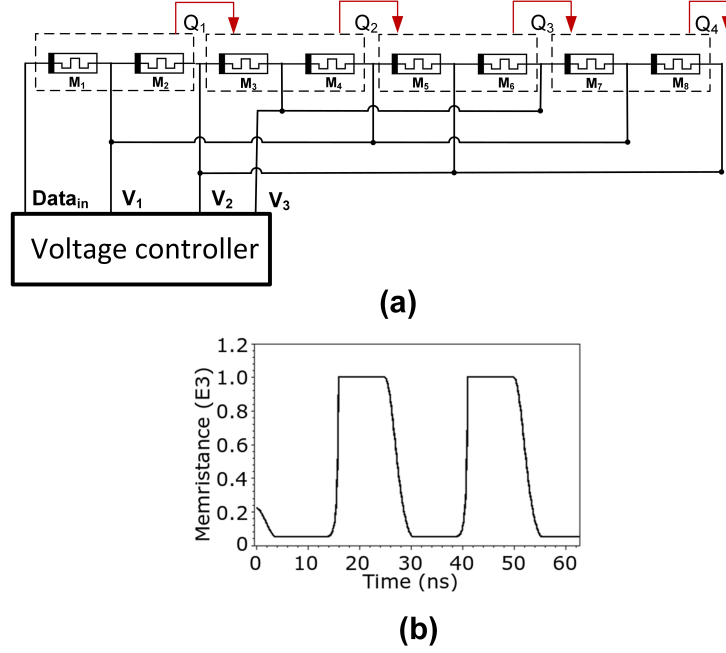


Figure 4.3: (a) Schematic of proposed shift register. (b) Memristance switching of one memristor device

In this case1, M_2 does not change and stays at R_{OFF}

- Case2 :if $R_{M_1} = R_{ON}$ then:

$$V_{M_2} = \frac{R_{OFF}}{R_{OFF} + R_{ON}} \times V_{in} \quad (4.3)$$

In this case2, memristance of M_2 starts changing and stops when the M_1 data is copied to M_2 . Therefore, the M_2 changes to LRS. Fig. 4.3 shows four memristive FFs which are cascaded with each other to build a shift register. The design procedure is as follows: The first step in this design is loading the data into first FF that is considered to be shifted to the first memristor in the second FF(M_3). The $Data_{in}$ in Fig. 4.3(a) is utilized for loading data to be shifted into the first FF. This achieved by applying positive voltage to the first memristor(M_1) which set to be always on LRS (R_{ON}). As shown in Table 4.1, the voltage sequences for computational steps are designed the way that in each step moves each bit to the right by one. Large enough input voltage is chosen to enable the memristors resistance state to switch

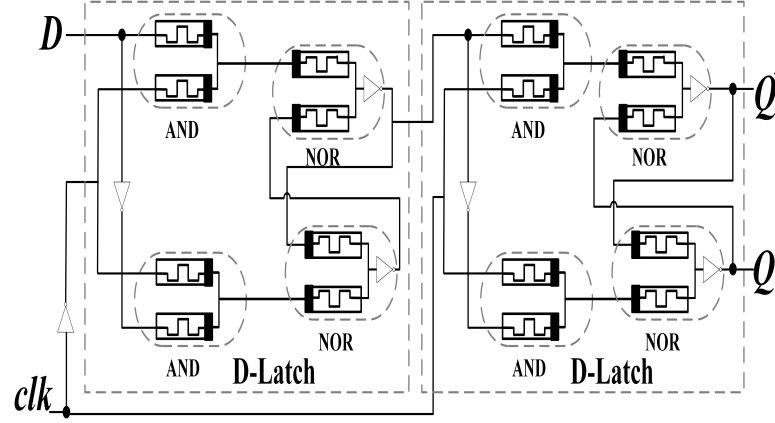


Figure 4.4: Hybrid CMOS-memristor based DFF design. Schematic of proposed D-Flipflop

from high to low and vice versa, as shown Fig. 4.3 (b). As stated in Eq. 4.2 and Eq.4.3 of FF, the first step in shifting mechanism of the designed register is loading data by applying voltage from $Data_{in}$ to copy the LRS ("0") in M_1 to M_2 . Then the shifting from first FF to second one is started when the voltage V_1 is applied. Here, the voltage at M_3 is calculated as the follows:

$$V_{M3} = \frac{R_{M3}}{R_{M3} + R_{M2}} \times V_1 \quad (4.4)$$

Subsequently, the memristance at M_3 changes to R_{ON} ("0"). After shifting the data to the second FF, voltage V_2 is applied to copy the data from M_3 into M_4 in the second FF. This process requires 8 computational steps to perform 4-bit shifting. This is a memristor-only design where each FF requires two memristors and two clock cycles to complete the shifting process of 1-bit of data. The voltage controller that exhibited in Fig. 4.3 (a) is implemented using HDL (Hardware Description Language) synthesis tools to control input voltages. The voltage controller is compiled in the Synopsys Design tool and it has a total area of $168 \mu m$ and power consumption of $17.9 \mu W$. The control voltages V_1 , V_2 and V_3 have two functions. First function is used to shift data from one FF to another and secondarily to reset the memristance of the memristors after shifting their data to its initial state HRS (R_{OFF}). For instance, V_1

Table 4.1: Voltage sequencing for computational steps of the Proposed shift Register. M_1 to M_8 indicate the memristance state of the design memristors, and Z means high impedance

Step	1	2	3	4	5	6	7	8	9
V_1	1	0	Z	1	0	Z	1	0	Z
V_2	Z	1	0	Z	1	0	Z	1	0
V_3	0	Z	1	0	Z	1	0	Z	1
M_1	R_{ON}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}
M_2	R_{OFF}	R_{ON}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}
M_3	R_{OFF}	R_{OFF}	R_{ON}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}
M_4	R_{OFF}	R_{OFF}	R_{OFF}	R_{ON}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}
M_5	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{ON}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}
M_6	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{ON}	R_{OFF}	R_{OFF}	R_{OFF}
M_7	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{ON}	R_{OFF}	R_{OFF}
M_8	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{OFF}	R_{ON}	R_{OFF}

is utilized to shift data from M_2 to M_3 and at the same time it resets the memristance in M_1 to become R_{OFF} .

4.3.2 4-bit CMOS-memristor based shift registers design

In this section, a shift register is a hybrid CMOS/memristor sequential logic circuit, designed based on hybrid memristor D flip-flops (DFF) that are serially cascaded and driven by the same clock. Memristor-based DFF is the main part of this proposal, and its schematic design is displayed in Fig. 4.4. The DFF consists of two D-latches with a serially-connected master and slave structure. The implementation of the D-latch requires 8 memristors and 6 MOSFETs and the DFF, therefore, includes 14 MOSFETs and 16 memristors.

4.4 Comparison and Evaluation

The implementation of the proposed shift registers was assessed in the Cadence Spectrum Environment. The Voltage Threshold Adaptive Memristor (*VTEAM*) model [13] was utilized for the simulation of the memristor-only-based shift register, and the

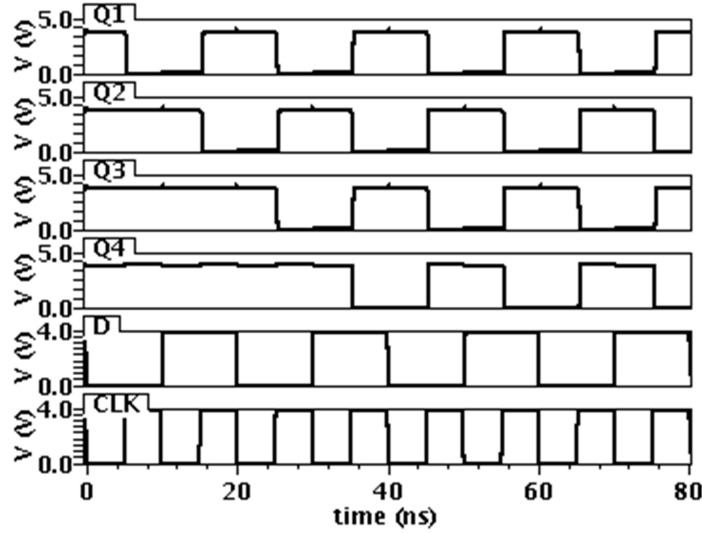


Figure 4.5: The simulation results of the proposed MRL shift register

Pt/TaOx/Ta memristor device [14] was used for MRL-based shift register simulation. The results of the proposed memristor-only shift register simulation are shown in Fig. 4.5. The memristor-only register requires 8 memristors while the MRL-based shift register requires 64 memristors and 64 CMOS transistors. The proposed model utilizes only 2 memristor to design the FF while the memristor-based Linear Feedback Shift Register Based on Material Implication Logic only utilizes 4 memristors. Moreover, the proposed memristor-only design achieved better speed than a linear-feedback shift register presented in [15], especially because it required only 8 computational steps to accomplish the 4-bit shifting operation while the method in [15] requires 55 computational steps. A comparative analysis of the memristor-only shift register design with a state-of-the-art designs is presented in Table 4.2. The power consumption of the proposed memristor-only shift register is reduced by 30.85% when comparing the lowest power consumed, as presented in Table 4.2. The total dynamic power consumed in the proposed memristor-only shift design is $83.85 \mu W$. This power consumption was measured with the presence of a voltages controller circuit. In the low power shift register design published in [16], the power consumed is $231.47 \mu W$, while

the design in [17] only consumed $114.43 \mu W$. In addition, the design presented in [18] consumed $570 \mu W$.

Monte Carlo simulation results:

process variation for utilized technology node in this proposal is important, to know the amount of variation that cells based memristor devices can tolerate without any fanout and degradation in power and delay. The Table 4.3 shows the statistical analysis including process and mismatch effects on CMOS inverter and memristor based logic gates used in memristor/CMOS shift register. The technology node 65nm were utilized to apply statistical spectre simulation to determine process variation on power consumption and delay for proposal characterized memristor based logic cells. The power variation results, using Cadence Analog Statistical Analysis for memristor only shift register and memristor/CMOS-based shift register are presented respectively, as shown shown in Fig. 4.6 (a) and (b).

4.5 Conclusion

In this chapter, a memristor-based shift register and memristor/CMOS-based shift register were designed and implemented. In the first design, the master and slave DFF was designed with only 2 memristors. In addition, by using an MRL design approach, a D-latch was implemented as a second step by utilizing eight memristors and six CMOS devices. Thus, a DFF was subsequently designed with 14 CMOS devices and 16 memristors. The proposed memristor-only-based shift register requires 8 memristors while the memristor/CMOS-based shift register demanded 64 memristor and an equivalent number of CMOS devices. Monte Carlo circuit simulations have been used to identify the power deviation of memristor cells utilized in both proposed design. A simulation confirmed the functionality of both proposals and exhibited acceptable process variation. The proposed memristor-only design occupies a small area, utilizes fewer devices, and is faster than the IMPLY-based shift register [15] and conventional CMOS-based design. Moreover, only the memristor shift register has lower power consumption than state-of-the-art designs.

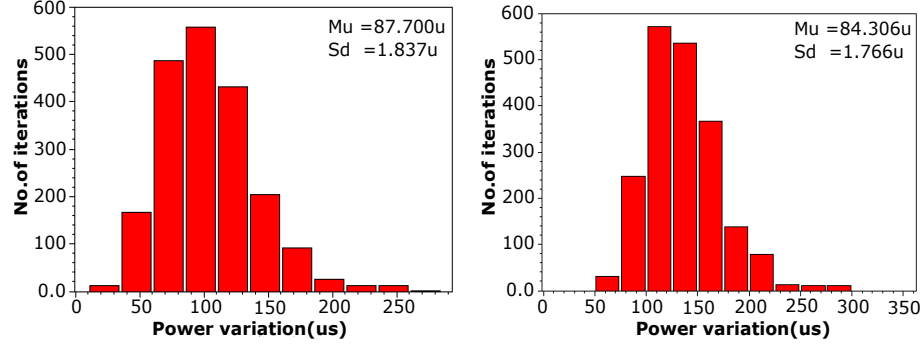


Figure 4.6: Analytical Monte Carlo Simulation of the power consumption for proposed designs (a) memristor based MRL shift register. (b)memristor only shift register

Table 4.2: Performance comparison of memristor only shift register and shift registers using cmos technology

Design	Freq.(MHz)	Operating voltage (V)	Power(μW)
LPSR [16]	50	2.5	231.47
HSSR [17]	62.5	1	114.43
PTLCP [18]	100	2	570
Only Memristor proposal	60	2	83.85
Only Memristor proposal	60	1	41.92
Only Memristor proposal	50	2.5	109.17
Only Memristor proposal	100	2	160.15
Proposed MRL shift register	60	3.8	87.70
Proposed MRL shift register	100	3.8	200.35

Table 4.3: The power consumption and delay of memristor based logic gates including the effects of process variation for 65nm technology

Gate	65nm					
	P(μW)		T _r (Ps)		T _f (Ps)	
	Mean	SD	Mean	SD	Mean	SD
Inverter	4.403	0.21	112.67	1.12	20.67	1.12
AND2X1	12.423	1.22	36.48	8.66	35.47	4.42
NOR2X1	26.20	1.51	411.40	2.62	150.30	2.29

References

- [1] L. Chua, “Memristor-the missing circuit element,” *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.
- [2] L. Guckert and E. E. Swartzlander, “Optimized memristor-based multipliers,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 2, pp. 373–385, Feb 2017.
- [3] Y. Yang, J. Mathew, S. Pontarelli, M. Ottavi, and D. K. Pradhan, “Complementary resistive switch-based arithmetic logic implementations using material implication,” *IEEE Transactions on Nanotechnology*, vol. 15, no. 1, pp. 94–108, Jan 2016.
- [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] 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.
- [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*, Aug 2012, pp. 1–6.
- [7] 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)*, Dec 2017, pp. 62–65.
- [8] 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, Nov 2018.

-
- [9] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Magicmemristor-aided logic,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, Nov 2014.
 - [10] L. Guckert and E. E. Swartzlander, “Mad gatesmemristor logic design using driver circuitry,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 2, pp. 171–175, Feb 2017.
 - [11] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky, “Logic design within memristive memories using memristor-aided logic (magic),” *IEEE Transactions on Nanotechnology*, vol. 15, no. 4, pp. 635–650, July 2016.
 - [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, thirdquarter 2016.
 - [13] 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.
 - [14] 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)*, June 2014, pp. 1420–1423.
 - [15] M. Teimoory, A. Amirsoleimani, A. Ahmadi, S. Alirezaee, S. Salimpour, and M. Ahmadi, “Memristor-based linear feedback shift register based on material implication logic,” in *2015 European Conference on Circuit Theory and Design (ECCTD)*, Aug 2015, pp. 1–4.
 - [16] K. V. S. S. Aditya, B. B. Kotaru, and B. B. Naik, “Design of low power shift register using activity-driven optimized clock gating and run-time power gating,” in *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCCE)*, March 2014, pp. 1–7.
 - [17] S. E. Razavi, J. Royaei, and M. Bahadorzadeh, “Design a low current and high speed shift register based on d type flip flop,” in *2015 Forth International Conference on e-Technologies and Networks for Development (ICeND)*, Sep. 2015, pp. 1–4.
 - [18] N. Staney and S. Anand, “Ptl-and clock-pulse circuit driven novel shift register architecture,” in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, May 2017, pp. 1072–1079.
-

Characterizing a Standard Cell Library for Large Scale Design of Memristive based Signal Processing

5.1 Introduction

Although the conventional CMOS technology scaling limitation was extended using FinFET architecture, FinFET is facing significant challenges due to different reasons such as doping damage, restriction in the logic chip design space, limitation of the electrostatics, and integration challenges [2, 3]. Therefore, substitutes to CMOS technology are in high demand. There are several alternative technologies, such as Double-Gate Tunnel FET [4], nanotube programmable devices [5], graphene transistors [6], and memristor devices[7]. Among those technologies, memristor devices are the most promising because of their great scaling ability, long-term data storage, low-power consumption, and CMOS compatibility [8, 9]. It is believed that these two terminal devices will play an essential role in the future fabrication of memory and information processing systems [10, 11]. Nevertheless, the number of memristor-based applications in today's circuit designs has been increasing exponentially. However,

the design and mapping of large-scale memristor-based applications is a challenging task due to the lack of comprehensive high-level design tools and simulation platforms. Currently, circuit design tools like SPICE, (H<) SPICE, ICAPS, and Cadence Virtuoso are not capable of providing designers with comprehensive design and simulation methodologies for memristors [12].

Xie et al. [13], presented a method for the automatic mapping of large-scale crossbar memristive-based Boolean logic circuits. This method involved the use of CMOS to control and drive the design. A programmable architecture for a large-scale neuromorphic-systems-based-memristive crossbar is proposed in [14]. The authors have proposed a framework for deep learning networks-based on the programming of spin electronics (spintronic devices). The framework mapping blocks consists of memristors and transistors to mimic spindle behavior. In the paper presented in [15], the authors introduced a design methodology for memristor crossbar architecture-based image compression. The author primary objective is to perform computational operations in a memristive crossbar and store the row-transformed image data in the same crossbar memory array. Therefore, the overall area, timing, and power of this architecture were reduced. The aforementioned methods were implemented based on memristive crossbars. Such design techniques presented real challenges, including those related to sneak path current and signal degradation. Moreover, memristive crossbar circuits require separate circuits to control input signals.

Material implication logic is also implemented to map memristor-based Boolean logic [16, 17]. In these works, implication logic was employed to reduce the number of memristor devices and operating cycles. However, the use of such methods is limited only to Boolean function implementation. Moreover, memristor-based crossbar and implication logic design methods are not synthesizable using Computer-Aided Design (CAD) synthesis tools [18]. In addition, above mentioned design methods require sequential computational steps to achieve a logic gate operation. In such process, execution of one logic computation requires more than one clock cycle.

Considering these challenges, a hybrid memristor/CMOS logic design is the most applicable method because it is CMOS compatible, delivers an optimal solution to

Table 5.1: The simulation parameters for ReRAM and RRAM devices

ReRAM PT/TAOx/TA Device [14]						
Parameters	$N_{min}(m^{-2})$	$N_{max}(m^{-2})$	$N_{init}(m^{-2})$	$C_{31}(pAm/V)$	$A(nm^2)$	$L_{Disc}(nm)$
Value	0.308	5	5	6	3.14	4
RRAM Device [21]						
Parameters	$I_0(A)$	$g_o(nm)$	$g_{max}(nm)$	$g_{min}(nm)$	$L(nm^2)$	$v_0(m/s)$
Value	$6.14(e^{-5})$	$2.75(e^{-10})$	$6(e^{-12})$	$3.14(e^{-14})$	5	150

eliminating signal degradation, and can be synthesized and mapped using CAD tools. However, it is impractical to manually design memristor-based large-scale circuits using currently-available methods due to design complexity and the limited number of memristors and transistors that CAD tools support [19].

In this chapter, a comprehensive automatic framework for the design and synthesis of large-scale memristor-CMOS circuits is proposed. This framework provides a synthesis approach that can be applied to all memristor-based Boolean logic designs. In particular, MATLAB, a high definition language (HDL) simulator, the Cadence Virtuoso environment, and Synopses software were utilized to implement parallel 8-bit adder/subtractor and 2D memristive based median filter. The filter was manually implemented on the Cadence Virtuoso schematic level and previously published in [7].

The rest of this chapter is organized as follows. A brief about choosing a proper memristor model is given in Section 5.2. Section 5.3 contains a description of cell library characterization. Section 5.4 contains a discussion of the CAD tools used for the automatic implementation of the proposal, case study and discussion of the proposed simulation results. Finally, Section 5.5 concludes this chapter.

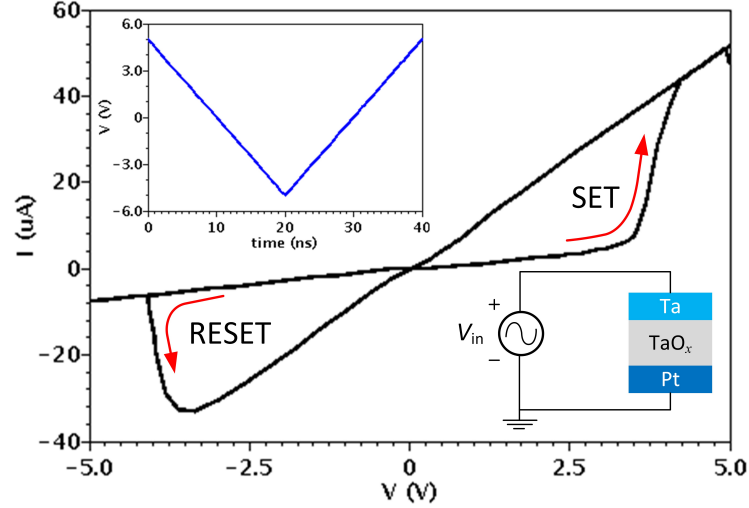


Figure 5.1: Memristor device I-V curve for ReRAM Pt/TaOx/Ta VCM device with a bipolar triangular input voltage of 5 V [14]

5.2 Memristor based-logic design

5.2.1 Memristor Modeling

For circuit testing and simulation, both Verilog-A models of RRAM that were presented in [21] and the ReRAM device that was presented in [14] were utilized to obtain the desired logic behavior for the proposed design. The accuracy levels of both memristive models provide the realistic required switching behavior. Both are simulated using the Verilog-A model in the Cadence Virtuoso environment. Due to the lack of real physical memristor device layout tools, it is important to choose an accurate memristor model [21] that simplifies the implementation of memristor-based applications and study cases for the creation of reliable simulations.

RRAM and ReRAM devices were simulated based on the parameters shown in Table 5.1. In this proposal, two factors were considered when differentiating and choosing between RRAM and ReRAM (Pt/TaOx/Ta) devices to implement memristor-based logic gates at the behavioral level. The first factor is device size and resistive layer: Both devices were designed based on small size of metal oxides that consume

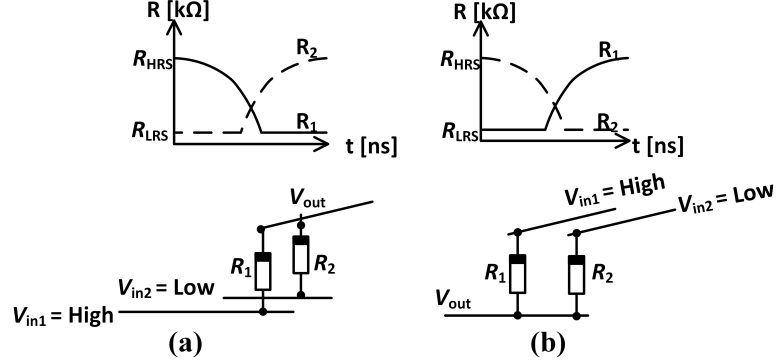


Figure 5.2: (a) MRL-based AND gate and its resistance progression. (b) MRL-based OR gate and its resistance progression.

less power. Small devices consume less power than large ones [22]. Therefore, the RRAM device is preferred due to its small size, which is < 10 nm, while the size of the ReRAM device is 11 nm. In addition, as seen in equation 5.1, the size of the metal has a direct effect on the capacitance of the device, which has a significant impact on the power dissipation and delay performance of the circuit

$$P = C_L V^2 f \quad (5.1)$$

where C_L , V , and f are load capacitance, voltage amplitude and frequency, respectively. The second factor is the amplitude of the input voltage. It is important to utilize an appropriate supply voltage to obtain low-power consumption and ensure high performance. Although having low voltage leads to significant increases in propagation delay, it significantly decreases power consumption. Thus, the RRAM device only has 2 V of input voltage supply, which is low compared to the ReRAM (Pt/TaOx/Ta) device, which has 4 V as shown in its I-V curve in Fig. 5.1.

5.2.2 Logic Design Approach

Memristor ratioed logic(MRL) is a hybrid CMOS-memristor-based logic [6]. It is a voltage-based design approach, unlike MAD [24] and Mirrored [25] logics, which are memristive-based. The compatibility of memristor devices with CMOS increases

circuit density and offers the best way to eliminate signal degradation in memristor logic of AND and OR gates. The CMOS inverter is added to the output of memristor-based OR and AND gates to achieve the desired NOR and NAND logic [26]. In MRL the voltage is perceived as logical states, high and low voltages, indicates logic "1" and "0" respectively. As shown in Fig. 5.2 (a) and (b), MRL AND and OR gate design structures. The voltages inputs V_{in1} and V_{in2} are applied to both memristors terminals that connected in parallel, and each memristor's set end is attached to the output terminal. In the test of AND gate circuit, if high voltage "1" and low voltage "0" are applied to terminals V_{in1} and V_{in2} respectively, then V_{out} can be Determined as:

$$V_{out} = \frac{R_{off}}{R_{off} + R_{on}} V_{high} \cong V_{high} \quad (5.2)$$

and when if low voltage "0" applied to both inputs terminals then the V_{out} can be calculated as:

$$V_{out} = \frac{R_{on}}{R_{on} + R_{off}} V_{high} \cong 0 \quad (5.3)$$

MRL logic design approach was exploited to implement the proposed circuit designs.

5.3 Synthesis methodology and implementation

Creating a memristor-based standard cell library is essential to exploring the potential of memristors in digital design using available CMOS synthesis tools. Using such tools requires an accurate cell characterization method for memristor-based logic gates. Synthesis tools involve the use of characterized gates library files to facilitate logic optimization, enhance design speed, and determine the area, timing, and power consumption. The characterization process for any memristor-CMOS cells can be described as follows:

Input/Output Capacitance : The measured capacitance values at each cell pin is the main factor used to estimate dynamic power and delay using synthesis tools. Input capacitance is calculated by measuring the charge flows into or out of each cell pin divided by the magnitude of the power supply. It can be mathematically

formulated as follows:

$$C_{pin} = \frac{1}{V_{dd}} \int_{t_1}^{t_2} i(t) dt \quad (5.4)$$

where $i(t)$ is the current flow into the pin and C_{pin} is the pin capacitance, measured as the amount of charge passing through the pin at the input voltage (rising swing from 0 to VDD and from VDD to 0) divided by the voltage supply. In the memristor based logic cells characterization method, the characterizing simulation utilizing a net of inverters as standard capacitive load which serially connected to the output pin of cell under characterization.

Power Measurement: The logic transition of cells input pins which are deployed in the proposed method consumes energy. The value of energy consumed by the proposed circuit was measured by calculating the current passing through the zero-DC source that was connected to the VDD. Then the consumed current integrated over each time transition using the Cadence Virtuoso calculator. The library table of each cell in the proposed design only contains energy values measured in joules, and the rest of the power consumption calculation was accomplished by the Synopsys synthesis compiler. The only measured power consumption in this method is dynamic power, which is mathematically described as follows:

$$P_{Dynamic} = \alpha C V_{DD}^2 f \quad (5.5)$$

where α , C , V_{DD} and f are the switching activity factor, capacitance, voltage source, and operating frequency, respectively

Delay Measurement: The non-linear delay simulation method was utilized to measure the propagation delay. With fan-out consideration, the delay measurement depends on transition time at the cell input pin and the capacitance of output pin. The specified slew threshold for the cell is set to be between 30%-70% of the power supply magnitude. In addition, it was defined as the time the signal rises from 30% to the 70% and fall from 70% to 30% of its VDD.

Area Estimation: As illustrated in [27] and [28] the memristor device can be fabricated on the top of the CMOS transistors. Therefore, the area was estimated depending on the size of the inverters utilized in each cell.

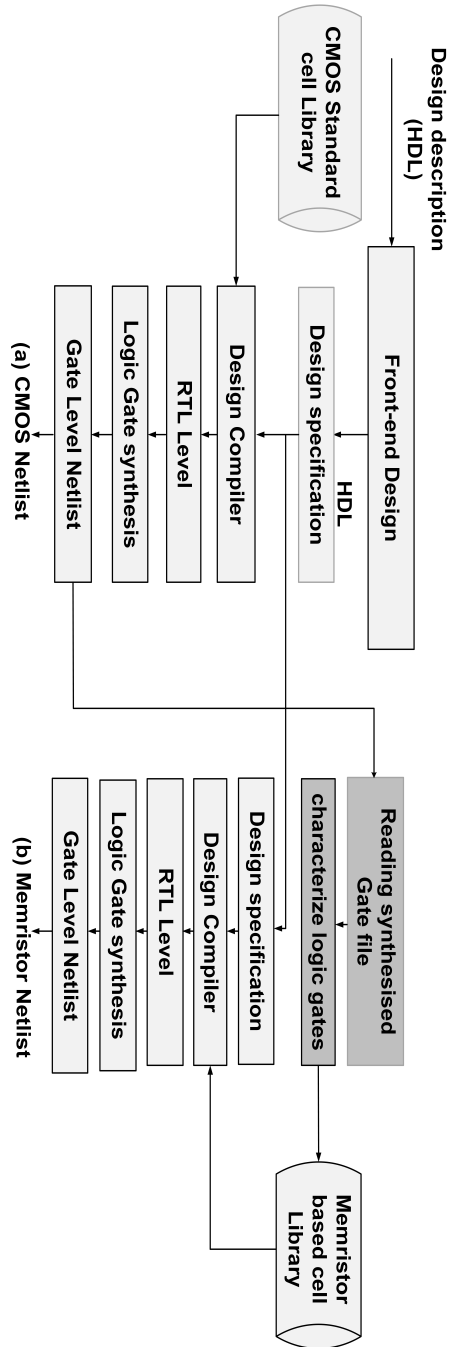


Figure 5.3: Flow chart displaying design flow based on Synopsys EDA tool for proposed framework

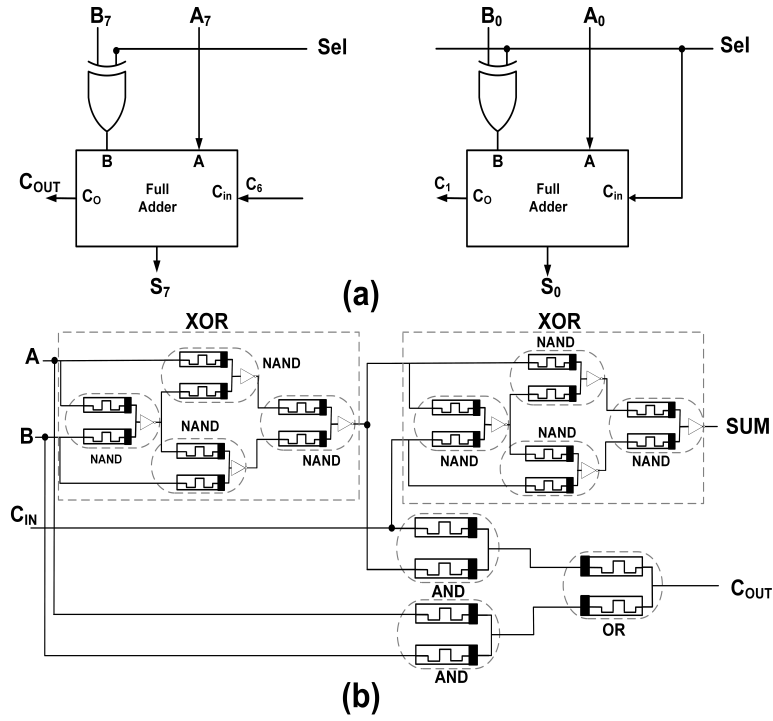


Figure 5.4: The memristor-based adder/subtractor. (a) 8-bit Adder/subtractor schematic circuit. (b) Implemented Memristor-based 1-bit Adder/Subtractor

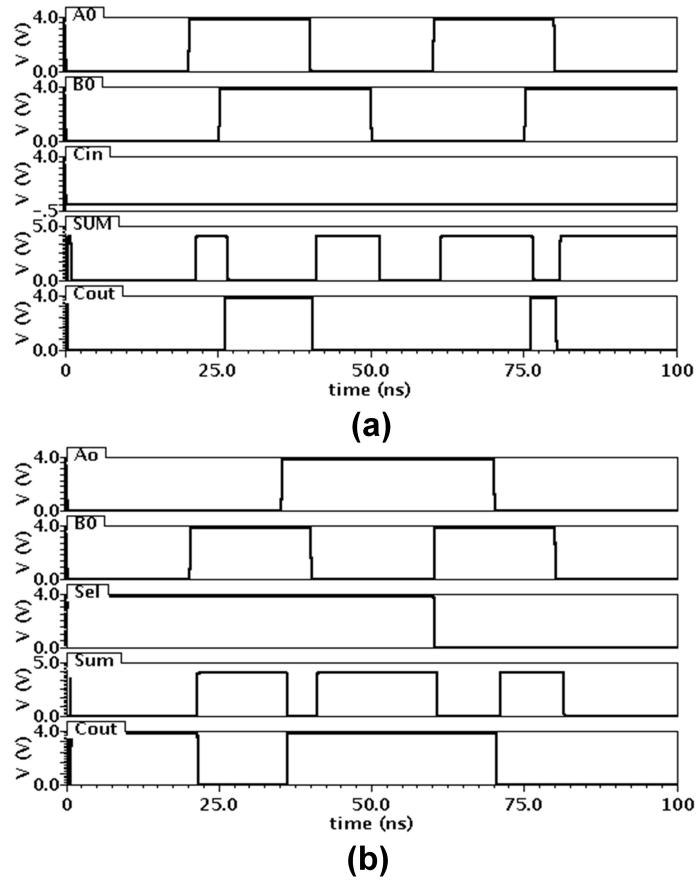


Figure 5.5: Simulation results of memristor-based 1bit adder and 1bit adder/subtractor. (a) MRL-based 1bit adder simulation results. (b) The simulations of the MRL-based 1bit adder/subtractor

5.4 Memristor based-logic design

5.4.1 Memristor Modeling

In the first step, the behavioral functions of the implemented cells at the schematic level were described using Verilog HDL and simulated using the Cadence NC-Verilog-XL simulator. The Verilog language can be used to read/write files from a storage environment. This feature makes 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. In the proposed framework, the signal processing applications require MATLAB encoder and decoder. Matlab function is needed to convert input signals into the form of hex arrays because Verilog only reads and writes ASCII character files, and then other Matlab function is used to import the processed data encoded by Verilog test bench to reconstruct it. In the second step, as shown in Fig. 5.3(a), after testing the design at the behavioral level, the implemented RTL was synthesized to the gate netlist level with the aid of a Synopsys Design Vision compiler. The design compiler uses a standard library that contains all information about the characteristics of logic cells to generate the final CMOS-based gate netlist file.

In the third step, the generated CMOS gate netlist was carefully inspected to realize the logic cells used to build the CMOS-based design. After the logic cells are produced by the Synopsys synthesis compiler, equivalent memristor-based logic gates are implemented at the schematic level, tested, and characterized using the MRL design method. Hence, at this stage of the design, the characterization process for memristor-based logic cells was obtained to build a standard memristor-based library for the Synopsys synthesis compiler, as presented in Fig. 5.3(b).

The most important characterized cells involved in the proposal are AND, NAND, OR, NOR, multiplexer, and other defined Boolean function circuits, as shown in Table 5.2. The built library provides a synthesis tool with information about cell logic function, area, input/output capacitance, delay, and power consumption.

5.4.2 Case Study 1

In this section, a memristor-based parallel 8-bit adder/subtractor is designed and analyzed using the proposed framework. It was implemented at both the schematic and behavioral levels. In other words, the implementation was done to establish and validate the design of the adder/subtractor at the schematic level using a Cadence Spice Spectre simulator, NC-Verilog at the behavioral level, and Synopsys Synthesis tools using Design Vision at the synthesis level. As part of the design, the adder/subtractor was chosen to clarify the proposed framework, design, and simulation. The following is a brief description of the framework.

The memristor-based 8-bit adder/subtractor was implemented with seven cascaded combinations of 1-bit memristor-based full adders. The schematic design of the adder/subtractor circuit is exhibited in Fig. 5.4 (a). The 1-bit memristor-based full adder logic circuit consists of two memristor-based AND gates, one memristor-based OR gate, and two memristor-based XOR gates, as shown in Fig. 5.4 (b).

The functionality of the designed adder/subtractor was proven by the simulation results in Fig. 5.5. As illustrated in Fig. 5.4 (a), the Sel line acts as a control signal to decide whether to use the adder or subtractor circuit modes. When $Sel = 1$, the Sel line acts as carry-in (Cin). Thus, all inputs of B will be reversed and 1 will be added to the LSB to determine the 2's complement. In addition, when $Sel = 0$, B XOR 0 will always produce B. Therefore, A and B will be added.

The adder/subtractor verilog description was verified in the Cadence NC-Verilog simulator, and the Synopsys compiler synthesis RTL description and then converts synthesized description to optimized gate-level. The produced gates file consists of several logic cells some of which are listed in Table 5.2. The characterization procedure for this proposed design was implemented at the schematic level by utilizing Cadence spice Spectre. This characterization process provides the required information for the memristor-based library that has been used by the Synopsys synthesis compiler to estimate the design area, delay, and power consumption. This information represented the design logic function and area. It also includes measurements of the design's input/output capacitance, delay, and power consumption. All this information was

Table 5.2: list of logic cells involved in the design of adder/subtractor

Cells	Description	Equation	MOSFET	Memristors
AND2X1	logic AND for two inputs	$Z = (A.B)$	0	2
OR2X1	logic OR for two inputs	$Z = A + B$	0	2
NOR2X1	logical NOR	$Z = \overline{(A + B)}$	1	2
MXI2X1	2 inputs multiplexer with inverted output	$Z = \overline{(\overline{S}.B) + (S.B)}$	2	6
XOR2X1	logic exclusive OR for two inputs	$Z = (A \oplus \overline{B}) + (\overline{A} \oplus B)$	5	8
XNOR2X1	logic exclusive NOR for two inputs	$Z = (A \oplus \overline{B}) + (\overline{A} \oplus B)$	4	8
AOI21X1	logical inverted OR of one AND gate and an additional input	$Z = \overline{(A_0.A_1)} + B_0$	1	4
NAND2X1	logical NAND of two inputs	$Z = \overline{(A.B)}$	1	2
NOR3X1	logical NOR of three inputs	$Z = \overline{(A + B + C)}$	2	4
INVX	logical inversion of single input	$Z = \overline{A}$	1	0

generated from the simulation of memristor-based cells at the schematic level.

5.4.3 Case Study 2

In this case study, the proposed framework was applied to implement a memristor-based median filter which was manually implemented and tested only at the Cadence Virtuoso schematic level and previously published in [7]. Image processing is very useful and has been extensively used in the areas of medicine, film and video production, photography, remote sensing, military target analysis, and manufacturing automation and control [29] [38] [30]. These applications usually require bright and clear images or pictures. Hence, corrupted or degraded images need to be processed to improve human interpretation, enhance visual pictorial information, and modify the data structure used for image representation to optimize it for data storage, transmission, or other representations for autonomous machine perception.

The main goal of any enhancement method is to obtain a more suitable result compared to the original. Digital images are represented as 2D arrays of numbers, where the value of each entry corresponds to the greyscale value of a pixel, ranging between 0 and 255 (255 being white). Thus, image enhancement techniques are transformed into 2D filtering operations. The 2D median filter replaces the value of each element based on the median value of its neighbor. The S_{xy} is a neighborhood concept with eight elements immediately surrounding the median element. Thus, the mathematical representation of an image $g(x;y)$ in the median filtering process is described as follows:

$$S_{xy} = \text{median}\{g_{(s,t)}\} \quad (s, t) \in S_{xy} \quad (5.6)$$

The implementation of a memristor-based median filter has two phases. The first phase is the schematic level implementation. At this stage, the median filter is manually designed and a 3×3 window is applied to verify the functionality of the proposal. In the second phase, due to the high design complexity, automated synthesis tools are required to make reliable and accurate simulations. Therefore, using a standard memristor cells library is essential to improving the accuracy of synthesis tools when

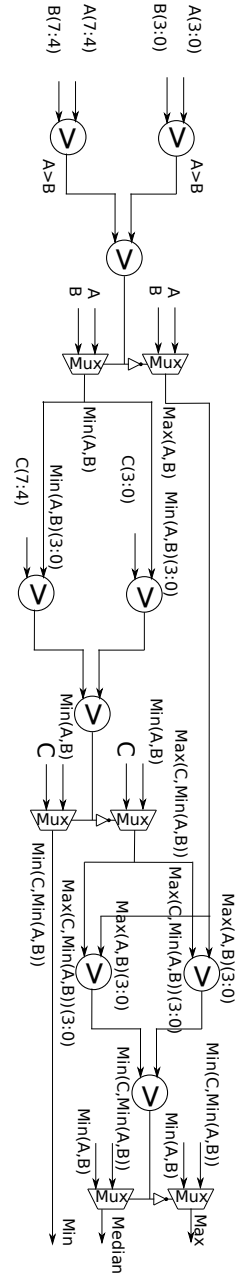


Figure 5.6: Proposed memristor based median filter sorting circuit

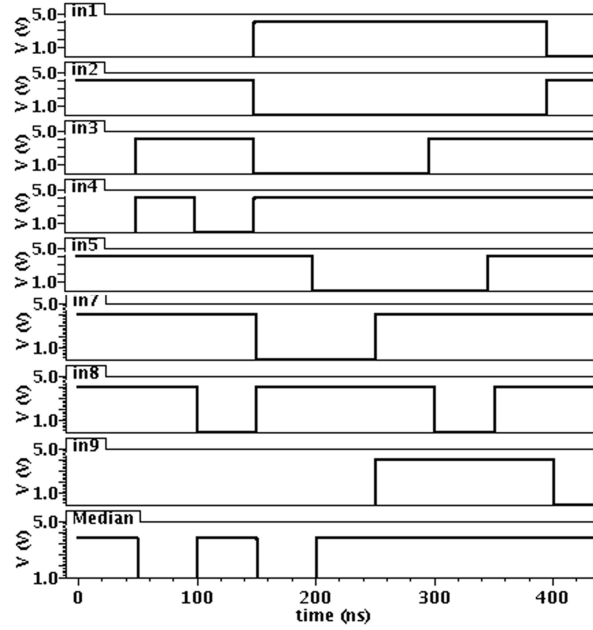


Figure 5.7: Proposed memristive median simulation detection in 3×3 window

they estimate power, area, and delay. Thus, the memristor cells involved in schematic implementation are characterized to create a memristor-based standard cell library.

Schematic Level Implementation : The sorting mechanism in this technique is to find median pixel from the surrounding neighborhood pixels. The execution steps of memristive median circuit detects the median pixel in a 3×3 window, and the simulation results for the circuit are shown in Fig. 5.7. This design was implemented using seven three-input 8-bit memristor-based comparators. Each of these comparators consists of three memristor-based two-input 8-bit magnitude comparators.

The 2 input 8-bit comparators were implemented as illustrated in Fig. 5.6, with two 4-bit memristor-based magnitude comparators to compare between two pixels (eight bits for each input). The schematic of this comparator is displayed in Fig. 5.8, and it was implemented based on a memristor-based MRL logic structure as shown in Fig. 5.9. The outputs of the two 4-bit comparators were compared again with those of the 2-bit comparator to find the largest pixel value between the two inputs. Then only one output value from the 2-bit comparator is split between two multiplexers

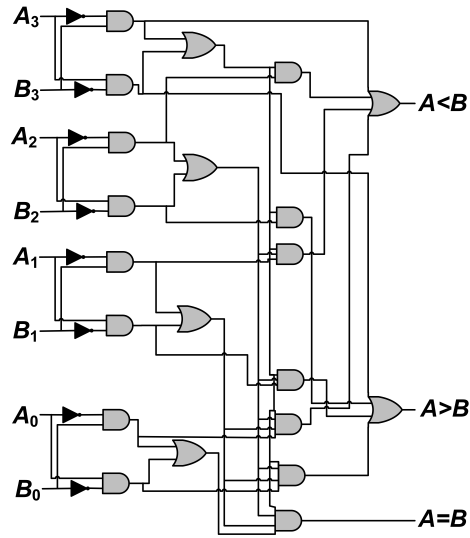


Figure 5.8: Schematic view of the implemented 4-bit memristor-based magnitude comparator.

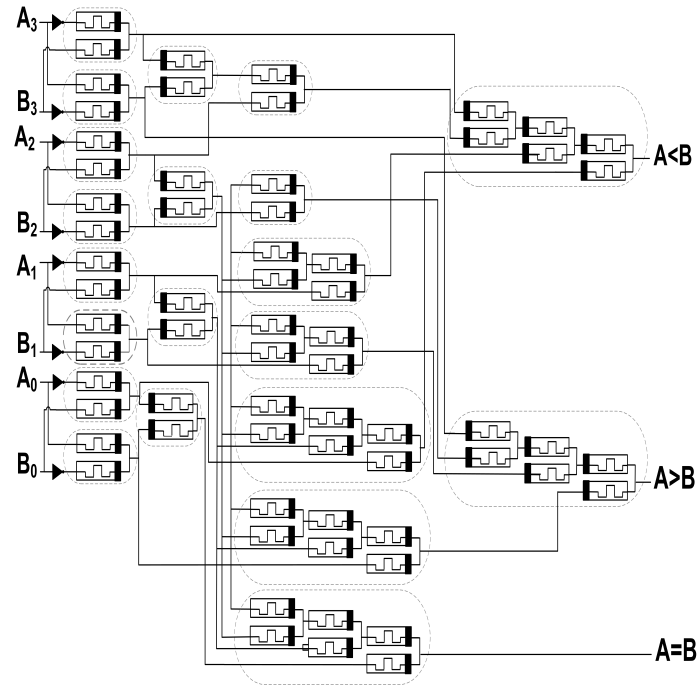


Figure 5.9: Implemented memristor-based 4-bit magnitude comparator.

and the other output is connected to the selector of the first multiplexer to decide which pixel has the maximum value, and the same output is inverted and connected to the second multiplexer to select the pixel with the minimum value.

The proposed filter proceeds nine inputs and determines the median value among them. This proposed architecture of the memristor-based median filter design was implemented and tested with Cadence Virtuoso environment at the schematic level using the memristor model presented in Verilog-A [14] and the parameters utilized for this model are shown in Table 5.1.

Automatic Implementation :

To prove the functionality of the proposed filter, first step was to describe the behavior of the median filter algorithm that was implemented at the schematic level using Verilog HDL and simulate it using the Cadence NC-Verilog-XL simulator. Unfortunately, Verilog only reads and writes ASCII character files. Therefore, it is not capable of reading images in standard formats, such as BITMAP or JPEG, directly from disk [31]. To resolve this problem, it is necessary to define a new image format to be used with a design test bench. The new image must be a HEX file that only contains information about RGB/grayscale vectors for each pixel of the input image. The data from hex-files are to be applied as stimuli to the point operations blocks described in Verilog language. The HEX characters are then elegantly converted to binary format by the Verilog HDL simulator.

In this part, the median filter implemented in Verilog was a behavioral model that removes the 'salt and pepper' noise of an input image and outputs the filtered image. The filtered image is then compared to an expected result that was created using the same filtering process in MATLAB for verification. The proposed filter flow chart of design verification is described as shown in Fig. 5.10. The development steps of the proposed method in the Verilog behavioral model includes four modules:

M1, M2, M3, and M4. M1 and M4 were set to load image data to memory and to write the filtered image data to a file. However, M2 was designated to buffer pixels, read addresses, filter input pixels, and sends out noise-free data while M3 was designed to generate addresses. In addition, to achieve an efficient processing time

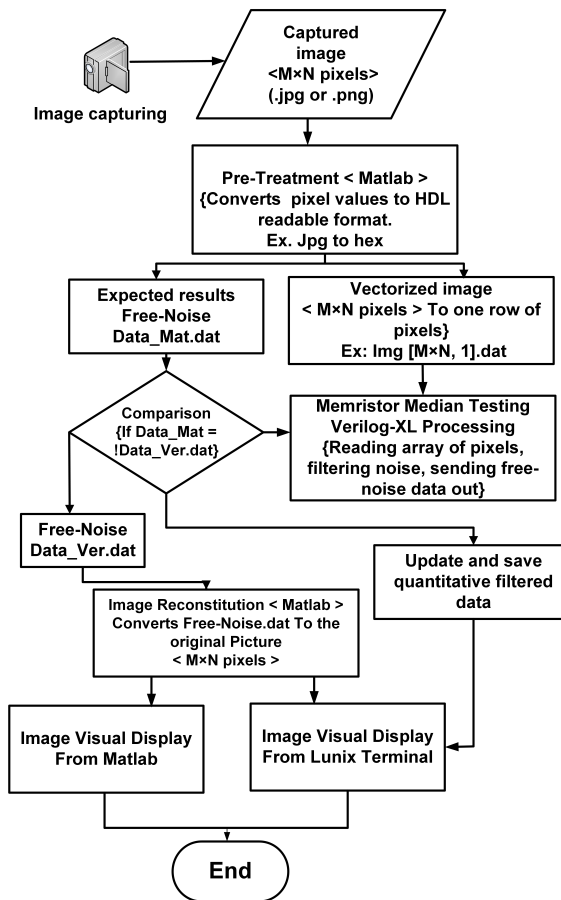


Figure 5.10: Flow chart of memristor based median filter design verification

Table 5.3: Comparison of synthesis results of proposed adder/subtractor design

Design method	Power (nW)	Delay (ns)	Area (μm)	MOSFET	Memristors
CMOS-based adder/subtractor	991.39	3.9	1247.40	322	—
Memristor-based adder/subtractor	525.82	4.23	1176.88	105	210

Table 5.4: Comparison of experimental results of proposed memristive median filter

Design method	Freq. (MHz)	Power (mW)	Delay (ns)	Area (μm)	MOSFET	Memristors
EIMF	129.58	2.85	7.72	-	-	-
LPAMF	714.58	5.52	-	12,937	-	-
Proposed CMOS	60.5	1.53	15.26	20,523	938/cell	-
Proposed Memristor	200	1.3	14.28	17,338	567	3213
Proposed Memristor	714.58	4.6	4.6	-	567	3213
Proposed Memristor	129.58	0.842	18.7	-	567	3213

and reduce power dissipation, all these implementation stages were pipelined under a unique clock signal. The flow chart of verification process for the design are shown in Fig. 5.10. In the second step, as shown in Fig. 5.3(a), after testing the design at the behavioral level, the implemented RTL is synthesized to the gate netlist level with the aid of a Synopsys Design Vision compiler. The design compiler uses a standard library that contains all information about the characteristics of logic cells to generate a final CMOS-based gate netlist file.

In the third step, the generated CMOS gate netlist is thoroughly inspected to realize the logic cells that were elaborated on in the CMOS-median-filter-based design. After the logic cells are produced by the Synopsys synthesis compiler, equivalent memristor-based logic gates are implemented at the schematic level, tested, and characterized using the MRL design method. Hence, in this stage of the design, the characterization process for memristor-based logic cells is obtained to build a standard memristor-based library for the Synopsys synthesis compiler, as presented in Fig. 5.3(b). The most important characterized cells involved in the proposal are AND, NAND, OR, NOR, multiplexer, and other defined Boolean function circuits that listed

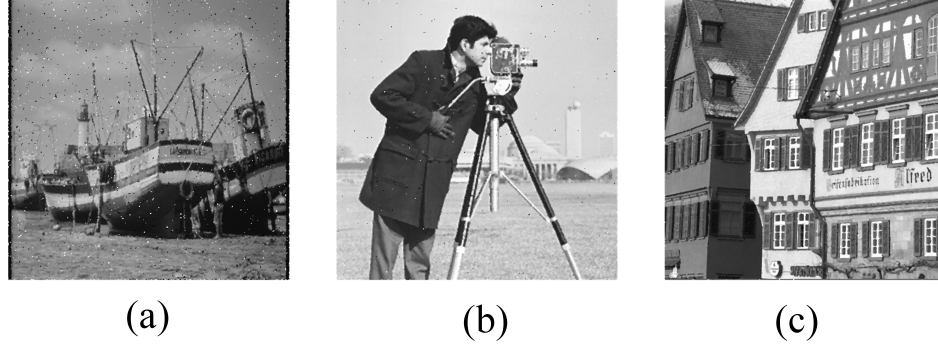


Figure 5.11: Performance of designed filter for Boat, Camera man and Houses 512×512 images contains salt and pepper with different noise density ratios. (a) Filtered Boat image with 30 % noise. (b) Filtered Camera man image with 20 % noise. (c) Filtered Houses image with 10 % noise.

in Table 5.2. The built library provides a synthesis tool with information about cell logic function, area, input/output capacitance, delay, and power consumption.

5.4.4 Simulation Results

Performance results: In this part, the filtering process of the proposed median filter is evaluated on both schematic and behavioral levels. The simulation results for the implemented memristor-based median filter at the schematic level were drawn in Cadence Virtuoso with 65 nm cell library. The RRAM and ReRAM Verilog-A models were utilized with the parameters shown in Table 5.1. To verify the performance of the memristive median filter on the schematic level, a 3×3 window was applied to the filter input to testify the sorting ability of the designed filter, and the simulation results are shown in Fig. 5.7 Then the proposed schematic was converted to the behavioral level, where it was simulated with Verilog-XL and NC-Verilog and synthesized with a Synopsys compiler before being tested for image denoising. In this test, standard 512×512 -pixel images (boat, cameraman, and houses) with various levels of salt and pepper noise density ratios (10 % up to 50%) were tested with the designed memristive median filter. The filter successfully removed the noise from the distorted

Table 5.5: The power consumption and delay of memristor based logic gate including the effects of process variation for 180 nm and 65 nm technology

Gate	180nm						65nm					
	$P(\mu W)$		$T_r(P_s)$		$T_f(P_s)$		$P(\mu W)$		$T_r(P_s)$		$T_f(P_s)$	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Inverter	5.29	0.96	22.69	4.39	18.75	1.263	4.403	0.21	112.67	1.12	20.67	1.12
AND2X1	19.58	0.53	109.36	6.94	35.23	2.16	12.423	1.22	36.48	8.66	35.47	4.42
NAND2X1	47.07	2.08	175.91	19.22	62.78	1.95	21.64	0.92	88.12	1.01	74.28	2.13
OR2X1	18.91	0.63	634.05	4.61	341.43	5.52	13.49	1.13	641.66	4.95	475.40	5.04
NOR2X1	36.58	2.57	411.40	2.62	150.30	2.29	26.20	1.51	411.40	2.62	150.30	2.29
XOR2X1	55.962	4.17	248.52	47.24	185.98	7.12	46.44	3.62	185.57	22.56	125.52	3.82

Table 5.6: Performance parameters for different salt & pepper noise variance applied on boat image

Boat Image					
Noise ratio	10%	20%	30%	40%	50%
MSE	8.916	11.6018	15.3932	20.8135	29.0203
PSNR	38.6313	37.5195	36.2915	34.9814.57	33.5360
MAE	1.2539	1.6510	2.5129	4.4779	8.0976

Table 5.7: Performance parameters for different salt & pepper noise variance applied on camera man image

Camera man image					
Noise ratio	10%	20%	30%	40%	50%
MSE	6.4582	8.7386	12.1256	17.3191	25.5203
PSNR	40.0637	38.7504	35.7796	34.0959	34.0959
MAE	0.9687	1.3793	2.4683	5.0225	10.0215

images. Samples with superimposed salt and pepper noise and recovered images are shown in Fig. 5.11. Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and Mean Absolute Error (MAE) were measured to evaluate the quality of the recovered images and calculated as follows:

$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N i(S_{i,j} - F_{i,j})^2}{M \cdot N}. \quad (5.7)$$

Where $S_{i,j}$ represents the noise-free image, $F_{i,j}$ is the recovered image, number of image column and row are represented M and N, respectively, and $M \cdot N$ is the total number of pixels in the image. Moreover, PSNR can be determined using the following equation:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right). \quad (5.8)$$

The sample word length for of each architecture was 8 bits with a 3×3 window size. The simulation results in Table 5.3 show that memristor-based adder/subtractor can

Table 5.8: Performance parameters for different salt & pepper noise variance applied on house image

House Image					
Noise ratio	10%	20%	30%	40%	50%
MSE	31.3554	35.5081	40.2278	45.7149	52.8390
PSNR	33.2017	32.6615	32.1195	31.5642	30.93530
MAE	4.0678	4.8980	6.2673	8.4359	12.1760

reduce power, area and dealy comparing with implemented CMOS-based adder/subtractor. In Table 5.4 it can be seen that the memristive median filter is the most time- and power-efficient design when compared to an efficient implementation of 1-d median filter (EIMF) [32] and a low-power architecture for the design of a low-power architecture for a one-dimensional median filter, (LPAMF) [33]. The power consumption reduced by 16.25% comparing with the lowest power consumed by other designs as shown in Table 5.4. Compared to the equivalent CMOS design, the area of the proposed architecture is significantly reduced by 16.82%. The visual simulation results for noisy images (10% up to 30%) recovered by the proposal filter are displayed in Fig. 5.11. Tables 5.6, 5.7 and 5.8 summarize the quantitative restoration results of PSNR, MSE, and MAE for boat, cameraman, and house images, respectively.

Monte Carlo simulation results:

In this proposal, the simulation results which presented in previous Section were exhibited an ideal outputs also the implemented memristor based logic cells with the same parameters and function are perfectly matched. However the process variations on memristor model parameters might be a reason to a consequential degradation. Therefore, understanding the impact of process variation for utilized technology node is important, to know the amount of variation that cells based memristor RRAM devices can tolerate without any fanout and degradation in power and delay. The Table 5.5 shows the statistical analysis including process and mismatch effects on CMOS inverter and memristor based logic gates. Two technology nodes 180nm and 65nm were utilized to apply statistical spectre simulation to determine process variation

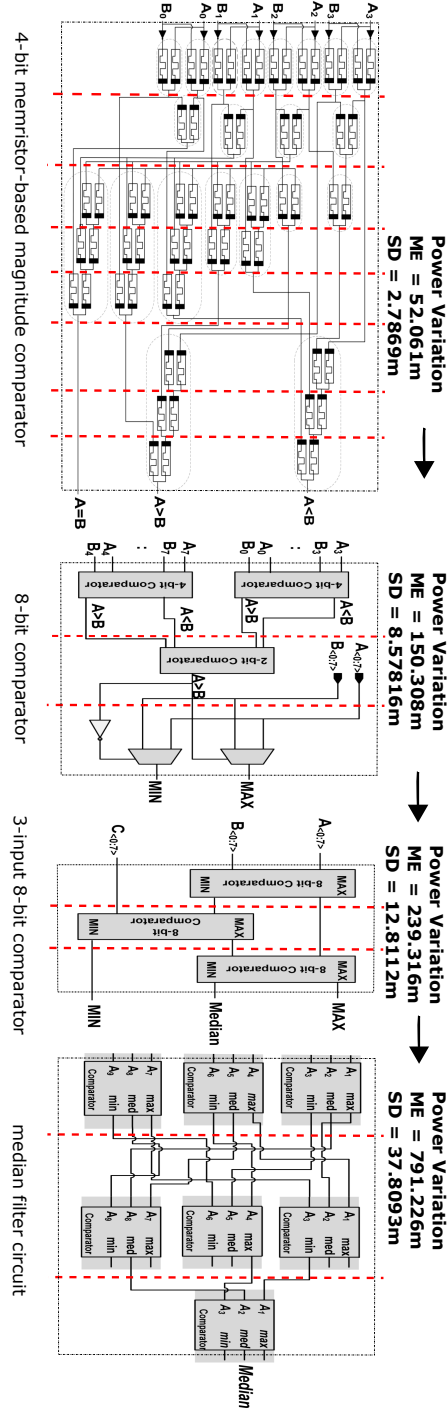


Figure 5.12: Hierarchical procedure for Process variation simulation . Simulation executes for 4-bit memristor-based magnitude comparator, 8-bit comparator, 3-input 8 bit comparator and finally median filter circuit.

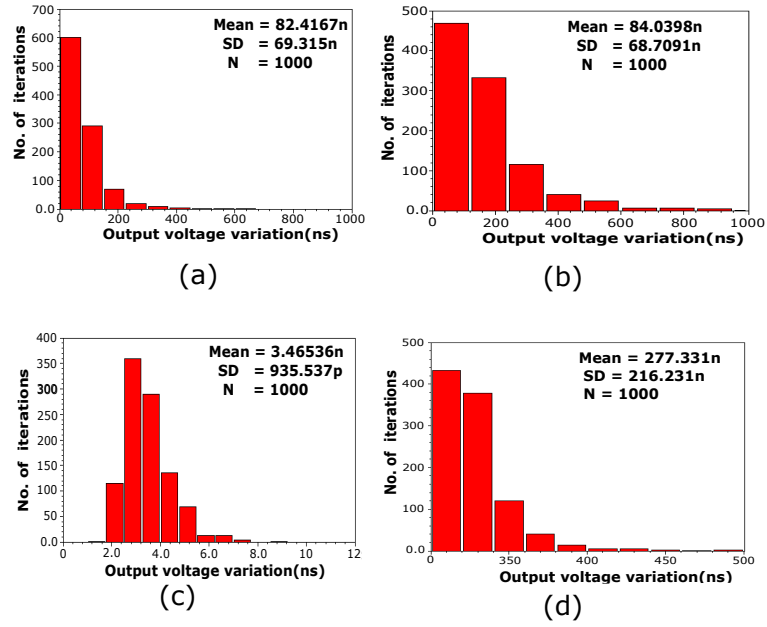


Figure 5.13: Analytical Monte Carlo Simulation of the output voltage for proposal basic logic gates. (a) CMOS inverter. (b) Memristor based AND gate.(c) Memristor based NAND gate.(d) Memristor based OR gate.

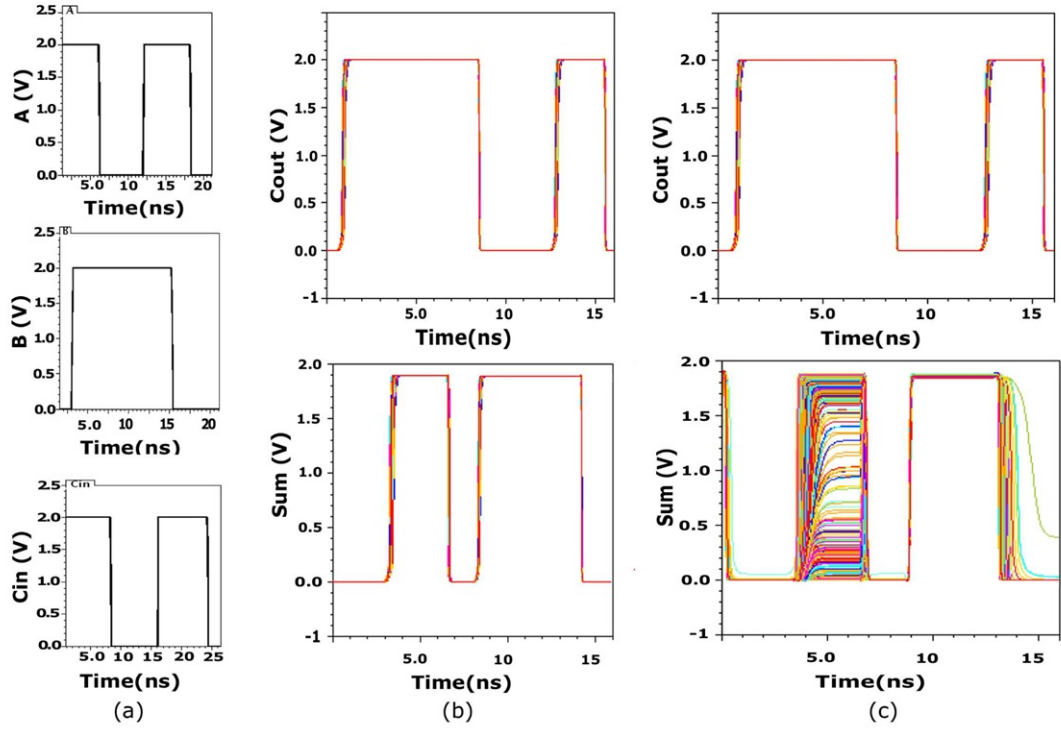


Figure 5.14: Monte Carlo output voltage variation of 1-bit adder/subtractor. (a) input A,B and Cin.(b)Optimized Monte Carlo simulation output voltage with buffer inserted. (c) Output voltage variation before buffers.

on power consumption and delay for proposal characterized memristor based logic cells. An ocean script has been writing in Cadence Spectre to preform Monte Carlo simulation on power consumption and delay of proposed large scale circuits. The Monte Carlo simulation procedure for proposed memristor based median filter circuit is achieved in a hierarchal order, as shown in Fig. 5.12. First step of the Monte Carlo simulation was preformed on memristor based logic gates such as OR, AND, NAND and NOR with 180nm and 65nm technology. In the second step, the mean (ME), standard deviation (SD), and number of iterations runs (N) in power consumption and delay for large scale circuits(8-bit adder/subtractor and 2D memristive median) implemented in this proposal are achieved. The voltage variation results, using Cadence Analog Statistical Analysis for inverter, memristor based AND, NAND, OR, NOR, and XOR are shown in Fig. 5.13. The Statistical Analysis indicates that a large circuits would not tolerate the fabrication standards and would not function as designed to do. Therefore, buffers are inserted at the outputs were the voltage dropped to correct the degradation issue. In the Fig.5.14 (a) input voltages for A, B and Cin pins, (b) output voltage variation for optimized 1-bit adder/subtractor circuit, and (c) untolaterated output voltage for 1-bit adder/subtractor circuit are exhibited.

5.5 Conclusion

In conclusion, this framework is a general methodology for designing large-scale CMOS/memristor-based circuits for digital logic. In particular, In this method MATLAB, a Hardware Description Language (HDL) simulator, the Cadence Virtuoso design environment, and Synopsys software were utilized in this framework. A low-power and high-speed memristor-based parallel 8-bit adder/subtractor and 2D memristive median filter were designed with RRAM and ReRAM devices. They were tested and verified in Cadence Virtuoso, Verilog-XL, Synopses Design Vision, and MATLAB. The low-power, low-area, and high-speed performance were achieved by generating a standard memristor-based cell library. The simulation results and verification process proved that the designed memristive behavioral model was able to

restore original images from distorted ones with 10 to 30% salt and pepper noise. The proposed design shows very significant enhancement in power consumption and delay compared to equivalent CMOS architecture, EIMF, and with LPAMF designs. Compared to the equivalent CMOS design, the area of the proposed architecture is significantly reduced by 32.79%.

References

- [1] 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)*, June 2014, pp. 1420–1423.
- [2] A. Razavieh, P. Zeitzoff, and E. J. Nowak, “Challenges and limitations of cmos scaling for finfet and beyond architectures,” *IEEE Transactions on Nanotechnology*, vol. 18, pp. 999–1004, 2019.
- [3] A. Malinowski, J. Chen, S. K. Mishra, S. Samavedam, and D. Sohn, “What is killing moore’s law? challenges in advanced finfet technology integration,” in *2019 MIXDES - 26th International Conference ”Mixed Design of Integrated Circuits and Systems”*, 2019, pp. 46–51.
- [4] K. Boucart and A. M. Ionescu, “Double-gate tunnel fet with high- κ gate dielectric,” *IEEE Transactions on Electron Devices*, vol. 54, no. 7, pp. 1725–1733, July 2007.
- [5] G. Agnus, W. Zhao, V. Derycke, A. Filoramo, Y. Lhuillier, S. Lenfant, D. Vuillaume, C. Gamrat, and J.-P. Bourgoin, “Two-terminal carbon nanotube programmable devices for adaptive architectures,” *Advanced Materials*, vol. 22, no. 6, pp. 702–706, 2010.
- [6] J. Kedzierski, P. Hsu, P. Healey, P. W. Wyatt, C. L. Keast, M. Sprinkle, C. Berger, and W. A. de Heer, “Epitaxial graphene transistors on sic substrates,” *IEEE Transactions on Electron Devices*, vol. 55, no. 8, pp. 2078–2085, Aug 2008.
- [7] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” *nature*, vol. 453, no. 7191, p. 80, 2008.
- [8] Y. Zhang, Y. Li, X. Wang, and E. G. Friedman, “Synaptic characteristics of ag/aginsbte/ta-based memristor for pattern recognition applications,” *IEEE Transactions on Electron Devices*, vol. 64, no. 4, pp. 1806–1811, April 2017.

-
- [9] N. Gergel-Hackett, B. Hamadani, B. Dunlap, J. Suehle, C. Richter, C. Hacker, and D. Gundlach, “A flexible solution-processed memristor,” *IEEE Electron Device Letters*, vol. 30, no. 7, pp. 706–708, July 2009.
 - [10] K. Eshraghian, K. Cho, O. Kavehei, S. Kang, D. Abbott, and S. S. Kang, “Memristor mos content addressable memory (mcam): Hybrid architecture for future high performance search engines,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 8, pp. 1407–1417, Aug 2011.
 - [11] H. Li and Y. Chen, “An overview of non-volatile memory technology and the implication for tools and architectures,” in *2009 Design, Automation Test in Europe Conference Exhibition*, April 2009, pp. 731–736.
 - [12] A. Sarmiento-Reyes, J. J. Leon, L. Hernandez-Martinez, and H. Vazquez-Leal, “A cad-oriented simulation methodology for memristive circuits,” in *2016 IEEE 7th Latin American Symposium on Circuits Systems (LASCAS)*, Feb 2016, pp. 103–106.
 - [13] L. Xie, H. A. D. Nguyen, M. Taouil, S. Hamdioui, and K. Bertels, “A mapping methodology of boolean logic circuits on memristor crossbar,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 2, pp. 311–323, Feb 2018.
 - [14] S. G. Ramasubramanian, R. Venkatesan, M. Sharad, K. Roy, and A. Raghunathan, “Spindle: Spintronic deep learning engine for large-scale neuromorphic computing,” in *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Aug 2014, pp. 15–20.
 - [15] Y. Halawani, B. Mohammad, M. Al-Qutayri, and S. F. Al-Sarawi, “Memristor-based hardware accelerator for image compression,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 12, pp. 2749–2758, Dec 2018.
 - [16] 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.
 - [17] F. S. Marranghello, V. Callegaro, M. G. A. Martins, A. I. Reis, and R. P. Ribas, “Improved logic synthesis for memristive stateful logic using multi-memristor implication,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 181–184.
 - [18] 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)*, July 2016, pp. 43–48.
-

-
- [19] B. P. Bhuvana, B. R. Manohar, and V. S. K. Bhaaskaran, "Standard cell characterization for reversible logic," in *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, Sep. 2016, pp. 534–538.
 - [20] A. Sasi, A. Amirsoleimani, M. Ahmadi, and A. Ahmadi, "A memristive taox-based median filter design for image processing application," in *2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, July 2018, pp. 85–88.
 - [21] 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.
 - [22] L. Deng, D. Wang, Z. Zhang, P. Tang, G. Li, and J. Pei, "Energy consumption analysis for various memristive networks under different learning strategies," *Physics Letters A*, vol. 380, no. 7-8, pp. 903–909, 2016.
 - [23] 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*, Aug 2012, pp. 1–6.
 - [24] L. Guckert and E. E. Swartzlander, "Mad gates-memristor logic design using driver circuitry," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 2, pp. 171–175, 2017.
 - [25] A. Amirsoleimani, M. Ahmadi, and A. Ahmadi, "Logic design on mirrored memristive crossbars," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2018.
 - [26] K. Cho, S.-J. Lee, and K. Eshraghian, "Memristor-cmos logic and digital computational components," *Microelectronics Journal*, vol. 46, no. 3, pp. 214–220, 2015.
 - [27] J. Cong and B. Xiao, "mrfpga: A novel fpga architecture with memristor-based reconfiguration," in *2011 IEEE/ACM International Symposium on Nanoscale Architectures*, June 2011, pp. 1–8.
 - [28] R. Huang, 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," *Applied Physics A*, vol. 102, no. 4, pp. 927–931, 2011.
 - [29] R. C. Gonzalez, R. E. Woods *et al.*, "Digital image processing," 2002.
-

- [30] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," *IEEE Signal processing letters*, vol. 18, no. 5, pp. 287–290, 2011.
- [31] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, and C. H. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 287–290, May 2011.
- [32] V. G. Moshnyaga and K. Hashimoto, "An efficient implementation of 1-d median filter," in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, Aug 2009, pp. 451–454.
- [33] R. Chen, P. Chen, and C. Yeh, "A low-power architecture for the design of a one-dimensional median filter," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 3, pp. 266–270, March 2015.

Hybrid Memristor-CMOS Based Finite Impulse Response Filter Design

6.1 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 element 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, 5]. Material Implication Logic (IMPLY) [4, 6] 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) [7] 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) [5] 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 [8]-[9] 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 [10] Memristor devices were used to provide weights for 6-tap FIR filter. In our 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. The introduction is described in Section 6.1. An introduction to the Memristor model and proposal design approach is described in Section 6.2. In Section 6.3, FIR filter design is presented. In Section 6.4, a case study of the DFF circuit was presented to show the utility of cell library designed. In Section 6.5, the verification of the proposed filter is explained. In Section 6.6, simulation results and comparison with CMOS based FIR is provided. Finally, remarks and conclusion are proposed in Section 6.7.

6.2 Design Approach and Device Modelling

6.2.1 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 6.1: RRAM memristor parameters for simulation

Parameters	$I_0(\text{A})$	$G_o(\text{nm})$	$L(\text{nm}^2)$	$G_{max}(\text{nm})$	$G_{min}(\text{nm})$
Value	$6.14(e^{-5})$	$2.75(e^{-10})$	5	$6(e^{-12})$	$3.14(e^{-14})$

6.2.2 Device Modeling

Metal oxide-based resistive switching memories RRAMs are intended for use in wide range application of non-volatile 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 6.1 Where I_e , G , and G_0 are hopping current density, gap length which can either minimum value G_{min} or maximum value G_{max} based on the applied voltage and window resistance coefficient.

6.3 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, 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 6.1.

$$S[n] = b_0X[n] + b_1X[n - 1] + \dots b_kX[n - k] \quad (6.1)$$

Several procedures are involved in the implementation of the proposed filter as shown in Fig. 6.2 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 netlist gates. The synthesized FIR memristor based netlist gates should be verified and tested again with the test bench.

6.4 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 requires implementing and validating all gates in the design. As an example, a sequential D-Flip Flop (DFF) circuit shown in Fig. 6.1. 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 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 6.2.

$$C = \frac{1}{V_{dd}} \int_{t_1}^{t_2} i(t) dt \quad (6.2)$$

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

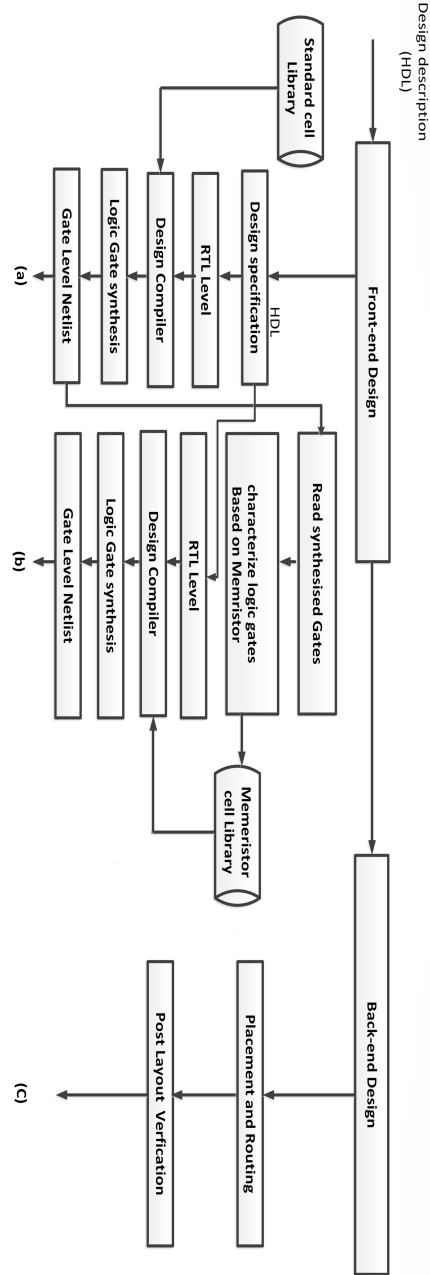


Figure 6.2: Flow chart displaying design flow based on Synopsys EDA tool for proposed FIR filter.(a) netlist CMOS Base (b) netlist memristive Based, (C) design layout.

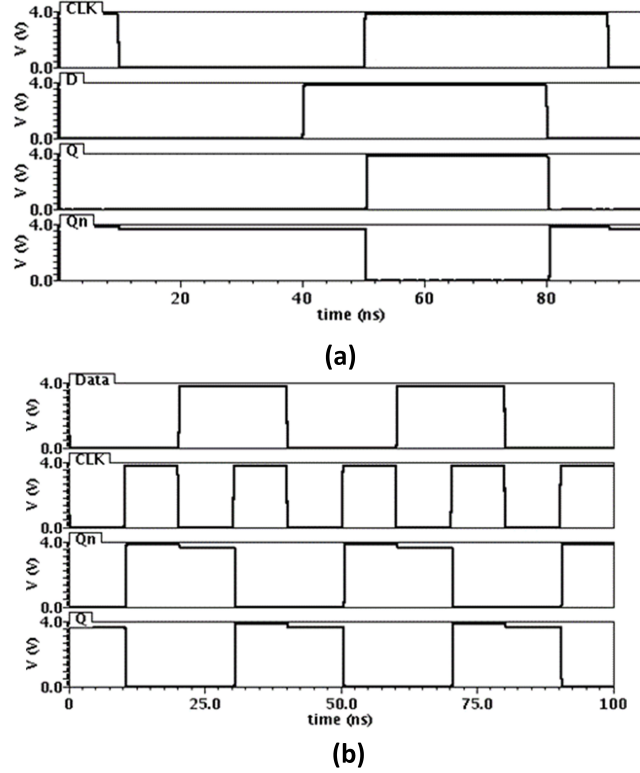


Figure 6.3: Simulation results in schematic level.(a) MRL-based D-Latch.(b) MRL-based DFF.

6.5 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. 6.4. It involves primarily several modules which are: *< Module1 >*, *< Module2 >*, *< Module3 >* and *< Module4 >*. *< Module1 >* and *< Module4 >* are responsible for loading input signals "8 bits every cycle" and write the filtered signals in the output file "hex decimal format" respectively. Whereas, *< Module2 >* is responsible for storing data temporarily and sending out undesired frequencies. And *< Module3 >* 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. $\langle \text{Module 1} \rangle$ input signals received and stored temporarily in register, and then signals are sent serially every clock cycle to $\langle \text{Module 2} \rangle$ where signals are stored again in another register, also $\langle \text{Module 2} \rangle$ communicates with $\langle \text{Module 3} \rangle$ in order to use the generated addresses to find the desired signals and pass it serially to $\langle \text{Module 4} \rangle$ as a final stage. At the end, the desired signals are written in the form of the array corresponding to the filtered signals.

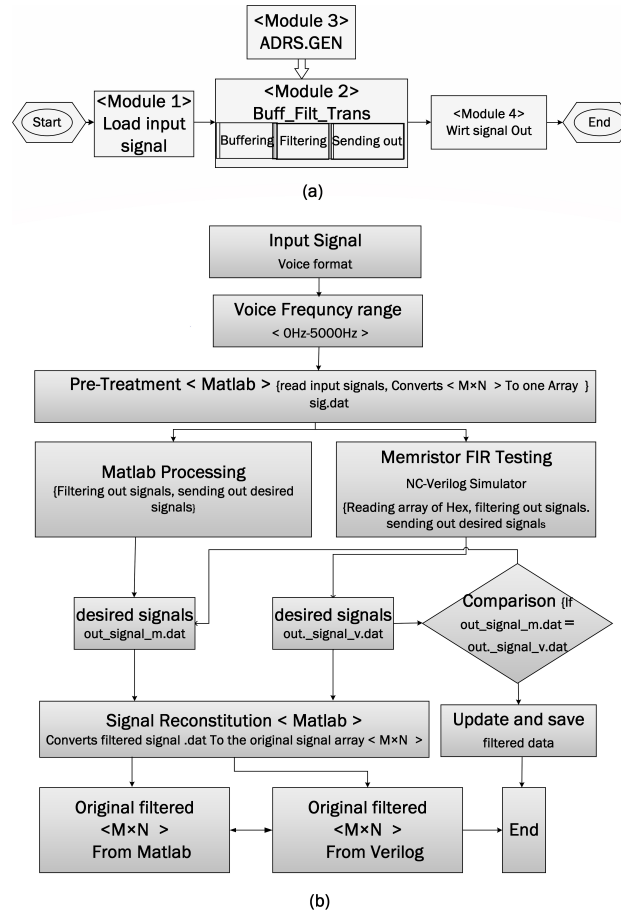


Figure 6.4: (a) Organization of the proposed FIR.(b) Verification flowchart for proposed.

Table 6.2: 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	$Y = \overline{(A.B)}$

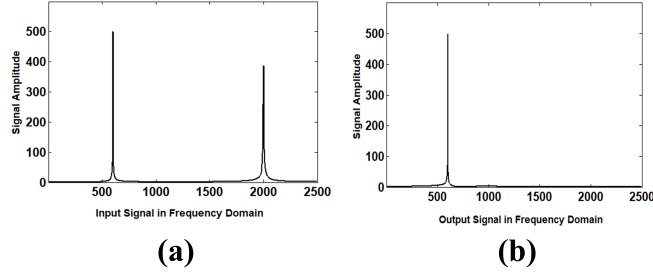


Figure 6.5: (a) Input signals in frequency domain at 600 and 2000 Hz.(b) Filter output signal in frequency domain at 600 Hz.

6.6 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 6.1, were employed to implement the Memristive behavior of the device. The filtering process of the proposed filter was tested by applying a set of signals with different frequencies. The simulation results have shown the behavioral model of the design was capable of distinguishing between all input signals and passes only signals with the desired frequency as shown in Fig. 6.5. 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 6.2, such cells

Table 6.3: Synthesis results of memristor based DFF

Design	Area(μm)	Power (mW)	Delay (ns)
CMOS	459.04	83.60	0.37
Memristor	379.20	41.29	0.16

Table 6.4: Synthesis results of memristor based FIR filter.

Design	Area(μm)	Power (mW)	Delay (ns)
CMOS	95813.62787	3.3099	0.36
Memristor	10973.69952	0.8776	0.31

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 6.3. The results have proven that Memristive DFF with 16 memristors and 14 MOSFET has less area comparing to DFF based CMOS. The design memristor base FIR appears to be compact comparing to CMOS base FIR. Consumed power and delay are comparably reduced as well, as shown in Table 6.4.

6.7 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 environment, NC-Verilog and Matlab. The simulation results have indicated that the filter was effectively able to distinguish between all inputs signals and allow only the desired signals to pass through the passband region and reject any signal beyond the stopband frequency.

References

- [1] H. A. 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, 2016, pp. 43–48.
- [2] S. Shin, K. Kim, and S.-M. Kang, “Memristor applications for programmable analog ics,” *IEEE Transactions on Nanotechnology*, vol. 10, no. 2, pp. 266–274, 2010.
- [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, 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)*. IEEE, 2014, pp. 562–565.
- [5] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “Mrlmemristor ratioed logic,” in *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*. IEEE, 2012, pp. 1–6.
- [6] 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.
- [7] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Magicmemristor-aided logic,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014.
- [8] V. Ntinis, I. Vourkas, and G. C. Sirakoulis, “Lc filters with enhanced memristive damping,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2015, pp. 2664–2667.

- [9] H. A. 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, 2016, pp. 43–48.
- [10] F. M. Bayat, F. Alibart, L. Gao, and D. Strukov, “A reconfigurable fir filter with memristor-based weights,” *arXiv preprint arXiv:1608.05445*, 2016.
- [11] P.-Y. 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, 2015.
- [12] R. Gonzalez and R. Woods, “Digital image processing, prentice hall, new jersey,” 2002.

Conclusion and Future Work

7.1 Summary

In this dissertation, several memristor models were utilized and developed for efficient memristor based digital implementation. This research provides solution for degradation and fanout challenges facing memristor circuit design. In addition, it provides a framework for mapping and synthesis of large-scale memristor-CMOS circuits.

In Chapter 2, a fast, low area and low power hybrid CMOSmemristor based LFSR design is proposed. As an example a 4- bit LFSR has been implemented by using MRL scheme with 64 CMOS devices and 64 memristors. The proposed design is more efficient in terms of the area when compared with CMOS-based LFSR circuits. The simulation results proves the functionality of the design. This approach presents acceptable speed in comparison with CMOS-based design and it is faster than IMPLY-based memristive LFSR. The propped LFSR has 841 ps delay. Furthermore, the proposed design has a significant power reduction of over 66% less than CMOS-based approach.

In Chapter 3, a memristive median filter was designed with Pt/TaOx/Ta device and it was verified and tested in Cadence environment, Verilog-XL and Matlab. The performance for restoring original images Lena, Bridge and Pepper from corrupted ones with 10% upto 50% random-valued impulse noise shows better PSNR results comparing with AMF and better denoising performance comparing with CWM al-

gorithm for Lena image. Furthermore, it has comparable MAE results comparing with SAMF, EMF, and CAFS. Due to the high density offers by utilizing hybrid CMOS/Memristor-based design and promising results in restoration of the noisy images, the proposed memristive median filter design can be an alternative to be applied in future image processing applications.

In Chapter 4, a fast and efficient area memristor-only-based shift register, as well as a hybrid CMOS/memristor-based shift register are proposed. Specifically, a 4-bit shift register with only 8 memristor devices and a hybrid CMOS /memristor with 64 memristor devices and 64 CMOS transistors were implemented and simulated using Cadence Virtuoso. The simulation results demonstrate the designs efficient functionality. Compared to the implementation of a CMOS-memristor based shift register, the implementation of the proposed design is more efficient when concerning area and speed with respect to the implementation of the Memristor Based-Material-Implication (IMPLY) memristive shift register. In addition, the shift register with only memristor-based has a significant power reduction compared to a CMOS design shift register.

In Chapter 5, a comprehensive automatic framework for the design and synthesis of large-scale memristor-CMOS circuits is presented. This framework provides a synthesis approach that can be applied to all memristor-based digital logic designs. In particular, it is a proposal for a characterization methodology of memristor-based logic cells to generate a standard cell library file for large-scale simulation. The proposed architecture is based on RRAM and ReRAM redox-based devices and the memristor ratioed logic (MRL) design approach. The proposed framework is implemented in the Cadence Virtuoso schematic-level environment and was verified with Verilog-XL, MATLAB, and the Electronic Design Automation (EDA) Synopses compiler after being translated to the behavioral level. The proposed method can be applied to implement any digital logic design.

In Chapter 6, 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 environment, NC-Verilog and Mat-

lab. The simulation results have indicated that the filter was effectively able to distinguish between all inputs signals and allow only the desired signals to pass through the passband region and reject any signal beyond the stop-band frequency.

In Chapter 6, a summary about the dissertation, conclusion and future work is presented.

7.2 Conclusion

In this dissertation, memristor-based digital systems design and architectures were presented.

a hybrid CMOS-memristor based LFSR is implemented by the MRL design method. In the first step, a D-latch is designed using 8 memristors and 6 CMOS devices. A D-flipflop is implemented with 16 CMOS and 16 memristors subsequently. The proposed LFSR requires 64 CMOS and 64 memristors. The functionality of the proposed LFSR was confirmed by simulation. This design shows how important the significant reduction of power consumption comparing with similar pure CMOS designs. Moreover, the proposed design uses small numbers of transistors comparing with conventional CMOS-based LFSR, which means it will occupy small layout

In another memristor-based circuit design, a memristive median filter was designed with Pt/TaOx/Ta device and it was verified and tested in Cadence environment, Verilog-XL and Matlab.

Further, The performance for restoring original images Lena, Bridge and Pepper from corrupted ones with 10% upto 50% random-valued impulse noise shows better PSNR results comparing with AMF and better denoising performance comparing with CWM algorithm for Lena image. Furthermore, it has comparable MAE results comparing with SAMF, EMF, and CAFS. Due to the high density offers by utilizing hybrid CMOS/Memristor-based design and promising results in restoration of the noisy images, the proposed memristive median filter design can be an alternative to be applied in future image processing applications.

A general methodology framework for designing large-scale CMOS/memristor-

based circuits for digital logic was designed with Hybrid CMOS/Memristor-based method "Memristor Ratioed Logic (MRL)". In particular, In this method MATLAB, a HDL simulator, the Cadence Virtuoso design environment, and Synopsys software were utilized in this framework. A low-power and high-speed memristor-based parallel 8-bit adder/subtractor and 2D memristive median filter were designed with RRAM and ReRAM devices. They were tested and verified in Cadence Virtuoso, Verilog-XL, Synopses Design Vision, and MATLAB. The low-power, lowarea, and high-speed performance were achieved by generating a standard memristor-based cell library. The proposed design shows very significant enhancement in power consumption and delay compared to equivalent CMOS architecture, EIMF, and with LPAMF designs. Compared to the equivalent CMOS design, the area of the proposed architecture is significantly reduced by 32.79%.

Overall, this thesis work contributes to design and implement of less area and high-speed large-scale memristor-based digital circuits. Moreover, It provides an over-coming solution for memristor-based circuits degradation and fanout.

7.3 Suggested Future Work

The work in this dissertation could be continued as following :

- In this dissertation memristor RRAM and ReRAM redox-based devices and the MRL design approach were utilized to implement different memristor based digital circuits. In addition, a framework for the design and synthesis of large-scale memristor-CMOS circuits was proposed.

The design approach in this dissertation was MRL design approach, in the future the aim is to design memristor-based arithmetic circuits using memristors only design Approach.

- The design and mapping of large-scale memristor-based applications is a challenging task due to the lack of comprehensive high-level design tools and simulation platforms. Therefore, this dissertation is the start to continue working to find an automatic mapping methodology of Boolean logic circuits on memristor crossbar.

VITA AUCTORIS

Abubaker Sasi received the B.Sc. degree in electrical engineering from the Institute of comprehensive professionals, Zawia, Libya, in 2003, and M.Sc. degree in electrical engineering from the Lviv Polytechnic National University, Lviv, Ukraine, in 2007. He is now Ph.D. student in the University of Windsor, Canada.

His current research interests include logic design, in-memory computing, RRAM, memristive circuits, and VLSI circuit design.