9-27-2023

New Memristive Architecture for Digital Circuits Design

Farzad Mozafari
University of Windsor

New Memristive Architecture for Digital Circuits Design

By

Farzad Mozafari

A Thesis

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 Master of Applied Science

at the University of Windsor

Windsor, Ontario, Canada

2023

New Memristive Architecture for Digital Circuits Design

by

Farzad Mozafari

APPROVED BY:

_____

Sh. Alirezaee
Department of Mechanical, Automotive and Materials 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

July 20, 2023

DECLARATION OF CO-AUTHORSHIP/PREVIOUS PUBLICATION

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

Chapter 3 of this thesis was co-authored, 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, 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 circuit designed, simulated, and analyzed in term of performance and resources by the author while Dr. Arash Ahamdi, and Prof. Majid Ahmadi provided supervision and edited 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.

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.

This thesis includes 4 original papers that have been previously published/accepted in conferences for publication, as follows:

Thesis Chapter	Publication Title	Publication status
Chapter 2	*F. Mozafari, Majid Ahmadi, Arash Ahmadi, M_J Sharifi, "A Novel Architecture for Memristor-Based Logic ", Proceedings the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), East Lansing MI, USA, Aug. 9-11, 2021, pp. 812-815.*	*Published*
Chapter 3	*Farzad Mozafari, Majid Ahmadi, Arash Ahmadi, "Design and Implementation of Full Adder Circuit Based on VTM-Logic Gates", to appear in the proceedings of the IEEE International Symposium on Circuits and Systems (MWSCAS), Phoenix, Arizona, USA, 2023.*	*In press*
Chapter 4	*Farzad Mozafari, Majid Ahmadi, Arash Ahmadi, "Design of A New Memristive-Based Architecture Using VTM Method", The IEEE International Symposium on Circuits and Systems (ISCAS), Austin Texas, USA May 28 - June 1, 2022, pp. 980-984.*	*Published*
Chapter 5	*Farzad Mozafari, Majid Ahmadi, Arash Ahmadi, " A Programmable Circuits Based on the Combination of VTM Cellular Crossbars", Proceedings of 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, July 3-5, 2023, Portugal, pp. 1-4.*	*Published*

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

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

Before 1971, all the electronics were based on three basic circuit elements. Until a professor from UC Berkeley reasoned that another basic circuit element exists, which he called memristor; characterized by the relationship between the charge and the flux-linkage. A memristor is essentially a resistor with memory. The resistance of a memristor (Memristance) depends on the amount of current that is passing through the device. In 2008, a research group at HP Labs succeeded to build an actual physical memristor. HP's memristor was a nanometer scale titanium dioxide thin film, composed of two doped and undoped regions, sandwiched between two platinum contacts. After this breakthrough, a huge amount of research started, with the aim of better realization of the device and discovering more possible applications of the memristor. 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 researcher's 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 elements 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 new Memristive structure for digital circuit design and to overcome some of the Memristor based logic design issues using convert Voltage to Memristance (VTM) method.

Keywords: Memristor, logic gates, convert voltage to Memristance (VTM), cellular crossbar arrays.

ACKNOWLEDGEMENTS

I would like to thank Dr. Majid Ahmadi for his advice and assistance. His thoughts, remarks, and recommendations have been essential. I record my deep sense of gratitude to my co-advisor, Dr. Arash Ahamdi, for his continuous suggestions in directing my research work. I would also like to thank Dr. Mohammed Khalid and Dr. Shahpour Alirezaee from my committee for their insightful remarks, recommendations, and feedback. Last but not the least; I would like to thank my family for their unconditional love, support, and spiritual guidance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CMOS	Complementary Metal-Oxide-Semiconductor.
VLSI	Very Large Scale Integration.
IMPLY	Material Implication.
MAGIC	Memristor Aided LoGIC.
MRL	Memristor Ratioed Logic **.**
MCM	Memristor-CMOS.
VTM	Voltage To Memristance.
SPICE	Simulation Program with Integrated Circuit Emphasis.
CIM	Computation-In-Memory.
1M1R	One-Memristor-One-Resistor.
BCI	Brain-Computer Interface.
RAM	Random-access memory.
NCCA	NAND Memristor-Based Cellular Crossbar Array.
ACCA	AND Memristor-Based Cellular Crossbar Array.
FPGA	Field Programmable Gate Arrays.
DRAM	Dynamic Random Access Memory.
FA	Full Adder.

Chapter 1:

Introduction And Preliminaries

In this chapter, we will begin by providing a comprehensive summary of the design of Memristor-based digital systems, including their architectures and applications. Subsequently, we will delve into various types of Memristor-based logic designs, providing detailed explanations for each logic type. Moving on to the second part of the chapter, we will engage in a thorough discussion concerning the challenges and design issues that arise in the field of Memristor-based logic design. Additionally, we will conduct a comprehensive literature review of advanced solutions proposed by researchers and practitioners in this domain.

Finally, in Section 1-8, we will present our own proposed solutions, building upon the existing knowledge and addressing the identified challenges to contribute further to the field of Memristor-based logic design.

1-1 Definition of Memristor:

The Memristor, known as the fourth basic element or the missing link, was discovered by Professor Leon Chua, from University of California Berkeley in 1971 [1, 2]. Previously, to describe the relationship between the four corner stones, the magnetic flux ϕ, electric charge q, current i, and voltage v, only the basic circuit elements: the resistor R, the inductor L, and the capacitor C have been used. But then based on the symmetry and the fact that the relations between four factors are described using three elements, he proposed that there exists another basic circuit element and called it "Memristor: the missing link". He even made a prototype of its active realization on a breadboard which was far away from being miniaturized. (Fig 1-1)

In 2008, a research group at HP Labs lead by Stanley Williams succeeded to fabricate the device in nanometer scale. Since then, the research being conducted on Memristors gained momentum and the number of publications has boosted quite rapidly.

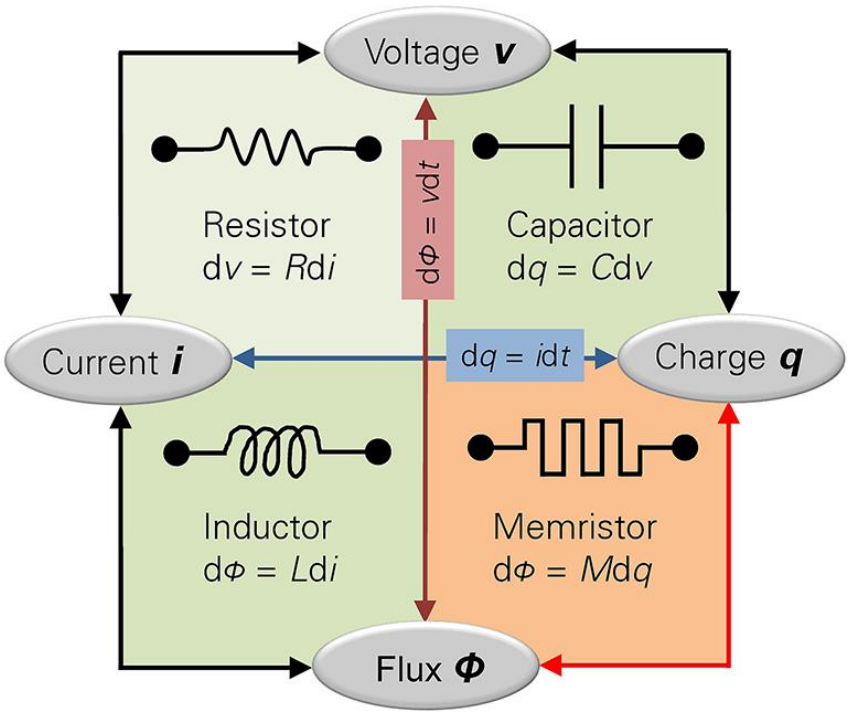


Fig. 1-1: Conceptual symmetries of resistor, capacitor, inductor, and memristor [1]

This Passive component with two-terminals has 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 ash 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.

Memristance (M) is defined as the functional relationship between charge and magnetic flux. The resistance of the Memristor will increase when the electrical charge flows through the Memristor in one direction and decreases when the current flows in the other direction. If the power shut down, which means the applied voltage turns off, the resistance of the Memristor freezes? In some sense the Memristor “remembers” its last resistance (state). If the power turns back on again, the resistance of the Memristor starts exactly from where

it was turned off. This is exactly the property that makes the Memristor exceptional. The symbol of a Memristor is shown in Figure 1.2.

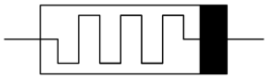


Fig. 1-2. Symbol of a Memristor [2]

If the flux and charge relationship is expressed as a function of charge, then it is a charge-controlled Memristor and when it is expressed as a function of the flux, it is a flux-controlled Memristor.

For a charge controlled Memristor,

$$\varphi = f(q) \tag{1-1}$$

Differentiating (1-1) yields:

$$\frac{d\varphi}{dt} = \frac{df(q)}{dq} \cdot \frac{dq}{dt} \tag{1-2}$$

Having the voltage as v(t)= d φ /dt and the current as i(t)= dq /dt , (1-2) can be rewritten as:

$$v(t) = M(q)\, i(t) \tag{1-3}$$

Where:

$$M(q) = \frac{df(q)}{dq} \tag{1-4}$$

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

$$q = f(\varphi) \tag{1-5}$$

Differentiating (1-5) yields

$$\frac{dq}{dt} = \frac{df(\varphi)}{d\varphi} \cdot \frac{d\varphi}{dt} \tag{1-6}$$

Having the current as i(t)= dq/dt and the voltage as v (t)= dφ/dt, (1-6) can be written as:

$$i(t) = W(q)\ v(t) \tag{1-7}$$

Where:

$$W(\varphi) = \frac{df(\varphi)}{d\varphi} \tag{1-8}$$

W(φ) is called the Memductance, and similar to conductance, it has the units of Siemens.

1-2 Why Memristor:

Memristors offer several compelling advantages, making them an attractive choice in various applications. Here are some reasons why Memristors are gaining increasing attention:

1. Scaling ability: Memristors have the potential for significant scaling, which means they can be fabricated at smaller sizes without losing functionality. This scalability makes them suitable for advanced nanoscale electronic devices and integrated circuits.
2. Non-volatile memory: Memristors have a unique property of retaining their resistance state even when the power is turned off. This non-volatility enables them to be used as non-volatile memory elements, which can store data even without continuous power supply, leading to more energy-efficient systems.
3. Low power consumption: Memristors generally consume lower power compared to traditional memory technologies, making them ideal for energy-efficient computing and storage devices.
4. High-speed operation: Memristors can switch between different resistance states rapidly, enabling faster data processing and reducing latency in memory access.

5. CMOS compatibility: Memristors can be integrated into existing CMOS processes, allowing them to be incorporated into conventional semiconductor fabrication techniques, making it easier to adopt them in existing systems.
6. Long-term data retention: Memristors offer excellent data retention capabilities, making them suitable for applications requiring long-term storage, such as archival storage and data backup systems.
7. Potential for neuromorphic computing: Due to their ability to mimic synapses' behavior in the human brain, Memristors hold promise for neuromorphic computing, enabling efficient and parallel processing for artificial intelligence and machine learning applications.
8. Fault tolerance: Memristors have the potential to be used in fault-tolerant systems due to their resistance state stability, which can enhance the reliability and robustness of computing systems.

Overall, the unique properties of Memristors make them a promising candidate for future memory and computation technologies, enabling advancements in various fields, such as artificial intelligence, Internet of Things (IoT), and high-performance computing. As research and development in Memristor technology continue, their impact on various industries is expected to grow significantly.

In addition, the computational capabilities of digital computers, based on Metal Oxide Semiconductors (CMOS) transistors, have shown remarkable improvement over the years, largely due to continuous shrinking of transistor dimensions, as predicted by Moore's Law [3]. However, the Von Neumann architecture, which involves assembling hardware components into a computer system, has undergone substantial changes since its inception in 1945. This architecture's modularity of engineering design has allowed thousands of engineers to build systems without needing to understand each component independently. With the advent of the internet of things, there has been a significant increase in data quantity, leading to crucial demands for improved energy consumption and processing speed for data-centered activities. The drawbacks of conventional digital computers have become a growing concern [4]. Leakage currents become problematic as the scaling limit is reached for the channel length and transistor gate dielectric thickness. The continuous data transfer between data processing and memory units in the Von Neumann architecture

results in a dramatic reduction in speed and energy performance, contributing to significant delays. To address these issues, enhanced systems have been introduced to boost programming skills and performance. Multi-core Graphics Processing Devices (GPUs) and high-performance interconnections are successful attempts to increase parallelism in computers [4, 5]. As an alternative to traditional programming schemes, in-memory computing, where processing takes place at the location where data is stored, has re-emerged. Memristors, electronic devices also known as resistance switches, have internal resistance states that depend on the past voltage or current applied to them.

Although FinFET architecture has extended the scaling limitations of conventional CMOS transistors, it is still facing significant challenges such as doping damage, logic chip design space restriction, electrostatic limitations, and integration challenges [2, 3]. As a result, there is a high demand for substitutes to CMOS technology. Various alternative technologies exist, such as Double-Gate Tunnel FET, nanotube programmable devices, graphene transistors, and Memristor devices [4 - 7]. Among these, Memristor devices show the most promise due to their great scaling ability, long-term data storage, low-power consumption, and compatibility with CMOS [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].

1-3 Memristor Models:

To be able to design, analyze and simulate Memristor based circuits and applications, a proper model is needed. Since May 2008, that HP Labs published their paper on Memristor implementation, several models have been proposed. In this report the major presented Memristor models will be pointed out.

1-3-1 Linear Ion Drift Model:

In order for HP to understand what they have built, they presented this model based on the physical structure of the device. It is assumed that the physical device of width D has two regions. One side has oxygen vacancies or in other word, is doped with positive oxygen ions; and the other side is undoped.

Each region is modeled with a resistor. The doped region of width W (which acts as the state variable), has lower resistance and hence more conductive, and the undoped region has high resistance. It is also assumed ohmic conductance, and that the field is uniform, the ion drift is linear, and the ions have equal average ion mobility μ_v. Figure 1-3 shows the model of the Memristor built by HP in 2008.

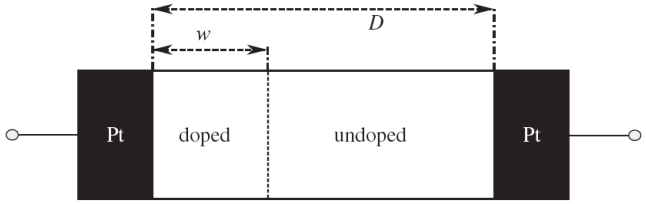


Fig. 1-3: Memristor model presented by HP [4]

There is a linear relationship between the drift and the diffusion velocities and the electric field, if we assume a uniform electric field across the device, the state equation can be written as:

$$\frac{1}{D}\frac{dw(t)}{dt} = \frac{R_{ON}}{\beta} i(t) \tag{1-9}$$

$$\frac{w(t)}{D} = \frac{w(t_0)}{D} + \frac{R_{ON}}{\beta} \cdot q(t) \tag{1-10}$$

Where $w(t_0)$ is the initial length of w. The amount of charge needed to move the boundary from $t_0(w(t_0))$ to $t_D(w(t_D))$ is defined as $Q_D = i \times t$. Hence, $Q_D = \beta / R_{ON}$, and (1-10) can be written as:

$$\frac{w(t)}{D} = \frac{w(t_0)}{D} + \frac{q(t)}{Q_D} \tag{1-11}$$

Substituting x(t)= w(t)/D , yields:

$$x(t) = x(t_0) + \frac{q(t)}{Q_D} \tag{1-12}$$

Where $q(t)/Q_D$ represents the amount of charge passing through the channel.

As it is stated by Strukov et al. in [12], we have:

$$v(t) = \left(R_{ON}\frac{w(t)}{D} + R_{OFF}\left(1 - \frac{w(t)}{D}\right)\right) i(t) \tag{1-13}$$

Or using x(t)= w(t)/D, it will be:

$$v(t) = \left(R_{ON} \cdot x(t) + R_{OFF}(1 - x(t))\right) \cdot i(t) \tag{1-14}$$

R_{ON} is the resistance when w(t)=D, and R_{OFF} is the resistance when w(t)=0, the term in the parentheses is the Memristance value or M(q).

$$M(q) = R_{ON}\frac{w(t)}{D} + R_{OFF}\left(1 - \frac{w(t)}{D}\right) \tag{1-15}$$

If we then assume $r = R_{OFF}/R_{ON}$ and $q(t_0)=0$, so that $w(t)= w(t_0) \neq 0$, then we can rewrite (1-15) as:

$$M_0 = R_{ON}\left(x(t_0) + r(1 - x(t_0))\right) \tag{1-16}$$

Where, M_0 is the Memristance value at t_0.

Figure 1-4 shows the current-voltage characteristic of the Memristor for three different frequencies.

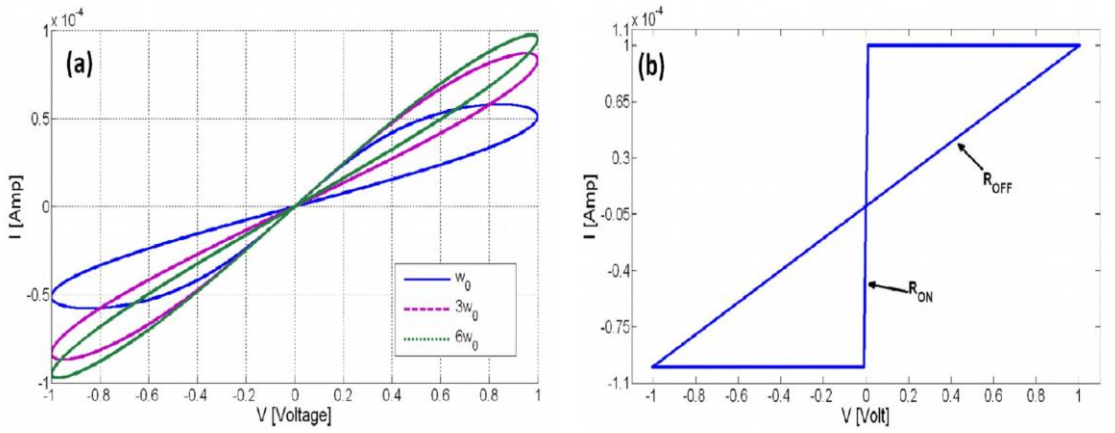


Fig. 1-4: The pinched hysteresis curve of a linear ion drift memristor for (a) sinusoidal waveform input in three different frequencies and (b) rectangular waveform input. It is clear that by increasing the frequency, the hysteresis shrinks. [12]

The hysteresis loop always crosses the origin due to no phase shift between current and voltage. Since there is an inverse relation between flux and frequency, at very high frequencies Memristor will practically act like a resistor. Figure 1-5 shows the hysteresis characteristics of the Memristor which has a varying resistance. This model assumes that the vacancies have freedom to move around the entire length of the device. Hence, one of the advantages of this model is its closed form solution and ease of use.

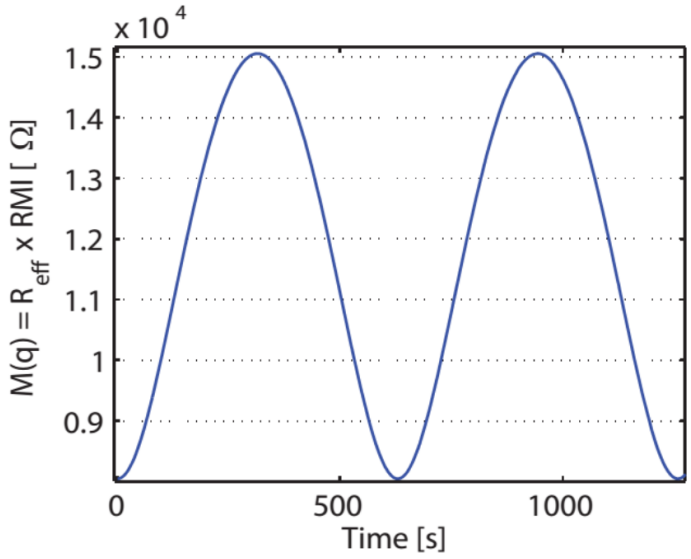


Fig. 1-5: The Memristance versus time, which shows the hysteresis characteristics of the Memristor. Depending on different parameters like RON and ROFF the Memristance value changes from a very low to a very high resistance [12].

1-3-2 Nonlinear Ion Drift mode

In the nonlinear ion drift model, a voltage controlled Memristor is assumed to have nonlinear dependency between the voltage and the internal state derivative. This model also assumes asymmetric switching behavior. The linear drift model produces the hysteresis characteristics at the Memristor, but it also has some limitations regarding basic electrodynamics. Studies and experiments have proved that the behavior of the implemented Memristors are quite nonlinear and the linear ion drift model is not accurate enough. For some applications like logic circuits, nonlinear characteristics are needed. Therefore, more suitable models are developed.

Basically, applying a small voltage across a thin film structure (e.g. 10 nm), will cause a very large electric field (e.g. 10^6 V/cm), which can create considerable nonlinearities in ionic transport. Consequently, it results in quick and significant reduction in energy barrier. These nonlinearities show themselves at the two ends of the device, where the boundary between the two regions gradually stops. In other words, when the Memristor is set to ON or OFF state, no other external stimulus can switch back the state. This is called the nonlinear dopant drift phenomenon. It should be noted that the ion boundary will not move all the way to each side of the device (it moves nonlinearly). If it does (e.g. undoped region occupies the whole device), it means that there will be no physical oxygen vacancies in the device and the length of the doped region is zero, which doesn't make sense. Similarly, the doped region cannot take up all the device length; since it will leave no undoped region and the device will not work. Therefore, the state variable should be between the boundaries $0<w_D<D$ or normalized within the interval [0.1]. Lehtonen et al. [13] proposed a model based on the results of [14]. The current-voltage relationship of this model is described by:

$$i(t)=w(t)^n \underbrace{\beta \sinh(\alpha v(t))}_{1} + \underbrace{\chi\left[\exp(\gamma v(t))-1\right]}_{2} \tag{1-17}$$

Where α, β, γ, and χ are experimental fitting parameters, and n determines how the state variable can affect the current. Here, the state variable w is normalized within the interval [0,1]. The model shows asymmetric switching behavior, in a way that during the ON state, w is near one and the first term of (1-17), (pointed by 1) is the dominant part of the current, which is a tunneling phenomenon. During the OFF state, w is near zero and the second term, (pointed by 2) has the dominant part of the current, which is similar to an ideal diode equation. The state variable differential equation is written as:

$$\frac{dw}{dt} = a \cdot f(w) \cdot v(t)^m \tag{1-18}$$

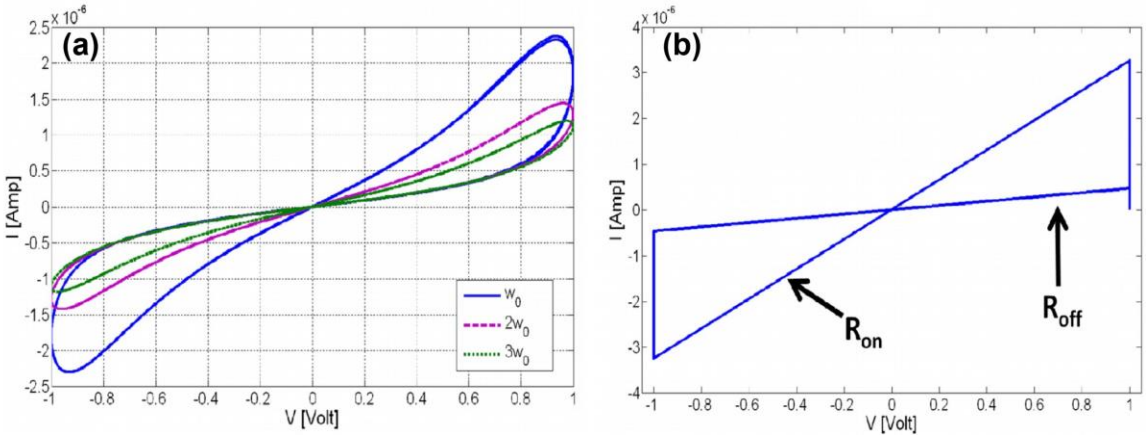


Figure 1-6: Current-voltage characteristic of a nonlinear ion drift model (a) sinusoidal input for three different frequencies and (b) rectangular voltage input. [16]

Where α, *m* are constants, f(*w*) is the window function and *m* is an odd integer. And there is a nonlinear dependency on voltage in (1-18). The current-voltage characteristic of a nonlinear ion drift Memristor for (a) sinusoidal and (b) rectangular inputs can be seen in Figure 1-6.

1-3-3 ThrEshold Adaptive Memristor model (TEAM)

The TEAM model, presented by Kvatinsky et al. [15] is a simple and general model. It represents the same physical model (Simmons tunnel barrier model [16]), but with much simpler expressions.

There are a couple of assumptions for analysis simplification and computational efficiency:

• Below a certain threshold the state variable does not change

• Instead of exponential dependence, there is a polynomial dependence involved between the Memristor current and the internal state drift derivative.

We can fit the TEAM model to any different Memristor model (such as Simmons tunnel barrier model). The dependence of the internal state derivative on current and the state variable itself can be modeled by multiplying two independent functions: one is a function of current and the other is dependent on state variable *x*. Therefore, the derivative of the state variable will be [15]:

$$\frac{dx(t)}{dt}=\begin{cases} k_{off}\cdot\left(\frac{i(t)}{i_{off}}-1\right)^{\alpha_{off}}\cdot f_{off}(x), & 0<i_{off}<i \\ k_{on}\cdot\left(\frac{i(t)}{i_{on}}-1\right)^{\alpha_{on}}\cdot f_{on}(x), & i<i_{on}<0 \\ 0, & otherwise \end{cases} \tag{1-19}$$

Where k_{off} , k_{on} , α_{off} and αon are constants ($k_{off}>0$, $k_{on}<0$). i_{off} , i_{on} are current thresholds and *x* is the internal state variable. $f_{off}(x)$, $f_{on}(x)$ act as the window function, constraining *x* to the bounds [x_{on} , x_{off}] . These two functions do not have to be equal; like the Simmons tunnel barrier model which the dependence on *x* is asymmetric. If we assume that the current-voltage characteristics are like (1-13), the Memristance changes linearly in *x* and we will have:

$$v(t)=\left[R_{ON}+\frac{R_{OFF}-R_{ON}}{x_{off}-x_{on}}(x-x_{on})\right]i(t) \tag{1- 20}$$

But if we use Simmons tunnel barrier current-voltage relationship, any change in the tunnel barrier width, will change the Memristance exponentially.

$$v(t)=R_{ON}e^{\frac{\lambda}{x_{off}-x_{on}}(x-x_{on})}i(t) \tag{1-21}$$

Where λ is a fitting parameter and R_{on}, R_{off} are effective resistances at the bounds, satisfying:

$$\frac{R_{off}}{R_{on}} = e^{\lambda} \tag{1-22}$$

According to [1-16] it is claimed that the TEAM model is accurate enough with a mean error of 0.2% and can boost the simulation runtime by 47.5%. It also satisfies the convergence conditions, computational efficiency required by simulation engines and also the requirements of a Memristive system. An advantage of this model is that it can be fit to any of the existing Memristor models, e.g. the linear ion drift model.

1-4 Memristors applications

Memristors can be used in quite extensive range of applications. In each application, different characteristics are expected from Memristor. For example, in logic and memory applications, an element that has the ability to compute, control and store the data after computation is needed. They need to have fast read and write times. The reading mechanism shouldn't change the data while reading. The difference between stored data should be large enough to avoid bad noise margins and have better sensitivity. Also for storing Boolean data in a Memristor, the ratio between Ron and Roff resistances should be high enough. There are other characteristics that are important for Memristor applications, such as good scalability, low power consumption and compatibility with conventional CMOS. Memristors have shown great promise in various applications, but they also come with their own set of design challenges. Here are some of the key applications of Memristors:

1. Non-volatile memory: Memristors can be used as non-volatile memory elements, providing high-density, low-power, and fast access storage solutions.

2. Neuromorphic computing: Memristors can mimic synaptic behavior in the human brain, making them suitable for building artificial neural networks and enabling efficient and parallel processing for neuromorphic computing.

3. In-memory computing: Memristors allow performing computation at the location where data is stored, reducing the data movement between memory and processing units and improving overall system performance.
4. Analog computing: Memristors can perform analog computations, enabling more efficient and faster signal processing tasks.

In addition, Since May 2008, when HP Labs published their paper in Nature, introducing the discovery of Memristor, a huge wave started in the research and engineering communities around the world to find applications for this new device. Attempts were either for using the Memristor in circuit architecture and use its features or tweaking the old configurations by adding a Memristor or utilizing the new properties of the Memristor and develop a new architecture. As a result, a wide range of articles and papers were published in the last few years. In this chapter, some of the more common or potential applications of the Memristor are described.

Two important characteristics of Memristor that is more interesting for researchers are its nanometer scale and the fact that it can have memory properties and latching capabilities. The nanometer dimensions enable us to build high density memories with less power consumption. Moreover, fabricating devices in nanometer scale is cheaper and easier compared to CMOS fabrication process [17]. Also, with the Memristor memory properties, nanocomputing methods will be possible.

Mazumder, Kang and Waser [18] published an article in a special issue of the Proceedings of the IEEE on Memristor device, models, and application. They presented taxonomy of possible Memristor application that is shown in Figure 1-7.

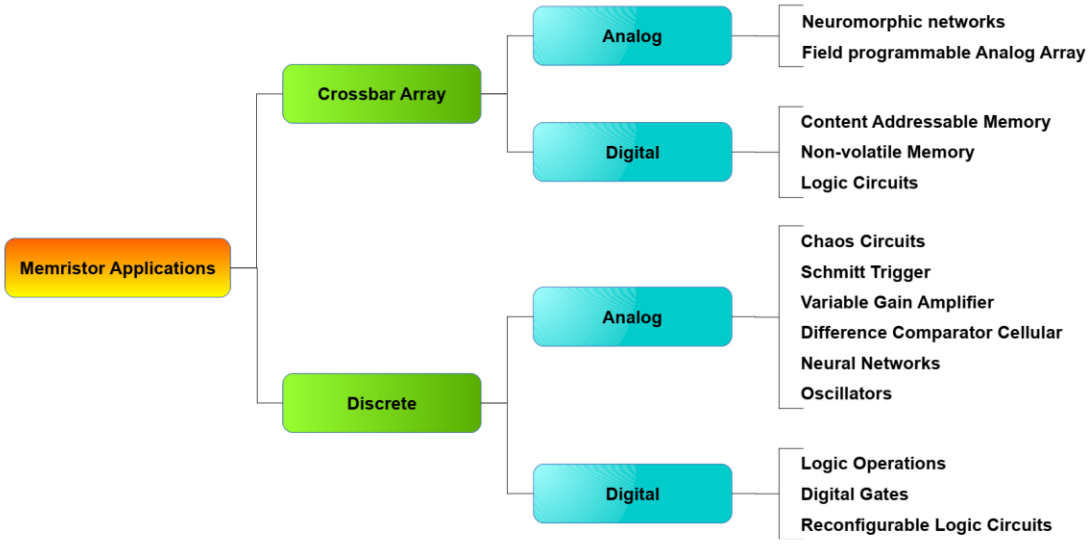


Figure 1-7: The taxonomy of Memristor applications [18]

1-4-1 Non-volatile memory

The first application of Memristor that is closer to reality and will possibly be seen in widespread use in the near future is non-volatile Memristor-based memories. The property of Memristor that can retain its previous state when it is in OFF mode (memory in a sense) is what makes Memristor a good candidate for nonvolatile random-access memories (NVRAMs). So far, there were several papers in the literature about using the Memristor in memory architectures (dense crossbar arrays). The technology for fabrication of 3 nm Memristors is already available. HP Labs fabricated the crossbar latch memory which is still ten times slower than DRAMs.

1-4-2 Resistive RAM (ReRAM)

ReRAMs or Resistance Switching Memories (also referred to as Memristor memories according to [19] is the subject of research in many companies around the world. Companies such as HP, IMEC, Fujitsu, Sharp and Unity Semiconductor has devoted some part of their research to develop different types of Memristive materials. The working

concept of this type of memories is based on the two logical values, “1” and “0” where they correspond to low resistance state (ON mode) and high resistance state (OFF mode) respectively [20].

Several materials were used as transition metal oxides in such structures. Resistive switching was discovered in NiO in 1964 [21]. Other materials were Titanium dioxide (TiO_2), Strontium Titanate ($SrTiO_3$), Zirconia (ZrO_2) and Hafnia (HfO_2). As reported by Xie [22], two switching mechanisms were observed in these materials: The unipolar and bipolar switching mechanisms. The state of the device can be changed by applying a potential across the device. For bipolar switching devices, to switch back to their previous state, a negative voltage should be applied. In case of unipolar switching, the switching voltage should be higher than two threshold voltages to change the state of the device [23],[24],[25].

1-4-3 Crossbar

The crossbar structure is one of the most promising nanostructures at the architectural level [26]. Due to the large number of switches, the crossbar is intrinsically defecting tolerant. It is also nanoscale and has potentially cheaper and easier fabrication process. Other advantages are scalability, flexibility, and its high density. Crossbar grids may be used to perform a large number of computations when research in the area progresses further.

It consists of a grid of nanowires which at every crosspoint or intersection of the horizontal and vertical wires, there is a Memristor switch. The size of each junction is about 2-3 nm (compared to the transistor junctions in the 90 nm technology that is about 60 nm). The fabrication method for implementing the nanowires is nanoimprint lithography (NIL), which is a simple, low cost and high-resolution nanolithography process. The necessary pattern is formed by means of mechanical deformation of an imprint resist (usually a polymer or monomer), where it will be cured using UV or heat in the imprinting phase.

As it is discussed in [27], in 3D stacked structures, extremely high capacity and ultra low power function are main advantages of this structure (or generally ReRAM) over DRAM.

The complexity of addressing grows logarithmically as the capacity grows. Hence, for extremely high capacities, the overhead addressing cost is less [27]. A schematic of the crossbar structure, showing a Memristor switch in each crosspoint is illustrated in Figure 1-8.

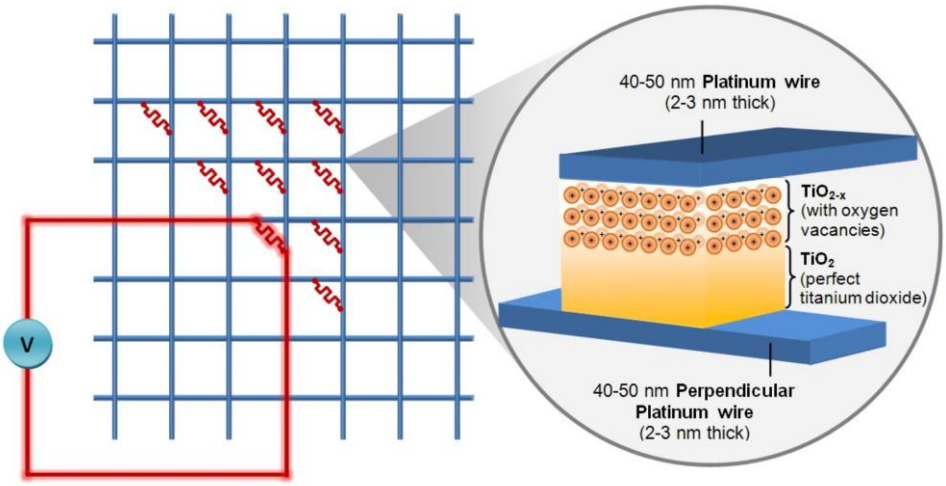


Figure 1-8: Schematic illustration of the connectivity of the crossbar structure [28]

1-4-4 Memory and storage

HP Labs made 3 nm Memristor in April 2010 that could switch in a nanosecond. They also found out that they can stack Memristor layers on top of each other, giving it a density of two to eight times more than the flash memory (or potentially 100 times more). So, having a gigantic amount of capacity of approximately 128 Tbytes/cm^2 (1 Pbits/cm^2) on a single chip can result in retiring hard disks. Imagine a camera memory in the size of a normal thumb drive that can store all the images and videos in a year. Memristor technology can lead to super-small memories that can have a vast amount of applications. A potential usage can be in Smart Dust Networks as a Memory Dust in a sense. Small sensors that need nonvolatile memory units for storing their acquired data might be another possible application.

1-5 Design Challenges of Memristors:

1. Variability and reliability: Memristors can exhibit significant device-to-device variability in their characteristics, which can affect circuit performance and reliability. Addressing this variability is crucial for achieving consistent and reliable operation.

2. Endurance and retention: Memristors may suffer from limited endurance and retention, meaning they have a finite number of write cycles and can lose data over time. Improving the endurance and retention characteristics is essential for long-lasting memory applications.

3. Fabrication reproducibility: Ensuring consistent fabrication of Memristors is challenging, as slight variations during manufacturing can result in significant differences in device behavior.

4. Non-ideal behaviors: Real-world Memristors may not precisely follow ideal mathematical models, leading to non-linearities, non-ideal switching characteristics, and sneak paths in crossbar arrays, which need to be accounted for in circuit design.

5. Material and process compatibility: Integrating Memristors into existing semiconductor processes and materials can be a challenge, as it requires compatibility with established technologies like CMOS.

6. Noise and interference: Memristor devices can be sensitive to noise and interference, which may affect their performance and reliability in practical applications.

7. Power consumption: While Memristors are generally known for their low-power characteristics, achieving the desired power efficiency in large-scale applications is still a challenge.
8. Hybrid integration: Combining Memristors with other emerging technologies, such as 2D materials or quantum devices, presents integration and compatibility challenges.

Despite these challenges, researchers and engineers are actively working to overcome them, and Memristors continue to hold great promise for revolutionizing various aspects of memory, computing, and beyond. As advancements in material science, device physics, and circuit design continue, Memristors are expected to find wider adoption in various applications, driving innovation in the field of electronics and computing.

1-6 Memristor Based Logic Circuits:

Types of implementations of basic logic functions using Memristors:

I. Material Implication (IMPLY)
II. Memristor Aided LoGIC (MAGIC)
III. Memristor Ratioed Logic (MRL)
IV. Memristor-CMOS (MCM)

1-6-1 IMPLY LOGIC

Material implication (IMPLY) [29], often represented by the symbol "→" or "⇒," is a logical connective used in propositional logic and mathematical logic to represent the relationship between two propositions. It is important to note that material implication does not necessarily reflect the everyday usage of the word "implies" or "if-then" statements in natural language.

In propositional logic, the material implication "P → Q" states that if proposition P is true, then proposition Q is also true, or in other words, whenever P is true, Q must be true as well. The only case where the material implication "P → Q" is considered false is when P is true and Q is false. In all other cases (P is false, or both P and Q are true), the material implication is considered true. (Table 1-1)

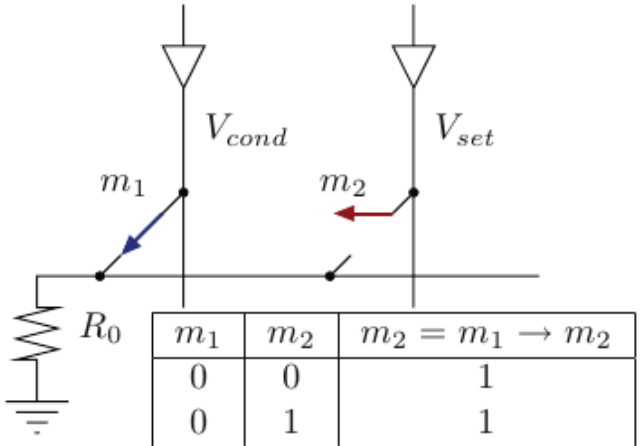
Table 1-1. Truth table to illustrate the material implication (P → Q)

P	Q	P → Q
True	True	True
True	False	False
False	True	True
False	False	True

Now, let's map the truth table to possible states of Memristors. For simplicity, let's assume that the resistance states of Memristors can be categorized into two levels: high resistance (representing 0) and low resistance (representing 1). When P is True (1) and Q is True (1), the Memristor representing "P → Q" should be in a low resistance state (1), indicating that the implication is true. When P is True (1) and Q is False (0), the Memristor representing "P → Q" should be in a high resistance state (0), indicating that the implication is false. When P is False (0), regardless of the state of Q, the Memristor representing "P → Q" should be in a low resistance state (1), indicating that the implication is true.

By setting the resistance states of the Memristors appropriately based on the input values (P and Q), we can realize the material implication operation using Memristor-based logic circuits. It's essential to recognize that material implication is an abstract concept used for formal logic systems and doesn't necessarily capture the full meaning of "implies" or "if-then" statements in natural language, where the context and real-world implications are more nuanced.

Based on Figure 1-9, it is obvious that only when m_1 is OFF, m_2 will stay ON. In all other cases, V_{set} is high enough to turn the device to ON state.



m_1	m_2	$m_2 = m_1 \rightarrow m_2$
0	0	1
0	1	1
1	0	0
1	1	1

Figure 1-9: Material implication using the memristor. [29]

1-6-2 MAGIC LOGIC

The Memristor Aided LOGIC (MAGIC) [30] is a novel computing paradigm that utilizes Memristors where inputs and outputs are treated as resistances. In MAGIC architecture, gate operations are carried out in two sequential stages, showcasing its unique approach to computing. One of the main design goals of MAGIC is to create a system that is compatible with crossbar designs, enabling efficient and compact circuitry.

In the MAGIC system, a high resistance value (R_{off}) is used to represent the logic value '0', while a low resistance value (R_{on}) corresponds to the logic value '1'. MAGIC leverages the properties of Memristors to create a computing system where logic operations are based on the resistance states, opening up new possibilities for energy-efficient computation.

Briefly, the properties of MAGIC Logic are listed below:

1. MAGIC is designed to exclusively utilize Memristors within logic gates.

2. The logical state within a MAGIC gate is symbolized by resistance, with high and low resistances representing logical '0' and '1' (ROFF and RON).
3. The inputs and outputs of these logic gates correspond to the logical states of the Memristors used.
4. Unlike IMPLY logic gates, distinct Memristors are needed for both input and output purposes.
5. In MAGIC gates, inputs originate from the initial logical state of input Memristors, while the output stems from the final state of the Memristor.
6. The operation of a MAGIC gate involves two sequential stages.
7. The initial stage sets the output Memristor to a predetermined logical state.
8. Subsequently, in the second stage, a voltage V_0 is applied across the logic gate. During V_0 application, the voltage across the output Memristor hinges on the logical states of input and output Memristors.

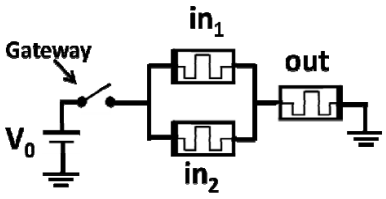


Fig 1-10: MAGIC NOR. Schematic of a two-inputs NOR gate. The logic gate consists of two input Memristors in_1 and in_2 and an output Memristor out. During execution, a voltage V_0 is applied at the gateway of the circuit. [30]

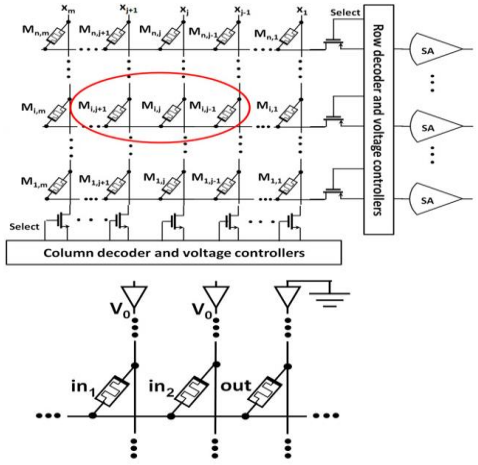


Fig. 1-11. MAGIC NOR gate within a crossbar array. [30]

1-6-3 MRL LOGIC

The Memristor Ratioed Logic (MRL) [31] utilizes CMOS inverter with Memristor to obtain various logic gates. AND and OR gates can be implemented by pure Memrisitive design. NOR and NAND gates are implemented by applying a CMOS-based inverter gate

to the output of Memrisitive OR and AND gates, respectively. In this logic, voltages are considered as logical states. Low voltage and high voltage denote 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 devices is determined by the voltage divider across both Memristors at the end of the logic operation. As shown in Fig. 1-12, the circuit schematics for hybrid CMOS Memristor based OR and AND are displayed.

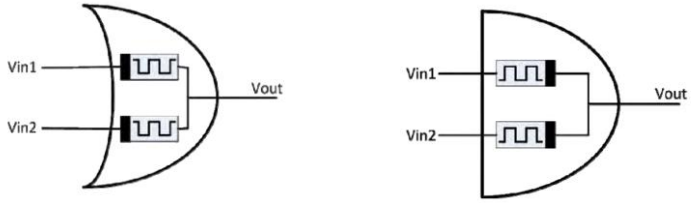


Fig. 1-12: Digital gates based on MRL, a) OR b) AND [31]

1-6-4 MCM LOGIC

Memristor–CMOS (MCM) [32] technology enables fabrication of thin film Memristors over the conventional CMOS devices and has the potential to significantly reduce the silicon-area and propagation delays in VLSI chips. These Memristors not only have an extremely very useful characteristic of non-volatile memory, but also has the advantage of significantly lesser ON resistance R_{on}.

Some of the properties of MCM Logic are listed below:

1. MCM circuits combine CMOS with nano-scale Memristive devices.
2. Fabrication of thin film Memristors over the conventional CMOS devices.
3. Reduction of silicon area.
4. Providing a promising option in the design of MCM based circuits.

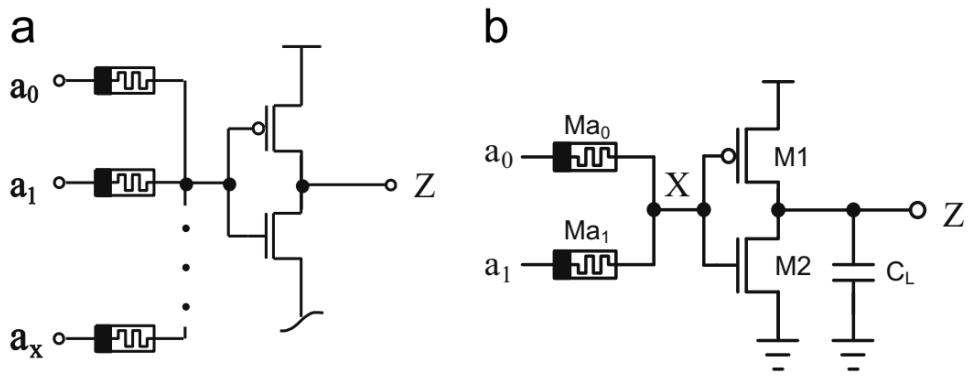


Fig. 1-13: Memristor-CMOS NAND logic (a) multi-input NAND logic (b) 2-input Memristor-CMOS logic gate. [32]

1-7 Memristor-based logic design obstacles

In this section, the main obstacles to logical design based on Memristor are listed:

IMPLY Logic

- Issues of delay and computational complexity due to its sequential nature.
- Requires more than one voltage level.
- Requires more than one clock cycle to execute the computation.
- Requires additional circuit components, such as an additional resistor.
- Requires complicated control circuitry.
- Both inputs and outputs are represented by Memristance values.

MAGIC Logic

- Challenges when connecting two or more logic gates together.
- Required two steps, including an initialization procedure prior to a computational step.

MRL Logic

- It suffers from voltage degradation.
- Both inputs and outputs are represented by voltage signals.

MCM Logic

- Problems of combining CMOS with nanoscale Memristive devices.

1-8 Proposed Solutions

Voltage To Memristance (VTM) Logic Gates

In the proposed method inputs are voltages and outputs are Memristances, which makes them an ideal candidate for interfacing between previously introduced logic circuits. The resistance of output Memristor can be considered as logic values, R_{off} and R_{on}, can be interpreted as logic zero and logic one respectively. Furthermore, all the basic logic gates can be implemented based on this approach. We consider the logical state, as stored data within the output Memristor. The proposed method for implementation basic functions is more complete than the IMPLY and MAGIC logic (NOT, AND/OR, NAND/NOR, XOR & XNOR) For example: (NAND/NOR) stateful logic gates with similar structure require two different input voltages in order to program the gate to NAND or NOR functions. Moreover, compared to the needed operation steps are significantly smaller than that of the IMPLY-based and MAGIC logic gates. Also gates operation performed by a one-step schedule. Proposed new Memristive-based architecture using VTM method, Memristive crossbar arrays can be created using the VTM method. The proposed circuits have the potential to replace volatile memories like RAM and can perform logical calculations within the memory. There is an improvement in the operation steps of the logic gates, resulting in higher switching speed and lower power consumption compared to previous works.

1-9 Outline of thesis and summary of contributions

The objectives of this thesis are to propose a new Memristive structure for designing digital circuits. 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 novel design for memristor-based logic gates. Universal logic gates NAND/NOR, with similar topologies are proposed. The proposed gates have advantages such as higher speed, fewer operation steps, and propagation delay less than its

counterparts. Seven basic logic functions, NOT, AND/OR, NAND/NOR, XOR and XNOR use simple connections among memristors.

In chapter 3, by using VTM based technique, a memristor-based XOR/XNOR logic gates with 5 memristor, which can execute operation in a single step, is presented. As a case study, a single bit full adder circuit by using proposed VTM logic gates is designed. Moreover, the functions of proposed logic gates and full adder are verified by PSPICE, respectively. The proposed full adder architecture based on XOR gates compared to the other full adders, has benefits of simpler architecture, higher speed, and lower power consumption.

Chapter 4 proposes a NAND Cellular Crossbar Array (NCCA) were proposed, which is a small unit for logic in-memory processing that consists of four NAND gates and peripheral circuits. By combining the NCCA, a novel cellular architecture of memristors to perform larger digital computations is proposed. In-memory diagonal computations is an efficient parallel and high-speed calculations method.

In chapter 5, by using VTM-NAND/AND gates, AND Cellular Crossbar Array (ACCA) are proposed. By combining the NCCAs and ACCAs, a novel programmable circuit architecture to perform larger digital computations was proposed. The novelty of the proposed circuit is use of the pipeline technique, which is divided into stages odd and even, and are connected with one another to form a pipe like structure. Pipelining is a technique for breaking down a sequential process into various sub-operations and executing each sub-operation.

1-10 References

[1] L. Chua, “Memristor-the missing circuit element," IEEE Transactions on Circuit Theory, vol. 18, no. 5, pp. 507{519, 1971.

[2] M. Khalid, “Review on various memristor models, characteristics, potential applications, and future works," Transactions on Electrical and Electronic Materials, pp. 1-10, 2019.

[3] G. E. Moore, “Cramming more components onto integrated circuits, reprinted from electronics”, volume 38, number 8, April 19, 1965, pp.114 _." 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-_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 programmable 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] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found,” Nature, vol.453, no. 7191, pp. 80‐83, 2008.

[13] E. Lehtonen and M. and Laiho, “CNN Using Memristors for Neighborhood Connections,” 12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), pp. 1‐4, 2010.

[14] J. J. Yang et al., “Memristive switching mechanism for metal/oxide/metal nanodevices,” Nature Nanotechnology, vol. 3, no. 7, pp. 429‐433, 2008.

[15] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, “TEAM: ThrEshold Adaptive Memristor Model,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. -, p. -, 2012.

[16] J. G. Simmons, “Generalized formula for the electric tunnel effect between similar electrodes separated by a thin insulating film,” Journal of Applied Physics, vol. 34, no. 6, pp. 1793‐1803, 2004.

[17] D. B. Strukov and K. K. Likharev, “CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices,” Nanotechnology, vol. 16, p. 888, 2005.

[18] P. Mazumder, S. M. Kang, and R. Waser, “Memristors: Devices, Models, and Applications,” Proceedings of the IEEE, vol. 100, no. 6, pp. 1911‐1919, 2012.

[19] L. Chua, “Resistance switching memories are memristors,” Applied Physics A, vol. 102, no. 4, pp. 765‐783, 2011.

[20] O. Kavehei, “Memristive Devices and Circuits for Computing, Memory, and Neuromorphic Applications,” PhD thesis, The University of Adelaide, December 2011.

[21] J. F. Gibbons and W. E. Beadle, “Switching properties of thin NiO films,” Solid-State Electronics, vol. 7, no.11, pp. 785–790, 1964.

[22] Y. Xie, “Modeling, Architecture, and Applications for Emerging Memory Technologies,” Design Test of Computers, IEEE, vol. 28, no. 1, pp. 44-51, 2011.

[23] O. Kavehei, A. Iqabal, Y. S. Kimi, K. Eshraghian, and S. F. Al-Sarawi, “The Fourth Element: Characteristics, Modeling and Electromagnetic Theory of the Memristor,” Proceedings of the Royal Society, vol. 466, no. 2120, pp. 2175-2202, 2010.

[24] O. Kavehei, “Memristive Devices and Circuits for Computing, Memory, and Neuromorphic Applications,” PhD thesis, The University of Adelaide, December 2011.

[25] O. Kavehei et al., “Fabrication and modeling of Ag/TiO2/ITO memristor,” IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1-4, 2011.

[26] R. I. Bahar et al., “Architectures for silicon nanoelectronics and beyond,” Computer, vol. 40, no. 1, pp. 25-33, 2007.

[27] D. L. Lewis and H.-H. S. Lee, “Architectural evaluation of 3D stacked RRAM caches,” IEEE International Conference on 3D System Integration, 3DIC, pp. 1-4, 2009.

[28] R. Williams, “How We Found The Missing Memristor,” Spectrum, IEEE, vol. 45, no. 12, pp. 28-35, 2008.

[29] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "‘Memristive’switches enable ‘stateful’logic operations via material implication," Nature, vol. 464, p .2010, 873. (IMPLY)

[30] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, et al., "MAGIC—Memristor-aided logic," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, pp. 895-899 ,2014. (MAGIC)

[31] 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), 2013, 12th International Workshop on, pp. 1-6, 2012. (MRL)

[32] K. Cho ,S.-J. Lee, and K. Eshraghian, "Memristor-CMOS logic and digital computational components," Microelectronics Journal, vol. 46, pp. 214-220, 2015. (MCM)

Chapter 2:

A Novel Architecture for Memristor-Based Logic

2-1 Introduction

Nanoscale devices such as Memristors have a great deal of scope for the next-generation VLSI systems. A Memristor is a nonlinear two-terminal electrical component which was introduced by Leon O. Chua in 1971 [1,2] as the fourth fundamental circuit element. The most interesting property of this new device is its memory-resistance (Memristance) characteristic. The major characteristics of Memristor are non-volatile memory, high switching speed, high integration density, low energy dissipation, and compatibility with CMOS technology. Memristors can be used for a wide range of applications, including logic gates and circuits [3,4], digital memory [5], and neuromorphic computing [6]. Several methods have been introduced to implement logic functions using Memristor devices.

In previously presented works such as Material Implication (IMPLY) [7] and Memristor Aided LoGIC (MAGIC) [8], only Memristors are used for logic gates implementation, while in Memristor Ratioed Logic (MRL) [9] and Memristor-CMOS (MCM) [10] a composition of Memristor and CMOS devices are used for logic implementation. In IMPLY logic gate implementation, special voltage sequences are applied to the Memristors where the Memristance of the Memristors represents the input and output values of the logic gates. In MAGIC gates, the inputs and outputs are also considered as resistance of Memristors, and gate operation consists of two sequential stages. In this logic the high resistance value (R_{off}) represents logic value of $'0'$, and a low resistance value (R_{on}), represents logic value of $'1'$. MRL is a hybrid CMOS-Memristive logic family that uses Memristors combined with CMOS inverters to design logic gates. In this logic, inputs and outputs are provided as voltage levels and Memristors are used as computational elements. MCM circuits combine CMOS with nanoscale Memristive devices enabling the reduction of silicon area thus providing a promising option in the design of MCM based circuits. It should be noted that all the various methods that have been proposed for designing Memristor-based logic gates, the inputs and outputs were either both voltages (such as MRL) or Memristances (such as IMPLY). This chapter proposes a Memristor-based NAND/NOR logic gates using convert VTM method. In the proposed logic gates, inputs are voltages and outputs are Memristance, which makes them an ideal candidate to be used as an interface between previously introduced logic designs to convert one kind of logic

representation to another. In this proposed method the resistance of output Memristor can be considered as logic values, R_{off} and R_{on}, can be interpreted as logic zero and logic one respectively. Furthermore, all the basic logic gates can be implemented based on this approach. We consider the logical state, as stored data within the output Memristor. The rest of this chapter is organized as follows: Section 2-2, presents the proposed NAND/NOR logic gates using VTM method. Simulation and results are presented in Section 2-3. Finally, Section 2-4 concludes the chapter.

2-2 Proposed NAND/NOR Logic Gates

The proposed NAND/NOR stateful logic gate, is shown in Fig. 2-1. The structure consists of two inputted Memristors M_{in1} and M_{in2} which are connected by two negative polarities, and an output Memristor M_{out}. V_{in1} and V_{in2} are gate input voltages and the Memristance of M_{out} stores the output. Also gate operation is performed by a one-step schedule. To explain the gate operation, let us assume that in initial state of all Memristors are set to R_{on}. V_{th} is the Memristors threshold voltage and V_m is the Memristors common node voltage. There are four cases for the write operation with this NAND/NOR gate.

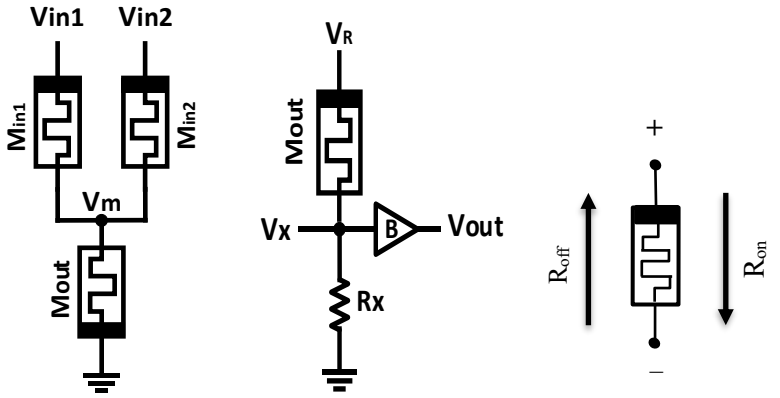


Fig. 2-1: a) Proposed NAND/NOR gates using convert VTM method b) Read circuit for the proposed Memristor-based gates c) polarity of the Memristor

2-2-1 Write operation.

Case 1: Both inputs are set to zero volt, no current flows through the Memristors, and the logic value stored in M_{out} does not change.

Case 2 & 3: The inputs change to opposite levels, V_{in1}= +V and V_{in2}= 0, by considering, Kirchhoff's voltage law and initial Memristance M_{in1}=M_{in2}= R_{on}, one can write:

$$V_m = \frac{(R_{on} \| R_{on})}{(R_{on} \| R_{on}) + R_{on}} V = \frac{\frac{1}{2} R_{on} V}{\frac{3}{2} R_{on}} = \frac{1}{3} V \qquad (2\text{-}1)$$

Case4: Both inputs are set to +V, V_m can be determined by:

$$V_m = \frac{\frac{1}{2} R_{on} V}{\frac{3}{2} R_{on}} + \frac{\frac{1}{2} R_{on} V}{\frac{3}{2} R_{on}} = \frac{2}{3} V \qquad (2\text{-}2)$$

To implement NAND operation, in cases 2&3, the relationship between the Memristor threshold voltage (V_{th}) and the common node voltage (V_m) must satisfy equation (2-2), so that the value of the initial output Memristance does not change.

$$V_m < V_{th},\ Vin < 3V_{th} \qquad (2\text{-}3)$$

In a similar way, to change the gate operation to the logical NOR gate:

$$V_m > V_{th},\ Vin > 3V_{th} \qquad (2\text{-}4)$$

In the fourth case, for both NAND and NOR gates, the M_{out} must be changed from Ron to R_{off}. As a result:

$$V_m > V_{th},\ Vin > \frac{3}{2} V_{th} \qquad (2\text{-}5)$$

By assuming V_{th} = 1(v):

$$\frac{3}{2} V_{th} < V_{in(NAND)} < 3V_{th} \Rightarrow 1.5 < V_{in(NAND)} < 3 \qquad (2\text{-}6)$$

$$V_{in(NOR)} > 3V_{th} \Rightarrow V_{in(NOR)} > 3 \qquad (2\text{-}7)$$

$V_{in\ (NAND)}$=1.8 (v) and $V_{in\ (NOR)}$= 3.3 (v) are used for the proposed NAND and NOR gates respectively. Nevertheless, in the proposed gates, NAND gate can be used by applying a smaller voltage and the Memristor-based gate serves as a NOR gate by applying larger voltage.

Please note that during NAND/NOR logic write operation, the node voltage V_m cannot be greater than the node voltage V_{in}, and the Memristances of the M_{in1} and M_{in2} do not change. Truth table of proposed NAND/NOR gates is shown in Table 2-1.

TABLE 2-1: Truth table of proposed NAND/NOR gates.

Inputs			Output (NAND)		Output (NOR)	
V_{in1}	V_{in2}	Logic	M_{out}	Logic	M_{out}	Logic
0	0	“00”	R_{on}	‘1’	R_{on}	‘1’
0	+V	“01”	R_{on}	‘1’	R_{off}	‘0’
+V	0	“10”	R_{on}	‘1’	R_{off}	‘0’
+V	+V	“11”	R_{off}	‘0’	R_{off}	‘0’

2-2-2 The Read operation.

Here, the read operation in the proposed NAND/NOR digital logic gates are explained. By using the read circuit for the proposed Memristor-based gates, the value of resistance stored in the output Memristor can be converted to voltage. It should be noted that the output Memristance value should not change during the reading process. The read circuit configuration is depicted in Fig. 2-1. The proposed circuit will avoid changing the state of the output Memristor after a read operation. The structure consists of an output Memristor M_{out}, which is connected in series to an external resistor (R_x). A buffer (B) is set to isolate the input voltage (V_x) from the output voltage (V_{out}), which can be applied to the inputs of

next gates. Also buffer circuit is to compensate for the voltage drop that usually occurs in Memristor circuits. V_R is designed read voltage required for read operation, which is determined considering V_{in}.

Two cases for the read operation are considered:

Case 1: $M_{out} = R_{off}$

$$V_x = \frac{R_x}{R_x + M_{out}} V_R, \quad V_x > V_R - V_{th} \qquad (2\text{-}8)$$

Using values V_R =1.8 (v), V_{th} = 1 (v), R_{off} =100 (kΩ) for the proposed NAND/NOR gates. R_x can be determined:

$R_x > 80$ (kΩ)

It is also assumed for the NAND/NOR gates

R_x= 90 (kΩ). (2-9)

Case 2: $M_{out} = R_{on}$

By considering the polarity of the Memristor shown in Fig. 2-1.c, even if V_R is bigger than the threshold voltage of the output Memristor, the state of M_{out} does not change.

2-3 Simulation Results

In this section, SPICE has been used to simulate the proposed Memristor-based NAND/NOR logic gates. A bipolar Memristive system with threshold Memristor model [11] is used in the simulations.

2-3-1 NAND write logic operations.

Fig. 2-2 shows the simulated results of the proposed two inputted NAND circuit, for all four possible input combinations. Gate output is the value of resistance stored in the output Memristor. M_{out} shows output Memristance changes. The initial state of the storage Memristor is ′1′.

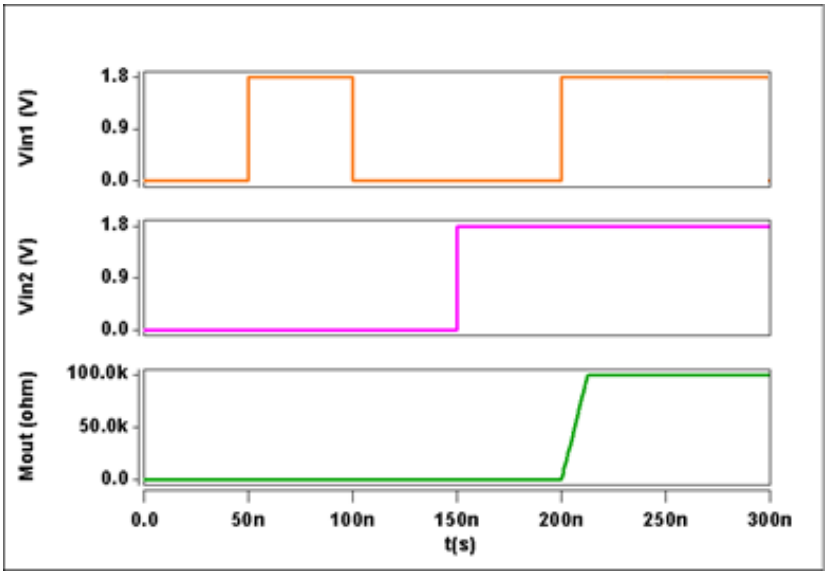


Fig. 2-2: Writing simulation results for the NAND operation.

2-3-2 NOR write logic operations.

Fig. 2-3 shows the simulated results of the two inputted NOR gate. All four different combinations of the input voltages are applied to the proposed gate. Simulation results clearly show the changes in the internal state of the output Memristor to the desired logic.

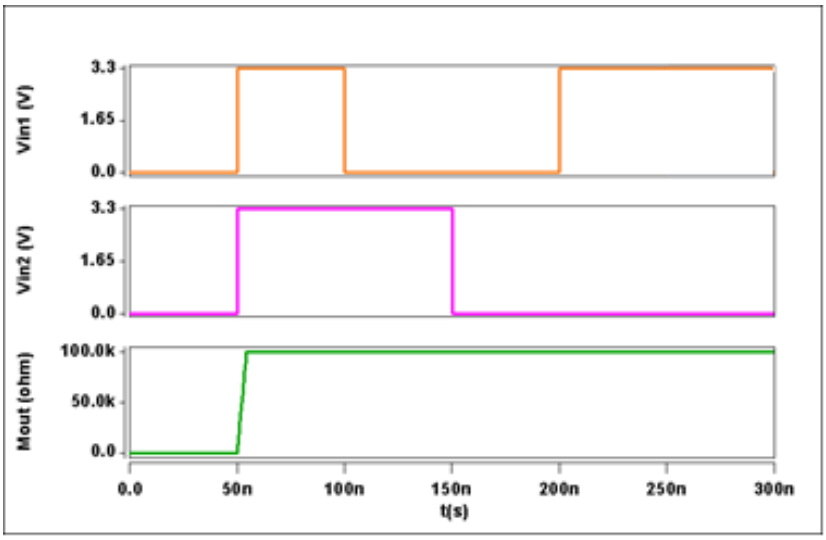


Fig. 2-3: Writing simulation results for the NOR operation.

2-3-3 Read circuit simulation results.

In the read process, as shown in Fig. 2-4, the input voltage has a positive pulse with a magnitude 1.8 (v). By considering $M_{out}= R_{off}$, the simulation result clearly shows the read processes for logic zero.

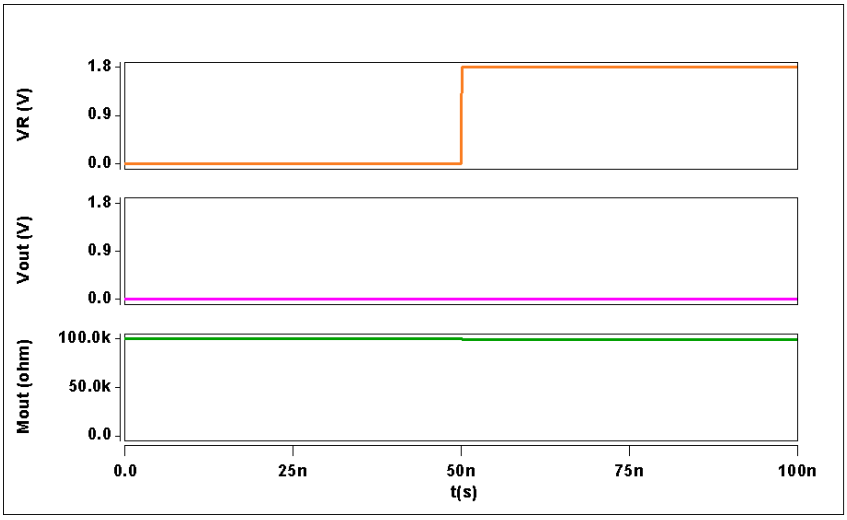


Fig. 2-4: Simulation results of proposed circuit reading ($M_{out}=R_{off}$).

When $M_{out}=R_{on}$, as depicted in Fig. 2-5, with time range from 0 to 50 (ns), no current flows to the output Memristor, therefore $V_{out}= 0$ (v). The stored logic value in the output Memristor is read out from 50 to 100 (ns), and V_{out} is equal of V_R. The simulation result shows the read processes for logic one.

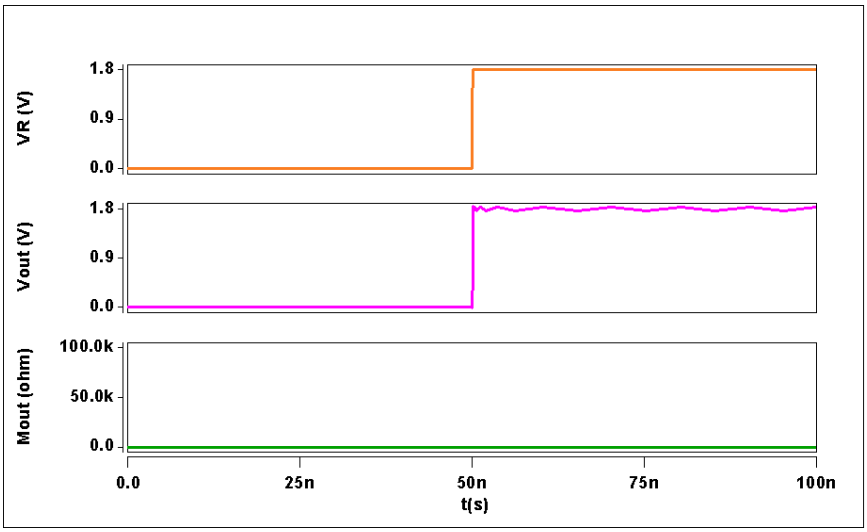


Fig. 2-5: Simulation results of proposed circuit reading ($M_{out}=R_{on}$).

Furthermore, the proposed Memristor-based logic gates are compared to the IMPLY-based [12] and MAGIC [4] logic gates, as shown in Table 2-2. One can see that the ability of the proposed method to implement basic functions is more than the IMPLY and MAGIC logic. Moreover, compared to the needed operation steps are significantly smaller than that of the IMPLY-based and MAGIC logic gates.

Table 2-2: Comparison Between IMPLY, MAGIC and VTM.

	IMPLY	MAGIC	VTM
No. of voltages	2 (V_{SET},V_{COND})	1 (V_0)	2 (V_{in1},V_{in2})
Separate input & output	No	Yes	Yes
Basic Functions	IMPLY (+ FALSE)	NAND, NOR, AND, OR,NOT	NAND/NOR, AND/OR, NOT, XOR, XNOR
No. of Memristors for NAND/NOR	3 (+ a resistor)	3	3
No. of clock cycles for NAND/NOR (Operation Step)	4	2	1
Within Memory	Yes	Yes (for NOR)	Yes (NAND/NOR, AND/OR)
Logically complete	Require FALSE	Yes (NOR, NAND)	Yes (NAND, NOR, AND, OR)

2-3-4 One-bit full adder – A case study

Considering the universality of the proposed gates, more complex circuits can be developed with higher speed and lower energy consumption. For example, a single-bit full adder is designed in Fig. 2-6, using the proposed NAND gates (see fig. 2-1a).

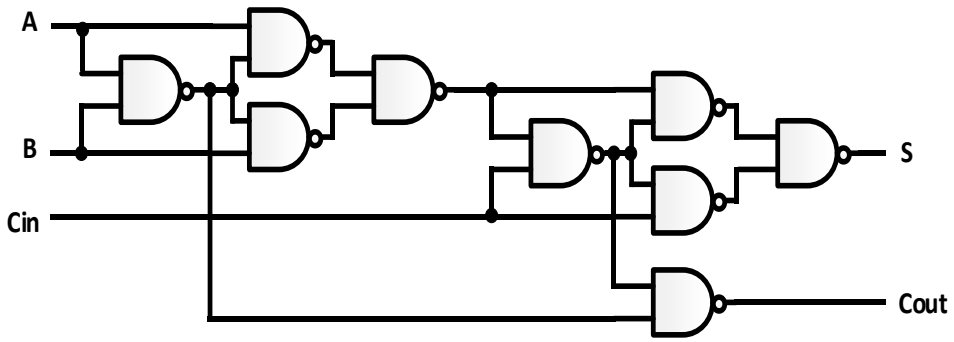


Fig. 2-6: Logic schematic of one-bit Full adder NAND equivalent.

In comparison with the proposed 1-bit CMOS-based full adder structures in [13], the proposed circuit is capable of having approximately 20% higher switching speed, 44% lower connections, and 28% lower power consumption. Table 2-3. shows the simulation results.

TABLE 2-3: Simulation results for proposed logic compared with CMOS.

Adder circuit	Power supply (v)	Propagation Delay (ps)	Static power (µW)	Dynamic power (pW)	Count (#)
VTM	1.8	202.1	3.99	79.8	27-Mem
CMOS	1.8	251.7	5.542	374.0	28-Tr

2-4 Conclusion

In this chapter, we presented a novel design for Memristor-based logic gates. Universal logic gates NAND/NOR, with similar topologies were proposed. The proposed gates have advantages such as higher speed, fewer operation steps, and propagation delay less than its counterparts. Seven basic logic functions, NOT, AND/OR, NAND/NOR, XOR and XNOR use simple connections among Memristors. As a case study, a single bit full adder is designed on the proposed cellular structure. In comparison with the 1-bit CMOS-based full adder, the proposed circuit can have approximately, 20% higher switching speed, 44% lower connections, and 28% lower power consumption.

2-5 References

[1] L. Chua, "Memristor-the missing circuit element," IEEE Transactions on circuit theory, vol. 18, pp. 507-519, 1971.

[2] I. Vourkas and G. C. Sirakoulis" ,Emerging memristor-based logic circuit design approaches: A review," IEEE Circuits and Systems Magazine, vol. 16, pp. 15-30, 2016.

[3] A. Madhavan, T. Sherwood, and D. B. Strukov, "High-Throughput Pattern Matching with CMOL FPGA Circuits: Case for Logic-in-Memory Computing," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, pp. 1-14, 2018.

[4] A. K. Maan, D. A. Jayadevi, and A. P. James, "A survey of memristive threshold logic circuits," IEEE transactions on neural networks and learning systems, vol. 28, pp. 1734-1746, 2017.

[5] Y.-F. Chang, F. Zhou, B. W. Fowler, Y.-C. Chen, C.-C. Hsieh, L. Guckert, et al., "Memcomputing (Memristor+ Computing) in Intrinsic SiO x-Based Resistive Switching Memory: Arithmetic Operations for Logic Applications," IEEE Transactions on Electron Devices, vol. 64, pp. 2977-2983, 2017.

[6] Y .V. Pershin and M. Di Ventra, "Neuromorphic, digital, and quantum computation with memory circuit elements," Proceedings of the IEEE, vol. 100, pp. 2071-2080, 2012.

[7] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'Memristive'switches enable 'stateful'logic operations via material implication," Nature, vol. 464, p .2010, 873. (IMPLY)

[8] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, et al., "MAGIC—Memristor-aided logic," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, pp. 895-899 ,2014. (MAGIC)

[9] 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), 2013, 12th International Workshop on, pp. 1-6, 2012. (MRL)

[10] K. Cho ,S.-J. Lee, and K. Eshraghian, "Memristor-CMOS logic and digital computational components," Microelectronics Journal, vol. 46, pp. 214-220, 2015. (MCM)

[11] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable SPICE simulations of memristors, memcapacitors and meminductors," arXiv preprint arXiv:1307.2717, 2013.

[12] S. Kvatinsky, 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) (in press).

[13] I. Hussain, S. Chaudhury, Performance comparison of 1-bit conventional and hybrid full adder circuits, in Advances in Communication, Devices and Networking. Lecture Notes in Electrical Engineering, eds. by R. Bera, S. Sarkar, S. Chakraborty, vol. 462 (Springer, Singapore, 2018)

Chapter 3:

Design and Implementation of Full Adder Circuit Based On VTM-Logic Gates

3-1 Introduction

The rapid progress of electronic technology has given rise to smaller and faster devices, primarily driven by Silicon based semiconductor technology such as CMOS. However, this trend towards miniaturization has led to new challenges and problems that impede the development of even smaller devices, including current leakage, high power densities, and the high cost of testing and manufacturing. To address these challenges, researchers have proposed variety of potential solutions including new devices such as Memristors [1,2].

Memristors are devices that behave like classic resistors but exhibit variable resistance, known as Memristance, when an electrical charge passes through them. These devices have the ability to retain their final Memristance value as memory, even when there is no electrical charge present, making them non-volatile memory components. Memristors offer several advantages, including good scalability, compatibility with CMOS devices, and the ability to enhance existing technologies such as memory, logic, and neuromorphic devices [3,4].

In addition, Memristors are gaining attention for their potential in solving challenges faced by the semiconductor industry, including current leakage, high power densities, and high costs of manufacturing and testing. One such application is in the implementation of logic gates, which is of particular interest due to its potential to reduce the number of Memristive devices, increase speed, and lower power consumption.

Furthermore, Memristors have the potential to enable promising applications such as crossbar architectures, neuroinformatic, brain-computer interface (BCI), and neuroprosthetics [5]. In addition, Memristors can be utilized to implement computing functionalities and stateful Boolean logic operations. Several techniques have been introduced for implementing logic gates using Memristive components, including Material Implication (IMPLY) [6], Memristor Aided LoGIC (MAGIC) [7], Memristor Ratioed Logic (MRL) [8], and Memristor-CMOS (MCM) [9]. While some techniques rely solely on Memristors for logic gate implementation, others combine Memristors with CMOS devices.

Memristor-based IMPLY logic operation provides a way to realize logic gates, including the XOR logic operation, which is an essential logic function for many other operations. The traditional CMOS XOR logic gate requires ten transistors, while the IMPLY-based XOR gate requires only five Memristors and 13 steps, which has been further improved to only six devices and two steps by Shin et al. [10]. Recently researchers have also developed a hybrid Memristor-CMOS XOR gate that requires only one Memristor and four voltage-controlled switches, providing a potential solution to reduce the number of devices and power consumption of XOR gates [11]. Given the significance of XOR gates as a base operation in computing, the development of XOR gates with fewer devices, higher speed, and lower power consumption is crucial.

Adders are essential components of Computation-In-Memory (CIM) architecture and variety of Memristor-based adders have been proposed using two main approaches [12-14]. The first approach involves using a special number system and multilevel storage based on Memristors, which divides Memristance into levels, with each level representing a certain value [12]. However, this approach has a drawback as the resistance values around the margin are possible to be inaccurate. The second approach involves using logic operations to create a full adder, which can overcome the previous drawback. However, this approach requires multistep operations, leading to complicated circuits and high-power consumption [13,14].

This chapter introduces a new design for XOR/XNOR logic gates using only 5 Memristors in a single step. In addition, a full adder circuit is designed based on the proposed logic gates, with advantages including simpler architecture, higher speed, lower voltage, and power consumption. The inputs of the logic gates are voltages, while the outputs are Memristance values R_{off} and R_{on}, that can be interpreted as logic zero and logic one. This chapter is structured into four sections, including the introduction, the proposed circuit design, simulation results and analysis, and a conclusion. The proposed circuit design and simulation results will be presented in Section 3-2 and 3-3, respectively. Section 3-4 concludes the chapter.

3-2 Proposed XOR and XNOR Logic Gates

3-2-1 Memristor-based XOR:

The proposed stateful XOR logic gate is shown in Fig. 3-1. The structure consists of 5 Memristors. There are four Memristors as inputs (M_1, M_2, M_3 and M_4) and one Memristor as the output (M_{out}). In_1 and In_2 inputs of the gate are voltage values, where the output is stored as the Memristance of M_{out}. In this approach, the gates' operation is carried out in a single step. To explain the gate operation, let us assume that the input Memristors are preset to the initial value of R_{on} and the output Memristor has the initial value of R_{off}. There are three cases for the write operation with the proposed XOR logic gate.

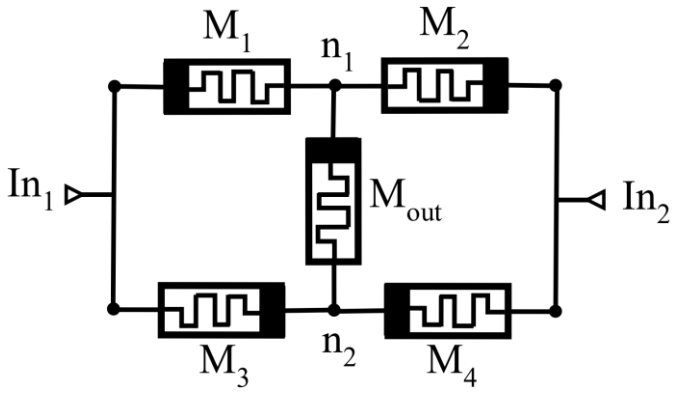


Fig.3-1: Proposed XOR logic gates using convert VTM method

3-2-1-1 Write operation for XOR.

Case 1: Both inputs are set to zero or set to +V volt, the voltage of nodes n_1 and n_2 are equal and as a result no current flows through the output Memristor and the logic value stored in M_{out}, does not change, which means logic '0'.

Case 2: The inputs change to opposite levels, Iin_1= +V and Iin_2= 0, the current direction will be such that the Memristances M_2 and M_3 increase and become to R_{off}, also the Memristances M_1 and M_4 decrease and change to R_{on}. As a result, it can be clearly shown that the voltage of node n_1 is higher than the voltage of node n_2, and with the condition that

the difference between these two voltages is greater than the V_{th} (V_{th} is the Memristors threshold voltage), the output Memristance change to R_{on} or logical '1'.

Case 3: $Iin_1 = 0$ and $Iin_2 = +V$, the current direction will be such that the Memristances M_1 and M_4 increase and become R_{off}, then the Memristances M_2 and M_3 decrease and change to R_{on}. As a result, in this case voltage of node n_1 is higher than the voltage of node n_2, and with the condition that the difference between these two voltages is greater than the threshold voltage of the Memristor (V_{th}), it causes the output Memristance change to Ron or logical '1'. The truth table of proposed XOR gate is shown in Table 3-1.

Table 3-1: XOR gate truth table designed by VTM method.

Input			Output	
I_{in1}	I_{in2}	Logic	M_{out}	Logic
0	0	“00”	R_{off}	‘0’
0	+V	“01”	R_{on}	‘1’
+V	0	“10”	R_{on}	‘1’
+V	+V	“11”	R_{off}	‘0’

3-2-2 Memristor-based XNOR:

XNOR logic gate can be implemented by changing direction of the Memristor M_{out} in XOR gate as shown in Fig. 3-2. Similar to the XOR gate, the proposed XNOR gate consists of 5 Memristors. There are four input Memristors (M_1, M_2, M_3 and M_4) and one Memristor as the output (M_{out}). The only difference between XOR and XNOR gates is in the polarity of the output Memristor. Analogous to the XOR gate, here also there are three cases for the write operation. To explain the gate operation, let us assume that the input Memristors are preset to the initial value of R_{off} and the output Memristor has the initial value of R_{on}.

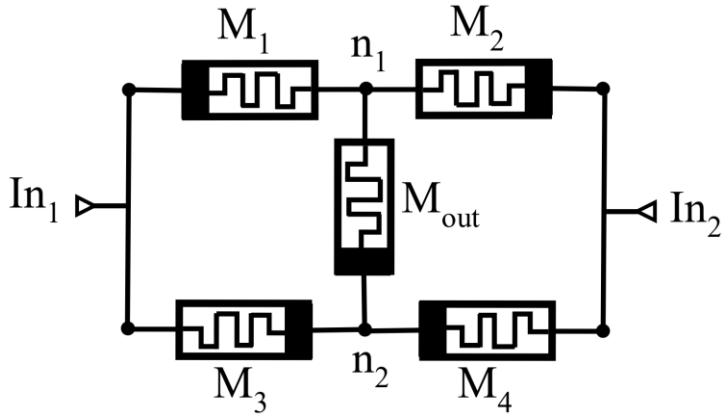


Fig. 3-2: Proposed XNOR logic gates using convert VTM method.

3-2-2-1 Write operation for XNOR.

Case 1: Both inputs are set to zero or set to +V volt, the voltage of nodes n_1 and n_2 are equal to each other and the logic value stored in M_{out}, does not change (R_{on}).

Case 2: The inputs change to opposite levels, Iin_1= +V and Iin_2= 0, the current direction will be such that the Memristances M_2 and M_3 increase and become Roff and the Memristances M_1 and M_4 decrease and change to Ron. Considering that the difference between the voltage of node n_1 and the voltage of node n_2 is greater than the threshold voltage (V_{th}), it follows that the M_{out}, change to R_{off}.

Case 3: Iin_1= 0 and Iin_2= +V, the current direction will be such that the Memristances M_1 and M_4 increase and become R_{off} and the Memristances M_2 and M_3 decrease and change to Ron. Considering (Vn_1 - Vn_2) > $V_{th,}$ the output Memristance change to R_{off} or logical '0'. Truth table of proposed XNOR gates is shown in Table 3-2.

Table 3-2: XNOR gate truth table designed by VTM method.

Input			Output	
I_{in1}	I_{in2}	Logic	M_{out}	Logic
0	0	“00”	R_{on}	‘1’
0	+V	“01”	R_{off}	‘0’
+V	0	“10”	R_{off}	‘0’
+V	+V	“11”	R_{on}	‘1’

3-2-3 Read operations for proposed XOR/XNOR gates.

In this section, the read operations for the proposed XOR/XNOR are explained. By using the read circuit for the proposed Memristor-based gates, the Memristance stored in the output Memristor should be converted to voltage. The proposed circuit will avoid changing the state of the output Memristor after a read operation. The structure consists of an output Memristor M_{out}, which is connected in series to an external resistor (R_x). A buffer (B) is added to isolate the input voltage (V_x) from the output voltage (V_{out}), which is applied to the inputs of next gates. Also, buffer circuits compensate for the voltage drop that usually occurs in Memristor circuits. V_R is designed to read voltage required for read operation, which is determined considering V_{in}. It should be noted that the output Memristance value should not change during the reading process. The read circuit configuration is depicted in Fig. 3-3.

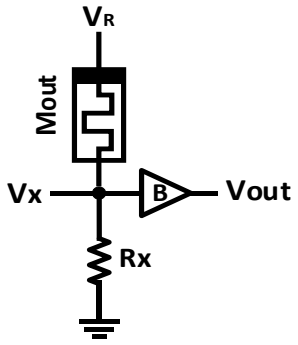


Fig. 3-3: Read circuit for the proposed XOR/XNOR memristor-based gates.

Two cases for the read operation are considered:

Case 1: $M_{out} = R_{off}$

$$V_x = \frac{R_x}{R_x + M_{out}} V_R, \quad V_x > V_R - V_{th} \qquad (3\text{-}1)$$

By using values V_R =1.8 (v), V_{th} = 1 (v), R_{off} =100 (kΩ) for the proposed XOR/XNOR gates. R_x can be determined:

Rx > 80 (kΩ)

It is assumed R_x= 90 (kΩ). (3-2)

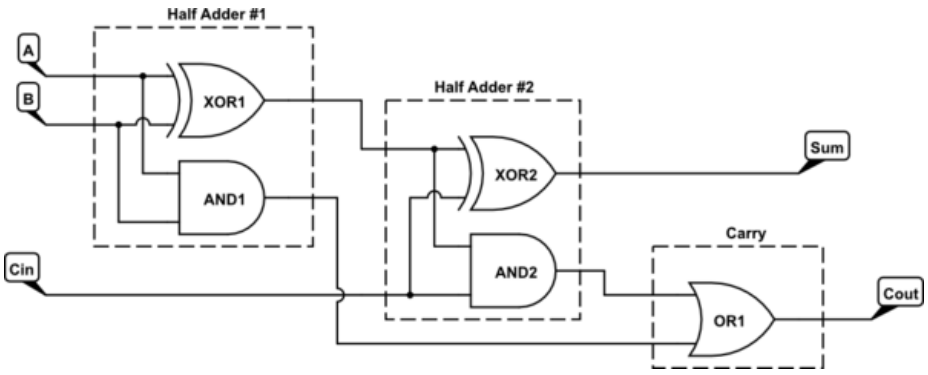
Case 2: $M_{out} = R_{on}$

By considering the polarity of the Memristor, even if $V_R > V_{th(Mout)}$, the state of M_{out} does not change.

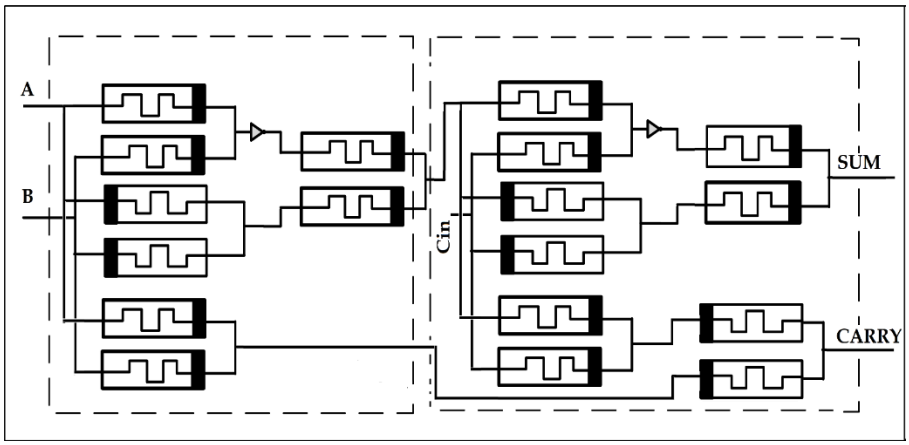
3-2-4 One-bit Full Adder circuit design using VTM logic gates: A case study.

Full adders are one of the most important elements in digital circuits. Therefore, providing an optimal design of this element will significantly help to improve the output parameters of these circuits. By exploiting the properties of the proposed VTM logic gates, a one-bit full adder circuit by using VTM logic gates is proposed in this section. As shown in Fig. 3-4, the proposed adder circuit consists of two proposed Memristor-based XOR, two AND gates and one OR [15] gate.

The logical values of the outputs sum, and the output carry are stored in the output Memristors (M_{out}) of XOR2 and OR1 gates, respectively.



a)



b)

Fig. 3-4: a) Design of proposed full adder. b) By using proposed VTM logic gates.

Truth table of proposed full adder is shown in Table 3-3.

Table 3-3: Truth table for full adder.

Inputs				Outputs	
A	B	C	Logic	Sum	Carry
0	0	0	“000”	R_{off}	R_{off}
0	0	+V	“001”	R_{on}	R_{off}
0	+V	0	“010”	R_{on}	R_{off}
0	+V	+V	“011”	R_{off}	R_{on}
+V	0	0	“100”	R_{on}	R_{off}
+V	0	+V	“101”	R_{off}	R_{on}
+V	+V	0	“110”	R_{off}	R_{on}
+V	+V	+V	“111”	R_{on}	R_{on}

3-3 Simulation Results

In this section, SPICE has been used to simulate the proposed Memristor-based XOR/XNOR logic gates. All the digital gates and proposed circuits are evaluated using an advanced Memristor model with a modified Biolek window and a voltage-dependent variable exponent [16].

3-3-1 XOR write logic operations.

Fig. 3-5 shows the simulated results of the proposed two inputted XOR circuit, for all four possible input combinations. Gate output is the value of resistance stored in the output Memristor. M_{out} shows output Memristance changes.

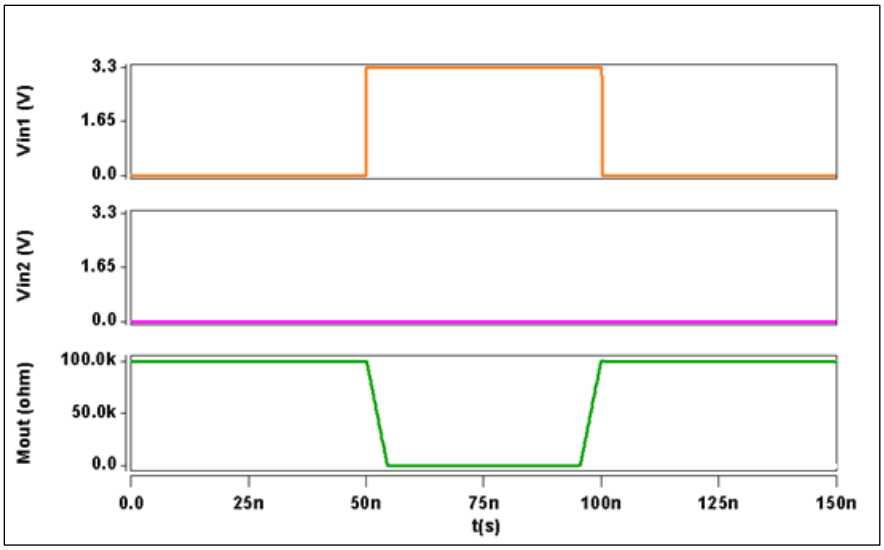


Fig. 3-5: Writing simulation results for the XOR operation

3-3-2 XNOR write logic operations.

Fig. 3-6 shows the simulated results of the two input XNOR gate. All four different combinations of the input voltages are applied to the proposed gate. Simulation results clearly show the changes in the internal state of the output Memristor to the desired logic.

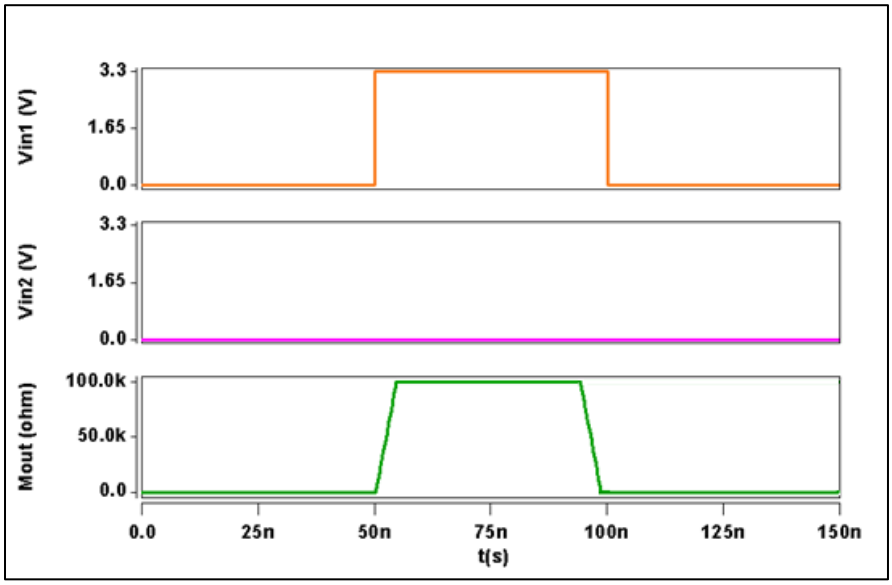


Fig. 3-6: Writing simulation results for the XNOR operation

Table 3-4 presents a comparison of the performance analysis of conventional full adder circuits with other adder circuits in terms of propagation delay, static and dynamic power [17][18]. The propagation delay between inputs and outputs of different adders, including VTM (NAND) [17], CMOS, CPL, and Hybrid logic [18], are compared, which are 202.1, 251.7, 188.0, and 233.64 ps, respectively. It is attainable from the table that the proposed circuit has the lowest propagation delay. Moreover, the static power consumption of the proposed circuit is lower than that of other full adder circuits, including VTM (NAND), CMOS, CPL, and Hybrid logic, with values of 3.99, 5.542, 6.520, and 4.189 μW, respectively [17]. Since dynamic power dissipation is the dominant component of power loss, and the CPL adder has the highest power dissipation due to the increased switching activity of the collector [18]. On the other hand, the circuit proposed in this paper has the least dynamic power dissipation.

Finally, the transistor counts for CMOS, CPL, and Hybrid adders are observed to be 28, 32, and 16, respectively [17][18], whereas the proposed adder circuit with Memristors exhibits higher speed, lower propagation delay time, and less energy consumption [18]. Notably, all simulation results are obtained at a voltage of 1.8 V [18]. Therefore, the proposed circuit can be a suitable alternative for conventional full adder circuits due to its superior performance in terms of propagation delay, static and dynamic power consumption, and transistor count [17][18].

Table 3-4: Simulation results for proposed logic compared with adder circuits at 1.8 (v)

Adder circuits	Power supply (v)	Propagation Delay (ps)	Static power (pW)	Dynamic power (μW)	Count (#)
VTM(XOR, AND, OR) 10-30 nm	1.8	173.4	68.6	3.01	20 Mem
VTM(NAND) 10-30 nm	1.8	202.1	79.8	3.99	27 Mem
CMOS Tec. (180-nm CMOS Level 49 Technology (BSIM))	1.8	251.7	374.0	5.542	28 Tr
CPL (32 nm-UMC technology)	1.8	188.0	1.383 (nW)	6.520	32+6 Tr
Hybrid (55 nm-UMC technology)	1.8	233.64	233.1	4.189	16 Tr

3-4 Conclusion

In this chapter, by using VTM based technique, a Memristor-based XOR/XNOR logic gates with 5 Memristor, which can execute operation in a single step, is presented. As a case study, a single bit full adder circuit by using proposed VTM logic gates is designed. Moreover, the functions of proposed logic gates and full adder are verified by PSPICE, respectively. The performance parameters of the proposed adder circuit such as power (dynamic and static), propagation delay was compared with conventional adders (CMOS, CPL) and combined adders (Hybrid). The proposed full adder architecture based on XOR gates compared to the other full adders, has benefits of simpler architecture, higher speed, and lower power consumption.

3-5 References

[1] L. Chua, "Memristor-the missing circuit element," IEEE Transactions on circuit theory, vol. 18, pp. 507-519, 1971.

[2] I. Vourkas and G. C. Sirakoulis" ,Emerging memristor-based logic circuit design approaches: A review," IEEE Circuits and Systems Magazine, vol. 16, pp. 15-30, 2016.

[3] Y. Li, Y. Zhong, Y. Deng, Y. Zhou, L. Xu, and X. Miao" ,Nonvolatile “AND,”“OR,” and “NOT” Boolean logic gates based on phase-change memory," Journal of Applied Physics, vol. 114, p. 234503, 2013.

[4] A. K. Maan, D. A. Jayadevi, and A. P. James, "A survey of memristive threshold logic circuits," IEEE transactions on neural networks and learning systems, vol. 28, pp. 1734-1746, 2017.

[5] Y .V. Pershin and M. Di Ventra, "Neuromorphic, digital, and quantum computation with memory circuit elements," Proceedings of the IEEE, vol. 100, pp. 2071-2080, 2012.

[6] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "‘Memristive’switches enable ‘stateful’logic operations via material implication," Nature, vol. 464, p .2010, 873.(IMPLY)

[7] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, et al., "MAGIC—Memristor-aided logic," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, pp. 895-899 ,2014.(MAGIC)

[8] 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), 2013, 12th International Workshop on, pp. 1-6, 2012.(MRL)

[9] K. Cho ,S.-J. Lee, and K. Eshraghian, "Memristor-CMOS logic and digital computational components," Microelectronics Journal, vol. 46, pp. 214-220, 2015.(MCM)

[10] Shin S, Kim K, Kang S M. "Memristive XOR for resistive multiplier" Electronics letters, 2012, 48(2): 78-80.

[11] Zhou Y, Li Y, Xu L, et al. A hybrid memristor-CMOS XOR gate for nonvolatile logic computation[J]. physica status solidi (a), 2015.

[12] El-Slehdar A A, Fouad A H, Radwan A G. Memristor based N-bits redundant binary adder. Microelectronics Journal, 2015, 46(3): 207-213.

[13] Yang Y, Mathew J, Pontarelli S, et al. Complementary Resistive Switch Based Arithmetic Logic Implementations Using Material Implication. IEEE Transactions on Nanotechnology, DOI 10.1109/TNANO.2015.2504841

[14] Siemon A, Menzel S, Waser R, et al. A complementary resistive switch-based crossbar array adder. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2015, 5(1): 64-74.

[15] Farzad Mozafari, Majid Ahmadi, Arash Ahmadi, “Design of A New Memristive-Based Architecture Using VTM Method“, The IEEE International Symposium on Circuits and Systems (ISCAS), 2022.

[16] Y. Mladenov, V.; Kirilov, S. Advanced Memristor Model with a Modified Biolek Window and a Voltage-Dependent Variable Exponent. Inform. Autom. Pomiary Gospod. Ochr. Sr. (IAPGO ´ S) 2018, 8, 15–20.

[17] F. Mozafari, Majid Ahmadi, Arash Ahmadi, M_J Sharifi, “A Novel Architecture for Memristor-Based Logic “, The IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021.

[18] Inamul Hussain, Saurabh Chaudhury, “Performance Comparison of 1-Bit Conventional and Hybrid Full Adder Circuits “, Advances in Communication, Devices and Networking, 2018, Volume 462, ISBN: 978-981-10-7900-9

Chapter 4:

Design of A New Memristive-Based Architecture Using VTM Method

4-1 Introduction

Recently, various types of digital and analog circuits based on Memristors have been proposed. Furthermore, various applications such as In-Memory Computing (IMC), reconfigurable digital circuits, neuromorphic circuits, image processing have been considered. The advantage of IMC is not only in the processing and storage of data on a single platform, but also parallel and complex computations can be performed in simple crossbars architecture.

Simultaneously with the development of some new non-volatile memories, an IMC architecture has been proposed to overcome the Von Neumann bottleneck [1]. Non-volatile memories include resistive memory (specifically Memristor), phase shift memory [2, 3], and magnetic tunnel junction memory [4]. Among these memory devices, Memristor will be the best choice due to its high operating speed, low power consumption, high stability, 3D integration and CMOS compatibility [5-7]. Fig. 4-1, shows the general structure of a memory architecture for IMC. All data processing, including writing, reading, and logical operations, can be performed in a single logic-in-memory unit which is dependent on a separate control unit.

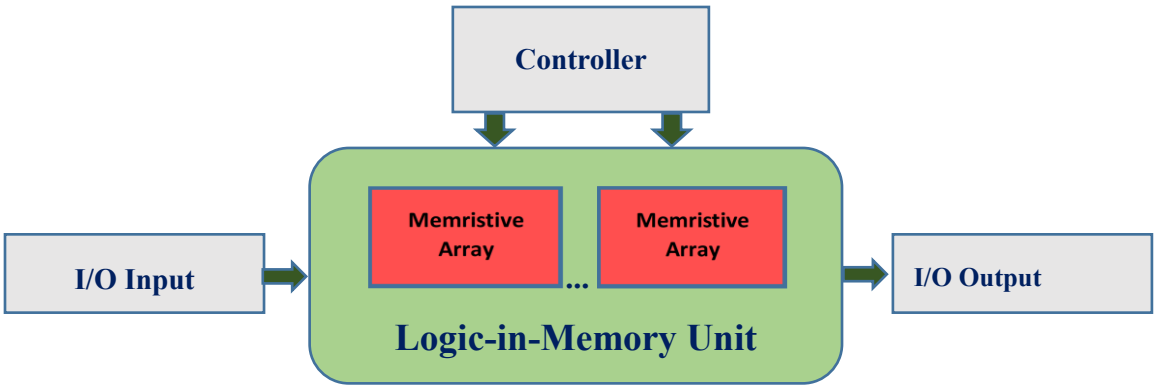


Fig. 4-1. Memory architecture for IMC [8].

In FPGAs, many logic blocks are connected to each other with programmable routing, but these connections cannot be controlled during operations [9]. In IMC architecture, several large-scale logical computations are performed in a homogeneous Memristor

crossbars structure. Individual logic operations can be programmed in a structure only by applying required voltages to different Memristors.

Recently, 16 Boolean functions have been implemented and designed in different methods [10, 11]. However, these methods still have several limitations that make them not desirable in practical applications. Firstly, the implementation of various Boolean functions with optimized computational complexities has not yet been properly studied. Secondly, most methods for cascaded designs are difficult due to the different physical states of the input and output. Finally, most studies have focused on single-device computing, while data exchanges and shifting in an array and malfunctions have rarely been investigated.

The authors in [12] used IMPLY logic as the basic logic operation and proposed a dynamically programmable logic method in Memristor crossbar arrays. Memristors in different rows and columns can be coupled and work together to perform the Boolean functions. This paper described how all 16 Boolean logical functions can be obtained with a simple grid of 4 Memristors located in a specific array. In [8], MAGIC logic is proposed for use as logic in the Memristive Memory Processing Unit (MPU), which describes algorithms for fully logical execution within the Memristive MPU with an example of a one-bit full adder.

To design and implement widely used digital circuits such as full adders in crossbar architecture, special methods have been proposed, and one of them is parallel adders [13]. One of the techniques that are considered for IMC requires parallel and large computing for which various solutions such as using CMOL circuits [14] is suggested. Reducing the amount of energy consumption and increasing the design dimensions to increase the desired efficiency are given in [15] and [16] respectively. In addition, crossbars have been developed so that they can perform in-memory computations based on the IMPLY method and the 1M1R solution [17]. This paper proposes a Memristive crossbar arrays by using the Voltage to Memristance (VTM) method. Proposed circuits can replace volatile memories such as RAM and has the ability to perform logical calculations within the memory.

The rest of this chapter is organized as follows: Section 4-2 presents the proposed NAND memristor-based cellular crossbar array. Simulation and results are presented in Section 4-3, and finally conclusions are given in Section 4-4.

4-2 Proposed AND/OR Memristor-Based Logic Gates & Cellular Crossbar Array

The stateful VTM logic gates in [18] are capable of being used in large structures such as Memristive-based crossbar arrays. To achieve this aim, it is focused on the NAND gate due to its lower input voltage (V_{in}= +V=1.8 volt) requirement.

4-2-1 Proposed NAND Memristor-Based Cellular Crossbar Array

As shown in Fig. 4-2, the proposed NAND Cellular Crossbar Array (NCCA) structure consists of four NAND gate cell in each row. In_1 and In_2 are connected to the applied voltages at columns one and two, while the result of the NAND logical operation is stored in last column (output Memristor) as a Memristance. In addition, all the Memristors are initialized to R_{on} and the initial input values of Memristors do not change after performing various digital processes.

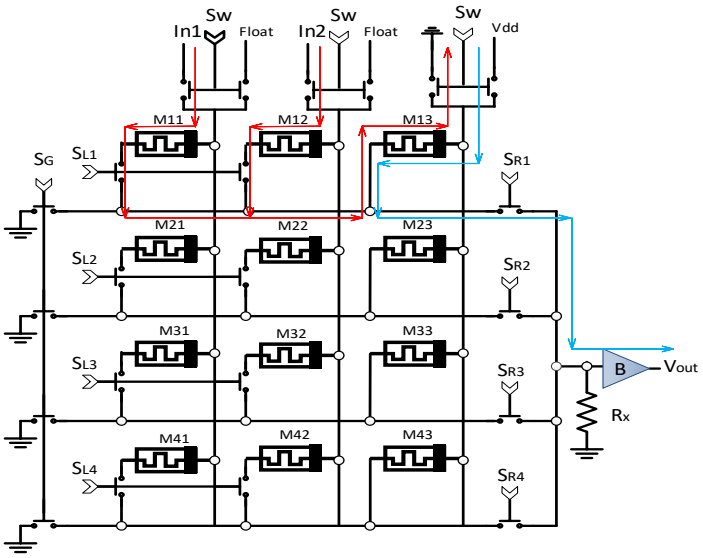


Fig. 4-2: Proposed NAND Cellular Crossbar Array (NCCA).

S_w is writing control switch, which is selected to allow writing operation by applying to input voltages. S_{L1}, S_{L2}, S_{L3} and S_{L4} are write operation control switches, which determine in each cycle which gates to perform the NAND operation. Also, S_{R1}, S_{R2}, S_{R3} and S_{R4} are read operating control switches, which are used to read logical states stored data within output Memristors (blue current path in Fig. 4-2). Finally, the output configuration of Memristors switches (S_G) to set initial state of output Memristors to R_{on}. For example, if the first and second column switches are connected to In_1 and In_2, and the third column connected to '0', first row NAND gate in the crossbar is started for write operations (current paths is shown as red in Fig. 4-2).

In the proposed NCCA, a switch is used to solve the sneak path current issue, also makes each gate in different rows independent of other NAND gates. In fact, input Memristors are series with an NMOS transistor, which acts as a switch.

4-2-2 Diagonal Computation

By combining the NCCAs, a new concept named cellular configuration is proposed, which can perform IMC. To prevent the gates from affecting each other within cellular configuration and to be able to convert the output Memristance to voltage simultaneously, the gates are selected diagonally within the cellular crossbars. The proposed cellular configuration has considerable lower wiring density, and increases the density of logic gates. As shown in Fig. 4-3 this configuration consists of 12 NAND gates. The gates highlighted by yellow, red, blue, and purple are selected diagonally and consecutively. By activating switches S_{L1}, S_{L2}, S_{L3} and S_{L4}, network performs write operation on NAND gates ($S_R =' 0', S_W = '1'$). By using proposed read-out circuits [18], stored logical state within output Memristors are converted to voltage for each row. ($S_R =' 1', S_W =' 1'$).

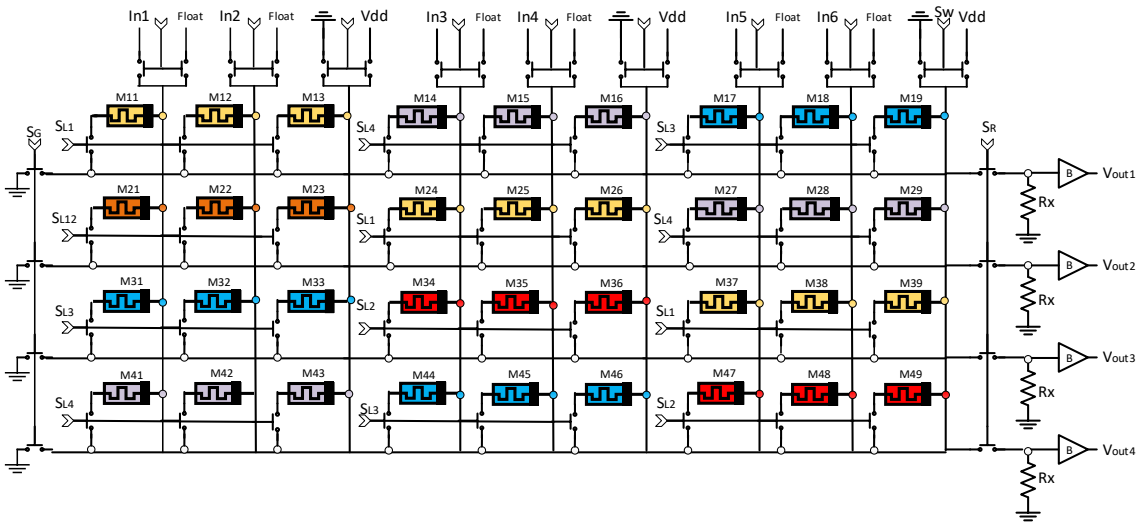


Fig. 4-3: NCCA with diagonal computations

4-2-3 Design and Implementation of Logic Function Using NAND Cellular Crossbar Arrays.

By considering the inputs and output of Fig. 4-2, the schematic of the NCCA can be shown in Fig. 4-4.

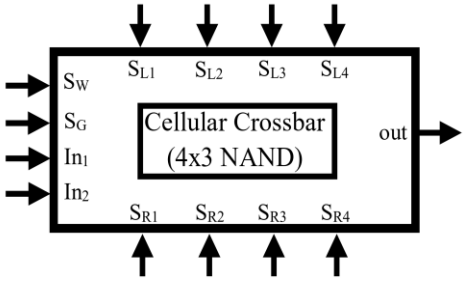


Fig. 4-4: Schematic of proposed NCCA.

In the following, as a case study, the following digital function is implemented using three NCCAs.

$$z = ab + c' \qquad (4\text{-}1)$$

To implement “z” in Eq. (4-1), conversion to NAND operation is done as:

$$z = ((ab)'c)' \tag{4-2}$$

Eq. (4-2), can be written in a simpler form below:

$$z = ((a\ NAND\ b)\ NAND\ c) \tag{4-3}$$

Fig. 4-5, shows the circuit designed for Eq. (4-3). The proposed circuit has three inputs and one output. For every 3 different inputs, there will be 2^3 different modes in the output. Three NCCAs are used to represent the function “z”, because all 8 output modes can be stored in the output Memristors of the last two units of Fig. 4-4. It should be noted that all 4 possible output states of the two-inputted NAND gates are stored in the first NCCA. Therefore, all the three capabilities of Memristors, consisting of reading, writing and the storing element, have been utilized. If the two input circuits, a and b are removed, the NAND operation of inputs is still stored in the output Memristors of the first NCCA.

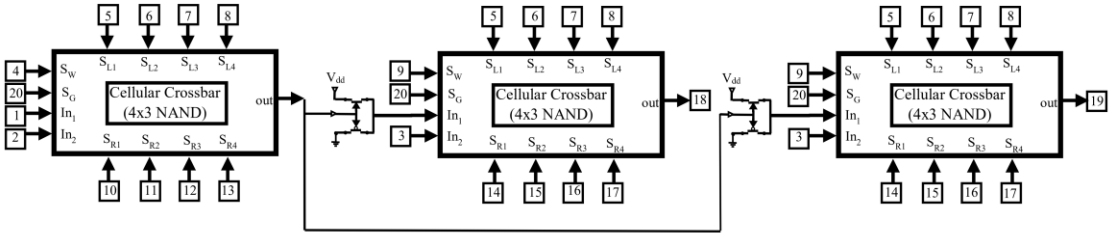


Fig. 4-5: Digital circuit design of Eq. (4-1) using three NCCA.

To start, the first unit of Fig. 4-4, performed the NAND process for 4 different states of two inputs ("00", "01", "10", "11") and store outputs within output Memristors as a Memristances. Then first NCCA starts the reading phase and the other two NCCAs enter the writing phase. The output of the first NCCA should be connected to one of the inputs of the second unit and to one of the inputs of the third unit. Here the output of the first unit is connected to the input of the second unit and the input of the third unit. It should be noted that the input of the second and the third NCCAs have a loading effect potential issue

caused by a fully Memristive crossbars, such as voltage drop. To compensate for this effect, a switch has been used for each input. In this case, the output of the first unit is connected to the control input of the switches. When output is '0', a voltage of zero volts is applied to the input of the other two units, and if output is '1', voltage +V are applied to the inputs. The advantage of these switches is the simultaneous use of one output at the input of several NCCAs, which is very valuable in parallel processing [19]. Note that all 8 output modes of the function “z” are stored in the second and third output unit of the 8 Memristors. Truth Table of proposed circuits is shown in Table 4-1. In this table, the top four states and the bottom four states of digital function “z” are stored in the second and third NCCAs respectively.

To get all of the states, firstly we set the second input of the second unit to logical '0' and the second input of the third unit to logical '1'. By using the logical '1' for the pin number 4 in the first NCCA, the writing phase starts. Then pins numbered with 5, 6, 7, and 8 perform all conditions of the NAND gate into the circuit, and store the results obtained in the output Memristors by the NAND process. Then, using pins numbered 10, 11, 12 and 13, these outputs are applied to the second and the third NCCAs by converting to voltage. Note that at this stage, pin 4 should be changed to logical value '0' and pin 9 to logical value '1'. The control of the first unit reading switches must also be compatible with the control of the digital process switches of the second and third NCCAs gates, so that pin numbered 10 with 5, pin numbered 11 with 6, pin 12 with 7 and pin numbered 13 with 8, are changed to '0' and '1’. By changing pin 9 to logical value '0', pin numbered 14, 15, 16 and 17 are used to read the second and third NCCAs outputs. As shown in Table 4-1.

Pins numbered 18 and 19 show the top four and bottom four states, consecutively. In addition, pin numbered 20 is for setting the output Memristor to Ron. If these pins are used, more NCCAs will need to be used again, and since in this case there is no need to reuse the units, this pin is '0' from the beginning to the end of the process.

Table 4-1: Truth table of digital function “z”

Inputs				Output (z)	
c	b	a	Logic	M_{out}	Logic
0	0	0	“000”	R_{on}	‘1’
0	0	+V	“001”	R_{on}	‘1’
0	+V	0	“010”	R_{on}	‘1’
0	+V	+V	“011”	R_{on}	‘1’
+V	0	0	“100”	R_{off}	‘0’
+V	0	+V	“101”	R_{off}	‘0’
+V	+V	0	“110”	R_{off}	‘0’
+V	+V	+V	“111”	R_{on}	‘1’

4-3 Simulation Results

This section presents simulation results for the proposed circuits using a nonlinear dopant drift Memristor model [20] and 180 nm complementary metal oxide semiconductor (CMOS) technology.

4-3-1 Proposed cellular configuration simulation results.

The simulation results of cellular configuration are shown in Fig. 4-6, V_{in1}, V_{in2}, V_{in3}, V_{in4}, V_{in5} and V_{in6} are inputs voltages and gates are selected via S_{L1}, S_{L2}, S_{L3} and S_{L4} control switches. All inputs are applied to the cellular configuration simultaneously and NAND gates writing process are performed in 4 cycles at time range 0 to 200 (ns). First during the time range 0 to 50 (ns), the gates highlighted by yellow in Fig. 4-3, are written. The gates marked in red, blue, and purple are written at 50 (ns) intervals, respectively. At time range 200 to 400 (ns), the read operation of the Memristance of output Memristors are performed. Also, V_{out1}, V_{out2}, V_{out3} and V_{out4} are output voltages.

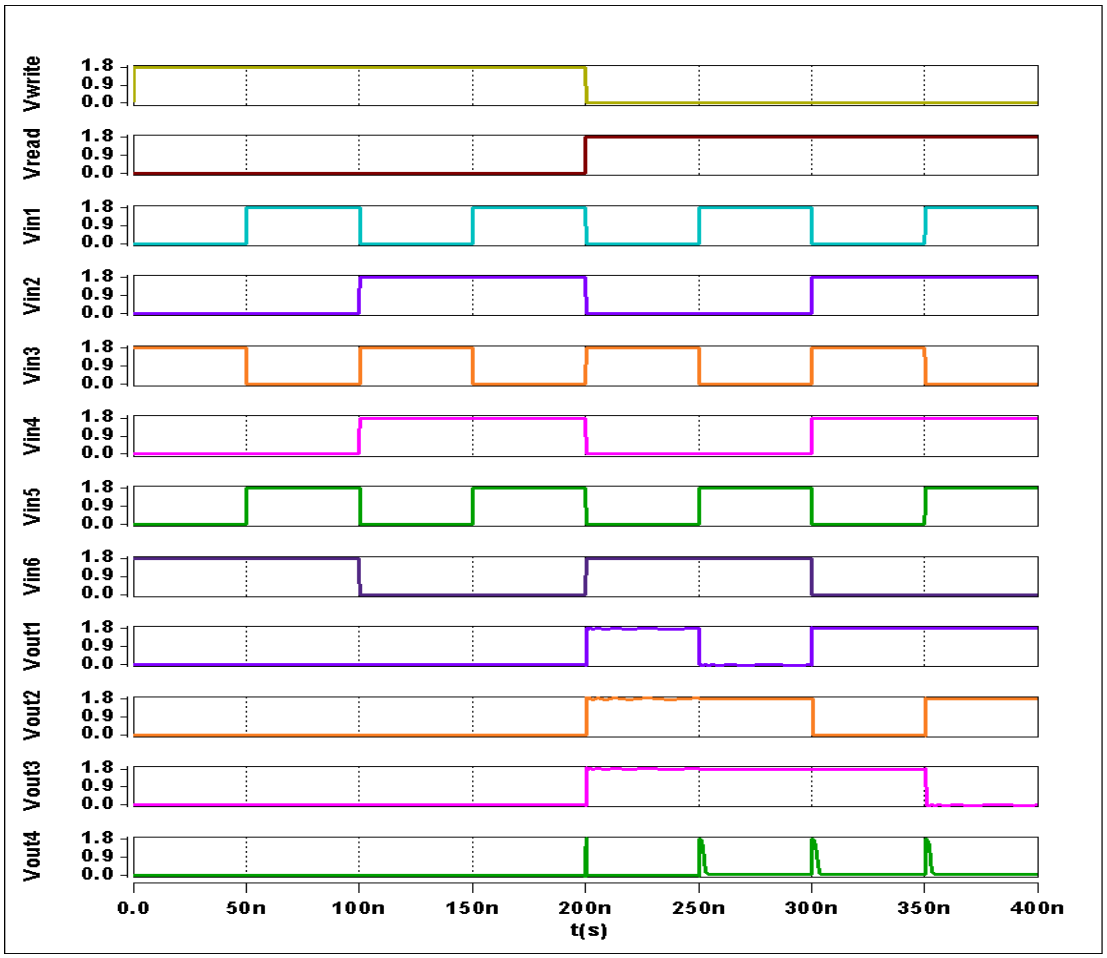


Fig. 4-6: Diagonal In-Memory Computation (IMC) simulations.

By considering the number of rows one unit greater than the number of columns, during the reading process, one of the outputs in each time cycle gives a don't-care state (X). Table 4-2 shows the operations sequences in the diagonal computing approach.

Table 4-2: Read simulations for diagonal computing.

	Column1	Column2	Column3	Column4
V_{Out1}	1 (200-250 ns)	1 (350-400 ns)	1 (300-350 ns)	X (250-300 ns)
V_{Out2}	1 (250-300 ns)	1 (200-250 ns)	1 (350-400 ns)	X (300-350 ns)
V_{Out3}	1 (300-350 ns)	1 (250-300 ns)	1 (200-250 ns)	X (350-400 ns)
V_{Out4}	0 (350-400 ns)	0 (300-350 ns)	0 (250-300 ns)	X (200-250 ns)

4-3-2 Simulation results of digital function “z” circuits.

The simulation results of Eq. (4-1) are given in this section. As shown in Fig. 4-7, at time range zero to 200 (ns), the NAND process is performed for four states of first input (a) and second input (b). The result obtained for the Vmid can be seen at 200 to 400 (ns). At time range 200 to 400 (ns), the third input (c) is NAND as a result of the previous process. Finally, 8 modes observable at 400 to 600 (ns) for V_{out1} (pin numbered 18), V_{out2} (pin numbered 19) by applying inputs V_{in3a} and V_{in3b}, respectively.

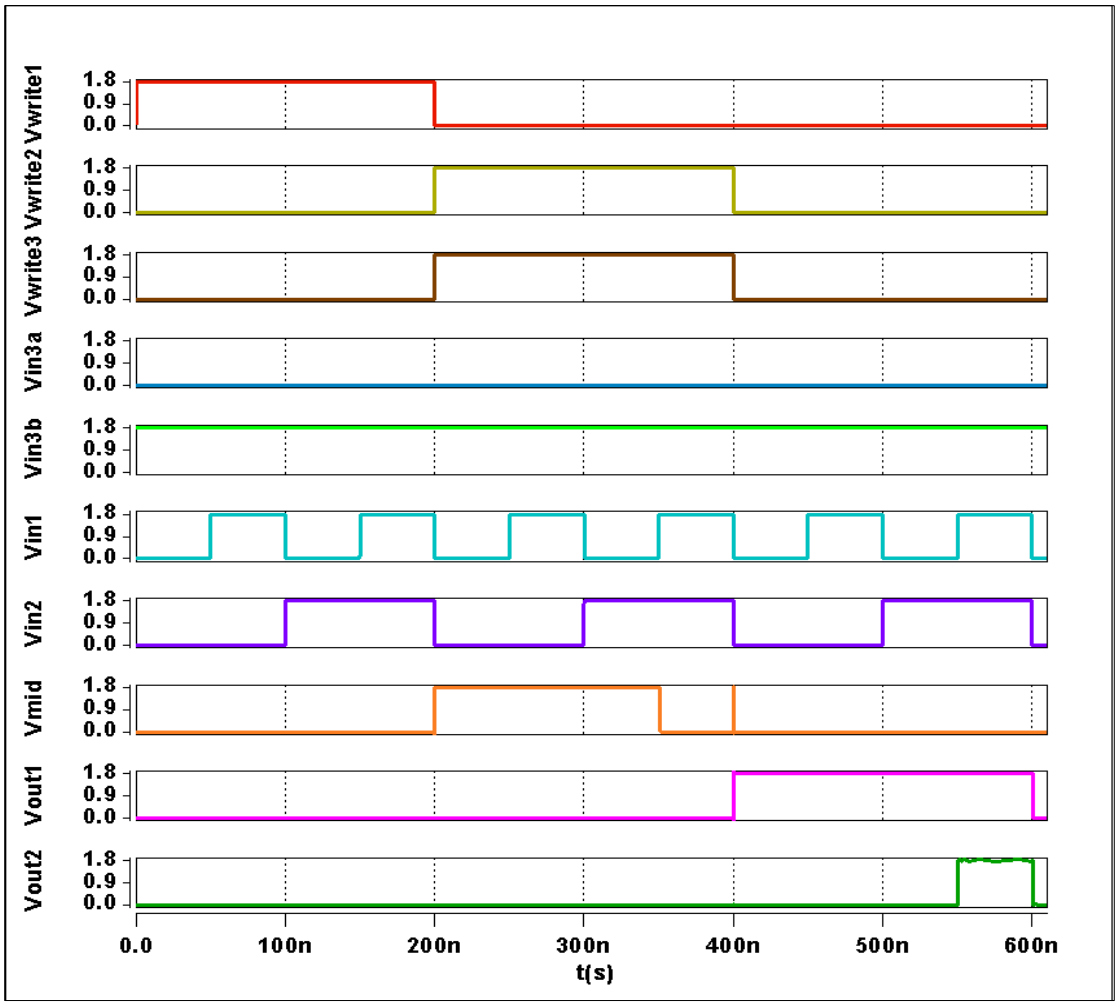


Fig. 4-7: Simulation results of the digital function “z”.

4-4 Conclusion

In this chapter, by using VTM-NAND gate, NAND Cellular Crossbar Array (NCCA) was proposed, which is a small unit for logic in-memory processing that consists of four NAND gates and peripheral circuits. By combining the NCCA, a novel cellular architecture of Memristors to perform larger digital computations was proposed. In the proposed cellular crossbar structure, to perform computations of larger digital functions, we had limited number of gates in each row. To achieve the desired result, a cellular configuration and diagonal processing method was introduced. In-memory diagonal computations are an efficient parallel and high-speed calculations method.

As a case study, a digital function by using three NCCAs was designed and simulated. Simulation results clearly show the changes in the output Memristor to the desired logic. Cellular configuration can both operate independently of each other, and communication lines can be connected to each other as needed to perform more complex digital computations simultaneously.

4-5 References

[1] H. Li, B. Gao, Z. Chen, Y. Zhao, P. Huang, H. Ye, et al., "A learnable parallel processing architecture towards unity of memory and computing," Scientific reports, vol. 5, p. 13330, 2015.

[2] M. Cassinerio, N. Ciocchini, and D. Ielmini, "Logic computation in phase change materials by threshold and memory switching," Advanced Materials, vol. 25, pp. 5975-5980, 2013.

[3] Y. Li, Y. Zhong, Y. Deng, Y. Zhou, L. Xu, and X. Miao" ,Nonvolatile “AND,”“OR,” and “NOT” Boolean logic gates based on phase-change memory," Journal of Applied Physics, vol. 114, p. 234503, 2013.

[4] J. Lee, D. I. Suh, and W. Park, "The universal magnetic tunnel junction logic gates representing 16 binary Boolean logic operations," Journal of Applied Physics, vol. 117, p. 17D717, 2015.

[5] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," Nature nanotechnology, vol. 8, p. 13, 2013.

[6] J. J. Yang and R. S. Williams, "Memristive devices in computing system: Promises and challenges," ACM Journal on Emerging Technologies in Computing Systems (JETC), vol. 9, p. 11, 2013.

[7] D. Ielmini and R. Waser, Resistive switching: from fundamentals of nanoionic redox processes to memristive device applications: John Wiley & Sons, 2015.

[8] L. Cheng, M.-Y. Zhang, Y. Li, Y.-X. Zhou, Z.-R. Wang, S.-Y. Hu, et al., "Reprogrammable logic in memristive crossbar for in-memory computing," Journal of Physics D: Applied Physics, vol. 50, 2017.

[9] M .Butts. "Future directions of dynamically reprogrammable systems," in Custom Integrated Circuits Conference, 1995., Proceedings of the IEEE 1995, pp. 487-494.

[10] Z.R. Wang, Y.-T. Su, Y. Li, Y.-X. Zhou, T.-J. Chu, K.-C. Chang, et al., "Functionally complete Boolean logic in 1T1R resistive random access memory," IEEE Electron Device Letters, vol. 38, pp. 179-182, 2017.

[11] Y. Li, Y.-X. Zhou, L. Xu, K. Lu, Z.-R. Wang, N. Duan, et al., "Realization of functional complete stateful Boolean logic in memristive crossbar," ACS applied materials & interfaces, vol. 8, pp. 34559-34567, 2016.

[12] L. Cheng, M.-Y. Zhang, Y. Li, Y.-X. Zhou, Z.-R. Wang, S.-Y. Hu, et al., "Reprogrammable logic in memristive crossbar for in-memory computing," Journal of Physics D: Applied Physics, vol. 50, p.505102, 2017.

[13] H. A. Du Nguyen, L. Xie, M. Taouil, R. Nane, S. Hamdioui, and K. Bertels, "On the implementation of computation-in-memory parallel adder," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 25 ,pp. 2206-2219, 2017.

[14] A. Madhavan, T. Sherwood, and D. B. Strukov, "High-Throughput Pattern Matching with CMOL FPGA Circuits: Case for Logic-in-Memory Computing," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, pp. 1-14, 2018.

[15] H. E. Yantır, A. M. Eltawil, and F. J. Kurdahi, "A Hybrid Approximate Computing Approach for Associative In-memory Processors," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 8(4), pp.758-769, 2018.

[16] H. E. Yantir, A. M. Eltawil, and F. J. Kurdahi, "A Two-Dimensional Associative Processor," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 26(9), pp.1659-1670, 2018.

[17] Y. X. Zhou, Y. Li, N. Duan, Z. R. Wang, K. Lu, M. M. Jin, et al., "Boolean and Sequential Logic in a One-Memristor-One-Resistor (1M1R) Structure for In-Memory Computing," Advanced Electronic Materials, p. 1800229, 2018.

[18] Farzad Mozafari, Majid Ahmadi, Arash Ahmadi, M_J Sharifi, “A Novel Architecture for Memristor-Based Logic “, The IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021.

[19] I. Vourkas and G. C. Sirakoulis, "On the generalization of composite memristive network structures for computational analog/digital circuits and systems," Microelectronics Journal, vol. 45, pp. 1380-1391, 2014.

[20] Z. Biolek ,D. Biolek, and V. Biolkova, "SPICE Model of Memristor with Nonlinear Dopant Drift," Radioengineering, vol. 18, no. 2, 2009.

Chapter 5:

A Programmable Circuits Based on the Combination of VTM Cellular Crossbars

5-1 Introduction

Implementation of digital processes with Memristors has provided new opportunities for researchers so that several methods of designing digital gates with Memristors have been proposed for use in crossbar array structures [1, 2]. The types of circuits designed in Memristor-based arrays can be divided into several categories. The first approach is to use a combination of CMOS and Memristor components in Boolean logic [3] and threshold logic calculations [4, 5]. For example, Sheldon Akers proposed the rectangular logic array [6]. In Akers logical array [6], the execution of each Boolean function is performed by passing data from basic logical cells, data is moved from each basic cell to the adjacent cells. This paper proposes a Memristive-architecture by using the combination of Voltage to Memristance (VTM) cellular crossbars. Proposed circuits have the pipeline process advantage that it can be used in the implementation of high complexity digital functions while the crossbar cells are never idle. The most important feature of this method is increasing the speed of calculations and saving data.

The rest of the chapter is organized as follows: Section 5-2 presents the proposed AND/OR logic gates using VTM method [8] and design of the architecture are based on the combination of NAND/AND cellular crossbars. Simulation and results are presented in Section 5-3. Finally, Section 5-4 concludes the paper.

5-2 Proposed AND/OR Memristor-Based Logic Gates & AND Cellular Crossbar Array (ACCA)

5-2-1 Proposed AND/OR Logic Gates

The proposed AND/OR stateful logic gates, is shown in Fig. 5-1. The structure of the proposed gate is similar to NAND/NOR [7], with the difference that the Memristors are connected by two positive polarities. This structure consists of Memristors with two inputs M_{in1} and M_{in2}, and an output Memristor M_{out}. V_{in1} and V_{in2} are gate input voltages and the Memristance of M_{out} stores the output.

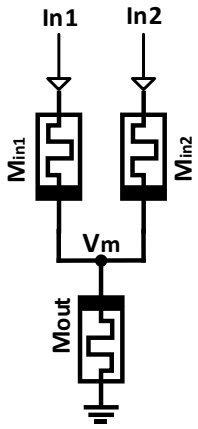


Fig. 5-1: Proposed AND/OR gates using convert VTM method.

Truth table of proposed AND/OR gates, is shown in Table 5-1.

TABLE 5-1: Truth table of proposed AND/OR gates.

Inputs			Output (AND)		Output (OR)	
I_{in1}	I_{in2}	Logic	M_{out}	Logic	M_{out}	Logic
0	0	"00"	R_{off}	'0'	R_{off}	'0'
0	+V	"01"	R_{off}	'0'	R_{on}	'1'
+V	0	"10"	R_{off}	'0'	R_{on}	'1'
+V	+V	"11"	R_{on}	'1'	R_{on}	'1'

This chapter proposes AND & OR gates using $V_{in\ (AND)}$ =1.8 (v) and $V_{in\ (OR)}$ = 3.3 (v). It is possible to program universal logic gates with similar structures to function as AND or OR gates by using two different input voltages. Yet, with the proposed gates, an AND gate can be used with a smaller voltage while an OR gate can be used with a higher voltage. As with NAND/NOR cellular crossbars [7], an output Memristor reading circuit is required for the AND/OR cellular crossbar, so that the resistance value can be converted into voltage. Fig. 5-2 shows that, like the NAND Cellular Crossbar Array (NCCA) [8], the proposed AND Cellular Crossbar Array (ACCA) consists of four AND gate cells across

each column. A memory resistance is stored in the last column (output Memristor) as the result of the logical operation performed on In_1 and In_2. Furthermore, the input Memristors' initial values are not changed after various digital processes are performed, as all of them are initialized once to R_{on}. The main aim of proposed cellular crossbars NAND/NOR and AND/OR in this chapter is to combine them to implement a wider cellular crossbar to perform larger digital functions. In this chapter, the name Memristive-architecture is chosen for this combination of NAND/AND cellular crossbars. Due to the low input voltage requirement (V_{in}=+V=1.8 volts), the NAND/AND gates are chosen for this purpose.

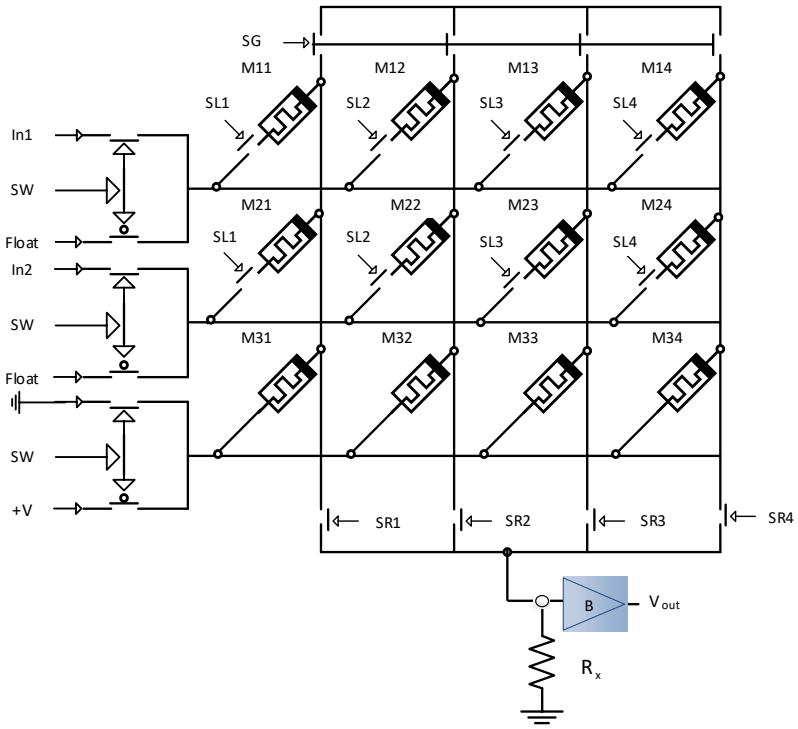
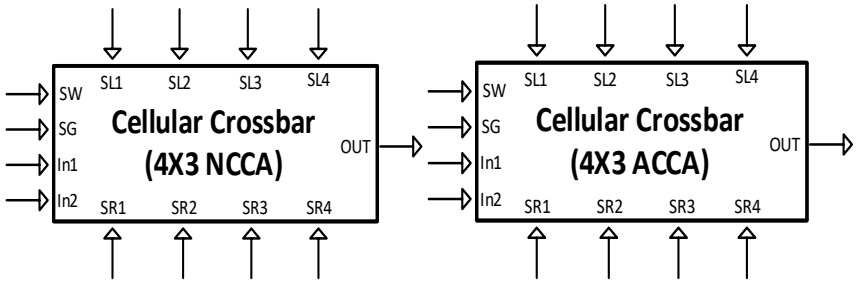


Fig. 5-2: Proposed Cellular Crossbar Array (ACCA).

5-2-2 Design and Implementation programmable circuit based on combination of NAND/AND Cellular Crossbar Arrays.

By considering the inputs and output of Fig. 5-2, the schematic of the NCCA and ACCA cells can be shown in Fig. 5-3.



a) NCCA b) ACCA

Fig. 5-3: Schematic of proposed NCCA and ACCA

By applying input voltages, Sw allows writing to take place. It is determined which gates will perform the NAND/AND operation in each cycle by the switches S_{L1}, S_{L2}, S_{L3}, and S_{L4}. These control switches also enable logical state data to be read from output Memristors by reading S_{R1}, S_{R2}, S_{R3}, and S_{R4}. Lastly, Memristors are configured to initiate their initial state to R_{on} by means of output configuration switches (S_G). In the same way, the schematic of a Memristive architecture is shown in Fig. 5-4.

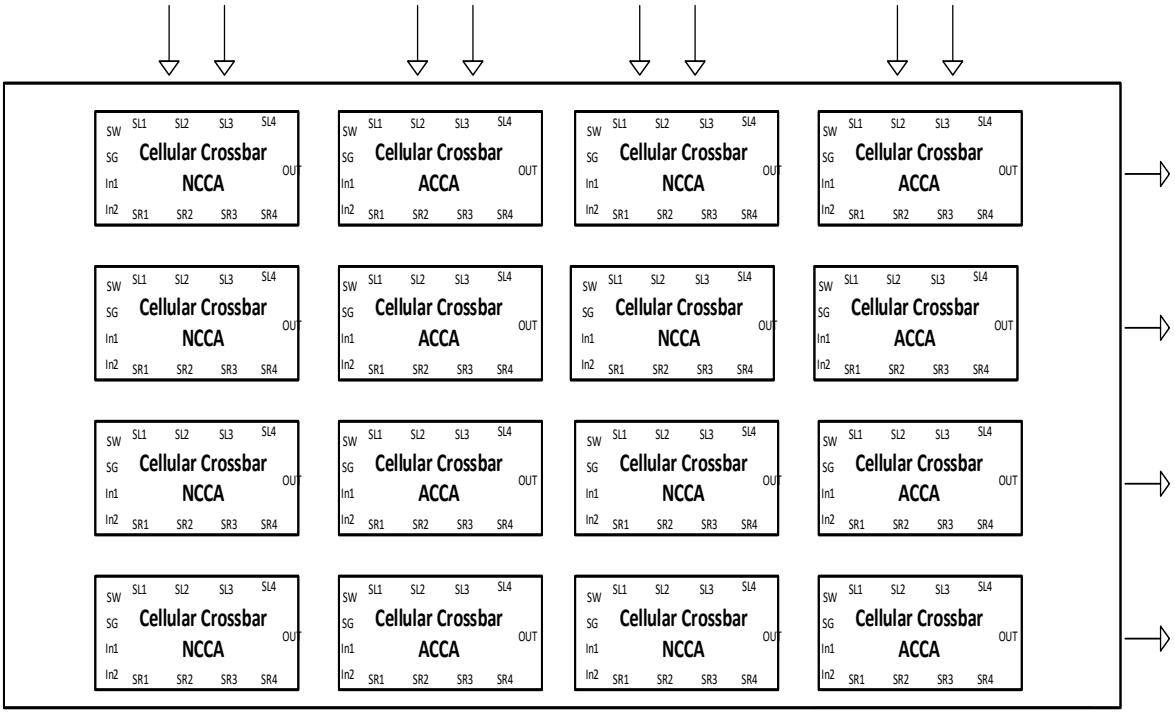


Fig. 5-4: Memristive architecture based on NAND/AND cellular crossbars.

In the proposed circuit, inputs of the cells can be considered from outside the architecture or from the output of the other cells inside the circuit. Also, the outputs of the cells can be applied to the inputs of other cells in the circuit or transferred to other elements outside of them. In the proposed architecture, several large-scale logical computations are performed in a homogeneous Memristor crossbars structure and individual logic operations can be programmed in a structure only by applying required voltages to different Memristors. The novelty of the proposed circuit is the use of the pipeline method, as depicted in Fig. 5-5, in which multiple instructions are overlapped during execution. Therefore, this causes none of the cells embedded in the circuit to be idle at any moment and the cells are always either in the writing step (odd) or in the reading step (even).

As a result, the number of commands performed per unit of time (throughput) will be increased by this proposed arrangement.

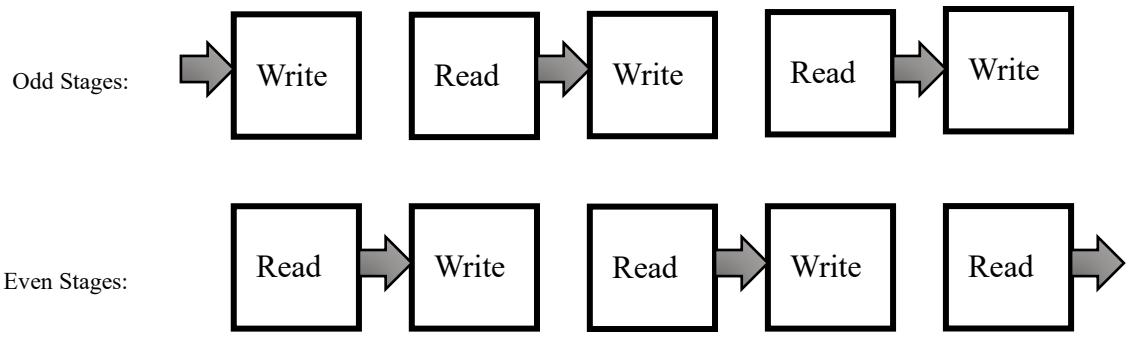


Fig 5-5: Pipeline method in the architecture based on cellular crossbars.

Complete details of the proposed circuit will be explained as follows. As shown in Fig. 5-6, several interconnections are used in the proposed architecture. Each of the vertical and horizontal lines has a special use. In this circuit, the cells of the architecture are arranged in such a way that they can receive input data from outside of the circuit, as well as from the output of their internal cells. First, the input and output lines are discussed.

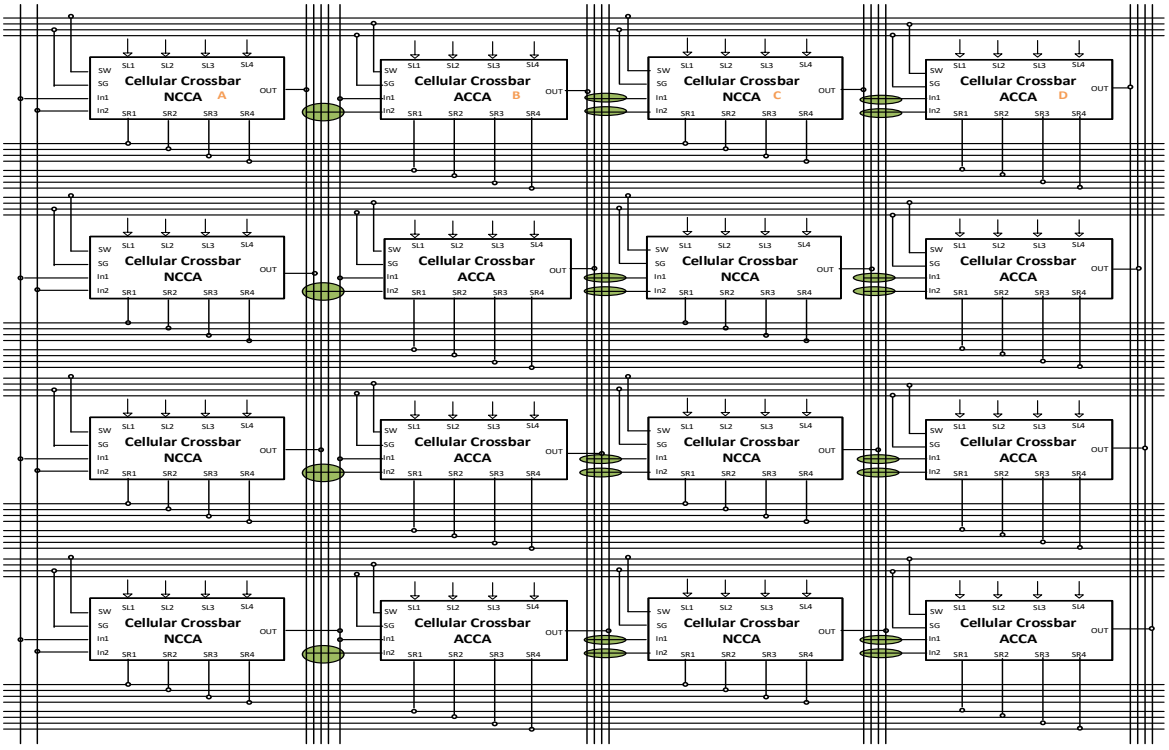


Fig. 6-6: Proposed architecture circuit based on NAND/AND cellular crossbars.

The first input of all 4 cells of the first column is connected to each other. Also, the second inputs are connected to each other in the same way. Each column contains 4 cells and a

total of 4 outputs that are connected to the next 4 vertical lines. For example, 4 vertical lines are considered for the outputs of the cells of the first column. The cell output of the first, second, third, and fourth rows are connected to the first, second, third, and fourth vertical lines, respectively. Meanwhile, these outputs can be used both in the circuit itself as input to the cells located in the second column and can be used outside of it. In the second column of this circuit, it can be seen that the first input of all 4 cells placed in this column are connected to each other and its digital value is provided from outside the architecture, the second input is also connected to one of the outputs of the first column. It should be noted that the second input in second column are shown with a green oval. This ellipse means that the specified input is connected to only one of the outputs of the first column in each time cycle. In the same way, third and fourth columns, both cell inputs receive data from the outputs of their previous columns (The inputs of the third column are supplied from the outputs of the second column and the inputs of the fourth column are supplied from the outputs of the third column). In Fig. 6-6, write operation control switches (S_{L1}, S_{L2}, S_{L3} and S_{L4}), are write operation control switches, that determine which gates to perform the NAND/AND operation. Also, all the cells are connected to each other and controlled by 4 lines from outside of the crossbar. (For the simplicity of the figure, lines are left as a free connection). S_W is a writing control switch, which is selected to allow writing operation by applying input voltages. In the proposed architecture S_W in each row is different from the other rows. A group of 4 horizontal lines is considered above each row of cells. The two upper lines are used for the S_W base. Of these two lines, the upper one is used for odd columns and the lower one is used for even columns. This means that the first and third columns enter the writing (or reading) process at the same time and, the second and fourth columns read (or write) simultaneously.

Finally, two lower lines of the 4 lines placed above the cells are assigned to S_G, which has the same function as S_W, with the difference that the initialization of the output Memristors to R_{on} in the odd and even columns is performed simultaneously. It should be noted that under each row of cells there are two groups of 4 horizontal lines. These lines are used to control the reading switches. 4 upper lines are used for cells located in odd columns and 4 lower lines are used for cells placed in even columns.

To clarify the explanation above, consider the first row of Fig. 6-6. First, the cells are named A, B, C and D from left to right. In the first step, cell A enters the writing phase, and after cycles, the output of NAND gates is kept in the output Memristors of this cell. In the second step, cell B (at the same time as cell A enters the reading phase) starts the writing phase. Again, in four cycles, the digital AND process is executed on the data entered from outside and inside of the circuit, and the outputs are kept in cell B. In the third step, cell A enters the output Memristor configuration phase, cell B enters the reading phase, and cell C starts the writing phase. Next, in the fourth step, cell B enters the output Memristor configuration phase, cell C enters the reading phase, and cell D starts the writing phase. Finally, cell C enters the output Memristor configuration phase, cell D enters the reading phase, and cell A enters the writing phase again. In the same way, the next steps are also implemented, and these processes are applicable for other columns and rows as well. It should be noted that the input of the B, C, and D cells have a loading effect potential issue caused by a fully Memristive crossbars, such as voltage drop. To compensate for this effect, a switch has been used for each line. In this case, the output of A cell is connected to the control input of the switches. When output is '0', a voltage of zero volts is applied to the input of the other cells, and if output is '1', voltage +V are applied to the inputs. The advantage of these switches is the simultaneous use of one output at the input of several cells, which is especially important in the proposed circuit [9].

As a case study, the following digital function is implemented using proposed programmable circuit.

$$F = [(a' + b')\ .c] \qquad (5\text{-}1)$$

Eq. (1) can be written in a form below:

$$F = [((a\ NAND\ b)\ AND\ c\ )] \qquad (5\text{-}2)$$

There are three inputs to the proposed function. The output will have 2^3 different modes for every 3 different inputs. Table 5-2 shows the truth table.

Table 5-2: Truth table of digital function “F”.

Inputs				Output (z)	
c	b	a	Logic	M_{out}	Logic
0	0	0	“000”	R_{off}	‘0’
0	0	+V	“001”	R_{off}	‘0’
0	+V	0	“010”	R_{off}	‘0’
0	+V	+V	“011”	R_{off}	‘0’
+V	0	0	“100”	R_{on}	‘1’
+V	0	+V	“101”	R_{on}	‘1’
+V	+V	0	“110”	R_{on}	‘1’
+V	+V	+V	“111”	R_{off}	‘0’

5-3 Simulation Results

In this section, SPICE has been used to simulate the proposed programmable circuit. Advanced Memristor model with a modified Biolek window and a voltage-dependent variable exponent [10] is used in the simulations.

5-3-1 Simulation results AND writes logic operations and digital function “F” circuits.

The simulation results are given in this section. The steps of the digital “F” function process were explained in the previous section. Considering that the F digital function with three inputs has 8 output states and because each cell is able to store 4 states in the output Memristors, it is required that all the states in the middle and right cells are kept in the first row of in Memristive architecture. First, the first-row cell executes the NAND process for 4 different states from the first and second inputs and stored in the 4 output Memristors as Memristance. Then this cell enters the reading phase (even) and the other cell enters the writing phase (odd). As shown in Fig. 5-7, the NAND process is performed for four states of V_{in1} (a) and V_{in2} (b). The result obtained can be seen at the time range zero to 200 (ns).

At time range 200 to 400 (ns), The third input (V_{in3}) with the results of previous NAND process becomes AND. As a result, the eight final states can be observed in V_{out1} and V_{out2}.

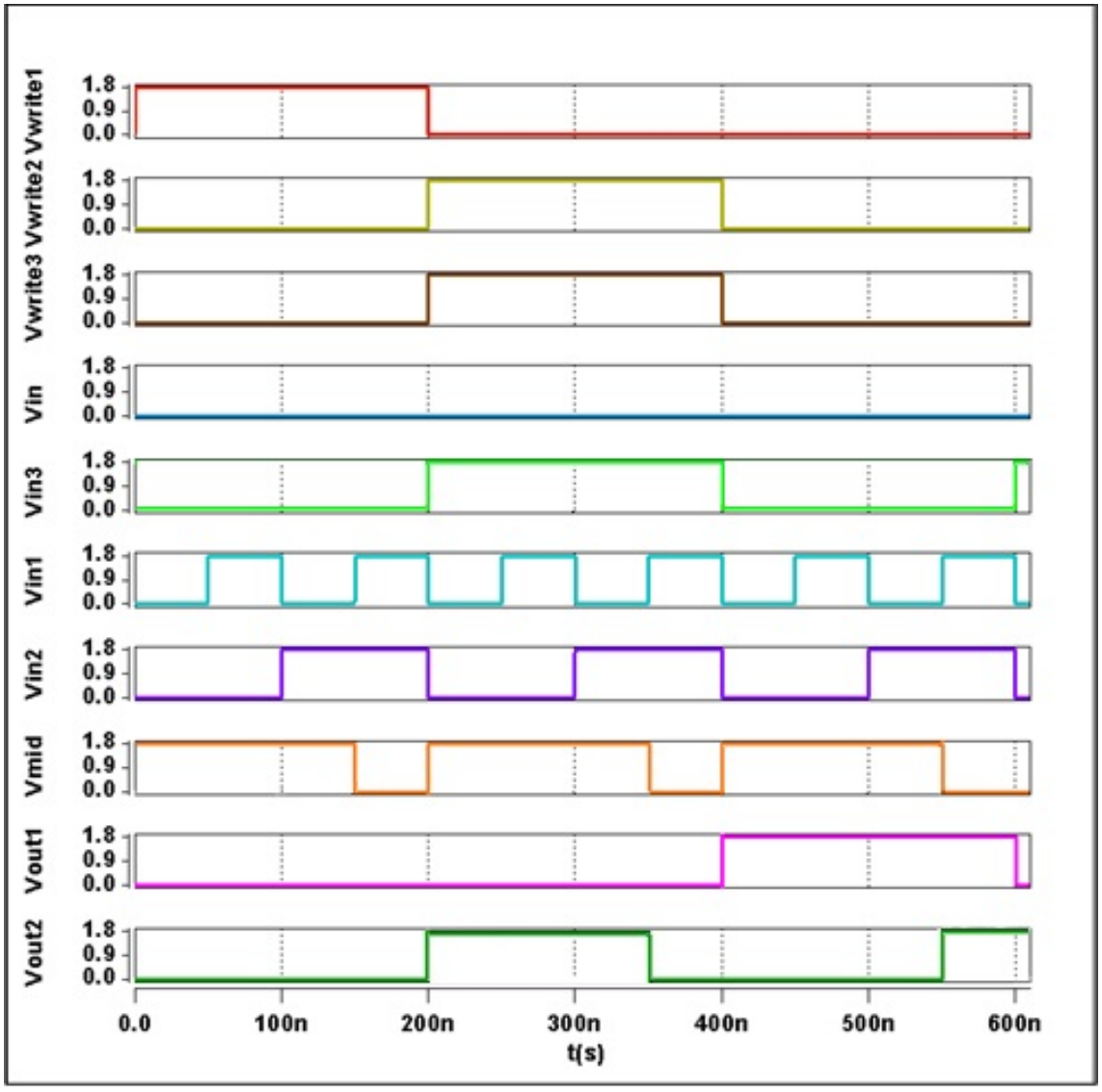


Fig. 5-7: Simulation results of the digital function “F” using proposed memristive- architecture.

5-4 Conclusion

In this paper, by using VTM-NAND/AND gates AND Cellular Crossbar Array (ACCA) were proposed. By combining the NCCAs and ACCAs, novel programmable circuit architecture to perform larger digital computations was proposed. The novelty of the proposed circuit is the use of the pipeline technique, which is divided into stages odd and even, and are connected with one another to form a pipe like structure. Pipelining is a technique for breaking down a sequential process into various sub-operations and executing each sub-operation. As a case study, a digital function by using three NCCAs and ACCAs was designed and simulated in the circuit. Simulation results clearly show the changes in the output Memristor to the desired logic. circuit configuration can both operate independently of each other and can be connected to each other as needed to perform more complex digital computations simultaneously as well. It should be noted that larger ACCA and NCCA can be constructed by the proposed 4x4 circuit.

5-5 References

[1] 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), 2013, 12th International Workshop on, 2012, pp. 1-6.

[2] N. Hashem and S. Das, "Switching-time analysis of binary-oxide memristors via a nonlinear model," Applied Physics Letters, vol. 100, p. 262106, 2012.

[3] J. Borghetti, Z. Li, J. Straznicky, X. Li, D. A. Ohlberg, W. Wu, et al., "A hybrid nanomemristor/transistor logic circuit capable of self-programming," Proceedings of the National Academy of Sciences, vol. 106, pp. 1699-1703, 2009.

[4] L. Gao, F. Alibart, and D. B. Strukov, "Programmable CMOS/memristor threshold logic," IEEE Transactions on Nanotechnology, vol. 12, pp. 115-119, 2013.

[5] J. Rajendran, H. Manem, R. Karri, and G. S. Rose, "Memristor based programmable threshold logic array," in Proceedings of the 2010 IEEE/ACM International Symposium on Nanoscale Architectures, 2010, pp. 5.

[6] S. B. Akers, "A rectangular logic array," in Switching and Automata Theory, 1971., 12th Annual Symposium on, 1971, pp. 79-90.

[7] Farzad Mozafari, Majid Ahmadi, Arash Ahmadi, M_J Sharifi, “A Novel Architecture for Memristor-Based Logic “, The IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021.

[8] Farzad Mozafari, Majid Ahmadi, Arash Ahmadi, “Design of A New Memristive-Based Architecture Using VTM Method“, The IEEE International Symposium on Circuits and Systems (ISCAS), 2022.

[9] I. Vourkas and G. C. Sirakoulis, "On the generalization of composite memristive network structures for computational analog/digital circuits and systems," Microelectronics Journal, vol. 45, pp. 1380-1391, 2014

[10] Y. Mladenov, V.; Kirilov, S. Advanced Memristor Model with a Modified Biolek Window and a Voltage-Dependent Variable Exponent. Inform. Autom. Pomiary Gospod. Ochr. Sr. (IAPGO ´ S) 2018, 8, 15–20.

Chapter 6:

Conclusions
And
Further Work

6-1 Summery

In chapter 2, we presented a novel design for Memristor-based logic gates. Universal logic gates NAND/NOR, with similar topologies were proposed. The proposed gates have advantages such as higher speed, fewer operation steps, and propagation delay less than their counterparts. Seven basic logic functions, NOT, AND/OR, NAND/NOR, XOR and XNOR use simple connections among Memristors. As a case study, a single bit full adder is designed on the proposed cellular structure. In comparison with the 1-bit CMOS-based full adder, the proposed circuit is capable of having approximately, 20% higher switching speed, 44% lower connections, and 28% lower power consumption.

In chapter 3, by using VTM based technique, a Memristor-based XOR/XNOR logic gates with 5 Memristors, capable of executing operation in a single step, is presented. As a case study, a single bit full adder circuit by using proposed VTM logic gates is designed. Moreover, the functions of proposed logic gates and full adder are verified by PSPICE, respectively. The performance parameters of the proposed adder circuit such as power (dynamic and static), propagation delay was compared with conventional adders (CMOS, CPL) and combined adders (Hybrid). The proposed full adder architecture based on XOR gates compared to the other full adders, has yielded simpler architecture, higher speed and lower power consumption.

In chapter 4, by using VTM-NAND gate, NAND Cellular Crossbar Array (NCCA) were proposed, which is a small unit for logic in-memory processing that consists of four NAND gates and peripheral circuits. By combining the NCCA, a novel cellular architecture of Memristors to perform larger digital computations was proposed. In the proposed cellular crossbar structure, to perform computations of larger digital functions, we had a limit on adding gates for each row. To achieve the desired result, a cellular configuration and diagonal processing method was introduced. In-memory diagonal computations are an efficient parallel and high-speed calculations method. As a case study, a digital function by using three NCCAs was designed and simulated. Simulation results clearly show the changes in the output Memristor to the desired logic. Cellular configuration can both operate independently of each other, and communication lines can be connected to each other as needed to perform more complex digital computations simultaneously.

In chapter 5, by using VTM-NAND/AND gates AND Cellular Crossbar Array (ACCA) were proposed. By combining the NCCAs and ACCAs, novel programmable circuit architecture to perform larger digital computations was proposed. The novelty of the proposed circuit is use of the pipeline technique, which is divided into stages odd and even, and are connected with one another to form a pipe like structure. Pipelining is a technique for breaking down a sequential process into various sub-operations and executing each sub-operation. As a case study, a digital function by using three NCCAs and ACCAs was designed and simulated in the circuit. Simulation results clearly show the changes in the output Memristor to the desired logic. Circuit configuration can both operate independently of each other and can be connected to each other as needed to perform more complex digital computations simultaneously as well.

6-2 Conclusion

In this thesis, a novel method for utilizing Memristors in digital circuits, named VTM, is proposed. The VTM method can be employed as a foundation for designing various types of digital circuits. In this thesis, gates including NOT, NAND, NOR, AND, OR, XOR, and XNOR were designed based on this method, and their correctness was verified through simulation using the HSPICE software. These gates offer advantages such as a reduced number of Memristors and the ability to perform digital processes in a single step. These gates achieve the desired digital operations in the shortest time, specifically 12 nanoseconds.

In addition, NAND and AND gates were utilized for crossbar cell design due to their lower operating voltage compared to other gates. The crossbar cell is a small unit for in-memory processing. It consists of four NAND gates and peripheral circuits for reading the Memristor outputs and configuring the Memristor outputs. Multiple crossbar cells are placed alongside each other, and with the use of control signals, various types of digital processing can be designed.

To address the limitation of the number of gates in each row of the proposed crossbar cell, a method called diagonal processing is introduced in this thesis. Diagonal processing is highly practical for parallel and high-speed computations. By arranging crossbar cells

together, an architecture known as a package is presented, which is employed for more complex computations. Finally, for more extensive processing tasks, the utilization of multiple packages is suggested.

6-3 Further Works

- Designing more complex circuits using the VTM method.
- Designing crossbar cells based on other gate types.
- Designing more complex circuits based on NAND crossbar cells.
- Practical application of diagonal processing.

6-4 References

[1] C. E. Shannon, "A Symbolic Analysis of Relay and Switching Circuits," Master Science, MIT, 1940.

[2] L. Chua, "Memristor-the missing circuit element," IEEE Transactions on circuit theory, vol. 18, pp. 507-519, 1971.

[3] I. Vourkas and G. C. Sirakoulis" ,Emerging memristor-based logic circuit design approaches: A review," IEEE Circuits and Systems Magazine, vol. 16, pp. 15-30, 2016.

[4] L. O. Chua and S. M. Kang, "Memristive devices and systems," Proceedings of the IEEE, vol. 64, pp. 209-223, 1976.

[5] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," nature, vol. 453, p. 80, 2008.

[6] S. D. Ha and S. Ramanathan, "Adaptive oxide electronics: A review," Journal of applied physics, vol. 110, p. 14, 2011.

[7] S .Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "The desired memristor for circuit designers," IEEE Circuits and Systems Magazine, vol. 13, pp. 17-22, 2013.

[8] Y. V. Pershin and M. Di Ventra, "Memory effects in complex materials and nanoscale systems," Advances in Physics, vol. 60, pp. 145-227, 2011.

[9] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'Memristive'switches enable 'stateful'logic operations via material implication," Nature, vol. 464, p .2010, 873.

[10] K. Eshraghian, O. Kavehei, K.-R. Cho, J. M. Chappell, A. Iqbal, S. F. Al-Sarawi, et al., "Memristive device fundamentals and modeling: applications to circuits and systems simulation," Proceedings of the IEEE, vol. 100, pp. 1991-2007, 201.

[11] F. L. Traversa, C. Ramella, F. Bonani, and M. Di Ventra, "Memcomputing NP-complete problems in polynomial time using polynomial resources and collective states," Science advances, vol. 1, p. e1500031, 2015.

[12] T. Chang, Y. Yang, and W. Lu, "Building neuromorphic circuits with memristive devices," IEEE Circuits and Systems Magazine, vol. 13, pp. 56-73, 2013.

[13] D. Kuzum, S. Yu, and H. P. Wong, "Synaptic electronics: materials, devices and applications," Nanotechnology, vol. 24, p. 382001, 2013.

[14] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, and R. Waser, "Beyond von Neumann—logic operations in passive crossbar arrays alongside memory operations," Nanotechnology, vol. 23, p. 305205, 2012.

[15] G. S. Rose, J. Rajendran, H. Manem, R. Karri, and R. E. Pino, "Leveraging Memristive Systems in the Construction of Digital Logic Circuits," Proceedings of the IEEE, vol. 100, pp. 2033-2049, 2012.

[16] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: device characteristics and design implications," in Proceedings of the 2009 International Conference on Computer-Aided Design, 2009, pp. 485-490.

[17] W. Zhao, J. M. Portal, W. Kang, M. Moreau, Y. Zhang, H. Aziza, et al., "Design and analysis of crossbar architecture based on complementary resistive switching non-volatile memory cells," Journal of Parallel and Distributed Computing, vol. 74, pp. 2484-2496, 2014.

[18] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits," in

Circuit Theory and Design, 2009. ECCTD 2009. European Conference on, 2009, pp. 563-566.

[19] 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, pp. 1857-1864, 2010.

[20] 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), 2013, 12th International Workshop on, 2012, pp. 1-6.

[21] N. Hashem and S. Das, "Switching-time analysis of binary-oxide memristors via a nonlinear model," Applied Physics Letters, vol. 100, p. 262106, 2012.

[22] A. Ascoli, F. Corinto, V. Senger, and R. Tetzlaff, "Memristor model comparison," IEEE Circuits and Systems Magazine, vol. 13, pp. 89-105, 2013.

[23] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Memristor SPICE modeling," in Advances in Neuromorphic Memristor Science and Applications, ed : Springer, 2012, pp. 211-244.

[24] D. Biolek, Z. Biolek, and V. Biolkova, "SPICE modeling of memristive, memcapacitative and meminductive systems," in Circuit Theory and Design, 2009. ECCTD 2009. European Conference on, 2009, pp. 249-252.

[25] Z. Biolek ,D. Biolek, and V. Biolkova, "SPICE Model of Memristor with Nonlinear Dopant Drift," Radioengineering, vol. 18, 2009.

[26] Y. Zhang, X. Zhang, and J. Yu, "Approximated SPICE model for memristor," in Communications, Circuits and Systems, 2009. ICCCAS 2009 .International Conference on, 2009, pp. 928-931.

[27] K. Zaplatilek, "Memristor modeling in MATLAB & SIMULINK," in Proceedings of the European computing conference, 2011, pp. 62-67.

[28] V. M. Mladenov and S. M. Kirilov, "Memristor Modeling In MATLAB & PSPICE," in ECMS, 2015, pp. 432-437.

[29] A. F. Adzmi, A. Nasrudin, W. F. H. Abdullah, and S. H. Herman, "Memristor Spice model for designing analog circuit," in Research and Development (SCOReD), 2012 IEEE Student Conference on, 2012, pp. 78-83.

[30] D. Biolek, V. Biolkova, and Z. Kolka, "Spice models of memristive devices forming a model of Hodgkin-Huxley axon," in Digital Signal Processing (DSP), 2013 18th International Conference on, 2013, pp. 1-5.

[31] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable SPICE simulations of memristors, memcapacitors and meminductors," arXiv preprint arXiv:1307.2717, 2013.

[32] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: Threshold adaptive memristor model," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, pp. 211-221, 2013.

[33] D. Biolek, V. Biolkova, and Z. Kolka, "Memristor model for massively-parallel computations," in Computing, Communication and Security (ICCCS), 2015 International Conference on, 2015, pp. 1-5.

[34] D .Biolek, Z. Kolka, V. Biolkova, and Z. Biolek, "Memristor models for spice simulation of extremely large memristive networks," in Circuits and Systems (ISCAS), 2016 IEEE International Symposium on, 2016, pp. 389-392.

[35] Z. Biolek, D. Biolek, J. Vavra, V .Biolkova, and Z. Kolka, "The simplest memristor in the world," in Circuits and Systems (ISCAS), 2016 IEEE International Symposium on, 2016, pp. 1854-1857.

[36] Y. V. Pershin and M. Di Ventra, "SPICE model of memristive devices with threshold," arXiv preprint arXiv:1204.2600, 2012.

[37] M. Mahvash and A. C. Parker, "A memristor SPICE model for designing memristor circuits," in Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on, 2010, pp. 989-992.

[38] H. Abdalla and M. D. Pickett, "SPICE modeling of memristors," in Circuits and Systems (ISCAS), 2011 IEEE International Symposium on, 2011, pp. 1832-1835.

[39] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Memristor SPICE model and crossbar simulation based on devices with nanosecond switching time," in Neural Networks (IJCNN), The 2013 International Joint Conference on, 2013, pp. 1-7.

[40] J. Secco, F. Corinto, and A. Sebastian, "Flux–charge memristor model for phase change memory," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, pp. 111-114, 2018.

[41] A. K. Maan, D. A. Jayadevi, and A. P. James, "A survey of memristive threshold logic circuits," IEEE transactions on neural networks and learning systems, vol. 28, pp. 1734-1746, 2017.

[42] Y .V. Pershin and M. Di Ventra, "Neuromorphic, digital, and quantum computation with memory circuit elements," Proceedings of the IEEE, vol. 100, pp. 2071-2080, 2012.

[43] H. A. Du Nguyen, L. Xie, J. Yu, M. Taouil, and S. Hamdioui, "Interconnect networks for resistive computing architectures," in Design & Technology of Integrated Systems In Nanoscale Era (DTIS), 2017 12th International Conference on, 2017, pp. 1-6.

[44] R. Gharpinde, P. L. Thangkhiew, K. Datta, and I. Sengupta, "A Scalable In-Memory Logic Synthesis Approach Using Memristor Crossbar," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, pp. 355-366, 2018.

[45] M. Imani, S. Gupta, and T. Rosing, "Ultra-efficient processing in-memory for data intensive applications," in Proceedings of the 54th Annual Design Automation Conference 2017, 2017, p. 6.

[46] A. Haj-Ali, R. Ben-Hur, N. Wald, and S. Kvatinsky, "Efficient Algorithms for In-Memory Fixed Point Multiplication Using MAGIC," in Circuits and Systems (ISCAS), 2018 IEEE International Symposium on, 2018, pp. 1-5.

[47] H. E. Yantir, A. M. Eltawil, and F. J. Kurdahi, "A Two-Dimensional Associative Processor," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2018.

[48] Y. X. Zhou, Y. Li, N. Duan, Z. R. Wang, K. Lu, M. M. Jin, et al., "Boolean and Sequential Logic in a One-Memristor-One-Resistor (1M1R) Structure for In-Memory Computing," Advanced Electronic Materials, p. 1800229, 2018.

[49] Y.-F. Chang, F. Zhou, B. W. Fowler, Y.-C. Chen, C.-C. Hsieh, L. Guckert, et al., "Memcomputing (Memristor+ Computing) in Intrinsic SiO x-Based Resistive Switching Memory: Arithmetic Operations for Logic Applications," IEEE Transactions on Electron Devices, vol. 64, pp. 2977-2983, 2017.

[50] Y. Zhang, Y. Shen, X. Wang, and L. Cao, "A novel design for memristor-based logic switch and crossbar circuits," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, pp. 1402-1411, 2015.

VITA AUCTORIS

NAME:	Farzad Mozafari
PLACE OF BIRTH:	Hamedan, IRAN
YEAR OF BIRTH:	1972
EDUCATION:	Dr. Hossein Fatemi High School, Kermanshah, Iran, 1990
	University of Sistan and Baluchestan , B.Sc., Behshahr, Iran, 1997
	Razi University, M.Sc., Kermanshah, Iran, 2011
	University of Windsor, M.Sc., Windsor, ON, 2023