Design, Analysis, and Testing of Memristive Logic and Authentication Architectures

---

By
Xiaohan Yang

A Dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

School of Engineering, Computing and Mathematics
Faculty of Technology, Design and Environment
of the
Oxford Brookes University

Oxford, UK

Spring 2021

Design, Analysis, and Testing of Memristive Logic and Authentication Architectures

Abstract

The semiconductor industry is already undergoing drastic changes since the transistor-based devices scaling are approaching their physical limits. For example, current technology of the finFET is going towards 5nm. Hence, any variation of the channel size could produce undesirable effects, which brings challenges to addressing the problem. However, memristive devices as alternative emerging technologies have the capability to scale smaller geometries and have found applications not only in high density memory design but also in neuromorphic systems, logic design, secure and crypto systems, etc.

Novel contribution of this thesis starts with a design of a low-complexity high-performance memristive multifunction logic architecture. This architecture is presented for low-power high-frequency operations in a single cycle, which does not require additional control input/logic and multicycle setup/operation. It can be seamlessly integrated with the existing CMOS technology with just one transistor and four memristors design and without additional overhead. This technique can realise both XOR/AND and XNOR/OR operations simultaneously. Experimental results prove that this technique significantly outperforms both CMOS and existing hybrid memristor-CMOS-based designs in terms of chip area, power consumption, and reliable performance especially at high frequencies. Then, we utilise our proposed logic architecture to design a delay-based arbiter physical unclonable function (PUF) by taking the memristive parasitic effects into consideration. The results prove that the

proposed memristive arbiter PUF provides a good performance in terms of uniqueness, uniformity, bit-aliasing and reliability. However, most of the delay-based arbiter PUFs are suffering from the low resistance of the modelling attack because of their linearity. Hence, we leverage the non-linear behaviour of memristor to devise a low overhead architecture for replicating a source memristor to a destination memristor, which can also be used for non-linear encoding/decoding. Based on this architecture, we present a novel PUF framework. This PUF framework is tested by connecting the proposed architecture with two different digital-to-analogue conversion circuits respectively. We demonstrate the effectiveness of the architecture in Challenge-Response-Pair (CRP) based authentication, and for its physical uncloneability. This architecture is highly versatile and can be implemented with a single encoder or a number of encoders running in parallel for extending the sizes of CRPs. Finally, we subject our PUF architectures to machine learning based modelling attacks. We found out that the proposed PUF provides better resistance to such attacks, even for smaller bit sizes and at reduced overheads.

Publications

Research Papers from this Thesis

X. Yang, A. A. Adeyemo, A. Jabir and J. Mathew, "High-performance single-cycle memristive multifunction logic architecture," in Electronics Letters, vol. 52, no. 11, pp. 906-907, 5 26 2016. doi: 10.1049/el.2015.4394

X. Yang, A. Adeyemo, A. Bala and A. Jabir, "Novel memristive logic architectures," 2016 IEEE 26th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), Bremen, Germany, 2016, pp. 196-199. doi:10.1109/PATMOS.2016.7833687 (**Best poster award**)

X. Yang, A. Adeyemo, A. Bala and A. Jabir, "Novel Techniques for Memristive Multifunction Logic Design" in special issue "Modeling, Optimization and Simulation for High Performance and Ultra-Low Power Circuits and Systems" with Integration, the VLSI Journal, available online 15 Sept, 2017. DOI: 10.1016/j.vlsi.2017.09.005

X. Yang, A. Adeyemo, A. Bala and A. Jabir, "Parasitic effects on memristive logic architecture," 2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), Thessaloniki, 2017, pp. 1-5. doi: 10.1109/PATMOS.2017.8106983

X. Yang, S. Khandelwal and A. Jabir, "A secure Memristor Replicator Architecture with physical Uncloneability," in Electronics Letters, Vol. 55, Issue. 24, pp.1275-1277,

doi: 10.1049/el.2019.1538.

X. Yang, S. Khandelwal, A. Jiang and A. Jabir, "A Modelling Attack Resistant Low Overhead Memristive Physical Unclonable Function," 33rd IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Frascati (Rome) Italy, Oct, 2020. DOI:0.1109/DFT50435.2020.9250762

Research Papers in Allied Areas

Adeyemo, **X. Yang**, A. Bala, J. Mathew and A. Jabir,"Analytic models for crossbar read operation," 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), Sant Feliu de Guixols, 2016, pp.3-4. doi:10.1109/IOLTS.2016.7604657

A. Adeyemo, **X. Yang**, A. Bala and A. Jabir,"Analytic models for crossbar write operation", 2016 IEEE Sixth International Symposium on Embedded Computing and System Design (ISED), India, Dec, 2016. Doi: 10.1109/ISED.2016.7977104

A. Bala, A. Adeyemo, **X. Yang** and A. Jabir, "High Level Abstraction of Memristor Model for Neural Network Simulation", 2016 IEEE Sixth International Symposium on Embedded Computing and System Design (ISED), India,Dec, 2016. Doi:10.1109/ISED.2016.7977105

A.Bala, A. Adeyemo, **X. Yang** and A.Jabir,"Learning Method for Ex-situ Training of Memristor Crossbar based Multi-Layer Neural Network" 2017 IEEE 9th International Congress on Ultra Modern Telecom and Control Systems, in Munich, Germany, October 2017, doi:10.1109/ICUMT.2017.8255181 (**Best Paper award**)

A. Bala, **X. Yang**, Adedotun Adeyemo and Abusaleh Jabir," A Memristive Activation Circuit for Deep Learning Neural Networks", IEEE 8th International Symposium on Embedded Computing and System Design (ISED), India, 2018, pp.1-5.

A.Bala, **X. Yang**, A. Adeyemo, A. Jabir,'Efficient and Low Overhead Memristive Activation Circuit for Deep Learning Neural Networks' Journal of Low Power Electronics (2019)ISSN: 1546-1998 eISSN: 1546-2005

Patents

A. Jabir, **X. Yang** and Adeyemo. Adedotun, Memristive multifunction logic architecture, UK Patent App. 1603089.2, Feb. 2016

A. Jabir, **X. Yang** and S. Khandelwal,Memristor-based Circuit (replicator/encoder), UK Patent App. 1905392.5, Apr. 2019

A. Jabir, S. Khandelwal and **X. Yang**, Sensor with Logic Architecture, UK Patent App. 1914221.5, Oct, 2019

Declaration

This Thesis is submitted to the Oxford Brookes University in accordance with the requirements of the award of Doctor of Philosophy in the Faculty of Technology, Design and Environment. It has not been submitted for any other degress or diploma of any examining body. Some parts of the work presented in this thesis have previously appeared in the published papers listed in the publications section. Except where specifically acknowledged, it is all the work of the Author.

Xiaohan Yang, Mar, 2021

Acknowledgments

This project is the result of weeks of hard work and perseverance. I would like to express my enormous gratitude to my supervisor and my research team members.

My sincere appreciation to Dr.Abusaleh Jabir, Dr.Adeyemo. Adedotun, Dr.Saurabh Khandelwal, Prof.Dhiraj Pradhan and Anu Bala who are always willing and able to give me the required support for the duration of this work. I also offer my big thanks to Aiqi Jiang who is the researcher in Cognitive Science Group, at Queen Mary University of London, for her support of the Machine Learning implementation in my project.

I am thankful to my friends: Cangli Duan, Lin Tan, Jianren Chen, Ming Li and Dan Cao, etc. They were always there for me especially during the session of writing this dissertation.

I also offer my profound gratitude to my wonderful families (Jianguo, Xiaohong, Xiaobai Chen, Tiankai, Zaizai and Menkai, etc). They always give me the encouragement and unconditional support.

Contents

Listing of figures

Listing of tables

Listing of Acronyms

1T-4M	One Transistor and Four Memristors
1T1R	One Transistor and One Resistor
APUF	Arbiter Physical Unclonable Function
ANN	Artificial Neural Network
CRP	Challenge Response Pair
CMOS	Complementary Metal-Oxide-Semiconductor
DIBL	Drain-Induced Barrier Lowering
EDA	Electronic Design Automation
FRAM	Ferroelectric Random Access Memory
GF	Galois Field
HRS	High Resistance State
IRDS	International Roadmap for Devices and Systems
I-V	current-voltage
LE	Linear Encoder
LR	Logistic Regression
LRS	Low Resistance State
MRAM	Magnetoresistive Random Access Memory
MNE	Memristive Non-linear Encoder

MLP	Multi-layer Perceptron
MRL	Memristor Ratio Logic
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MOM	Metal-Oxide-Metal
MVL	Multiple Valued Logic
PCM	Phase Change Memory
PUF	Physical Unclonable Function
RBF	Radial Basis Function
ReRAM	Resistive Random Access Memory
RF	Random Forest
STTRAM	Spin-Transfer-Torque Random Access Memory
SOI	Silicon-On-Insulator
SVM	Support Vector Machine

1

Introduction

1.1 Motivation & Background

The conventional metal oxide semiconductor (MOS) transistor plays a significant part in the integrated circuits since the late 1950s. The observation, made by Gordon E. Moore in 1965, is that the number of transistors in a silicon chip circuit doubles approximately every 18 months, though the cost of the device is decreasing by a certain factor with time [12]. The prime part of this observation indicates that computing devices become smaller, faster, and cheaper. Since then, researchers started to scale down the physical feature size (gate length) of a MOS transistor to reduce the manufacturing cost, power consumption and also increase the yield and operation speed of the devices. Fig. 1.1 shows the transistor scaling trends over the past few decades. As the transistor gate length has shrunk (red line), the transistor density of each chip (blue line) has increased drastically. However, scaling rate has slowed down recently according to the data collected from the International Roadmap for Devices and Systems (IRDS) [13], as transistors approach their fundamental physical limit. This scaling challenge is caused by multiple factors. For example, in the 3-dimension aspect, as we shrink the channel length of the transistor, the threshold voltage goes down as we expect. However, one of the strengths of and beauties of the MOSFET transistors is that we have two degrees of freedom. Not only can the length of the transistor be changed, but also the width of the transistor can be changed. As the width of the transistor also continues to shrink, the depletion region formed on the edge sides of the transistor becomes the larger fraction of the total depletion region and a higher gate voltage is needed to maintain this larger depletion region. Therefore, the threshold voltage increases which reduces gate overdrive. This is known as narrow-width effect. Another undesirable factor that increases the scaling challenge is leakage current. When the source and drain are close to each other and two depletion regions start to meet. In this case, the barrier is lowered. According to the

Boltzmann or Fermi Dirac distribution, a little lowering of the potential barrier results in a lot more leakage. This also results in the failure to switch off the transistor properly and causes more power dissipation. Some other unwanted short-channel effects also limit the scaling such as drain-induced barrier lowering (DIBL) and velocity saturation among others. MOS transistors are also affected by their parasitic capacitances, which limit their performance, especially when operating at high frequencies. In addition, these parasitic effects are amplified as the size of the transistor is further reduced [14]. Apart from these issues, the economic challenges have erected another barrier to the MOS transistors' scaling. In the chip manufacturing process, the most expensive part is the lithography mask. These masks have to be made with electron beam lithography and cannot be processed in batches. In the most advanced manufacturing process, the number of masks will increase as the transistor technology node continues to shrink.

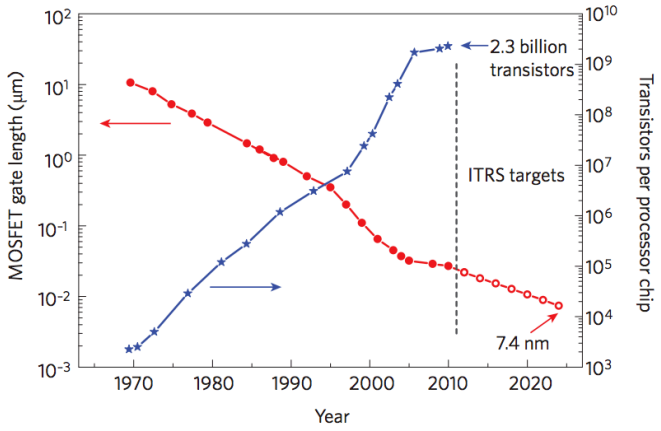


Figure 1.1: MOSFET scaling trends and number of transistors per chip [1].

Considering the pros and cons of the existing CMOS technology, chip manufacturers are beginning to invest huge resources to explore alternative technologies for the evolution of computing devices that might sustain Moore's law for another decade. The most advanced technologies are devoted to developing new device structures and

implementing new materials. These novel technologies can be separated into two categories as follows:

1. *Modification of the conventional MOS transistor structures (CMOS extension):* Nowadays, researchers are trying to exert the maximum control over the channel by mechanically making the gate oxide thinner and geometrically making the gate wrapped around the channel as much as possible such as in FinFETs [15]. Another novel FET structure is called silicon-on-insulator (SOI), which is a small piece of the silicon sitting on a dielectric (SiO_2) [16]. In the conventional CMOS technology, an n-well is created in a p-type substrate to isolate the different MOSFETs, which also increases the parasitic capacitances since the PN junction is formed here. For the SOI structure, however, the transistor can be implemented directly inside of SiO_2 instead of having the PN junction on the bottom side. Hence, this effectively eliminates the parasitic capacitances which are usually created by n-well and p-type substrate. Additionally, this technology also allows people to develop a very compact device.

2. *Development of the new device structures (Beyond-CMOS devices):* The aforementioned challenges and limitations of the MOS scaling also give impetus to investigations of various 'emerging technologies'—technologies that may bring the semiconductor industry into the era of 'Beyond CMOS'. According to the report by IRDS [13], these emerging technologies can be categorised as follows:

    (a) *Emerging Logic Devices:* IRDS divides emerging logic devices into two groups based on state variables: charge-based devices (e.g., Tunneling FETs [17], Ferroelectric-FETs [18], and graphene pn junction devices [19]) and non-charge based devices. The most representative of non-charge based devices are spintronic devices (e.g., Spin Wave Devices [20], Domain Wall Logic Devices [21], and All-Spin Logic Devices [22]).

(b) *Emerging Memory Devices:* In 1971, Leon Chua found the missing link between charge(q) and flux(φ) and theorised a new electronic element, memristor (short for 'memory-resistor'). This element shows the inherent hysteresis I-V relationship and retains its previous state after power off. Furthermore, the element can also be fabricated in the size of the nanoscale and intersected into the crossbar architecture to build a highly dense device structure. Owing to these superiorities, the memristive device is a highly promising technology as an alternative to extending Moore's law. So far, the number of novel memory devices is designed in this fashion such as Resistive Random Access Memory (ReRAM) [23], Phase Change Memory (PCRAM) [24], Spin-Transfer-Torque RAM (STTRAM) [25] and Memristor [26] which this work focusses on. In addition to these memristive devices, there are still a large number of novel devices included in this category (e.g.,Magnetoresistive RAM (MRAM) [27] and Ferroelectric RAM (FRAM) [28]) .

(c) *Emerging Architectures:* There are a number of novel architectures which have been proposed beyond conventional von Neumann architecture. Those emerging architectures include cellular automata, co-located memory logic [29] (e.g., processor-in-memory and computational memory) and cognitive computing [30](e.g., neuromorphics and machine learning).

Among the emerging technologies mentioned above, memristive devices have shown great potential for building the next generation devices not only in the design of the high-density memory devices but also in other aspects such as logic circuit, neural networks, cryptosystems (e.g., physical unclonable function) and memristive gas sensor etc. In this project, the author mainly focusses on the design of memristor-based

logic and authentication architectures.

1.2 Methodology

This work covers the research, technology and application categories in the area of design of the logic circuit and security systems. All the designs were implemented and simulated in spectre-based Cadence Virtuoso 6 Electronic Design Automation (EDA) and LTspice. All the diagrams including experimental demonstrations are either drawn using Xcircuit or automatically generated from Cadence Virtuoso EDA tools, MATLAB and Python3 by the author.

This thesis is written with the LaTeX document preparation system.

1.2.1 Author's Contributions to Subject Area

1. The author presents techniques for realising reliable logic functions and more complex systems based on the switching characteristics of memristors. The author also shows that memristive circuits have inherent properties for realising multiple valued MIN-MAX operations over the post algebra [31]. Then, an efficient hybrid 1T-4M multifunction logic architecture for seamless integration with existing CMOS technology is presented. Although memristors are usually considered to operate at lower frequencies, however, recent advances in technology show their potential at high frequencies. Hence, the author also explores the effects of high frequencies on their performance and thereby proposed reliable high frequency design techniques based on our 1T-4M architectures. Experimental results, based on the design of full adders and multipliers over Galois Field (GF), show that our proposed design requires significantly less power while maintaining reliable performance at low as well as at high frequencies compared to the existing techniques.

2. Most of the proposed memristor models have not taken the parasitic effects into the consideration. Hence, a general memristor model is applied to the proposed 1T-4M architecture to emulate the practical memristive logic architecture. Experimental results demonstrate that the proposed architecture almost eliminates those oscillating effects which are generated by the parasitic components. Meanwhile, the propagation delay and the variation of the architecture are increased. With the observation and analysis on cascaded memristive logic design, the author shows that the proposed architecture can build a more robust system compared to the existing memristive logic technique. Hence, the author proposed a delay-based arbiter Physical Unclonable Function (PUF) afterwards, which leverages the physical variations of 1T-4M architecture.

3. Most of the delay-based arbiter PUFs are vulnerable to the machine learning based modelling attack since the mapping from challenges to responses shows a certain degree of linearity. Hence, a framework for non-linear memristive PUF using a non-linear memristive digital-to-analogue conversion circuit conjunction with a memristive non-linear encoder is proposed and analysed. The memristive digital-to-analogue circuit is implemented by the two different techniques (the proposed memristive reverse decoder and memristive chaos circuit) respectively. The architecture of memristive non-linear encoder (MNE) is not only used for encoding but also can be used for replicating the resistance of the memristor. For the PUF application, the multiple MNEs can also place in parallel to increase the size of responses and reduce the quantisation error. This also offers higher flexibility in terms of hardware requirements. Finally, the author applied machine learning based modelling attacks e.g. Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), as well as Artificial Neural Network (ANN) classifiers. The experimental results show that the proposed PUF architecture can resist the machine learning-based modelling

attacks.

1.3 Outline of this Thesis

Subsequent chapters of this work are organised as follows:

- Chapter 2 presents the background concepts related to the memristor theory and describes the physical model of the memristor. Some existing memristor models are introduced in this chapter. This chapter also extensively summarises memristor-based applications. Some existing memristive applications related to logic architecture and the Physical Unclonable Functions (PUFs) are explained concretely.

- Chapter 3 presents novel techniques for realising low overhead logic functions and more complex systems based on the memristors. Firstly, the chapter begins with memristive multiple-valued operations (MIN-MAX operations over the post algebra). Then, an efficient hybrid 1T-4M logic architecture for dual XOR/AND and XNOR/OR functionality was proposed. A comprehensive analysis of the proposed architecture was carried out specifically for the high frequency operations. This chapter also presents a novel implementation and analysis of some memristive logic systems such as the full adder and Galois Field (GF) Multiplier.

- Chapter 4 constitutes two parts. The first part demonstrates the parasitic effects of the memristive 1T-4M logic architectures. By comparing with some existing memristive logic architectures, the proposed architecture shows more robustness under the parasitic effects. In the second part, a delay-based memristive arbiter PUF is proposed by utilising the 1T-4M logic architectures. The

systematic analysis (uniqueness, uniformity, bit-aliasing and reliability) was carried out to determine the performance of the proposed PUF architecture.

- In Chapter 5, the author firstly demonstrated a lightweight and highly versatile architecture that can realise memristive non-linear encoding/decoding process and memristive replicating process. Then, this architecture was joined to the memristive reverse decoder and the memristive chaotic circuit respectively to form two different PUF architectures (e.g., PUF with reverse decoder and PUF with chaos circuit). After that, the systematic analysis was carried out to measure the performance of the proposed PUFs. Finally, various machine learning-based modelling attacks were applied to demonstrate the highly modelling attack resistance of the proposed PUF architectures.

- Chapter 6 concludes and summarises the work presented in this thesis. Other possible extension of the work carried out in this thesis was also identified.

2

Background and Literature Review

2.1 Introduction

This chapter introduces the basic concepts of memristors. A brief description of the physical structure of the memristor device is given in Section 2.2.1. Then, some existing memristor models with their current-voltage (I-V) characteristics are discussed in Section 2.2.2. This thesis mainly focuses on memristor-based logic design and unclonable memristive architecture. Therefore, we give explanations of the existing techniques for the memristor-based logic design in Section 2.3.1 and an overview of the different types of physical unclonable functions as described in Section 2.3.2.

2.2 Background

2.2.1 Memristor Overview

The idea of the memristor was first theorised by Leon Chua in 1971 as the fourth two-terminal electronic device along with the existing resistor, inductor and capacitor [26]. It specifies the relationship between charge (q) and flux (φ).

The four basic circuit variables voltage (v), current (i), flux (φ) and charge (q) create six possible links as shown in Fig. 2.1. Three relationships had already been realised and described as electronic components such as resistor, inductor and capacitor. As shown in Fig. 2.1, resistor (R) represents the link between the voltage and current as $dv = Rdi$. Inductor (L) represents the link between current and flux as $d\varphi = Ldi$ and capacitor (C) represents the link between the voltage and charge as $dq = Cdv$. However, none of the existing devices could define the missing link between charge (q) and flux (φ)until Leon Chua published his paper "Memristor-The Missing Circuit Element" [26]. In that paper, he claimed that there is a fourth circuit element, memristor (short for 'memory-resistor'), exists and can be defined as $d\varphi = Mdq$, where M represents the memristance of the memristor. Hence, memristance could be described

as follows:

$$M = d\varphi/dq \tag{2.1}$$

Here, φ and q can be expressed as follows:

$$\varphi(t) = \int_{-\infty}^{t} v(t)dt \tag{2.2}$$

$$q(t) = \int_{-\infty}^{t} i(t)dt \tag{2.3}$$

Then, substituting Eq. (2.2) and Eq. (2.3) into Eq. (2.1):

$$M(q) = \frac{d\varphi/dt}{dq/dt} = \frac{v(t)}{i(t)} \tag{2.4}$$

Here, we have the memristance $M(q)$ described by Ohm's Law.

Similarly, memductance $W(\varphi)$ can be defined by swapping numerator and denominator of Eq. (2.4):

$$W(\varphi) = \frac{dq/dt}{d\varphi/dt} = \frac{i(t)}{v(t)} \tag{2.5}$$

Memristor thus depends on the time integral of voltage or current. Hence, it can retain its previous memristance value after power has been removed, thereby remaining non-volatile. Leon Chua also argued that the memristor, as a passive device, cannot store any energy as capacitors or inductors. Hence, both $v(t)$ and $i(t)$ of memristor must be in phase. In other words, the voltage must be zero whenever the current equals to zero and vice versa. Fig. 2.2 is the I-V characteristic of the memristor which shows that the voltage and current are pinched at the origin point. Therefore, Fig. 2.2 is also known as the memristor pinched hysteresis loop. However, the loop becomes narrow continuously as frequency (ω) increases as shown in Fig. 2.3 [2]. At high frequencies, the memristor does not have enough time to adjust its memristance. Finally, the

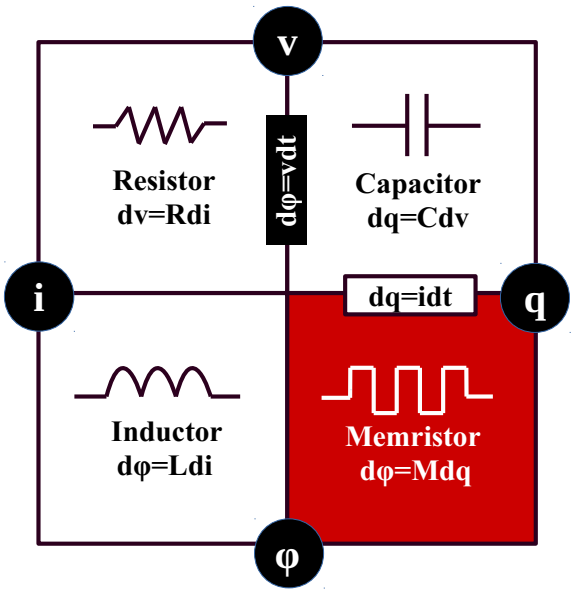


Figure 2.1: 4 elements (voltage (v), current (i), flux (φ) and charge (q)) relationships.

pinched hysteresis loop tends to be a straight line as $\omega \to \infty$, which behaves like a linear resistor [32].

2.2.1.1 Titanium Dioxide (TiO_2) Based Memristor

One of the most widely used technologies for fabricating memristive devices is based on oxides such as NiO, TiO_2, HfO_x and SiO_x [3, 33]. However, the most well-known physical memristive device is based on Titanium Dioxide and was fabricated by Hewlett-Packard Lab in 2008 [3]. The simplified physical structure is shown in Fig. 2.4 (a). It is composed of a heavily doped TiO_{2-x} layer with oxygen vacancies, placed above a zero doped TiO_2 layer and sandwiched between a pair of metallic electrodes. Here, D is the width of the device and w is the width of the doped region. w acts as the state variable which depends on the previously applied voltage. The symbol of the memristor is illustrated in Fig. 2.4 (b), where p and n represent the positive and the negative terminals respectively. This symbol will represent the memristor for the rest

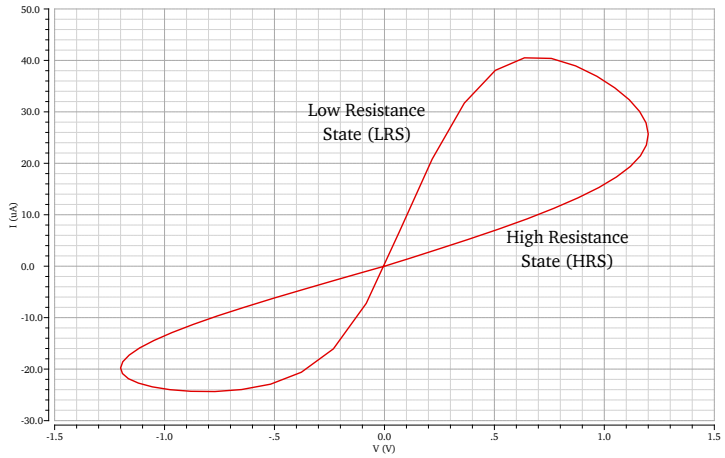


Figure 2.2: Memristor voltage-current characteristics.

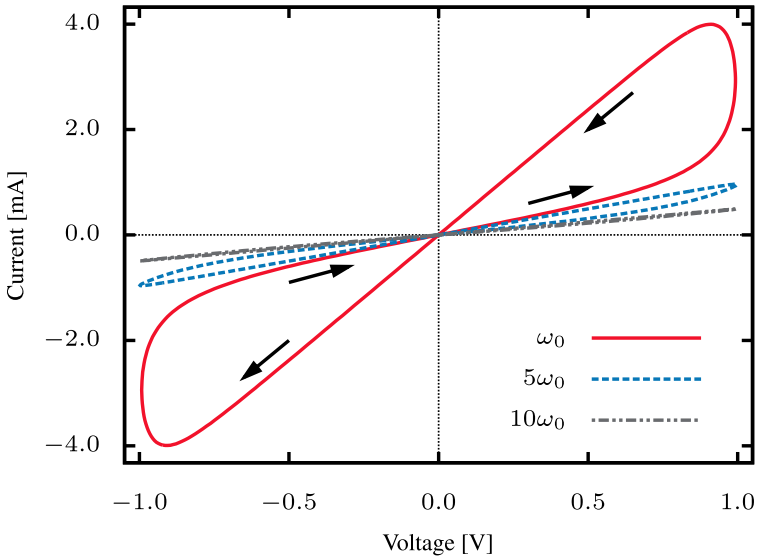


Figure 2.3: Frequency response of pinched hysteresis loop [2].

of the thesis.

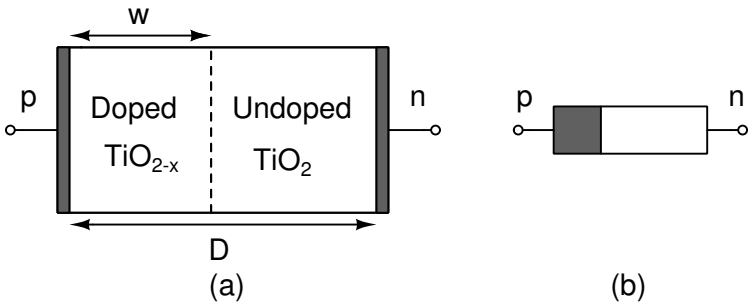


Figure 2.4: Memristor: (a) Physical structure of the memristor fabricated by HP Lab; (b) Symbol of the single memristor.

When a positive voltage is applied to the p-terminal (and a negative voltage is applied to the n-terminal) of this device, the oxygen vacancies, which constitute positive charges, drift into the zero doped region. It increases the width, w, of the TiO_{2-x} layer. This results in the device switching to a low resistance state (LRS).

However, when a negative voltage is applied to the p-terminal (and positive voltage is applied to the n-terminal) the oxygen vacancies are attracted to the p-terminal. This results in the TiO_2 layer $D - w(t)$ widening and TiO_{2-x} layer $w(t)$ decreasing. Hence, the device switches to a high resistance state (HRS).

In contrast, when there is no voltage applied to either terminal, the boundary between the two titanium dioxide layers freezes. This allows the memristor to retain its previous state.

To generalise a basic mathematical model, several assumptions are made: ohmic electronic conduction, linear ion drift in uniform field, and the ions have equal average ion mobility. Hence, memristance can be defined as follows [3]:

$$M(t) = R_{\text{on}}\frac{w(t)}{D} + R_{\text{off}}(1 - \frac{w(t)}{D}), \tag{2.6}$$

where, R_{off} is the resistance of the TiO_{2-x} layer (undoped region) and R_{on} is the

resistance of the TiO_2 layer (doped region). From Eq. (2.6), $M(t) = R_{on}$ when $w(t) = D$, and $M(t) = R_{off}$ when $w(t) = 0$. Ideally, $R_{on} \ll R_{off}$. However, the actual values of R_{on} and R_{off} depend on different memristive devices [6]. [3] developed the memristor model with the resistance ratio (R_{off}/R_{on}) around 160∼380. The state variable $w(t)$ in Eq. (2.6) can be defined as:

$$\frac{dw}{dt} = \mu_n \frac{R_{on}}{D} i(t), \tag{2.7}$$

$$w(t) = \mu_n \frac{R_{on}}{D} q(t), \tag{2.8}$$

where, μ_n represents the average ion mobility. Eq. (2.8) is only valid for w in the interval [0, D]. $w(t)$ is proportional to q that passes through the device until its approaches D. Fig. 2.2 shows the I-V characteristics of this device and the switching behaviour between LRS and HRS. This figure clearly shows hysteresis, which is the key characteristics of a memory device.

2.2.1.2 Applications of Memristors

Since Hewlett-Packard Lab fabricated the TiO_2 based physical memristive device, there has been increasing interest in different aspects of its applications [34]. To this end, memristors have been applied in areas such as high-density memory design [35, 36], neuromorphic systems [37, 38], secure and crypto (e.g. Physical Unclonable Functions (PUF) etc.) systems [39, 40], and logic design [2, 7, 9, 11, 41, 42]. In the high-density memory structure, memristors are designed as the memory cells on the crossbar layers and can be manipulated by the CMOS-based peripheral control circuit [43]. Neuromorphic systems design is another application area, which has multiple displays of using memristors. In this case, a signal memristor can be acted as a two-terminal synapse structure, which is like a signal connection between the neurons in the human brain. The resistance or conductance of the memristor can

act as a synaptic weight. Section 2.3.1 and Section 2.3.2 overview the memristor applications in logic design and cryptosystem respectively.

Memristor can be scaled to very small geometries and can be fabricated in layer upon layer, thereby providing a 3D structure for memristor chips. Memristors can also be interfaced with the existing CMOS technology, owing to the fact that they both share similar fabrication properties [44]. The ability to fabricate chips in 3D in this fashion can help eliminate a key shortcoming of the CMOS technology, which suffers from the difficulties associated with 3D fabrication [45]. Hence, these types of interfacing allows 3D fabrication of hybrid memristor-CMOS designs, where the CMOS can be fabricated along with the bottom layer and the memristors are stacked over the CMOS layer [46, 47].

2.2.2 Memristor Models

Over the past few years, memristive devices are characterised in various models owing to the effective usages for the different applications. One of the memristor applications mentioned in Section 2.2.1.2, is using the memristor as a switch in Memristor Ratio Logic (MRL), which is similar to MOSFET. The operation is based on switching the memristance (R_{on} to R_{off}) according to the input voltages and by changing the voltage at the output. This requires that the memristor works with voltage or current thresholds, and also needs to provide fast switching speed between two different resistance states. For the high-density memory application, apart from fast read and write access required, the memristor must work with low sensitivity to various parameters and operating conditions, especially in the reading process. Both memristive logic and high-density memory design, require that the memristor has high R_{off} to R_{on} ratio and can operate in low power consumption. Therefore, various memristor models proposed to emulate the physical properties as those applications required. In this section, the different models are discussed in the following paragraphs.

2.2.2.1 Linear Ion Drift Model

The linear ion drift model was introduced in [3] and it came from the titanium based physical memristive device which was fabricated by HP group. They assume that the memristor consists of two regions: doped region TiO_{2-x} with the width of w and undoped region TiO_2 with the width of $D-w$ as shown in Fig. 2.4 (a). Therefore, the memristor is modeled as two regions or resistors connected in series under assumptions of ohmic conductance and linear ion drift with the average ion mobility μ_n. The basic equation of the memristance indicated in Eq. (2.6). Hence, the voltage across the memristor can be represented as follows:

$$v(t) = M(t)i(t) \tag{2.9}$$

$$v(t) = (R_{on}\frac{w(t)}{D} + R_{off}(1 - \frac{w(t)}{D}))i(t) \tag{2.10}$$

The I-V characteristic curve is shown in Fig. 2.5.

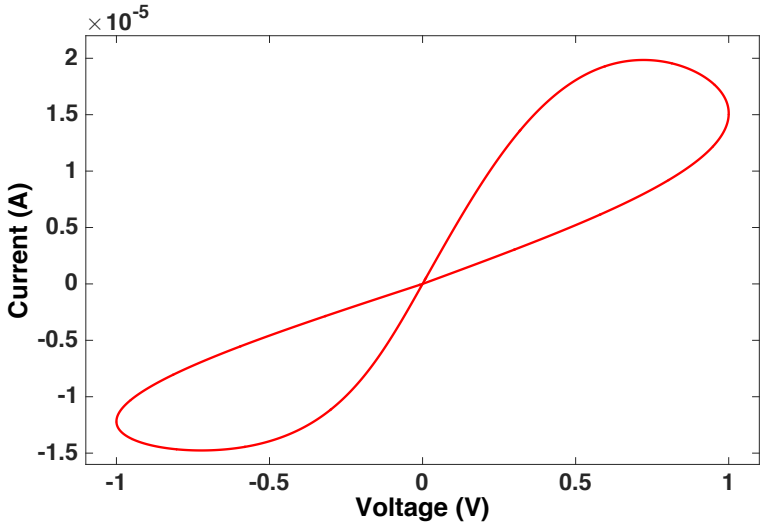


Figure 2.5: Memristor voltage-current characteristics for linear ion drift model. R_{on}=2.5kΩ, R_{off}=100kΩ, D=10nm, and μ_n=10e-10$m^2s^{-1}V^{-1}$ [3].

2.2.2.2 Non-linear Ion Drift Model

As most fabricated memristors manifest non-linear behaviour, a non-linear model is required [4]. Unlike the ion drift model as shown in Fig. 2.5, the current and voltage relationship of non-linear model can be described as follows:

$$i(t) = w(t)^n \beta \sinh(\alpha\ v(t)) + \chi[exp(\gamma\ v(t)) - 1] \tag{2.11}$$

where $\beta \sinh(\alpha\ v(t))$ describes the state when the electrons tunnel through a barrier (ON state); the second term $\chi[exp(\gamma\ v(t)) - 1]$ describes the state when the memristor is switched off. α, β, χ and γ are fitting constants which are used to characterise the behaviours of the specific physical device. n is interpreted as the non-linearity of the ion drift velocity of the memristive device. The I-V characteristics is shown in Fig. 2.6.

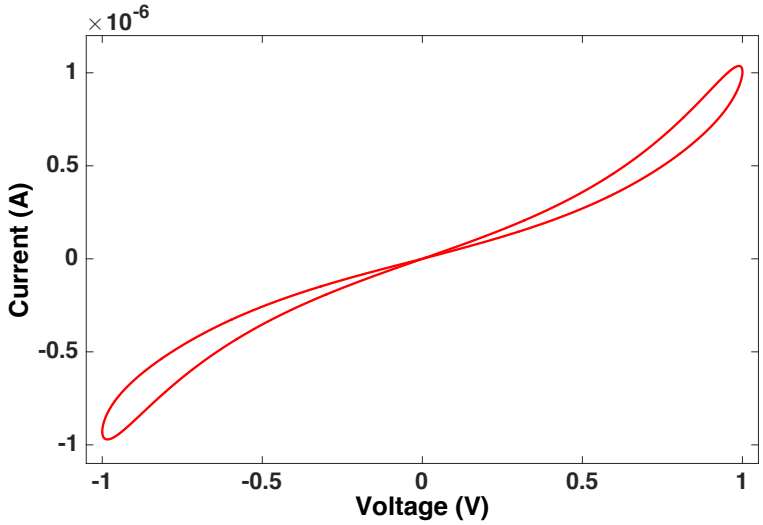


Figure 2.6: Memristor voltage-current characteristics for non-linear ion drift model. The values of these fitting constants are $\alpha = 2$, $\beta = 9$, $\chi = 0.01$, and $\gamma = 4$ [4].

2.2.2.3 Simmons Tunnel Barrier Model

This model assumes an asymmetric and non-linear switching behaviour as defined in [48] because of the exponential dependence of ionised dopants. This model is different from the drift models discussed above in the way it models the device, instead of two serial resistors, a resistor is connected in series with an electron tunnel barrier as shown in Fig. 2.7. The width of the tunnel barrier is represented by the state variable x. The I-V characteristics is based on the Simmons tunnelling model [49] defined by:

$$\begin{aligned} i(t) =& \tilde{A}(x, v_g)\phi_1(v_g, x)exp(-B(v_g, x).\phi_1(v_g, x)^{1/2}) - \tilde{A}(x, v_g)(\phi_1(v_g, x) + e|v_g|) \\ & \times \exp(-B(v_g, x).(\phi_1(v_g, x) + ev_g)^{1/2}) \end{aligned} \tag{2.12}$$

where, $\tilde{A}$ is the barrier cross-sectional area; e is the electron charge; ϕ_1 is the barrier height; B is the function related to uncertainty principle, and v_g is the voltage in undoped region. Here, v_g is defined as follows:

$$v_g = v - i(t)R_s \tag{2.13}$$

where, v is the internal voltage in the device.

2.2.2.4 ThrEshold Adaptive Memristor (TEAM) Model

The TEAM model is regarded as a generic memristor model because the I-V characteristics is undefined and can be in a linear or an non-linear form. The TEAM model was proposed in [5] as an adaptation of the Simmons Tunnel Barrier Model and has since gained a high degree of acceptance because of its simplicity and accuracy. A unique benefit of the TEAM model is its undefined I-V relationship. Users can freely

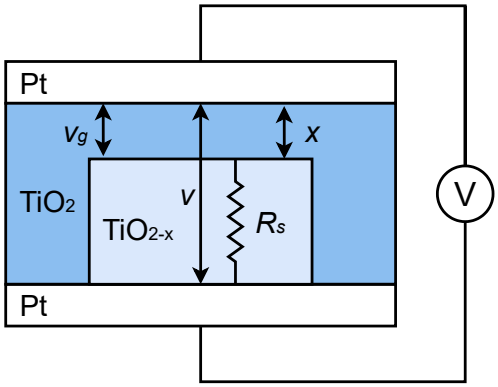


Figure 2.7: Physical model of Simmons tunnel barrier memristor. The state variable x is the width of the oxide region, and V is the applied voltage on the device which does not necessarily equal to v [5].

choose any existing I-V relationship. Fig. 2.8 shows the I-V characteristic of the TEAM model. The I-V relationship is given by following equation if memristance changes linearly in state variable of the memristor x:

$$v(t) = [R_{\text{on}} + \frac{R_{\text{off}} - R_{\text{on}}}{x_{\text{off}} - x_{\text{on}}}(x - x_{\text{on}})] \cdot i(t), \tag{2.14}$$

If any changes in memristance of TEAM model depends exponentially on state variable x, the I-V relationship becomes

$$v(t) = R_{\text{on}} e^{(\lambda / x_{\text{off}} - x_{\text{on}})(x - x_{\text{on}})} \cdot i(t), \tag{2.15}$$

where λ is a fitting parameter and R_{on} and R_{off} are the resistance at the boundary of the memristor, which satisfy

$$\frac{R_{\text{off}}}{R_{\text{on}}} = e^{\lambda}. \tag{2.16}$$

The state variable x is defined by:

$$\frac{dx(t)}{dt} = \begin{cases} k_{\text{off}} \cdot (\frac{i(t)}{i_{\text{off}}} - 1)^{\alpha_{\text{off}}} \cdot f_{\text{off}}(x), & 0 < i_{\text{off}} < i \\ 0, & i_{\text{on}} < i < i_{\text{off}} \\ k_{\text{on}} \cdot (\frac{i(t)}{i_{\text{on}}} - 1)^{\alpha_{\text{on}}} \cdot f_{\text{on}}(x), & i < i_{\text{on}} < 0 \end{cases} \tag{2.17}$$

where, k_{on} k_{off} α_{off} and α_{on} are fitting parameters. i_{off} and i_{on} are current thresholds. The functions $f_{\text{off}}(x)$ and $f_{\text{on}}(x)$ behave as the window functions described in Table 2.1.

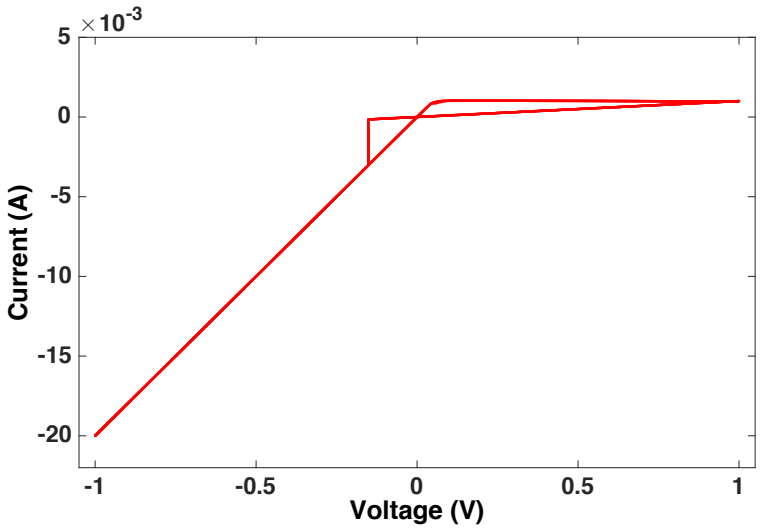


Figure 2.8: Memristor voltage-current characteristics for TEAM model. R_{on}=50kΩ, R_{off}=1kΩ, x_{off}=3nm, i_{on}=-1e-6A, i_{off}=1e-6A, k_{on}=-8e-13m/s, k_{off}=8e-13m/s, α_{on}=3 and α_{off}=3 [5].

2.2.2.5 Voltage ThrEshold Adaptive Memristor (VTEAM) Model

The VTEAM model was proposed as an extension of the TEAM model by the same research team. The VTEAM model [6] exhibits threshold voltages as against the TEAM model that exhibits current thresholds(i_{off}, i_{on}). Hence, the VTEAM model is more appropriate for certain memory and logic circuit applications. The derivative

of the state variable w for the VTEAM is

$$\frac{dw(t)}{dt} = \begin{cases} k_{\text{off}} \cdot (\frac{v(t)}{V_{\text{off}}} - 1)^{\alpha_{\text{off}}} \cdot f_{\text{off}}(w), & 0<V_{\text{off}}<\text{v} \\ 0, & V_{\text{on}}<\text{v}<V_{\text{off}} \\ k_{\text{on}} \cdot (\frac{v(t)}{V_{\text{on}}} - 1)^{\alpha_{\text{on}}} \cdot f_{\text{on}}(w), & \text{v}<V_{\text{on}}<0 \end{cases} \tag{2.18}$$

where, k_{on} k_{off} α_{off} and α_{on} are fitting parameters. V_{on} and V_{off} are voltage thresholds. The functions $f_{\text{off}}(x)$ and $f_{\text{on}}(x)$ behave as the window functions described in Table 2.1. [6] also shows that the VTEAM model can successfully fit different memristive devices (e.g. Pt-Hf-Ti, Ferroelectric, and Metallic nanowire memristor). In this thesis, most of our proposed memristive logic designs are based on this VTEAM model. Fig. 2.9 shows the current-voltage characteristic of the VTEAM model.

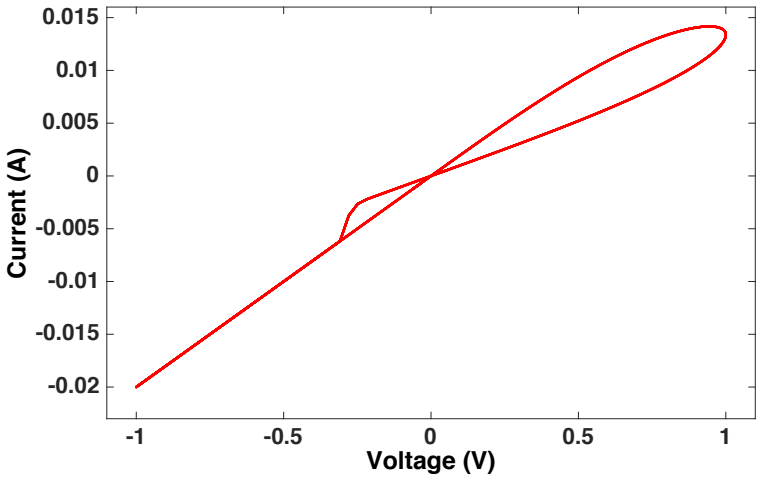


Figure 2.9: Memristor voltage-current characteristics for VTEAM model. R_{on}=50Ω, R_{off}=100Ω, x_{off}=3nm, V_{on}=-0.2V, V_{off}=0.02V, k_{on}=-10m/s, k_{off}=5e-4m/s, α_{on}=3 and α_{off}=1 [6].

Table 2.1: Comparison of different window functions

Window Function	**Biolek** [50]	**Joglekar** [51]	**Prodromakis** [52]	**TEAM** [5]	**VTEAM** [6]
Function $f(w)$	$1-(w/D-stp(-i))^{2p}$	$1-(2w/D-1)^{2p}$	$j(1-[(w-0.5)^2+0.75]^p)$	$exp[-exp(\|x-x_{on,off}\|W_c)]$	$exp[-exp(\|w-w_{on,off}\|W_c)]$
Compatible Memristor Models	Linear, Non-linear ion drift & TEAM	Linear, Non-linear ion drift & TEAM	Linear, Non-linear ion drift & TEAM	TEAM	VTEAM
Symmetric	Yes	Yes	Yes	No	No

2.2.2.6 Window Functions

The target of using a window function is to ensure that the state variable of memristor w or x does not exceed its specific boundary. Table 2.1 shows a comparison between the different window functions.

2.3 Literature Review

In this thesis, we have designed a novel multifunction logic architecture based on the memristors and come up with some novel memristive cryptographic systems. Hence, in this chapter, Section 2.3.1 provides some of existing architectures and techniques for the memristor based logic circuit. Section 2.3.2 gives an overview of some conventional Physical Unclonable Functions (PUFs) and existing memristive PUFs.

2.3.1 Existing Memristive Logic Architectures

The previous sections demonstrate some memristor characteristics (e.g., switching, non-volatility, etc.). Therefore, this section shows how these characteristics can be used to design a memristor-based logic circuit. For the memristive logic design, most of the existing techniques can be categorised as stateful logic and non-stateful logic. Here, the author outlines both of these techniques as follows:

2.3.1.1 Memristive Stateful Logic Architectures

In contrast to the CMOS logic, the memristors shown in this section can not only store logic values but also perform logic operations which is defined as stateful logic.

- *IMPLY Logic Gate:* One of the earliest memristive logic architectures is the IMPLY logic function, which could be implemented as a memristor-based crossbar array [7]. Fig. 2.10(a) shows the basic IMPLY logic structure, where the two memristors P and Q are connected by the same horizontal wire with different voltages V_{COND} and V_{SET} applied simultaneously. The initial resistance of P and Q are the inputs of the logic gate. After applying V_{COND} and V_{SET}, the output of the logic gate is the final changed resistance of the memristor Q. Here the amplitude of V_{COND} is smaller than V_{SET} ($|V_{\mathrm{COND}}| < |V_{\mathrm{SET}}|$), which prevents the device Q from switching to the previous state when $P = 1$ (low resistance). For example, if memristor P is in low resistance state, the voltage on the common terminal is $V_{\mathrm{COND}} - V_{Ron} \approx V_{\mathrm{COND}}$, where, V_{Ron} is the voltage across P. Hence, the voltage on Q is approximately $V_{\mathrm{SET}} - V_{\mathrm{COND}}$, which is sufficiently small to maintain the logic state of Q (Case 3 and 4 of Table 2.2). In the case of $P = 0$ and $Q = 0$ (high resistances), V_{SET} is applied on Q, which is switched to the low resistance state ($Q = 1$). In the case of $P = 0$ and $Q = 1$, the Q is maintained in its previous state (Case 1 and 2 of Table 2.2). The complete truth table of imply logic function is shown in Table 2.2. However, this technique is designed on the memristor-based crossbar. It suffers from leakage current effect, known as the "sneak paths" effect, which is common in memristive crossbar memory architectures [53]. The "sneak paths" effect is the leakage current sneaking through the undesired cells with smaller resistances which are data dependent (See Fig. 2.11). These resistances cause overlapping of the two regions of the '0' and '1', which significantly reduce the noise mar-

Table 2.2: Truth Table of imply logic function [7].

Case	P	Q	P→Q
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

Table 2.3: Logical operation of an IMPLY-based NAND [7].

Step	Voltages
Step 1: $S = 0$	$V_s = V_{CLEAR}$
Step 2: $P \rightarrow S$	$V_P = V_{\mathrm{COND}}$ $V_s = V_{\mathrm{SET}}$
Step 3: $Q \rightarrow S$	$V_Q = V_{\mathrm{COND}}$ $V_s = V_{\mathrm{SET}}$

gin (reliability). Hence, this deteriorates the accuracy of the read operation in crossbar architectures [8]. It is quite a challenge to effectively eliminate the sneak-path without massively extending the circuit size. Additionally, this technique requires multiple clock cycles to operate the logic functions. For example, a simple IMPLY NAND gate as shown in Fig. 2.10(b) requires three sequential steps, which will slow down the speed of the computation. Table 2.3 shows the sequence of a two input NAND.

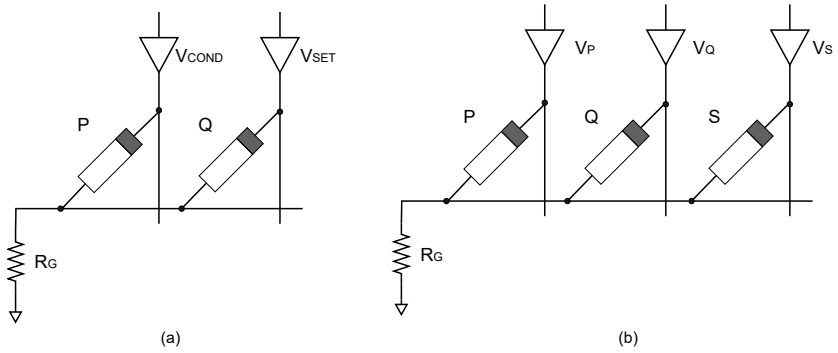


Figure 2.10: IMPLY logic structure [7]. (a) Basic structure IMPLY logic gate. (b) Schematic of IMPLY-based NAND gate. In NAND gate, two memristors (P and Q) act as inputs in the initial stage, one memristor S acts as the output in the last stage.

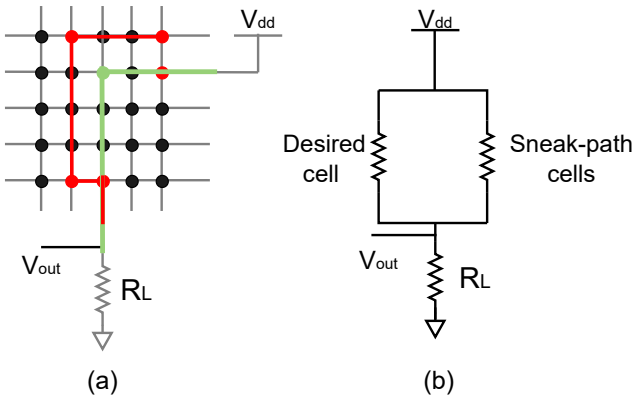


Figure 2.11: Demonstration of sneak-paths effect in crossbar architecture [8]. (a) Every node in the grid represents a memristor. The green line is the desired path, and red line is a sneak path. (b) Equivalent circuit model of sneak-paths effect.

- *MAGIC Logic Gate:* Another crossbar based memristive logic architecture is the MAGIC logic gate [9] as shown in Fig. 2.12. The inputs of the MAGIC gates are the initial resistances of input memristors In_1 and In_2 respectively; the output is the final resistance of memristor *Output*. Operation of this technique consists of two sequential stages. Firstly, the output memristor has to be initialised to a certain logic state which depends on the specific logic operation need to be realised. Then, V_0 is applied across the logic gate and the output resistance of memristor depends on the logical state of both input and output memristors. The advantage of the MAGIC technique is that all the basic boolean functions can be realised by this technique as shown in Fig. 2.13. However, this technique also suffers from sneak-path effect as the designs are based on the memristor crossbar array. The implementations also require that the memristors be initialised to a known state which could complicate the control circuit. All of these factors make it difficult to integrate with the existing CMOS technology.

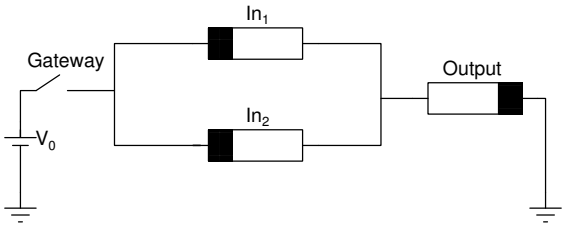


Figure 2.12: Basic structure of MAGIC logic NOR gate [9].

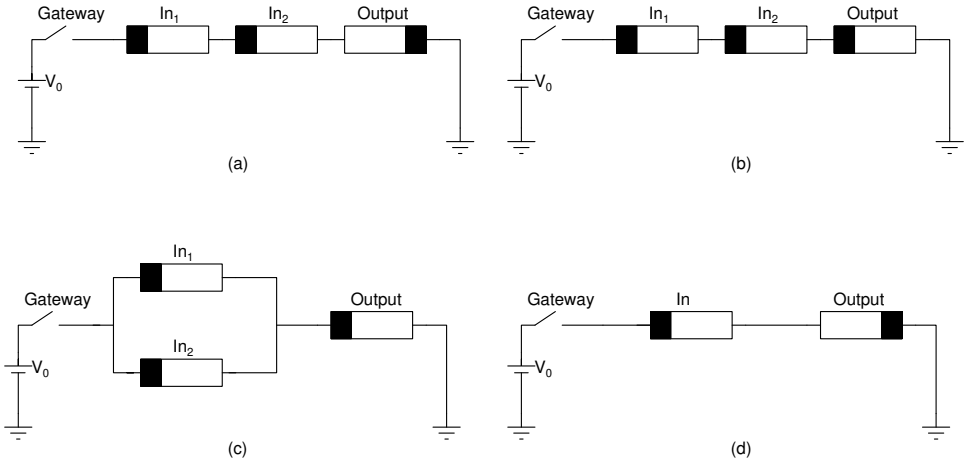


Figure 2.13: Basic structure of MAGIC logic gate [9]: (a)MAGIC NAND gate. (b)MAGIC AND gate. (c)MAGIC OR gate. (d)MAGIC NOT gate.

- *1M-4T XOR Gate:* The technique of [42] proposed a hybrid memristive XOR function block with one memristor and four transistors (1M-4T) as shown in Fig. 2.14. In this block, apart from the two normal inputs, A and B, as conventional logic gates have, a third input C is added as the control signal for controlling the read and write operations. The resistance of the memristor is the output of this block. Table 2.4 describes the logic operation of the XOR functionality, where HRS (R_{off}) and LRS (R_{on}) represent the high resistance state and the low resistance state respectively. Again, this technique requires multiple sequential steps to operate the logic computation.

According to the previous discussions, the advantages and challenges of the existing

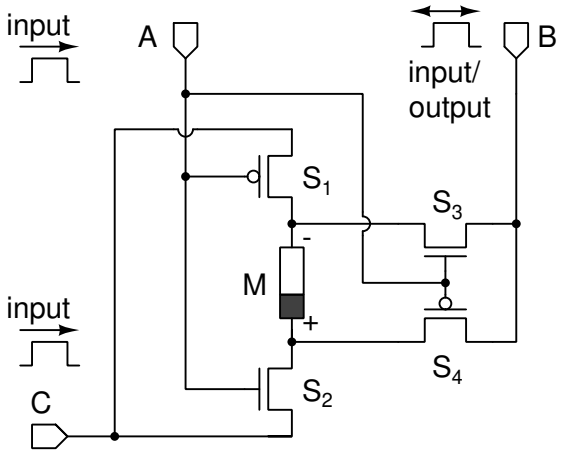


Figure 2.14: Hybrid 1M-4T logic XOR gate [42].

Table 2.4: The logic operation for 1M-4T XOR gate.

A	**B**	**State of S_1, S_2, S_3 and S_4**	**State of M**
0	0	S_1 and S_4 are closed; S_2 and S_3 are opened.	HRS $\Longrightarrow$ 0
0	1	S_1 and S_4 are closed; S_2 and S_3 are opened.	LRS $\Longrightarrow$ 1
1	0	S_1 and S_4 are opened; S_2 and S_3 are closed.	LRS $\Longrightarrow$ 1
1	1	S_1 and S_4 are opened; S_2 and S_3 are closed.	HRS $\Longrightarrow$ 0

stateful-logic techniques are summarised as follows:

- *Advantages:* The memory and the processing units are usually the separated blocks in the conventional computation systems (e.g. von-Neumann architecture [54]). These separating structures limit the speed of data transmission between different blocks. However, the stateful architecture provides an opportunity to enable highly efficient computation due to the combined non-volatile memory and logic operations. It can massively save time and energy from transferring the data between the different blocks. Compared with contemporary silicon technology, [55] shows the circuit based on stateful architecture can improve the speed by 76.8% and the power dissipation by 60.3%. Additionally, the circuit also demonstrates 700 times reduction in the area consumption.

- *Challenges:* As we have seen the previous techniques, all of them require the read/write operations which complex the control circuitry. Additionally, if the architecture is based on the memristor crossbar, the sneak-path issue will increase the challenge for the read operation. Finally, most of the techniques (IMPLY, MAGIC and 1M-4T etc.), require multiple clock cycles to finish a simple logic operation.

2.3.1.2 Memristive Non-stateful Logic Architectures

In this section, we introduce some existing memristive non-stateful logic designs. Here, the memristors serve as switches like the conventional CMOS-based logic designs. The logic states are presented by using voltage instead of resistance.

- *Memristor Ratioed Logic (MRL) Gate:* Unlike the previous techniques, Memristor Ratioed Logic (MRL) architecture [41] is based on the output voltage as the logic state. It uses the programmable resistance of a memristive device to ensure that the circuit operates like a voltage divider for realising the Boolean OR and AND functions as shown in Fig. 2.15(a) and Fig. 2.15(b) respectively. In this diagram a and b are the two inputs and V_{x} and V_{y} are the OR operation and AND operations respectively. For the case where the inputs are different, *i.e.*, $a = 1$, $b = 0$, current flows from the high voltage to the low voltage, thus changing the state of M_1 and M_2. In Fig. 2.15(a), M_1 and M_2 change to R_{on} and R_{off} respectively. Hence, V_{x} is determined by:

$$V_{\mathrm{x}} = \frac{R_{\mathrm{off}}}{R_{\mathrm{off}} + R_{\mathrm{on}}} V_{dd} \approx V_{dd} \tag{2.19}$$

Similarly, in Fig. 2.15(b), M_1 and M_2 change to R_{off} and R_{on} respectively. Hence, V_{y} is determined by:

$$V_{\text{y}} = \frac{R_{\text{on}}}{R_{\text{off}} + R_{\text{on}}} V_{dd} \approx 0 \tag{2.20}$$

where, $R_{\text{off}} \gg R_{\text{on}}$. For the case where both inputs a and b are identical. There is no current flow within the circuit for both OR and AND gates. Hence, output voltage V_{x} or V_{y} is equal to the input voltage. This MRL technique is capable of operating the logic computation within a single clock cycle. However deriving the Boolean 'NOT' operation by the purely memristive device is challenging with this technique. To extend this to more logic gates, e.g. NAND/NOR, a CMOS inverter is applied as 'NOT' operation combined with the MRL OR/AND logic.

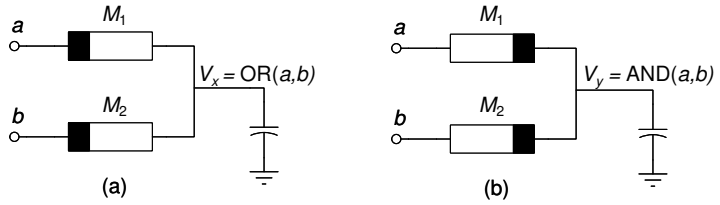


Figure 2.15: Memristor Ratioed Logic (MRL) architecture [41]. (a)MRL OR gate. (b)MRL AND gate.

The technique of [2] proposed a bufferless 4-transistor and 6-memristor (4T-6M) hybrid memristive XOR gate by using the MRL AND and OR gates and CMOS inverters based on the expression $A \oplus B = (A \wedge \overline{B}) \vee (\overline{A} \wedge B)$ directly. The area complexity of this technique is higher than pure CMOS XOR implementations [56], because it requires ten elements whereas the CMOS design requires six elements for bufferless design.

- *2T-2M XOR Gate:* Based on the MRL OR gate (Fig. 2.15(a)), the technique of [11] proposed a hybrid memristive XOR architecture as shown in Fig. 2.16.

It consists of two memristors (M_1 and M_2), and a pair of pull-down NMOS transistors. When both of the inputs A and B are equal to logic 1, the two transistors pull down the voltage appearing at the output to ground. This reduces the output to $\geq$ 0.2V, where as it should be $\approx$ 0V. This happens since the two NMOS transistors are connected in series, which results in their drain-to-source saturation voltage to be $V_{\text{DS}} + V_{\text{DS}} \geq 0.2\text{V}$ when both of the inputs to the XOR gate are at logic 1. This may be too high to switch off a device connected in the following stages, especially with smaller technology nodes. This technique requires 2T-6M and 2T-4M elements for fully buffered XOR and XNOR functionality respectively. Additionally, it inherently requires memristors with very high R_{on} and even higher R_{off} to keep the transistors properly biased. This has the effect of slowing down any design owing to very large RC time constants.

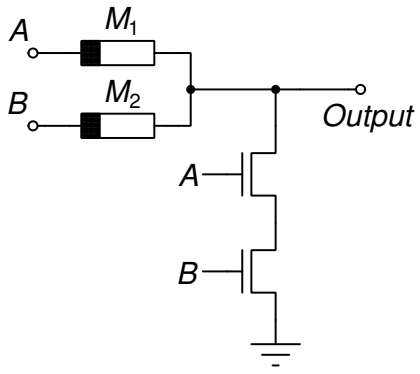


Figure 2.16: Bufferless 2T-2M XOR gate [39].

2.3.2 Physical Unclonable Functions

2.3.2.1 PUF Background

In the present era, electronic devices are prevalent in our daily lives, leading to high demand for implementing a reliable cryptographic system. Conventional crypto-system, based on the mathematical algorithm, is remaining the binary keys stored in memory elements secretly. Software security solutions have advantages such as low cost and easy implementation. However, these keys can be exposed by various hacking techniques and are also vulnerable to bugs. Hardware-based security solutions based on integrated circuit leverage cryptographic primitives to provide device authentication and confidential information protection. These solutions offer superior security, however are vulnerable to versatile cryptographic attacks such as Modelling attacks, Side-Channel attacks and Invasive attacks etc. [57, 58]. Most of these hardware security technologies require extra chip area, high power consumption, high manufacturing cost. To create a lightweight, low power and the low-cost system, as well as to maintain the high-security level is a challenge. Physical Unclonable Function (PUF) as an emerging technology was proposed recently [59, 60]. It uses the manufacturing variation of the physical device as sources of the security primitives [40, 61]. For example, uncontrollable random variation of the doping concentration and undesirable difference in the physical dimension of the device create the distinct properties which make the devices extremely difficult to re-fabricate, thereby providing physical uncloneability.

PUFs can be used in challenge-response authentication and key generation applications. The fundamental concept of the PUF is an implementation in which the mapping between a challenge and its response is dependent on the physical variation which is beyond the control of the manufacturer [58]. Fig. 2.17 shows the basic principle of the PUFs. The same stimulus called Challenge (C) is applied from *Chip-1*

to *Chip-n*, where each Chip represents a PUF instance. The output from each chip which is called the response (e.g. *R-1, R-2..R-n*) is unique due to the manufacturing variation. Therefore, the formed unique challenge-response pair (CRP) corresponds to a specific PUF device. Fig. 2.18 [10] shows the process of PUF authentication. All possible challenges are applied to device A to generate the corresponding responses. Then, these generated CRPs are stored by trusted party in a database. Finally, the trusted party chooses one of the challenges and applies to a device which need to be authenticated. If $Response' = Response$, the device is authentic.

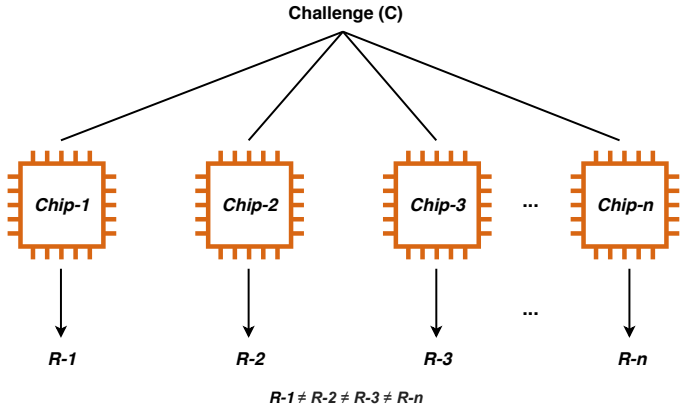


Figure 2.17: The basic principle of PUF.

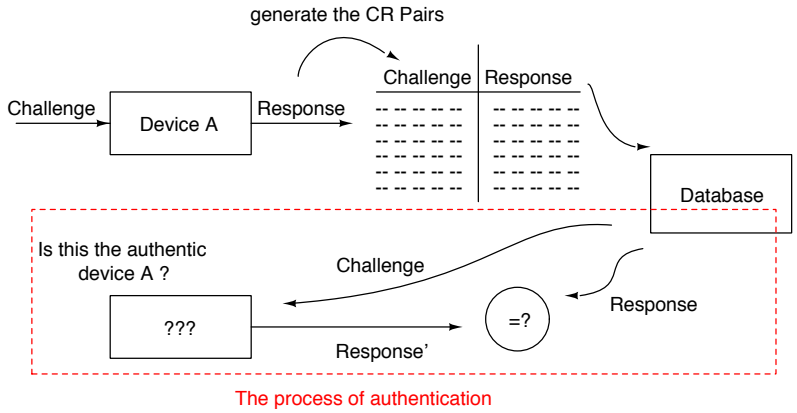


Figure 2.18: The process of PUF authentication [10].

2.3.2.2 PUF Performance Evaluation

In this thesis, the author applied a systematic method [62, 63] to evaluate the performance of PUFs. Hence, this section provides a brief review of this method. There are four parameters which need to be considered as shown in follows:

- **Uniqueness** specifies the ability of PUF to uniquely distinguish a particular chip among a total number of PUF instances of the same type. For example, two chips, i and j, have m-bit responses, R_i and R_j respectively for the same challenge C. Hence,

$$\begin{aligned} Uniqueness &= \frac{2!(k-2)!}{k!} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \\ &= \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \end{aligned} \tag{2.21}$$

  where, k is the total number of PUF instances; n represents the total number of bits in the response of a PUF instance; $HD(R_i, R_j)$ is the hamming distance between two bit-strings R_i and R_j. The ideal value of uniqueness is 50%, indicating that all PUF responses are distinguishable (e.g. half the response bits are expected to differ between any two PUFs). Otherwise, there is some probability that various PUFs generate identical (inverse) responses if uniqueness is biased towards 100% (zero).

- **Uniformity** is defined as the proportion of '1' and '0' in the responses bits of a PUF. This parameter ensures the randomness of the response. Assume $r_{i,l}$ is the l-th bit of a response string of the chip-i. n is the total number of bits in the response of a PUF instance. Hence, the uniformity of the chip-i is defined as follows:

$$Uniformity|_i = \frac{1}{n} \sum_{l=1}^{n} r_{i,l} \times 100\% \tag{2.22}$$

when the challenge C is kept constant. The ideal value of the proportion should be 50%.

- **Bit-aliasing** represents the affinity of a response bit towards either '1' or '0'. If bit-aliasing happens, different chips may produce similar response for the same position which is undesired effect. Assume $r_{i,l}$ is the l-th bit of a response string of the chip i. k is the total number of the PUF chips. Hence, the bit-aliasing across a group of chips can be evaluated as follows:

$$Bit - aliasing|_l = \frac{1}{k}\sum_{i=1}^{k} r_{i,l} \times 100\% \tag{2.23}$$

when the challenge C is kept constant. The ideal value of the bit-aliasing is also 50%. To calculate the average bit-aliasing value, we average the bit-aliasing values for all the bit positions.

- **Reliability** represents how efficient a PUF instance can reproduce the response bits under different operating condition e.g. different temperature and certain range of the voltage fluctuations. Ideally, the PUF should be able to produce exact the same response bits under the different operating condition. To estimate the reliability, the equation is formed as follows:

$$Reliability = 100\% - \frac{1}{m}\sum_{t=1}^{m} \frac{HD(R_i, R_{i,t}^{'})}{n} \times 100\% \tag{2.24}$$

where, R_i is the n response bits produced under nominal operating condition. $R_i^{'}$ is the n response bits extracted at a different operating condition. m indicates the total number of the samples which collected. Then we calculate the hamming distance HD between the R_i and response bits from other samples. $R_{i,t}^{'}$ in Eq. (2.24) represents the t-th sample of $R_i^{'}$. The ideal value of reliability is 100%.

2.3.2.3 Types of PUFs

PUFs can be classified into non-electronic PUFs and electronic PUFs. Electronic PUFs can be categorised into analogue-electronic PUFs, delay-based PUFs and memory-based PUFs as shown in Table 2.5. A brief overview of different types of PUFs is given below:

Table 2.5: The Classification of PUF.

Classification for a range of PUFs		
PUF	Non-electronic PUFs	Optical PUFs [64]
		Paper PUFs [65, 66]
		CD PUFs [67]
		.....
	Analogue-electronic PUFs	V_T PUFs [68]
		Power Distribution PUFs [69]
		Coating PUFs [70]
		LC PUFs [71]
	Traditional Memory-Based PUFs	SRAM PUFs [72, 73]
		Butterfly PUFs [74]
		Flash PUFs [75]
		DRAM PUFs [76]
	Delay-Based PUFs	Arbiter PUFs [77, 78]
		Ring-Oscillator PUFs [79]
		.....
	Emerging Memory Based PUFs	PCM PUFs [80–83]
		STTRAM PUFs [84–88]
		ReRAM PUFs [89–91]
		Memristor PUFs [92, 93]

1. **Non-electronic PUFs**

   - *Optical PUFs:* Optical PUFs is one of the earliest unclonable identification systems based on transparent media which proposed in paper [64]. It uses

different laser orientation as the challenge to reflect the corresponding speckle pattern through an optical token. However, the orientation of the laser beam must be precise to guarantee the same speckle pattern established. Hence, the whole mechanical procedure is very tedious and difficult to implement.

- *Paper PUFs:* Another early PUF is called paper PUFs [65, 66], which mainly use for an anti-counterfeiting strategy for currency notes. It uses the irregular fibre structure of the paper as a fingerprint of the specific document to prevent fraud. However, digital currencies have gradually replaced traditional currencies due to the increasing demands of the Internert of Things (IoT). By not having physical banknotes, the virtual currencies allow for instantaneous transactions, and also effectively provide counterfeit prevention.
- *CD PUFs:* CD PUFs was introduced in [67]. This proposed technique uses the deviation of the length of lands and pits during the manufacturing procedure to extract unique fingerprints from CDs. This deviation generated by manufacturing variability is large enough to be observed by the photodetector.

These non-electronic PUFs have difficulties in terms of fabrication cost and readiness to be integrated to computing devices.

2. **Analogue-electronic PUFs**

- *V_T PUFs:* The first technique to assign the identification to the integrated circuit was proposed in [68] and it was named Integrated Circuit Identification (ICID) or V_T PUF. ICID constructs on an array of addressable MOSFETs, with sequentially selected drains, driving a resistive load. Because of the mismatch variations of the threshold voltage, V_T, the current

through the drains will be different. Hence, the voltage across the resistive load will be random. Then, this voltage is measured and converted to a bit string with an auto-zeroing comparator. However, this method has limited CRPs (IDs). Hence, hackers can readout the ID and clone the chip.

- *Power Distribution PUFs:* Another application for IC-based security authentication is called "Power Distribution PUF". [69] proposed a PUF that leverages the inherent resistance variations during the manufacturing process in the metal layers defining the power grid of the chip. However, this technique requires massive Stimulus/Measure Circuits (SMCs) to sense the voltage drop. Additionally, external instruments are also required to measure the voltage.

- *Coating PUFs:* In cryptography, the black-box method is no longer sufficient since the invasive attacks can access many physical parameters from the black-box to derive the secret information. To resist invasive attacks, paper [70] proposed a "coating PUF". As the name implies, this PUF has a protective coating that sprays on top of an IC. The coating is opaque and consists of a matrix of aluminophosphate with a mixture of TiO2 and TiN particles. These particles are random in terms of size, shape and location. The top metal layer of the IC also accommodates an array of aluminium sensor as the challenge that can evaluate the local capacitance of the coating and generate a corresponding response. Each response is protected against invasive attacks from outside by the coating on the top of the IC. However, this technique requires extra manufacturing steps e.g. coating, which is highly specialised processing step. Hence, this increases the fabrication costs and difficult to implement with computing or communication devices.

- *LC PUFs:* The capacitor also can be utilized as a unique identifier and can be placed in a radio frequency (RF) electromagnetic field. In paper [71], it uses a small glass plate sandwiched between two metal plates, connected with a metal coil, emulating a capacitor and an inductor. This LC circuit will absorb an amount of power that not only depends on frequency but also accurate properties of the physical devices. Hence, a frequency sweep appears the peak of resonance of the circuit that will be slightly mismatched due to manufacturing variations.

3. **The Challenges of Non-electronic PUFs and Analogue-electronic PUFs**

   A number of non-electronic PUFs and analogue-electronic PUFs were discussed previously. Some earlier discussed techniques require external instruments to measure the responses (e.g. paper PUF, CD PUF, Power distribution PUF etc.). Others, for example, optical PUF and coating PUF need additional manufacturing processes or some specific components to support their implementation. To overcome these challenges, a number of PUFs, which are able to integrate on digital IC have been proposed. Some of the research papers define them as intrinsic PUFs. The biggest advantage of these PUFs is that their outputs (responses) can directly be used by other applications running on the same device since the PUFs are integrated on the same chip. The following parts presents two types intrinsic PUFs, which are based on memory and delay.

4. **Intrinsic PUFs: Memory-Based**

   Numbers of conventional intrinsic PUFs can be classified according to their characteristics of the circuits (e.g. memory cells or propagation delay for each gate). In this section, we mainly focus on some PUFs based on digital memory primitives. Since a huge number of different types of memory cells can contrive

PUFs such as DRAM [76], Flash [75] and SRAM, etc. Therefore, we choose the most representative memory-based PUFs (SRAM PUF and Butterly PUF) to show as follows:

- *SRAM PUFs:* SRAM Static Random Access Memory (SRAM) was considered as a PUF in [72, 73]. A single SRAM cell as shown in Fig. 2.19 (a) consists of 6 transistors. M_1 to M_4 are used to implement two cross-coupled inverters and M_5 and M_6 are used for read/write operations. The technique uses the mismatch variation of the threshold voltage, V_{th}, caused by two cross-coupled inverters (Fig. 2.19 (b)) to create an unstable state. In this case, the physical mismatch between two inverters remains as small as possible. The outputs depend on the preferences of the memory cells. Some cells store logic one and others store logic zero. This process is completely random when the supply voltage comes up. Hence the final output state is determined by the power-up behaviour or sometimes determined by stochastic noise. The advantage of this PUF is that there is no need to quantise the output since it already produces a binary number. However, the only one-bit response generated makes the SRAM PUF extremely weak. Moreover, SRAM PUFs are also sensitive to the noise. The erroneous output could occur because of the ambient temperature and fluctuations of the supply voltage.
- *Butterfly PUFs:* [74] presents another memory-based PUF called butterfly PUF as shown in Fig. 2.20 which behaves similarly to the SRAM PUF. It consists of cross-coupled latches. The output Q of each latch turns to 1 when a preset (PRE) signal is on high or it turns to 0 when a clear (CLR) signal is on high. The data D is passed to Q as CLK is high. In this Butterfly PUF, the PRE of Latch 1 and CLR of latch 2 are constantly set to low (0). The excite signal is connected to CLR of Latch 1 and PRE of

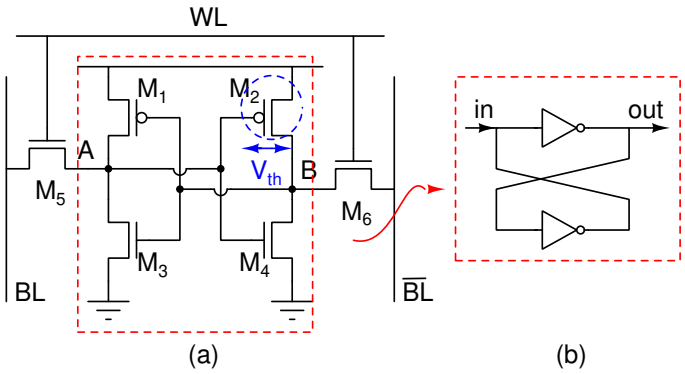


Figure 2.19: (a) SRAM PUF [73]; (b) Cross-coupled inverter [74].

Latch 2. The process of PUF starts with a high excitation signal which brings the circuit to a unstable state (*metastability*). The excite signal is set to low after a few clock cycles. This causes the PUF circuit to achieve a preferred stable state, 0 or 1. The preferred stable state of the circuit is determined by the physical mismatch of the internal connection. Again, this technique does not need quantisation step, but only 1 CRP generated makes PUF almost impossible to prevent opponents from acquiring information inside the device.

5. **Intrinsic PUFs: Delay-Based**

In this section, we introduce another type of intrinsic PUF which exploits the random variation in propagation delays of gates on silicon. The Arbiter and Ring Oscillator PUFs are the most representative techniques which are implemented based on this principle [77–79].

- *Arbiter PUFs:* Fig. 2.21 indicates the fundamental structure of arbiter PUF. The circuit consists of switching components (multiplexers) and an arbiter component (e.g. latch, D flip-flop and XOR gate, etc.). The

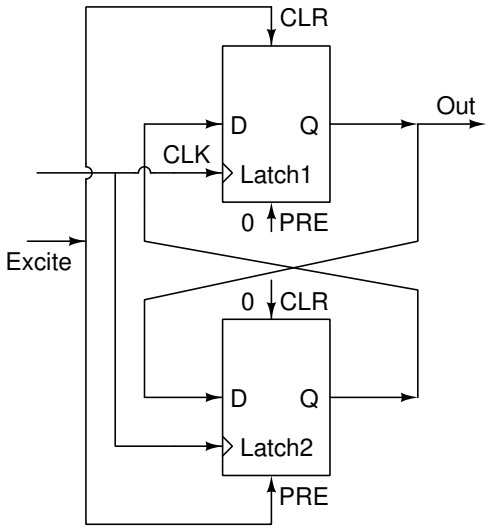


Figure 2.20: The butterfly PUF [74].

basic idea behind Fig. 2.21 is that the input signal is fed into the two symmetric delay paths simultaneously. The delay paths, in this case, are established by a chain of switching components and could be manipulated by the challenge bits C_0-C_n. Due to the manufacturing variability, the propagation delay of each switching element is slightly different. Hence, the rising edges of the signal placed in two delay paths are also different. The arbiter component determines the output response depending on which rising edge arrives faster as shown in Fig. 2.21. The switching components in Fig. 2.21 do not necessarily use MUX, but can also be replaced by XOR gates. In [11], they used their memristive XOR gates to implement this type of PUF. As we explained in Section 2.3.2.2, some environmental parameters (e.g. die temperature and supply voltage) affect the PUF responses. However, this was not a big effect for arbiter PUF since the differential measurement of the output is determined by considering two symmetric parallel delay paths simultaneously, thereby

providing good reliability.

However, one of the disadvantages for the arbiter PUFs is the linear characteristic which is vulnerable to machine learning based modelling attacks [11]. These attacks try to learn certain parameters of the PUF circuit. For example, the Support Vector Machine (SVM) technique which is commonly applied to the arbiter PUFs, tries to analyse the propagation delay of each stage (e.g. MUX, XOR gate). Since the output response depends on the total delay of the propagating signals which is simply a linear summation of the stage delays, the SVM is extremely successful for an delay based arbiter PUF. In [94–98], with a certain amount of observed CRPs, machine learning can successfully predict the behaviour of arbiter PUFs. [77] proposed a feed-forward arbiter PUFs where some challenges are fed by the output of the intermediate stage. This modified arbiter PUF can resist the simple modelling attacks due to non-linear behaviour but not for the all advanced modelling techniques [94, 98].

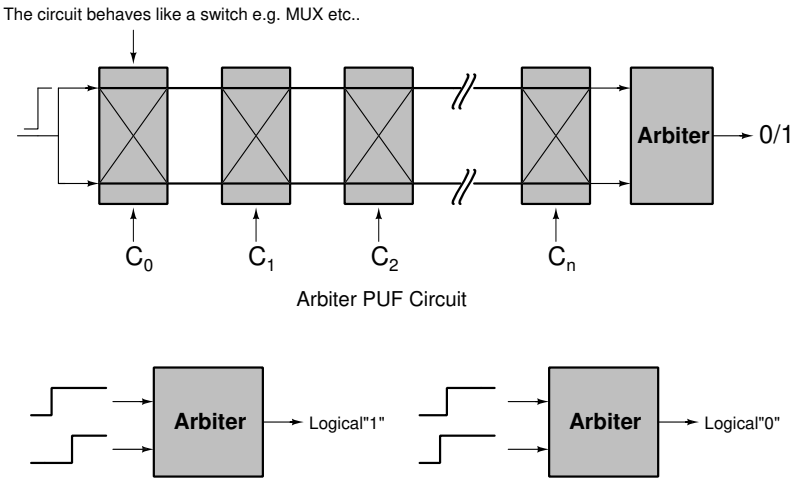


Figure 2.21: The basic circuit of delay-based arbiter PUF [78].

- *Ring-Oscillator PUFs:* Another delay-based PUF called Ring Oscillator

PUF as proposed in [79]. It consists of the number of ring oscillators with slightly different frequencies due to the variation of the delay line. The output is obtained by measuring and comparing the frequencies of each possible pair of oscillators as shown in Fig. 2.22. However, this technique requires at least 8 pairs of oscillators to obtain the most stable response bit, which relatively increases the implementation cost. The small probability of errors can be corrected using an error correcting code. Different error-correcting techniques have been summarised in [97]. Again, Ring-Oscillator based PUFs are also vulnerable to modelling attacks. [99] shows that the logistic regression (LR) algorithm is still able to attain 90% accuracy.

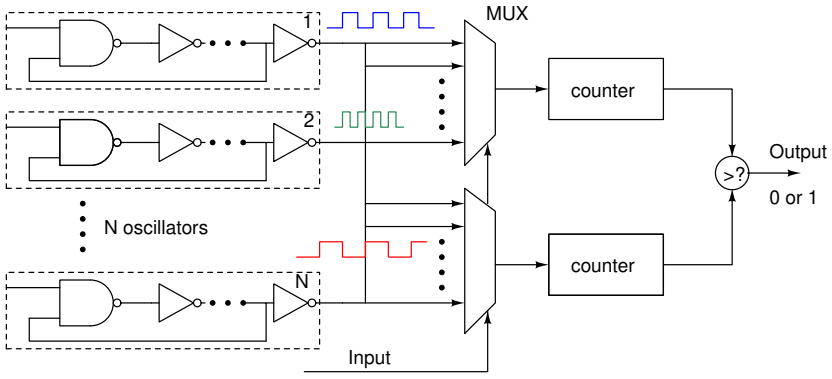


Figure 2.22: Ring Oscillator-based PUF circuit. [79]

6. Emerging Memristive Devices * Based PUFs

*In 2011, Leon Chua argued that all of the resistance switching memories including ReRAM can be considered as memristors [100]. However, [101, 102] provide the opposite argument by showing an experimental proof. Whether the resistive switching memories (ReRAM, STTRAM, PCM) can be included in the memristors is still a subject of debate. Hence, in this thesis, the author use the term 'memristive device' to refer to the resistance switching memories.

Most of the conventional hardware security solutions are based on CMOS devices. In recent years, the diversity of materials and devices are being explored for use in nanoscale electronics such as spin-transfer torque random access memory (STTRAM) [25], phase-change memory (PCM) [24, 103], memristors [3, 26], and resistive random access memory (ReRAM) [23, 104], etc. Because of their minuscule dimensions, these devices usually consume ultra-low power and operate relatively fast compared with CMOS devices. This part gives brief explanation of nano-electronic devices and describes how their properties can be utilized in the security primitives like physical unclonable functions [40, 61].

- *PCM-based PUFs:* Phase Change Memory (PCM) [24, 103] as shown in Fig. 2.23(a) use a phase change material to switch the device between the crystalline phase (low resistance state) to the amorphous phase (high resistance state). On applying a short duration of high current pulse to a heating element (heater) generally made of titanium nitride, the PCM set into the amorphous phase. Hence, the device turns into a high resistance state. However, to set the PCM into the crystalline phase, a longer current pulse with relatively less amplitude is applied to the heater. During both the reset and set cycles, the variability can occur and comes up with partial set/reset operations. Besides, the initial resistance of the PCM not only depends on the previous state of the cell but also varies because of the process variations and local material variations. Hence, PCM cells have difficulty to leave at the programmed resistance levels and the deviation of the resistance will accumulate in following operating cycles. These unpredictable partial operations and other manufacturing variations are the downside for memory design but can leverage the PCM in PUF applications or true random number generators. One of the earliest PCM-based PUFs was proposed as a concept in [80]. They divided

the resistance of the PCM cell into several logical intervals. Each of them again can separate into many subintervals, which can obtain a multiple bit response. Since the variability can occur during the programming from cycle to cycle, a random state can be extracted from a PCM cell. Apart from this, there are other techniques such as [81, 82] and [83], which also discuss the PUFs based on PCM.

- *STTRAM-based PUFs:* A spin-transfer torque RAM is another emerging memory, which consists of a magnetic tunnel junction (MTJ) as a barrier. The barrier lies between the two magnetic layers (a fixed layer and a free layer), and the whole structure is sandwiched between the two electrodes as shown in Fig. 2.23(b) [25, 40]. The resistance states of SST-RAM depends on the relative magnetization orientation of the two magnetic layers. If the current (spin-polarized current) directly passes through the free layer, the orientation of its magnet alters to be parallel or anti-parallel with the fixed layer, thereby providing low resistance and high resistance state respectively. During the state switches, each device shows the different programming currents due to the manufacturing variations. Hence, such switching characteristics can leverage PUF applications as proposed in [84]. This paper investigated the spin-transfer-switching characteristics and proposed a method to extract the PUF signature. In [85], they used variations in high resistance state of the STT-RAM cell to extract a PUF response by comparing the measured current with a reference current. The reference current indicates an average current, coming from many specific selected STT-RAM cells. If the measured current is higher than the reference current, the logic "1" is given as a response, otherwise, logic "0".

- *ReRAM-based PUFs:* Fig. 2.23(c) demonstrates the structure of the re-

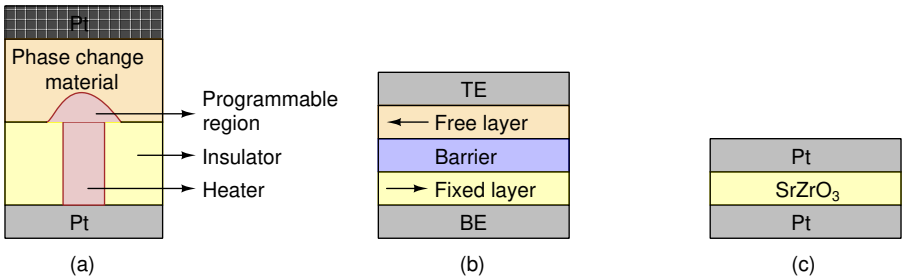


Figure 2.23: Structure of emerging memory devices: (a) Phase change memory (PCM) [24]. (b) Spin-transfer torque random access memory (STTRAM) [25]. TE and BE represent two electrodes. (c) Resistive random access memory (ReRAM) [23].

sistive random access memory [23, 104] (ReRAM) which consists of the resistance switch material e.g. ternary oxide $SrZrO_3$, sandwiched between the two electrodes. After applying a sufficient voltage, the metal-oxide-metal (MOM) structure is broken down and many current paths or conductive filaments are formed by oxygen vacancies between two metal layers. Hence, the device changes from the high resistance state to the low resistance state (set). By applying a voltage with the same polarity, the ReRAM switches back to the high resistance state (reset). The dimension of the paths or filaments varies from device to device due to the formation and rupture of the oxygen vacancies. Hence, in this case, the variability of ReRAM is not only in the manufacturing process but also in its physical switching mechanism. [89] observed that the current of high resistance state is mainly dominated by the tunnelling mechanism, which means a tiny variation of the tunnelling distance results in a significant difference in the resistance. This wide resistance distribution in the high resistance state provides sufficient randomness for PUF applications. [89] implemented the PUF in a fabricated 1kb (128×8) 1-transistor-1-resistor (1T1R) array. Here, a forming pulse is performed in the array to ensure the low resistance state across cells. Then, the same pulse is applied to

flip the resistance state of the cells, which causes random variation as the signature of the ReRAM PUF. [90] also implemented the stochastic nature of the ReRAM physical switching mechanisms to a 64Kbit array of 1T1R ReRAM cells. They found that the sensitivity of the reset probability can achieve 50% by increasing the drain voltage or the gate voltage of the transistor to a certain point. Therefore, this voltage-dependent switching probability can be applied as entropy for the PUF application. In [90], all the cells are set to the low resistance state. Then, by biasing the transistor properly, each cell starts to flip the state to the high resistance with the 50% reset probability randomly distributed in the array. Both techniques, as proposed in [89, 90], used the variability of the high resistance state as a source of the randomness. However, the bit flipping issue caused by fluctuation of the read voltage and temperature is still not addressed. [91] eliminated bit flipping issue because of sufficient read margin by proposing a new ID-generation algorithm. The cells initially stay at the high resistance state. By applying the forming voltage, some cells successfully switch to the low resistance state due to the formation of conductive filaments. Others which have not formed properly stay at the initial state because of the random nature of the switching characteristics. The paper claimed that the two resistance states which come from initial high resistance cells and low resistance cells have stronger reliability. 50% forming rate can be achieved by adjusting the forming voltage.

- *Memristor-based PUFs:* Memristor can also be utilised for PUF applications. The physical structure and the characteristics of the memristor are described in the previous section. [92] exploited the effects of variations in the width of a memristor as the inherent random source to create a PUF, which named NanoPUF. The basic structure of this NanoPUF is indicated

in Fig. 2.24. It consists of a memristor crossbar array, a challenge circuit and a response circuit. The row decoder applies a write pulse of magnitude V_{dd} to the particular row and the column decoder selects the specific column and connect it to the load resistor R_L. All other unselected rows and columns remain floating. In the response circuit, the comparator compares the output current from the selected memristor with the reference current. The reference current is the minimum current which can switch the memristor state from high to low. For the PUF operation, the memristors were first set to the high resistance state; then, a write pulse of a fixed duration which can provide the reference current is applied to a selected memristor. Due to the manufacturing variation of the physical dimension, some memristors might need a longer pulse to switch successfully, thereby remaining in high resistance state (logic "1"). If the actual duration of the write pulse is greater than the reference, the memristor will switch to the low resistance state (logic "0"). Another technique for memristor-based PUF was proposed in [93]. It takes both duration and amplitude of the write pulse into consideration to utilize a weak-write mechanism. Hence, unpredictable cell behaviour influenced by process variation can be obtained and usable as a PUF response.

According to the previous discussions, the advantages and challenges of the emerging memristive device-based PUFs are summarised as follows:

- *Advantages:* The emerging memristive devices e.g. PCM, ReRAM and memristors, etc. possess inherent unique properties that can be leveraged for PUF. i) Nonlinearity: The I-V characteristics of memristive devices are highly nonlinear. ii) Process variations: The resistance of the memristive device is affected nonlinearly by changes in its dimensions and dopant

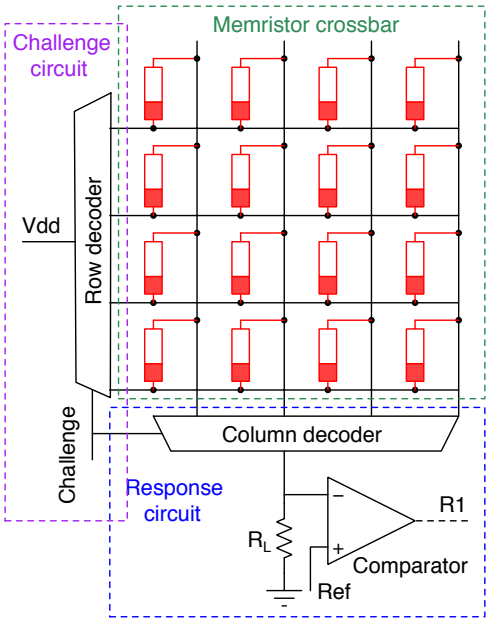


Figure 2.24: The structure of the memristor-based PUF [92].

concentration caused by process variations. iii) Radiation-hardness: Some of the memristive devices are radiation-hard owing to their material properties compared to the CMOS devices.

Another advantage is the low area consumption. [40] reports that NanoPUFs generate more response bits for the same amount of area as memristive devices are denser than conventional CMOS devices e.g. SRAM PUFs. Additionally, these devices themselves also consume less power.

- *Challenges:* Most of the memristive device-based techniques are implement on the crossbar array, which suffer the sneak-path effect. Hence, in order to operate PUF, these techniques usually require more complex control circuits to reduce the effect, that consume more power than the PUF circuit itself. Another challenge of these PUFs is the entropy. Ideally, the number of the response should be exponential to the number of elements

in the circuit. However, in most of the emerging memristive PUF designs, the number of response bits is linear to the number of devices.

2.4 Summary

This chapter gives an overview of the memristor in Section 2.2.1, which includes the basic concept, memristor-based applications and different memristor models. For a better understanding of the contributions of this thesis, Section 2.3.1 and Section 2.3.2 review the existing works related to memristive logic architecture and different types of physical unclonable functions respectively. These reviewed works demonstrates their advantages and disadvantages. The proposed memristive architectures targeted to address these issues are presented in upcoming chapters.

3

Memristive Single-cycle Multifunction Logic Architecture

3.1 Introduction

Most of the techniques for designing logic circuit with the memristor require multiple sequential steps (clock cycles) and complex control logic to realise even the simplest logic function as described in Chapter 2. While there are some existing work on single cycle operation, these techniques fail to operate with realistic resistance values and also require power comparable to conventional CMOS designs. In this chapter, firstly we propose a purely memristive logic architecture, consisting of only 4 memristors, for realising the XOR and inversion functions. Then we extend this architecture with just one transistor for seamless integration with the CMOS technology, thereby resulting in a hybrid 1T-4M architecture with dual XOR/AND and XNOR/OR functionality.

Based on the technological advances thus far, some of the materials considered for fabrication of memristors limits their operation to relatively lower frequencies. However, recent technological advances are seeing memristor operating at much higher frequencies [33]. This chapter also explores this aspect and proposes parametric selection in their compact modelling to enable the devices to work at higher frequencies.

As we know, the CMOS technology suffers from significant parasitic capacitance, which reduces its reliability at high frequencies [105, 106]. To circumvent this problem, one of our design objectives is to reduce the total number of transistors in the CMOS layer of our hybrid designs. This not only helps to enhance reliability at higher frequencies, but it also frees up chip area in the CMOS layer, which can only grow along the XY-axis, where additional CMOS exclusive functionality can be incorporated.

One of the key difficulties with traditional CMOS technology is its lack of ability for realising multiple valued logic (MVL). MVL has many applications, e.g. for memory and field programmable array designs, redundant number systems, etc. MVL can also help reduce interconnections on and off chip [107]. We show that certain configurations of memristors allow very efficient and simple realisation of MVL as

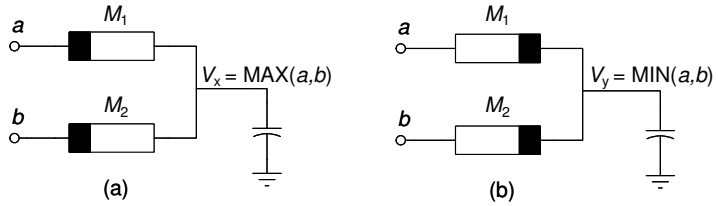


Figure 3.1: Memristive MIN-MAX (AND-OR) functionality; (a) OR/MAX operation, (b) AND/MIN operation.

MIN-MAX post algebra.

This Chapter is organised as follows. We show that memristors have inherent properties for realising operations over MIN-MAX post algebra in Section 3.2. In this section, we also propose a one transistor and four memristors (1T-4M) multifunction logic architecture which can be seamlessly integrated with the CMOS technology. In Section 3.3, we analyse the effects of frequencies on the physical parameters of memristors, thereby propose parametric selections in the memristive models for high frequency operations. With the help of more complex systems, Section 3.4 presents the proposed architecture can result in highly compact, low power and reliable systems capable of operating at low as well as high frequencies. Section 3.5 presents the experimental results.

3.2 Proposed Memristive Logic Architecture

In this section, we propose a 1T-4M multifunction logic architecture which can be seamlessly integrated with the CMOS technology. First, for completeness, we show that the Memristor Ratio Logic (MRL) architecture naturally exhibits multiple valued MIN-MAX functions over post algebra.

3.2.1 MIN-MAX Functionality

We have the following regarding the MIN-MAX functionality of MRL.

Theorem 1 *The MRL in Fig. 3.1(a) realises the MAX operation and that in Fig. 3.1(b) realises the MIN operation in post algebra as defined below:*

$$MAX(V_a, V_b) = \begin{cases} V_a, & V_a \geq V_b; \\ V_b, & otherwise. \end{cases} \tag{3.1}$$

$$MIN(V_a, V_b) = \begin{cases} V_a, & V_a \leq V_b; \\ V_b, & otherwise. \end{cases} \tag{3.2}$$

where, V_a and V_b denote the voltage of the inputs a and b respectively.

Proof: We prove each case separately.

MAX Operation (Fig. 3.1(a)): If $V_a > V_b$, then the voltage appearing at the positive terminal of M_1 is higher than its negative terminal. Thus M_1 switches to R_{on}. For the state of M_2, the voltage appearing at the positive terminal is smaller than its negative terminal. Thus M_2 switches to R_{off}. Hence V_{x} follows the input a and approximately equals to $V_a - V_{\text{M1}}$, where V_{M1} is the voltage drop across M_1. If $V_a = V_b$, then M_1 and M_2 appear in parallel. Hence, the output voltage V_{x} approximately equals to V_a or V_b. In this case, no current flows between the inputs. If $V_a < V_b$, then M_1 switches to R_{off} and M_2 switches to R_{on}. Hence, the voltage V_{x} equals to $V_b - V_{\text{M2}}$, where V_{M2} is the voltage drop across M_2.

MIN Operation (Fig. 3.1(b)): If $V_a > V_b$, then the voltage appearing at the positive terminal of M_1 is smaller than its negative terminal. Hence, M_1 switches to R_{off} and M_2 switches to R_{on}. The voltage V_{y} in this case follows V_b. If $V_a = V_b$, then it is similar to the case in the MAX operation. M_1 and M_2 appear in parallel and, hence, the output voltage approximately equals to V_a or V_b. In this case, no current flows between the inputs. If $V_a < V_b$, then M_1 switches to the R_{on} state and M_2 switches to the R_{off} state. The voltage V_{y} follows V_a. [QED]

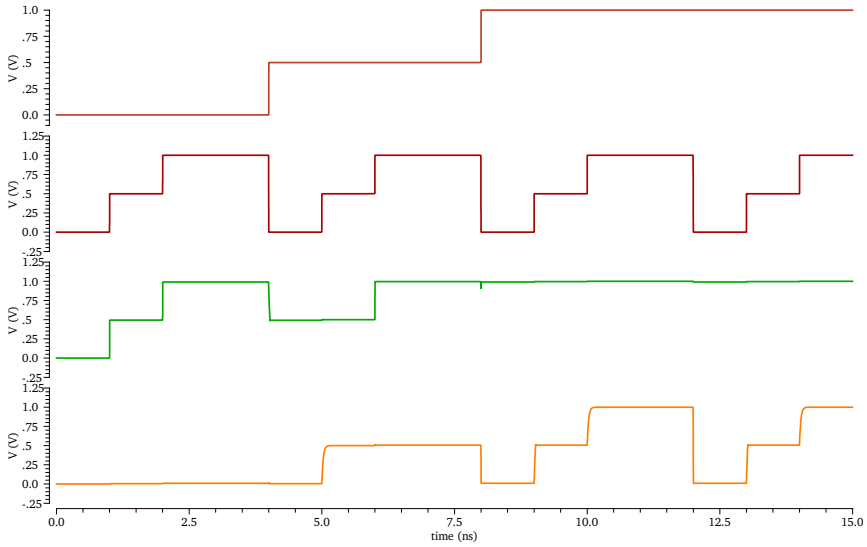


Figure 3.2: Input-Output Response of MIN-MAX Operation: Top two signals are the inputs a and b respectively, and the third signal is the output of the MAX operation. The bottom signal represents the MIN operation.

Theorem 1 reduces to the following for two valued logic.

Corollary 1.1 *The MRL in Fig. 3.1(a) becomes an OR gate and that in Fig. 3.1(b) becomes an AND gate for two valued logic.*

Proof: Follows trivially. [QED]

The input/output behaviour of MIN and MAX operations based on the architecture in Fig. 3.1 is presented in Fig. 3.2 for three valued logic. Here, we assumed 0V to be logic 0, 0.5V to be logic 1 and 1V to be logic 2. The memristor model used for MIN-MAX operation is based on VTEAM. The parameters of VTEAM are $V_{\text{on}} = -0.2\text{V}$, $V_{\text{off}} = 0.02\text{V}$, $R_{\text{off}} = 80k\Omega$, and $R_{\text{on}} = 500\Omega$. Based on Theorem 1 this circuit will operate for any reasonable values of logic.

3.2.2 Memristive XOR and Inversion Functionality

Most existing techniques for realising logic functions require multiple clock cycles to operate [7, 9, 42], and some require a hybrid of memristors and CMOS devices [2, 39].

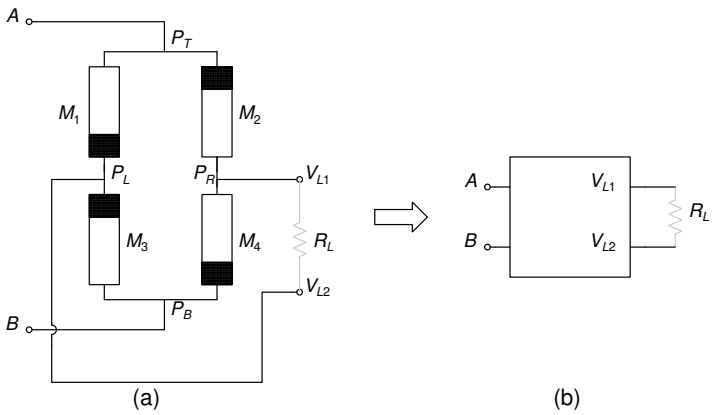


Figure 3.3: Memristive XOR functionality. (a) Purely memristive XOR functionality, (b) Symbol used in the rest of the paper.

Considering the merits and the demerits of these approaches, we propose a purely memristive XOR architecture as shown in Fig. 3.3(a). Fig. 3.3(b) shows the symbol for this architecture, which is used in the rest of the paper.

The purely memristive XOR architecture in Fig. 3.3(a) consists of four memristors M_1, M_2, M_3 and M_4. Here, M_1 and M_3 are connected for logical AND or MIN operation. M_2 and M_4 are connected for logical OR or MAX operation. The voltage difference between $V_{\rm L1}$ and $V_{\rm L2}$ behaves like XOR operation as shown in Table 3.1. In Table 3.1, the logical behaviour of the purely memristive XOR circuit is based on whether or not current flows from $V_{\rm L1}$ to $V_{\rm L2}$ through the load resistance R_L. In this table V_1 ($>$ the memristor threshold voltate $V_{\rm on}$) is assumed to be equivalent to $V_{\rm DD}$ in CMOS logic and represents the ON-state voltage, i.e. logic 1. Correctness of this architecture is summarised in Theorem 2.

Theorem 2 *The pure memristor circuit in Fig. 3.3(a) realises the XOR functionality depicted in Table 3.1.*

Proof: We consider each row in Table 3.1 separately.

Row-1: Follows trivially because no current flows in the circuit.

Table 3.1: Pure memristive XOR functionality.

Row	A	B	Output
1	0	0	$V_{L1} - V_{L2} = 0$V. $\implies$ No current flows from V_{L1} to V_{L2} $\implies$ Logic 0.
2	0	V_1	$V_{L1} - V_{L2} \approx V_1$V. $\implies$ Current flows from V_{L1} to V_{L2} $\implies$ Logic 1.
3	V_1	0	$V_{L1} - V_{L2} \approx V_1$V. $\implies$ Current flows from V_{L1} to V_{L2} $\implies$ Logic 1.
4	V_1	V_1	$V_{L1} - V_{L2} \approx 0$V. $\implies$ No current flows from V_{L1} to V_{L2} $\implies$ Logic 0.

Row-2: In this case, memristors $M_2 = M_3 = R_{\text{off}}$ because 0V appears at the positive terminal of M_2 through A, and V_1V appears at the negative terminal of M_3 through B. In contrast, memristors $M_4 = M_1 = R_{\text{on}}$ because their positive terminals are closer to V_1V and negative terminals closer to 0V. Because of voltage division the voltage at P_R rises towards V_1V, and that at P_L falls towards 0V. Thus, the current flows from $B \rightarrow P_B \rightarrow P_R \rightarrow V_{L1} \rightarrow V_{L2} \rightarrow P_L \rightarrow P_T \rightarrow A$. Hence, this is logic 1.

Row-3: This is similar to Row-2. Here, $M_1 = M_4 = R_{\text{off}}$, while $M_2 = M_3 = R_{\text{on}}$. Again the voltage at P_R rises towards V_1V, and that at P_L falls towards 0V. The current flows from $A \rightarrow P_T \rightarrow P_R \rightarrow V_{L1} \rightarrow V_{L2} \rightarrow P_L \rightarrow P_B \rightarrow B$. Hence, this is logic 1.

Row-4: In this case, both P_L (V_{L2}) and P_R (V_{L1}) are at the same voltage level and no current flows through R_L. Hence, this is logic 0. [QED]

These operations can take place in the same clock cycle as the inputs. Hence, the circuit in Fig. 3.3 exhibits XOR functionality in Table 3.1 in a single cycle. We note that $\forall A, B \in \{0, V_1\}$, the following are true for the circuit in Fig. 3.3.

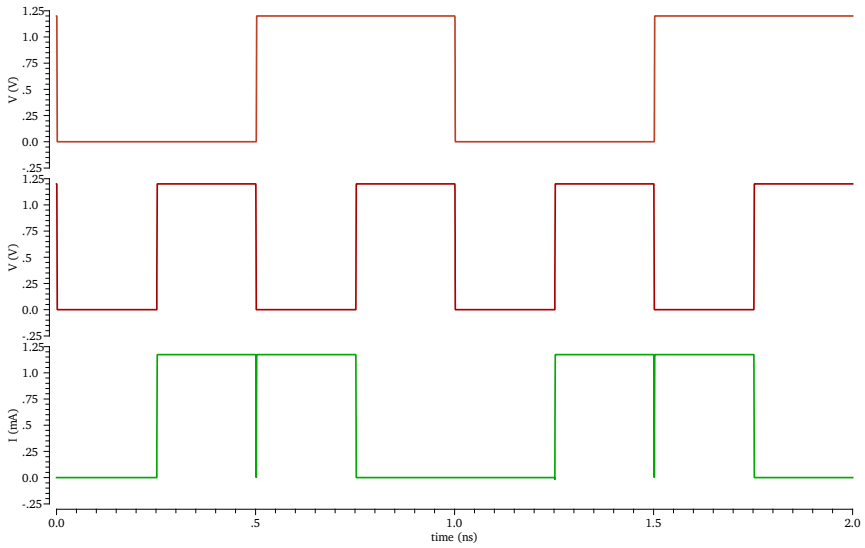


Figure 3.4: Purely memristive XOR functionality. The top two signals represent the two input voltages and the bottom signal is the current which flows into load R_{L}.

$$V_{L1} = A \vee B \tag{3.3}$$

$$V_{L2} = A \wedge B \tag{3.4}$$

$$V_{L1} \geq V_{L2} \tag{3.5}$$

Fig. 3.4 presents the output of the architecture in Fig. 3.3 for all the input combinations (i.e. 11, 10 01, 00). If we interpret the current flowing from V_{L1} to V_{L2} as logic 1, and the absence of this current as logic 0, then clearly this is XOR functionality.

Our proposed architecture trivially allows pure memristive inversion via XOR operation in a single cycle. For example, in Fig. 3.3(a), if we set $A = 1$ (i.e. $A = V_1$), then the output will correspond to the inverse of B, as shown by Rows 3 and 4 in Table 3.1, i.e. $\overline{B} = 1 \oplus B$. Hence, we have the following.

Corollary 2.1 *The pure memristive logic architecture in Fig. 3.3(a), which realises the XOR functionality depicted in Table 3.1, becomes an inverter when either of its*

inputs A or B is assigned logic 1 (V_1).

Proof: Follows trivially from Table 3.1. [QED]

3.2.3 1T-4M Hybrid CMOS-Memristive Logic Architecture

In Section 3.2.2, we proposed a novel pure memristive XOR logic architecture. Although this architecture works as an XOR gate, however, we interpreted the output logic value based on whether or not the current is flowing from V_{L1} to V_{L2}. Clearly, this is not directly compatible with the existing CMOS technology. Our aim is to integrate our system so that it works seamlessly with the existing CMOS technology. To this end, we present in this section a highly efficient way of integrating our pure memristive architecture with the CMOS technology by using only a single MOS transistor (MOST). This results in a 1-transistor 4-memristor (1T-4M) architecture as shown in Fig. 3.5(a) and (c). By keeping the number of MOST to only one, we are also able to maintain reliable performance at higher frequencies, when parasitic capacitance begins to degrade performance.

The logic architectures for our proposed 1T-4M XOR/AND and XNOR/OR dual functionality appears in Fig. 3.5(a) and Fig. 3.5(c) respectively. Compared with existing memristive logic techniques (e.g. IMPLY and MAGIC), the 1T-4M architecture seamlessly integrates with MOS technology without requiring extra control circuitry and the logic computation also finishes within one clock cycle.

First we have the following.

Lemma 1 *The NMOST in Fig. 3.5(a) and the PMOST in Fig. 3.5(c) realise the following logic operations respectively.*

$$V_{\mathrm{XOR}} = V_{L1} \wedge \overline{V_{L2}} \tag{3.6}$$

$$V_{\mathrm{XNOR}} = \overline{V_{L1}} \vee V_{L2}. \tag{3.7}$$

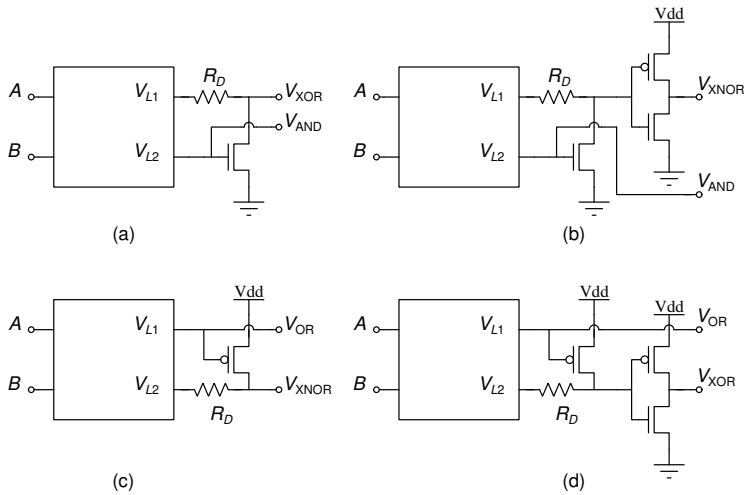


Figure 3.5: Memristive XOR and XNOR gates with dual functionality. (a) 1T-4M bufferless XOR/AND dual functionality, (b) 3T-4M fully buffered XNOR functionality, (c) 1T-4M bufferless XNOR/OR dual functionality and (d) 3T-4M fully buffered XOR functionality.

Proof: The proof follows by firstly noting Eq. (3.5). The only time $V_{\rm XOR}$ in Fig. 3.5(a) is at logic 1 ($\approx V_1$V) is when $V_{L1} \approx V_1$V and $V_{L2} \approx 0$V. From Table 3.1, clearly this happens when either $A = V_1$ and $B = 0$ or $A = 0$ and $B = V_1$. At all other times either $V_{\rm XOR} = 0$ ($V_{L1} = V_{L2} = 0$) or the NMOST goes into saturation and $V_{\rm XOR} \approx 0.1V$ ($V_{L1} = V_{L2} \approx V_1$V). Again, from Table 3.1 this only happens when either $A = 0$ and $B = 0$ (i.e. no current flows in the circuit, thus resulting in $V_{\rm XOR} = 0$) or $A = V_1$ and $B = V_1$ (i.e. this drives the NMOST into saturation).

Similarly, in Fig. 3.5(c) the only time $V_{\rm XNOR} \approx 0$V is when $V_{L1} \approx V_1$V and $V_{L2} \approx 0$V. At all other times $V_{\rm XNOR} \approx V_1$V. Hence the proof follows. [QED]

We now show that our proposed architecture realises XOR/AND and XNOR/OR dual functionality.

Theorem 3 *The 1T-4M multifunctional logic architecture in Fig. 3.5(a) realises the XOR/AND dual functionality and that in Fig. 3.5(c) realises the XNOR/OR dual functionality.*

Proof: To prove that the circuit in Fig. 3.5(a) realises the XOR operation, we

substitute V_{L1} and V_{L2} from Eq. (3.3) and Eq. (3.4) into Eq. (3.6) respectively, which yields

$$V_{\mathrm{XOR}} = V_{L1} \wedge \overline{V_{L2}} = (A \vee B) \wedge (\overline{A \wedge B}) = A \oplus B. \tag{3.8}$$

Now, regarding Fig. 3.5(c), Eq. (3.7) is merely the inverse of Eq. (3.6), i.e. $V_{\mathrm{XNOR}} = \overline{V_{\mathrm{XOR}}} = 1 \oplus A \oplus B$.

To show that the circuits also exhibit AND and OR operations while realising XOR and XNOR operations, firstly we note that according to Eq. (3.4) and Eq. (3.3), the circuits in Fig. 3.5(a) and (c) trivially realise the AND and OR operations at the gates of the NMOST and PMOST respectively. Since, neither gates of the NMOST and PMOST draw any current owing to gate isolation, therefore, this architecture realises the dual functionality of V_{AND} with V_{XOR} and V_{OR} with V_{XNOR} simultaneously as shown in Fig. 3.5(a) and (c) respectively.

Hence, the proof follows. [QED]

To ensure that the NMOST operates in saturation region (when A and B are different), the following design rule must be satisfied:

$$\begin{aligned} I_{D,sat} &= \frac{V_{\mathrm{RD}}}{R_{\mathrm{D}} + (R_{\mathrm{on}} || R_{\mathrm{on}})} = \frac{V_{\mathrm{DD}} - V_{DS,sat}}{R_{\mathrm{D}} + (R_{\mathrm{on}} || R_{\mathrm{on}})} \\ \implies R_{\mathrm{D}} + \frac{R_{\mathrm{on}}}{2} &= \frac{V_{\mathrm{DD}} - V_{DS,sat}}{I_{D,sat}} \end{aligned} \tag{3.9}$$

where, V_{RD} is the voltage across R_{D} and $I_{D,sat}$ is the NMOST saturation current. To ensure that the NMOST can also go into cut off region (when $A = B$), we have,

$$V_{\mathrm{DD}}(\frac{R_{\mathrm{on}}}{R_{\mathrm{on}} + R_{\mathrm{off}}}) < V_{\mathrm{TH}} \tag{3.10}$$

where, V_{TH} is the threshold voltage of the NMOST.

Similar design rule applies to the PMOST in Fig. 3.5(c).

Our 1T-4M architecture employs only one transistor, thereby ensuring that the

output is ≈ 0.1V when the transistor saturates, e.g. when both of the inputs are at logic 1. In contrast, the technique of [39] 2T-2M cannot ensure output voltage any less than 0.2V when both of its output transistors saturates. This larger voltage can be a critical factor, especially when driving very small technology nodes (Chapter 2).

We have tested with several memristor models [5, 6], and the designs worked correctly for a range of R_{on} and R_{off}. For low R_{on} a higher R_{D} maybe necessary to bias the transistor properly, while for higher R_{on}, R_{D} may be eliminated. We demonstrated the performance of Fig. 3.5(a) by choosing different values of R_{D} as shown in Fig. 3.6. It is clearly shown that the range of R_{D} $(15k\Omega - 24k\Omega)$ which is selected based on Eq. (3.9) and Eq. (3.10) provides the accurate performance, while with the smaller R_{D} e.g. $50\Omega - 800\Omega$, the transistor cannot be biased properly. In contrast to this, the technique of [39] (Fig. 2.16) can only operate with very high R_{on} and R_{off} (Chapter 2) thereby limiting its applications to certain types of memristors and to low frequencies only.

It should be noted that the 1T-4M XOR/AND architecture in Fig. 3.5(a) is more power efficient compared to the 1T-4M XNOR/OR architectures in Fig. 3.5(c). This is because the former architecture is not directly drawing power from the supply voltage (V_{DD}), whereas the latter architecture is.

Both of these circuits in Fig. 3.5(a) and (c) are weak, i.e. they may not guarantee a full voltage swing with sufficient current drive for a following stage. A two-transistor CMOS inverter buffer can be added at the outputs for both 1T-4M XOR and XNOR gates to obtain full-voltage swings as shown in figures (b) and (d). This yields 3T-4M 'strong' XOR/XNOR gate as compared to the 1T-4M bufferless XOR/XNOR gates. However, in line with the reasons stated in the previous paragraph, the 3T-4M fully buffered XNOR gate in figure (b), which draws power directly from the power supply only at the buffer stage, is more power efficient compared to the XOR gate in figure (d), which draws power in the pre-buffer stage also.

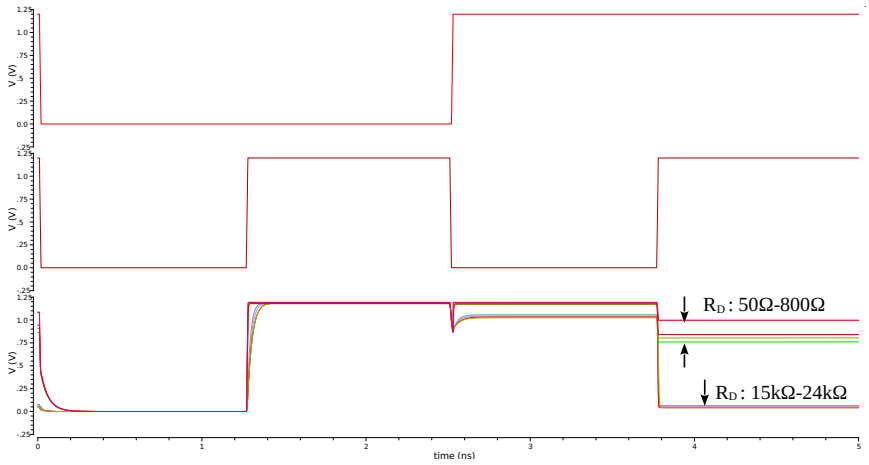


Figure 3.6: The performance of XOR architecture affected by a selection of the load resistor R_{D}. Here, $R_{\mathrm{off}} = 80k\Omega, R_{\mathrm{on}} = 500\Omega$ and the On Resistance, $R_{\mathrm{ds,on}}$, of NMOST is about $1k\Omega$. The top two signals represent the two input voltages and the bottom singals are the output V_{XOR} of Fig. 3.5(a) with different values of R_{D}.

3.3 High Frequency Operations

Memristors are usually considered to operate at lower frequencies owing to the limitations of the materials and the way they are used for their fabrication. However, low frequency ultra low power electronic device have critical applications, e.g. in medical electronics. To this end, the proposed technique is well suited for such applications owing to their low power requirements (Table 3.2 and Section 3.4). However, this is a highly evolving area and new materials are investigated for fabrication of more efficient and better performing memristors, especially at high frequencies. For example [33] presented a novel SiO_x based memristor with much better frequency characteristics, which also allows better integration with the CMOS technology. In line with these, we present the effects of high frequency on memristor models and the modification necessary for high frequency operations.

Various memristor models have been proposed in literature, e.g. [4–6, 48, 49]. However, not all of these models are suitable for logic design. The model proposed in [6],

which our models and designs are based on, is a flexible memristor model that uses voltage as a threshold parameter and it is also based on modified Simmons tunnel barrier model [49], owing to the fact that the latter model is applicable only to specific materials and hence cannot be directly generalised.

To analyse the effects of frequency on the performance of this model, we need to consider the derivative of the state variable w as follows [6]:

$$\frac{dw(t)}{dt} = \begin{cases} k_{\text{off}} \cdot (\frac{v(t)}{V_{\text{off}}} - 1)^{\alpha_{\text{off}}} \cdot f_{\text{off}}(w), & 0<V_{\text{off}}<\text{v} \\ 0, & V_{\text{on}}<\text{v}<V_{\text{off}} \\ k_{\text{on}} \cdot (\frac{v(t)}{V_{\text{on}}} - 1)^{\alpha_{\text{on}}} \cdot f_{\text{on}}(w), & \text{v}<V_{\text{on}}<0 \end{cases} \tag{3.11}$$

where k_{off} (a positive number), k_{on} (a negative number), α_{on} and α_{off} are constants; and, V_{off} and V_{on} are voltage thresholds. Here, k_{off} and k_{on} are in meters per second, whereas α_{on} and α_{off} do not have any unit [6].

The k and α parameters are fitting parameters which are directly related to the physical behviour of the materials used to fabricate the memristors, as shown in [6]. For example, for ferroelectric memristors, k_{off} and k_{on} are considered to be 10^{-4}m/s and -30m/s respectively and the parameter α_{off} and α_{on} are considered to be 5, etc.

The functions $f_{\text{off}}(w)$ and $f_{\text{on}}(w)$ behave like a window function which constrains the state variable w to bounds of the device as defined in [5]. This derivative of the state variable $\frac{dw(t)}{dt}$ indicates the rate of change of w which in this case describes how rapidly the memristor could switch between low and high resistance states. Based on memristor characteristics and Eq. (3.11), the width w is directly dependent on the parameter k_{off} (k_{on}). Here, apart from the other parameters, k_{off} and k_{on} mainly influence the rate of change of the magnitude of w. Fig. 3.7 demonstrates the variation of the state variable w for different applied values of k_{off}, where the memristor is driven by a 20MHz sinusoidal input with 1.2V amplitude. When k_{off} is increased, the slope of w becomes steeper, which results in the device switching rapidly to a high resistance

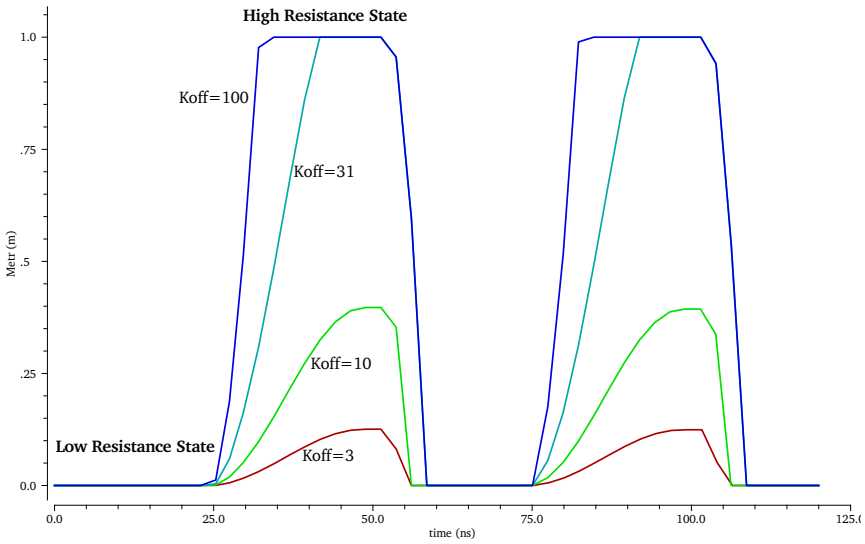


Figure 3.7: Dynamic behaviour of state variable w for a selection of the parameter k_{off}. A 20MHz 1.2V sinusoidal voltage is applied in this simulation. Here, $V_{\text{on}} = -0.2\text{V}$, $V_{\text{off}} = 0.02\text{V}$, $R_{\text{off}} = 80k\Omega, R_{\text{on}} = 500\Omega$ and $k_{\text{on}} = -200\text{m/s}$.

state.

Similarly, when $|k_{\text{on}}|$* is increased, the device quickly switches to a low resistance state as shown in Fig. 3.8. We also demonstrate this characteristic via applying more realistic stimulus such as square wave as shown in Fig. 3.9 and Fig. 3.10. The parameters k_{off} and k_{on} are fitting parameters which with correct values can be used to simulate the behaviour of a wide range of memristive devices with different materials. Hence, with the proper values of these parameters a memristor can also be modeled for operating at high frequencies, e.g. [33].

Based on these parametric selections, we compared the performance of the proposed architectures with other hybrid memristive logic designs, e.g. [39], at high frequencies. We used the same memristive model for all the designs for fairness. We also compared the performance of our designs with pure CMOS designs, e.g. in [56]. To this end, Table 3.2 presents the performance of our fully buffered 3T-4M XOR and XNOR gates

*The symbol $|x|$ represents the absolute value of a signed number x.

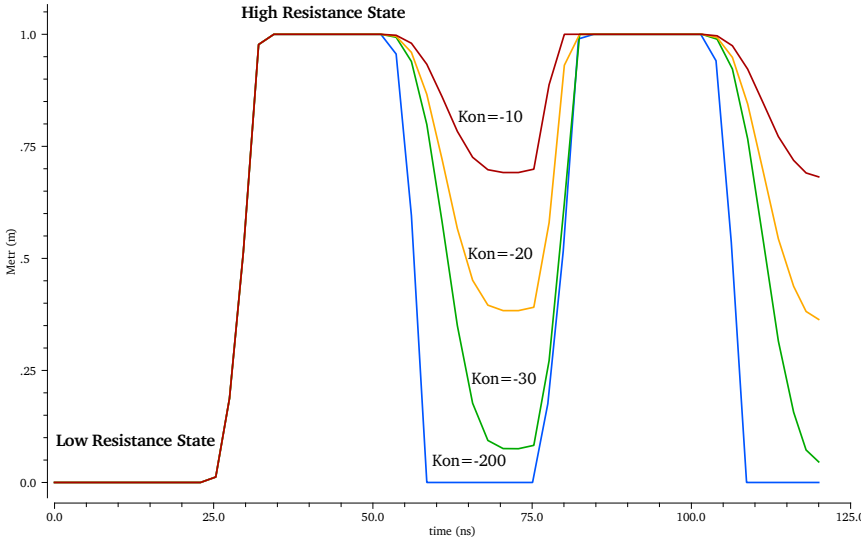


Figure 3.8: Dynamic behaviour of state variable w for a selection of the parameter k_{on}. A 20MHz 1.2V sinusoidal voltage is applied in this simulation. Here, $V_{\text{on}} = -0.2\text{V}$, $V_{\text{off}} = 0.02\text{V}$, $R_{\text{off}} = 80k\Omega, R_{\text{on}} = 500\Omega$ and $k_{\text{off}} = 100\text{m/s}$.

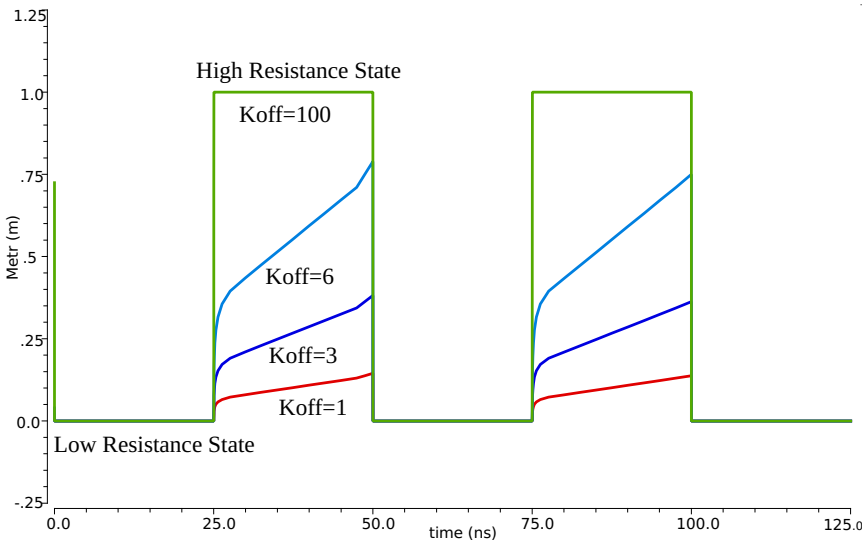


Figure 3.9: Dynamic behaviour of state variable w for a selection of the parameter k_{off}. A rectangular square wave of 20MHz with amplitude of 1.2V is applied in this simulation. Here, $V_{\text{on}} = -0.2\text{V}$, $V_{\text{off}} = 0.02\text{V}$, $R_{\text{off}} = 80k\Omega, R_{\text{on}} = 500\Omega$ and $k_{\text{on}} = -200\text{m/s}$.

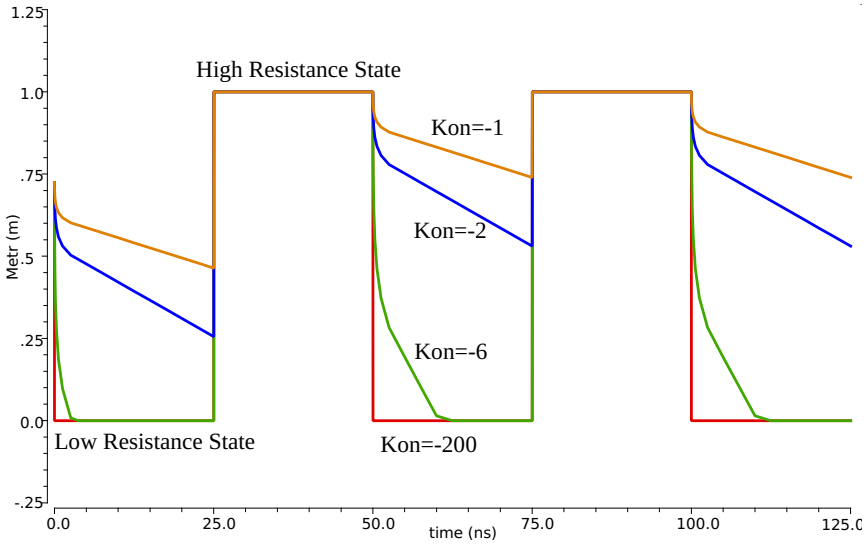


Figure 3.10: Dynamic behaviour of state variable w for a selection of the parameter k_{on}. A rectangular square wave of 20MHz with amplitude of 1.2V is applied in this simulation. Here, $V_{\text{on}} = -0.2\text{V}$, $V_{\text{off}} = 0.02\text{V}$, $R_{\text{off}} = 80k\Omega$, $R_{\text{on}} = 500\Omega$ and $k_{\text{off}} = 100\text{m/s}$.

at different frequencies as compared to (i) fully buffered 10T and 8T pure CMOS XOR (C-XOR) and XNOR (C-XNOR) gates [56], (ii) fully buffered AOI22 XOR and AOI22 XNOR gates [108], and (iii) fully buffered 6T-2M hybrid memristive XOR and 4T-2M XNOR gates proposed in [39].

Throughout the thesis, we measured the average power drawn by the circuits as follows. The average power P measured here is the energy E consumed by the entire circuit divided by the simulation time as shown below:

$$E = \int_0^T V_{\text{DD}} \cdot i(t)dt \tag{3.12}$$

$$P = \frac{E}{T}, \tag{3.13}$$

where V_{DD} is the supply voltage, $i(t)$ is the total instantaneous current drawn from V_{DD} by a circuit at time t. T is the simulation time.

The 10T/8T buffered CMOS XOR/XNOR gate constitutes a bufferless 6T stage, as

shown in Fig. 3.11 [56]. The 6T-2M/4T-2M buffered hybrid memristive XOR/XNOR design constitutes a 2T-2M bufferless design as shown in Fig. 2.16 [39]. For the proposed 3T-4M XOR gate, $R_{\mathrm{D}} = 24k\Omega$, $R_{\mathrm{on}} = 500\Omega$ and $R_{\mathrm{off}} = 40k\Omega$ are calculated based on Eq. (3.9) and Eq. (3.10). The same values of R_{on} and R_{off} are also considered for the design of the 6T-2M XOR gate [39]. The parameters $k_{\mathrm{on}} = -200\mathrm{m/s}$ and $k_{\mathrm{off}} = 50\mathrm{m/s}$ are selected based on the simulation results as shown in Fig. 3.7 and Fig. 3.8. The values of the k parameters which we considered in this paper are similar to those considered in [109]. As shown in [6], this appears to refer to Pt-Hf-Ti memristive devices [110].

Table 3.2 shows that our 3T-4M structure requires significantly less power than the 10T CMOS designs, while maintaining reliable performance even at high frequencies by comparing with the AOI22 designs.

For the 6T-2M hybrid structure, R_{on} has to be high enough to bias the two NMOSTs which are connected in series as shown in Fig. 2.16. Hence, in this case, this circuit required $R_{\mathrm{on}} = 30k\Omega$ and $R_{\mathrm{off}} = 80k\Omega$. However, the 6T-2M hybrid structure was unable to reliably operate at frequencies higher than 1GHz for reasons explained in Chapter 2 and Section 3.2.3. Additionally, our 3T-4M structure also outperformed the 6T-2M structure in terms of power requirement, with our 3T-4M XNOR gate requiring significantly less power compared with the 4T-2M XNOR gate of [39].

Table 3.2 also shows that, clearly, the 10T/8T CMOS XOR/XNOR gate failed at 4GHz, and the AOI22 XOR/XNOR gate failed as frequencies are reaching 8GHz, whereas our 3T-4M gate reliably operated at frequencies of 8GHz or higher as shown in Fig. 3.12.

In Section 3.4, we show that our multifunction memristive XOR/XNOR architecture offers compact and efficient design of more complex circuits.

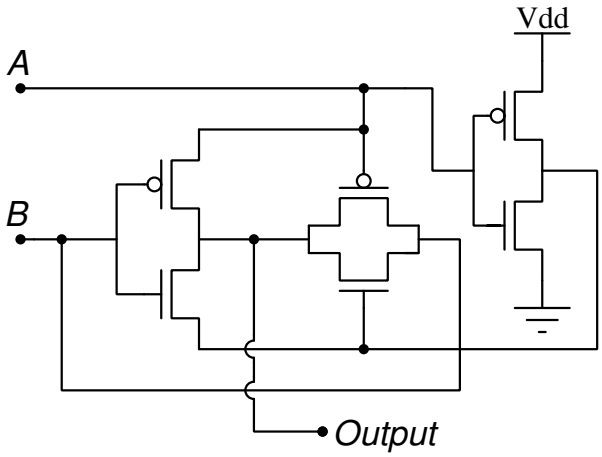


Figure 3.11: Bufferless 6T CMOS XOR (C-XOR) gate [56].

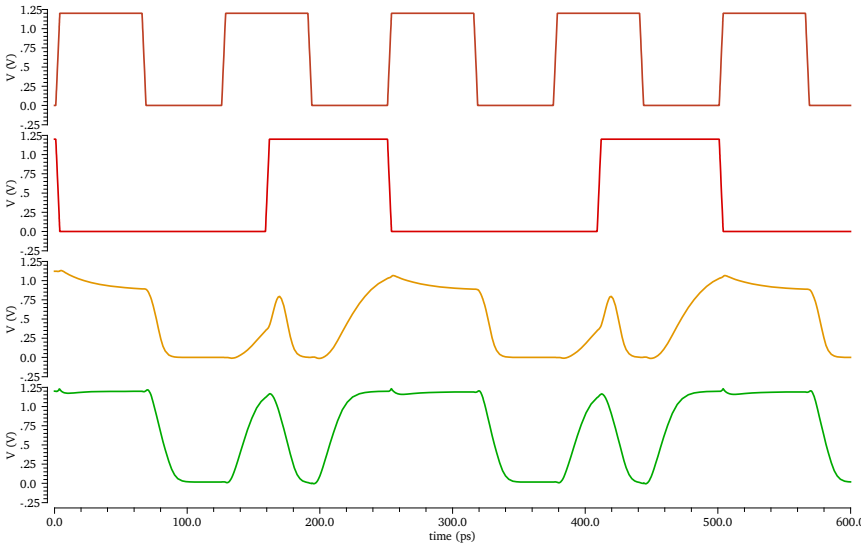


Figure 3.12: Buffered XOR performance for 10T and 3T-4M XOR gate at 8GHz: top two signals are inputs; third: 10T CMOS XOR gate [56]; fourth: proposed 3T-4M XOR gate.

Table 3.2: Performance analysis compared to existing techniques.

Frequency → **Architecture** ↓	**1GHz**		**2GHz**		**4GHz**		**8GHz**	
	Power (μW)	**Stat**	**Power** (μW)	**Stat**	**Power** (μW)	**Stat**	**Power** (μW)	**Stat**
3T-4M XOR	14.38	Pass	14.49	Pass	16.79	Pass	25.62	Pass
3T-4M XNOR	1.71	Pass	2.518	Pass	4.483	Pass	8.414	Pass
10T C-XOR[56]	61.11	Pass	67.93	Pass	72.01	Fail	76.57	Fail
8T C-XNOR[56]	50.56	Pass	54.55	Pass	56.50	Fail	59.78	Fail
AOI22 XOR [108]	7.32	Pass	12.99	Pass	24.29	Pass	46.80	Fail
AOI22 XNOR [108]	6.077	Pass	10.84	Pass	20.35	Pass	39.28	Fail
6T-2M XOR[39]	19.14	Pass	–	Fail	–	Fail	–	Fail
4T-2M XNOR[39]	14.69	Pass	–	Fail	–	Fail	–	Fail

3.4 Designing Systems

In this section, we demonstrate that our 3T-4M multifunction logic architecture can be used to design highly compact, reliable, and efficient systems. To this end, firstly we consider a full adder design, and then we present the design of multipliers over the Galois Fields.

3.4.1 Memristive Full Adder

A 2T-10M 'weak' full adder can be formed as shown in Fig. 3.13. This bufferless full adder mainly consists of two 1T-4M XOR/AND multifunction stages (Fig. 3.5(a)). The sum (S) and carry (C_o) outputs are formed as follows:

$$S = A \oplus B \oplus C_i \tag{3.14}$$

$$C_o = C_i(A \oplus B) \vee (A \wedge B) \tag{3.15}$$

In Eq. (3.15), the terms $A \wedge B$ and $C_i(A \oplus B)$ are shared from the first and second 1T-4M stages respectively.

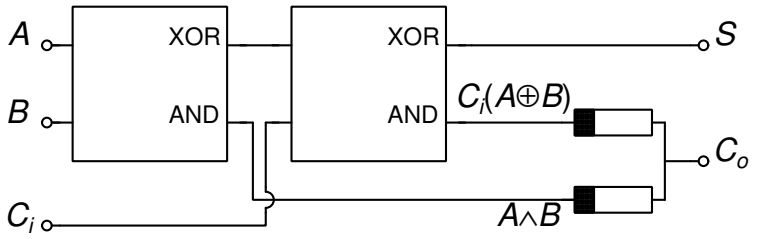


Figure 3.13: 2T-10M 'weak' adder design.

The multifunction properties of the proposed architecture also permit highly compact fully buffered full adder designs. For instance, to obtain the sum output S, the output of a 3T-4M XNOR gate (Fig. 3.5(b)) can be fed into another 3T-4M XNOR stage, thus giving us

$$S = (1 \oplus A \oplus B) \oplus (1 \oplus C_i) = A \oplus B \oplus C_i.$$

The output C_o can be formed by ORing the shared term $A \wedge B$ from the first 3T-4M XNOR stage with $C_i(A \oplus B)$ and then double inverting the output. In the term $C_i(A \oplus B)$, the term $A \oplus B$ is also a shared term from the first 3T-4M XNOR stage before the CMOS inverter. Hence, we only need two more memristors to simply form an AND gate. This yields the fully buffered 10T-12M structure as shown in Fig. 3.14.

For completeness, we have designed both bufferless and fully buffered full adders in all possible correct configurations of the proposed bufferless 1T-4M XOR/XNOR gates as well as the fully buffered 3T-4M XOR/XNOR gates. The total number of elements for the bufferless designs varies from 12 to 16 elements, whereas the total number of elements for the fully buffered designs varies from 22 to 24 depending on how the XOR and XNOR stages were connected. Table 3.3 shows the results of the designs. In this table the columns 'Architecture', 'Unbuffered', and 'Buffered' represent which architecture combination we used, and whether the adder is unbuffered or fully buffered respectively. The columns 'Shared', 'Elem', and 'Tot' represent the shared

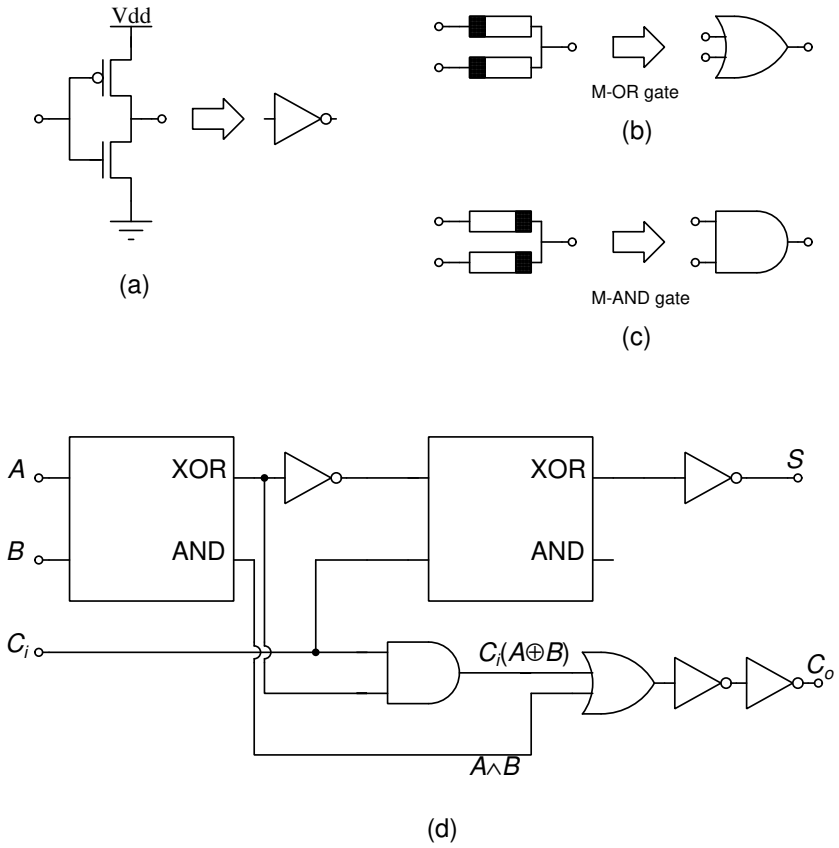


Figure 3.14: 10T-12M fully buffered adder circuit; (a) CMOS inverter; (b) Memristive OR gate (M-OR); (c) Memristive AND gate (M-AND); (d) 10T-12M adder.

terms, the total number of memristors and transistors used, and the total number of elements respectively. Table 3.3 clearly shows that the XOR-XOR architecture (Row-1) required the fewest number of elements. This design is also a significant improvement over the existing full adder designs e.g. 16T-18M [2] and 27T-2M [42].

3.4.2 Memristive Galois Field Multiplier

The arithmetic operations over finite fields (or Galois fields), i.e. over the set GF(2^m), where m is a non zero positive integer, have critical applications in public-key cryptography systems and error-correcting codes among others [111]. Addition and mul-

Table 3.3: Full adder design with proposed architecture.

Architecture	Unbuffered			Buffered		
	Shared	**Elem**	**Tot**	**Shared**	**Elem**	**Tot**
XOR-XOR	$A \wedge B$, $C(A \oplus B)$	2T-10M	12	$A \wedge B$, $A \oplus B$	10T-12M	22
XOR-XNOR	$A \wedge B$, $C(A \oplus B)$	4T-12M	16	$A \wedge B$, $A \oplus B$	12T-12M	24
XNOR-XOR	$C(A \oplus B)$	4T-12M	16	$C(A \oplus B)$	12T-12M	24
XNOR-XNOR	$A \vee B$	2T-14M	16	$A \vee B$/ $A \oplus B$	10T-14M	24

tiplication are the two basic arithmetic operations over these fields. While an m-bit adder over GF(2^m) only requires m XOR gates working in parallel, multiplication is much more complex and requires a combination of many AND and XOR gates [112]. In this section, we present design methods for hybrid memristive polynomial basis multipliers over GF(2^m). We begin by showing a multiplier design over GF(2^2) and extend this design over GF(2^4).

3.4.2.1 2-bit Multiplier Over GF

Let us assume that the two inputs of the 2-bit multiplier are $A = (a_1, a_0)$ and $B = (b_1, b_0)$. The polynomial representation of the elements over GF(2^2) is $A(x) = a_1x + a_0$, and $B(x) = b_1x + b_0$. Here, '+' represents addition over GF, which is equivalent to the XOR operation (i.e. '$\oplus$') over GF(2). Then $A(x) \cdot B(x) = (a_1x + a_0) \cdot (b_1x + b_0) = a_1b_1x^2 + (a_0b_1 + a_1b_0)x + a_0b_0$. Here, the element $a_1b_1x^2$ with exponent greater than 1 is not in the field. Hence it needs to be reduced modulo the primitive polynomial, which in this case is $P(x) = x + 1$. The final result is given as follows:

$$A(x) \cdot B(x) \mod P(x) = (a_0b_1 + a_1b_0 + a_1b_1)x + (a_0b_0 + a_1b_1). \tag{3.16}$$

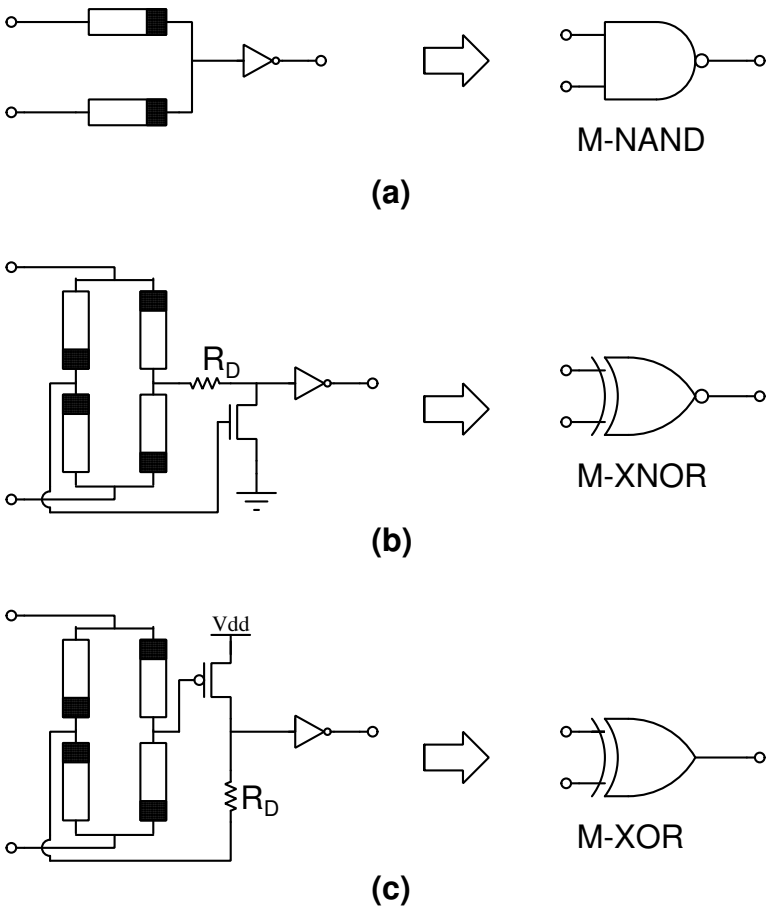


Figure 3.15: Memristive logic symbol; (a) Memristive logic NAND gate; (b) Memristive logic XNOR gate; (c) Memristive logic XOR gate

To simplify Eq. (3.16), we denote C_0, C_1..., C_{m-1} to be the coefficients corresponding to the terms associated with x^0, x^1..., x^{m-1} respectively. Then, Eq. (3.16) becomes

$$A(x) \cdot B(x) \mod P(x) = C_1 x + C_0. \tag{3.17}$$

This 2-bit GF multiplier consists of four AND and three XOR gates based on Eq. (3.16). As we have shown in Section 3.2.3 and Table 3.2, our 3T-4M XNOR gate is more power efficient compared to our 3T-4M XOR gate. Therefore, it is reasonable

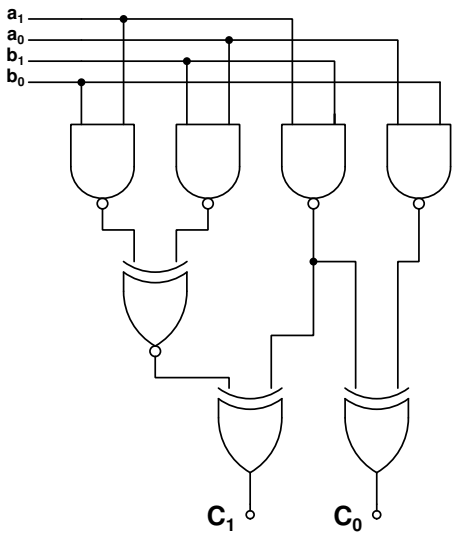


Figure 3.16: Memristive GF(2^2) Multiplier.

to replace as many XOR gates as possible with XNOR gates. To this end, we ensure reliable performance with low power by redesigning the circuit as follows:

- We use buffered memristive NAND (M-NAND) gates (Fig. 3.15 (a)), instead of buffered AND gates. The buffered M-NAND gates require a single inverter, which is more power efficient compared to a buffered M-AND gate.

- We replace as many buffered 3T-4M XOR gates (Fig. 3.5 (d)) as possible with our power efficient 3T-4M buffered XNOR gates (Fig. 3.5 (b)).

Therefore, we modify Eq. (3.16) as follows to accommodate the above optimisation:

$$\begin{aligned} C_0 &= (a_1 b_1 \oplus 1) \oplus (a_0 b_0 \oplus 1) \\ C_1 &= \overline{(a_0 b_1 \oplus 1) \oplus (a_1 b_0 \oplus 1)} \oplus (a_1 b_1 \oplus 1). \end{aligned} \tag{3.18}$$

The resulting design appears in Fig. 3.16.

Although in this design we replaced only one XOR gate with a power efficient XNOR gate in Fig. 3.16, the number of XOR gates which can be replaced is much

higher as the circuits are scaled up. We demonstrate this by designing a 4-bit multiplier over GF(2^4) as follows.

3.4.2.2 4-bit Multiplier Over GF

For designing a polynomial basis GF(2^4) multiplier, we multiply the two inputs $A = (a_3, a_2, a_1, a_0)$ and $B = (b_3, b_2, b_1, b_0)$ and then modulo the result by a primitive polynomial, which in our case is $P(x) = x^4 + x + 1$. The final result appears in Eq. (3.19).

$$\begin{aligned} A(x) \cdot B(x) \mod P(x) = \\ &(a_0b_3 + a_1b_2 + a_2b_1 + a_3b_0 + a_3b_3)x^3 + (a_0b_2 + a_1b_1 + a_2b_0 + a_2b_3 + a_3b_2 + a_3b_3)x^2 \\ &+(a_0b_1 + a_1b_0 + a_1b_3 + a_2b_2 + a_2b_3 + a_3b_1 + a_3b_2)x \\ &+(a_0b_0 + a_1b_3 + a_2b_2 + a_3b_1). \end{aligned} \tag{3.19}$$

This can be simplified as follows.

$$A(x) \cdot B(x) \mod P(x) = C_3x^3 + C_2x^2 + C_1x + C_0. \tag{3.20}$$

Now we replace all the AND gates with M-NAND gates and as many XOR gates as possible with M-XNOR gates. The new expression of each coefficient appears in the following:

$$\begin{aligned} C_0 &= \overline{(a_0b_0 \oplus 1) \oplus (a_1b_3 \oplus 1)} \oplus \overline{(a_2b_2 \oplus 1) \oplus (a_3b_1 \oplus 1)} \\ C_1 &= \overline{\overline{(a_0b_1 \oplus 1) \oplus (a_1b_0 \oplus 1)} \oplus \overline{(a_1b_3 \oplus 1) \oplus (a_2b_2 \oplus 1)}} \oplus \overline{(a_2b_3 \oplus 1) \oplus (a_3b_1 \oplus 1)} \oplus (a_3b_2 \oplus 1) \\ C_2 &= \overline{\overline{(a_0b_2 \oplus 1) \oplus (a_1b_1 \oplus 1)} \oplus \overline{(a_2b_0 \oplus 1) \oplus (a_2b_3 \oplus 1)}} \oplus \overline{(a_3b_2 \oplus 1) \oplus (a_3b_3 \oplus 1)} \\ C_3 &= \overline{\overline{(a_0b_3 \oplus 1) \oplus (a_1b_2 \oplus 1)} \oplus \overline{(a_2b_1 \oplus 1) \oplus (a_3b_0 \oplus 1)}} \oplus (a_3b_3 \oplus 1). \end{aligned}$$

The resulting circuit appears in Fig. 3.17. As we can see in this figure, we need

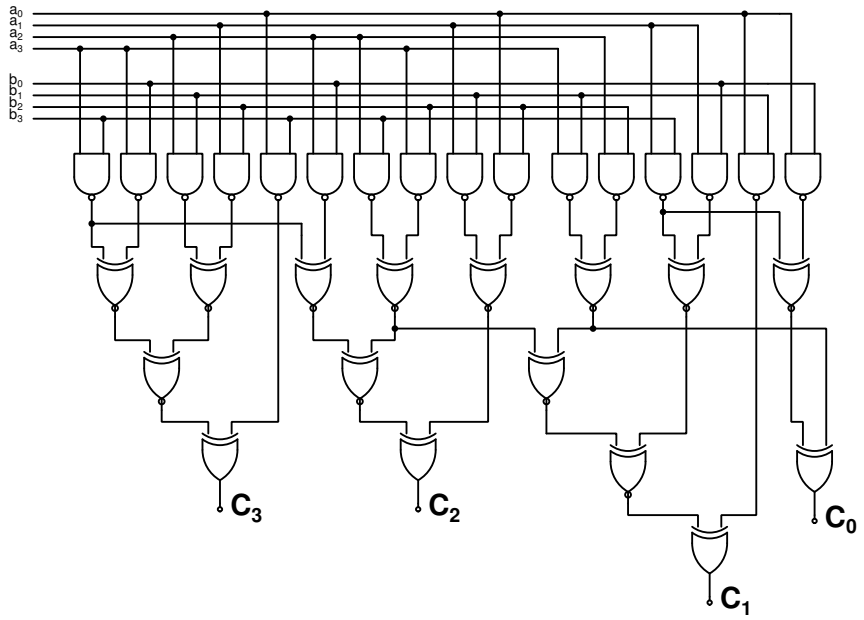


Figure 3.17: Memristive GF(2^4) Multiplier.

twelve 3T-4M XNOR gates and only four 3T-4M XOR gates.

The design methods presented in this section can be adopted for the design of m-bit multipliers over GF(2^m), e.g. by modifying/combining with the methods presented in [112].

The performance of the systems developed in this section is analysed in Section 3.5.

3.5 Experimental Results

All the memristors used in this Chapter are modelled based on [6] and coded in Verilog-A. All designs were implemented and simulated in Cadence Virtuoso. The threshold voltages of the memristor assumed are $V_{\text{on}} = -0.2\text{V}$ and $V_{\text{off}} = 0.02\text{V}$. The supply voltage considered is $V_{\text{DD}} = 1.2\text{V}$ and the transistors used here are based on the 32nm CMOS technology node. The saturation current for n-type and p-type transistors are $I_{D,sat} = 46.63\mu\text{A}$ and $I_{D,sat} = 47.6\mu\text{A}$ respectively. All the circuits

in this paper, where appropriate, were tested with a load capacitance of 1fF and all the power measurements are based on the sum of the static and dynamic powers as reported by Cadence.

3.5.1 Multifunction Gates

As we already demonstrated in Section 3.3, the power and reliability performance of our design clearly outperforms both the CMOS based 10T design as well as the technique of [39] (Table 3.2 and Fig. 3.12). In addition, the techniques of [7, 9, 42] require multiple clock cycles to operate, whereas our technique can be operated in a single clock cycle. Hence, for fairness these were not compared with. The technique of [2] proposed a 8T-6M XOR gate design, which requires similar power as pure CMOS design simply because it requires more than twice the number of transistors and two more memristors compared to our designs. Hence, this technique is clearly less power efficient compared to the proposed designs.

3.5.2 Full Adder Designs

For the full adder designs in Section 3.4.1 we considered $R_{\text{on}} = 500\Omega$, $R_{\text{off}} = 40k\Omega$, and $R_{\text{D}} = 10K\Omega$ based on Eq. (3.9) and Eq. (3.10). A reliable performance at high frequencies could be achieved with only 90.98μW power consumption when $k_{\text{on}} = -300$, $k_{\text{off}} = 52$. The input and output wave forms for our 10T-12M fully buffered full adder design appear in Fig. 3.18 at 8GHz. Of course, needless to say, the design also works at lower frequencies. Adders designed with both pure CMOS [56] and memristive hybrid gates based on [39] failed at these frequencies and their power performance could not be compared. Hence, we also tested our adder design at 2GHz, i.e. the frequency where the CMOS XOR/XNOR gate worked (Table 3.2). At 2GHz the proposed adder required 40.77μW, which is clearly better than a single CMOS XOR and XNOR gate at this frequency.

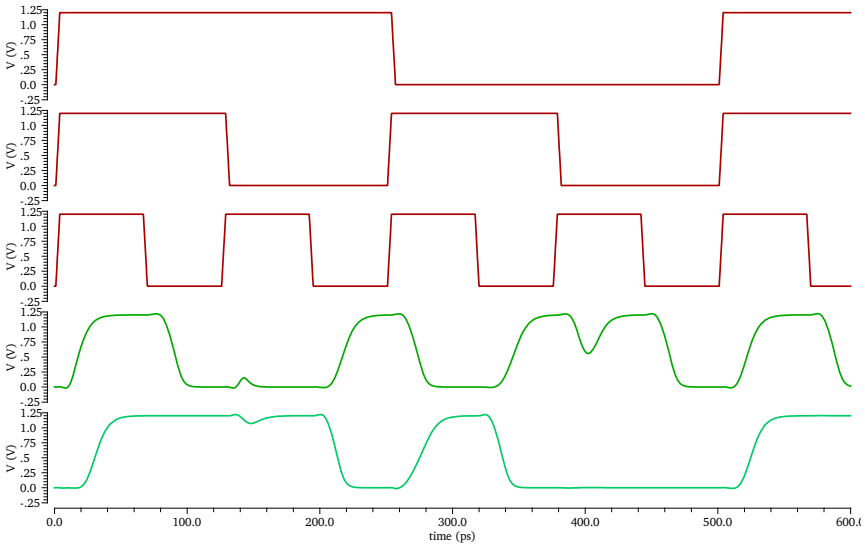


Figure 3.18: 10T-12M buffered full adder at 8GHz frequency. Top three signals are the inputs A, B and input carry, C_i, respectively, and the bottom two signals are the sum (S) and output carry (C_o) respectively.

3.5.3 Multiplier Over GF Designs

For the multiplier designs over GF in Section 3.4.2, the power consumption of each individual memristive logic gate M-NAND, M-XNOR and M-XOR in Fig. 3.15 are 1.016μW, 1.1774μW and 29.11μW at 1GHz respectively. The input and output wave forms of our GF(2^2) multiplier circuit is shown in Fig. 3.19, which consumes 63.401μW power. The proposed memristive GF(2^4) multiplier results in considerably smaller area and requires lower power compared with pure CMOS implementation. The entire circuit requires approximately 153.984μW power, which is much lower than 535.2μW consumed by the same multiplier designed with pure CMOS technology. The total number of elements used for the memristive GF(2^4) multiplier is 176 (96M-80T) which is much fewer than 264T used by the pure CMOS design. We also tested the 4-bit multiplier based on the XOR/XNOR gate in [39]. The latter design required 168 (64M-104T) elements, but it also required significantly more transistors compared

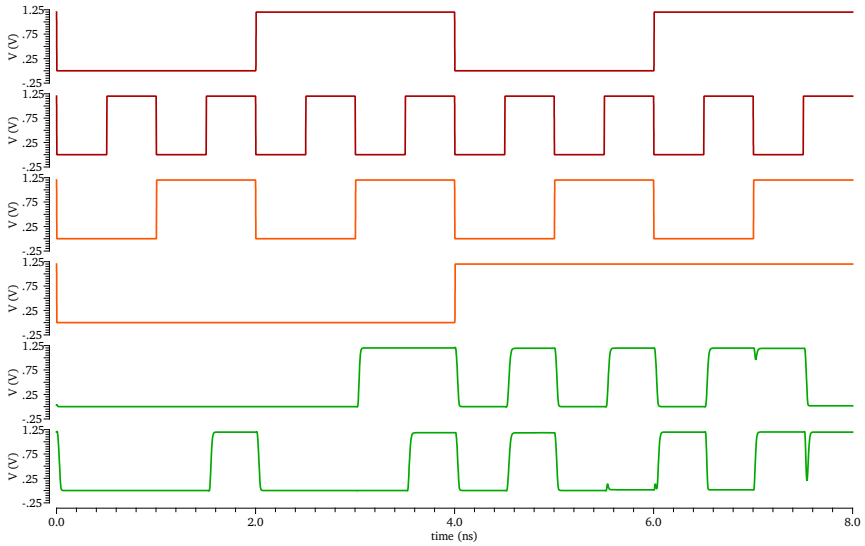


Figure 3.19: Memristive GF(2^2) Multiplier at 1GHz. Top four signals are the inputs a_0, a_1, b_0 and b_1 respectively. Bottom two signals are the outputs C_0 and C_1.

to the proposed design. However, this design failed to operate at a relatively high frequency of 1GHz and hence the power figures could not be obtained. The technique of [2] yields designs with much higher number of transistors and memristors for the multiplier designs (e.g. 128M-136T or 264 elements for the GF(2^4) multiplier).

Because of the storage limitation of the simulation environment, it is difficult to implement some larger multipliers (e.g. GF(2^8), GF(2^{16})...GF(2^m)). However, the power consumption of each individual memristive logic gate M-NAND, M-XNOR and M-XOR are considerably less. Hence, it can be estimated that even for a GF(2^m), the circuit consumes less power compared to CMOS technology.

3.6 Summary

This chapter proposed novel memristive logic architectures for low power and reliable performance at low as well as high frequencies. Firstly, with the help of MRL, we showed that memristors can naturally represent multiple valued logic as MIN-MAX

post algebra. We proposed an efficient multifunction logic architecture which can operate in single clock cycle with considerably fewer components and less energy. It is also capable of seamless integration with the CMOS technology for hybrid CMOS memristive chip fabrication in 3D [44]. We also demonstrated that with the proper selection of k_{on} and k_{off} parameters, this multifunction logic architecture can reliably operate at high frequencies where both CMOS devices as well as existing hybrid-memristor gates start to fail. The proposed logic architecture consumes significantly less power when compared with pure CMOS as well as existing hybrid memristive logic circuits.

We used the proposed 1T-4M multifunction architecture as the essential building block to implement a highly compact full adder design with very low power requirement. Finally, we presented design techniques for highly efficient memristive bit parallel multipliers over GF(2^2) and GF(2^4). These designs required considerably lower power compared to pure CMOS designs. The latter design technique can be generalised to m-bit multipliers, e.g. by combining with existing techniques [112]. Our experimental results demonstrated that the proposed memristive multifunction logic architecture can be effectively used to design power efficient low complexity systems alongside the CMOS designs.

Apart from applications in logic design, memristors with strong security primitives are finding applications in PUFs for the emerging hardware security solutions [40, 113]. The XOR arbiter PUF is one of the most common delay-based PUFs, e.g. that in [39] which utilises a 6T-2M memristive XOR structure. [39] also demonstrated that a high degree of randomness could be derived by designing PUF with memristors. Our proposed logic structure has two more memristors and one fewer transistor than the XOR structure proposed in [39]. The presence of extra memristors in our structure increases the variation level which makes it well suited for delay-based PUFs as demonstrated in Chapter 4.

4

Analysis of Memristive Parasitic Effects on Logic Architecture

4.1 Introduction

A number of memristor models have been proposed rapidly [3–6, 48–50, 114] with their own attributes (e.g. symmetricity and operating frequency etc.) and have been applied in different applications as shown in Section 2.2.1. However, the memristor models used for those applications did not take the parasitic effects into consideration. Therefore, [115, 116] proposed a generic memristor model, with parasitic components built in and it corresponds more closely to the memristive device in reality. In this chapter, we show how the memristor parasitic components effect our proposed memristive logic architectures in Section 4.2. Then, we use 3T-4M buffered memristive XOR gate, which combines with parasitic components, to propose a delay based arbiter Physical Unclonable Function (PUF) in Section 4.3. We show that our 3T-4M logic architecture has certain advantages over the existing 6T-2M logic architecture [39], particularly in dealing with the cascaded XOR structure e.g. delay based Physical Unclonable Function (PUF). By comparing with other CMOS-based arbiter PUFs, our proposed PUF demonstrates some improvements in terms of uniqueness, uniformity and bit-aliasing.

4.2 Memristive Parasitic Effects in Hybrid Systems

4.2.1 Memristor Model with Parasitic Components

As we discussed in the previous chapters, the hysteresis loop of the memristor is pinched at the origin of I-V plane. However, the pinched hysteresis loop of the real memristive devices may not pass through the origin as we expected. In addition, the pinched point may disappear when the memristive circuit exceeds a certain frequency [115]. Hence, to emulate a real physical memristive device properly, [115] proposed a generic memristor model as shown in Fig. 4.1(a), where m represents the

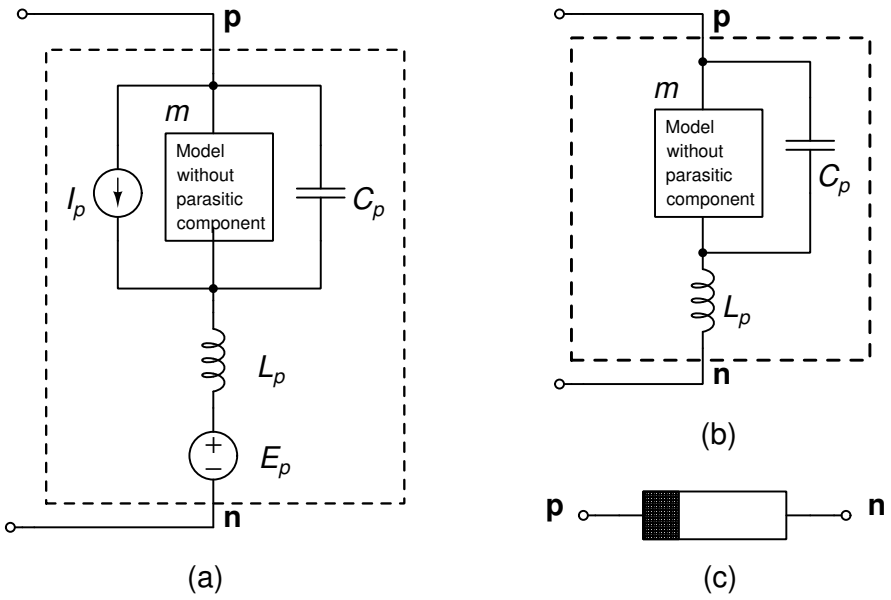
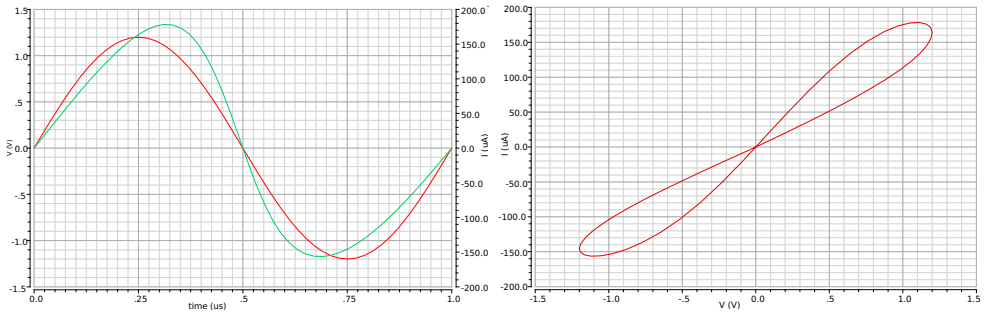


Figure 4.1: (a) Memristor with parasitic components.

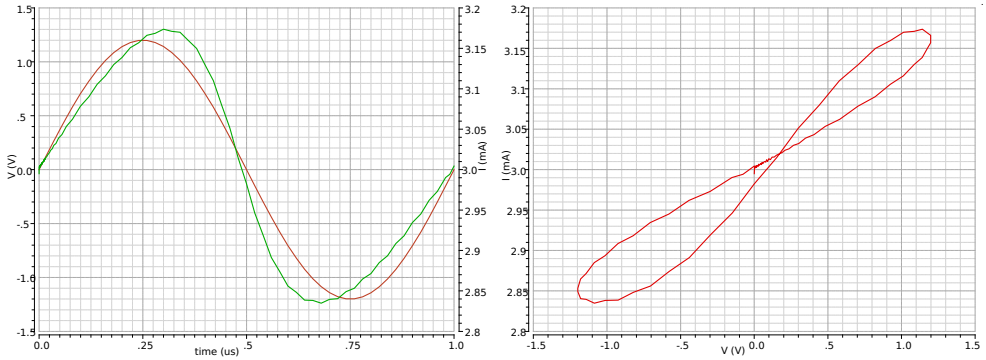
memristor model without parasitic components. This generic model consists of the basic memristor model in parallel with a parasitic capacitor C_p and a current source I_p. Then, a parasitic inductor L_p along with a voltage source E_p is connected in series as depicted in Fig. 4.1(a).

In order to ensure that the pinched point of the memristor's hysteresis loop does not disappear, we assume that $C_p = 1p\text{F}$, $I_p = 3m\text{A}$, $L_p = 10n\text{H}$ and $E_p = 30m\text{V}$. We applied a sinusoidal voltage, $v(t) = A\sin(2\pi ft)$, to both the ideal memristor (the memristor without any parasitic component) and Fig. 4.1(a) at 1MHz, where $A = 1.2\text{V}$. The result in Fig. 4.2 (a) indicates that there is no phase shift between the input voltage and output current. Hence, the memristor is pinched exactly at the origin as demonstrated in Fig. 4.2(b). However, the result in Fig. 4.2(c) observes that parasitic components induce the phase shift to the current response. Therefore, the pinched point drifts from the origin as shown in Fig. 4.2(d), thereby non-ideal pinched hysteresis loop. The voltage source E_p and the current source I_p directly emulate the



(a) Voltage and current curves of the ideal memristor.

(b) I-V characteristic of the ideal memristor.



(c) Voltage and current curves of the non-ideal memristor.

(d) I-V characteristic of the non-ideal memristor.

Figure 4.2: (a) and (c) show the input voltage and the current response of the ideal memristor and Fig. 4.1(a) respectively, where the red signal is the sinusoidal voltage input and green signal is the current response; (b) and (d) show the I-V pinched hysteresis loop for the ideal memristor and Fig. 4.1(a) respectively.

phase shift of the current response. Hence, Fig. 4.1(b) represents the simplified model which only takes the capacitor C_p and the inductor L_p as the parasitic components into consideration. The symbol of the Fig. 4.1(b) is illustrated in Fig. 4.1(c).

The model applied for m of Fig. 4.1(b) needs to be asymmetric and suitable for logic design. Therefore, one of the most flexible models, VTEAM, has been considered as m and coded in Verilog-A for all the memristive logic architectures described in this chapter.

4.2.2 Unit Step Response for the Single Memristor

To evaluate the parasitic effects on the memristive logic architecture, we demonstrated the unit step response for a single memristor model as shown in Fig. 4.1(b). The impedance of the memristor for the Low Resistance state (LRS) and the High Resistance State (HRS) are shown in Eq. (4.1) and Eq. (4.2) respectively.

$$Z_{\text{on}} = (R_{\text{on}}||Z_{\text{C}}) + Z_{\text{L}} = \frac{R_{\text{on}}\frac{1}{sC}}{R_{\text{on}} + \frac{1}{sC}} + sL \tag{4.1}$$

$$Z_{\text{off}} = (R_{\text{off}}||Z_{\text{C}}) + Z_{\text{L}} = \frac{R_{\text{off}}\frac{1}{sC}}{R_{\text{off}} + \frac{1}{sC}} + sL \tag{4.2}$$

Z_{C} represents the effective impedance of C_p which is $\frac{1}{sC}$. Similarly, Z_{L} represents the effective impedance of L_p which equals to sL. Where the $\frac{1}{s}$ is the Laplace operator for the integration, and s is used for the differentiation. Hence, the current response of the memristor is calculated as follows:

$$I(s) = \frac{V(s)}{Z_{\text{on}}(s)} = \frac{\frac{1}{s}}{\frac{R_{\text{on}}\frac{1}{sC}}{R_{\text{on}}+\frac{1}{sC}} + sL} \tag{4.3}$$

Eq. (4.3) can be rewritten as a polynomial equation, $H(s) = \frac{\sum_{i=0}^{m} b_i s^{m-i}}{\sum_{i=0}^{n} a_i s^{n-i}}$. Hence, Eq. (4.4) also describes Eq. (4.3).

$$I(s) = \frac{V(s)}{Z_{\text{on}}(s)} = \frac{R_{\text{on}}Cs + 1}{R_{\text{on}}CLs^3 + Ls^2 + R_{\text{on}}s} \tag{4.4}$$

Eq. (4.4) indicates the third-order system with at least one real pole and a complex conjugate poles. The poles of the denominator are located in left half of the s-plane. This implies that a memristor will produce the attenuated oscillation as illustrated in Fig. 4.3. Therefore, any change in input signal results in memristor producing damped oscillation.

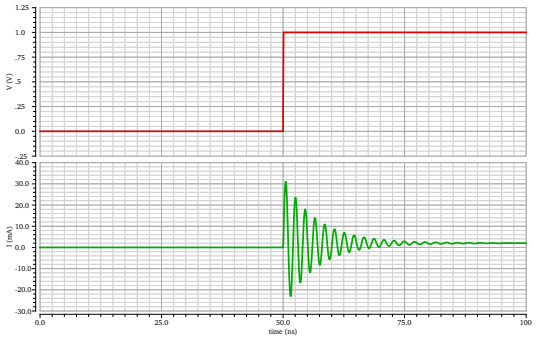


Figure 4.3: Step response of the generic memristor model which is shown in Fig. 4.1(b): The first signal is the unit step input; The second signal is the current response of the memristor.

4.2.3 Parasitic Effects in Memristive XOR Logic Architecture

Previously, we applied a step function to the single memristor to investigate the decaying oscillation of the current response. In this section, the parasitic effects on memristive logic design will be analysed by implementing the new generic memristor model to the existing MRL [41] and our proposed memristive logic architecture.

The non-ideal memristor model as presented in Fig. 4.1(b) is applied to the Memristor Ratioed Logic (MRL) architecture [41]. As discussed earlier in Chapter 2, the MRL is a voltage divider for realising the Boolean OR and AND functions as shown in Fig. 2.15(a) and Fig. 2.15(b) respectively. The input/output behaviour of OR and AND operations based on the MRL architecture in Fig. 2.15 is presented in Fig. 4.4. Because the parasitic components of the memristor have been taken into account, the output logic OR and AND operations demonstrate the decaying oscillation effects which are shown in Fig. 4.4, especially when the logic inputs at the switching stages. Therefore, these memristive parasitic effects bring the challenges to the design of stable logic circuits with MRL technique.

However, the proposed purely memristive XOR architecture as shown in Fig. 3.3 (a) can reduce the unwanted effects. M_1 and M_3 constitute a logical AND gate, while M_2 and M_4 constitute a logical OR gate. The output of the XOR operation can be

directly generated by taking the voltage difference between V_{L1} and V_{L2}. Hence, additional oscillations produced by the AND gate and the OR gate have been eliminated as shown in Fig. 4.4. To realise the 1T-4M XOR and XNOR operations, the NMOST and PMOST are connected as shown in Fig. 3.5 (a) and (c) respectively. The transistor connected here can help further smoothen the output signal as demonstrated in Fig. 4.4. Therefore, the proposed 1T-4M architecture can naturally reduce the parasitic effects of the circuit. However, these parasitic components still cause a large amount of propagation delay, which is beneficial to the design of delay-based PUF.

4.3 Proposed Memristive Arbiter PUF

4.3.1 Delay Analysis of Cascaded Memristive XOR Architecture

Since the memristor contains parasitic elements, such as the C_p and L_p as shown in Fig. 4.1 (b), a single XOR gate generates a larger propagation delay due to the increase in RC time constant. This feature makes the proposed XOR gate a good option for delay-based PUF design. In addition, by combining with the parasitic components, the memristive XOR gate can also increase the degree of the process variation as shown in Fig. 4.5. The proposed buffered 3T-4M XOR circuit shows large variation in delay due to process variations, as depicted in Fig. 4.6. Fig. 4.6 demonstrates the variation in rise and fall times of the XOR gate obtained from a 5000 samples Monte-Carlo simulations. This can be a disadvantage for designing a reliable circuit in most of the cases. However, a large degree of variation leverage the randomness of the PUF to a certain extent. In this section, we first cascade memristive XOR gates for few stages to observe the effects of the propagation delay. The experimental results shows that the proposed 3T-4M buffered XOR architecture in Fig. 4.7 (a) provides robust performance compared with other existing 6T-2M XOR architecture [11].

To evaluate the time delay for each stage of the cascaded XOR gates, we utilise fully

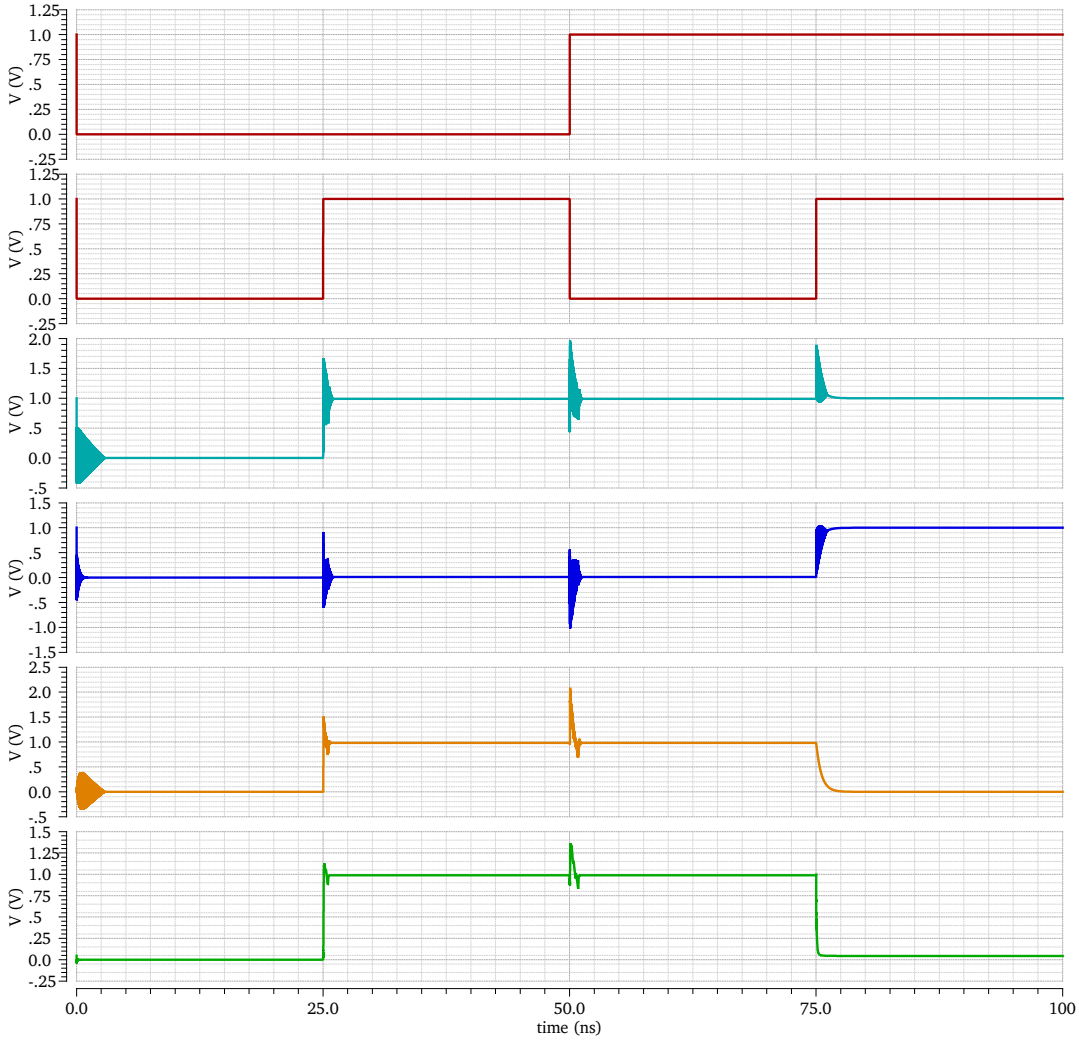
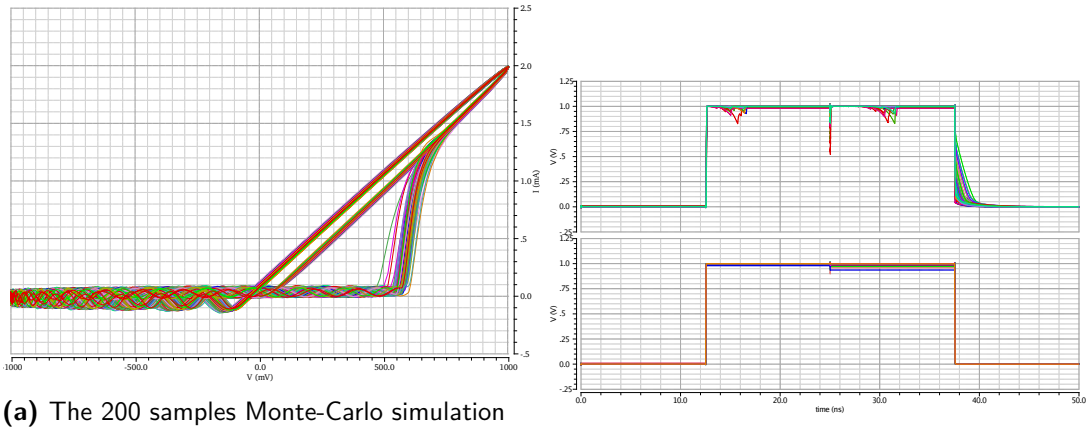


Figure 4.4: Top two signals are the inputs a and b respectively; third: the output of V_{L1} (OR gate); fourth: the output of V_{L2} (AND gate); fifth: the voltage difference between the V_{L1} and V_{L2}; sixth: the output of the 1T-4M XOR architecture.



(a) The 200 samples Monte-Carlo simulation of hysteresis loops of a single memristor: The nominal memristor parameters: $R_{\text{off}} = 80k\Omega$, $R_{\text{on}} = 500\Omega$, $C_p = 20$fF, $L_p = 10$nH, $V_{\text{off}} = 0.02$V, $V_{\text{on}} = -0.2$V, $D = 3$nm and $R_{\text{D}} = 16k\Omega$. The variation of parameters (e.g., C_p, L_p, and D) and threshold voltages (e.g.,V_{on} and V_{off}) are $\pm 5\%$.

(b) Monte Carlo simulation on single memristive XOR architecture at 25MHz. The top signals are the outputs from the XOR gate with parasitic components; the bottom signals are the outputs from the XOR gate without parasitic component. $\pm 5\%$ variation on the size of the memristor (3nm) has been taken here.

Figure 4.5: Monte-Carlo simulation for a single non-ideal VTEAM memristor and proposed 1T-4M memristive XOR gate.

buffered 3T-4M XOR gate as shown in Fig. 4.7 (a) and construct them in the fashion presented in Fig. 4.7(c). It will become the basic delay-based arbiter PUF circuit if the D-type flip-flop which appears with a dashed line is preceded by the cascaded XOR gates. Here, C_0, C_1, C_2, C_3 and C_4 are the challenge bits. The output from the D-type flip-flop q is the response bit of the PUF. For the sake of simplicity, in this case, the challenge bits have been given the same logic level, $C_0 = C_1 = C_2 = C_3 = C_4$. This results in the same XOR behaviour with the propagation delay between the node x and the node y which are labeled in Fig. 4.7 (c). Assuming $R_{\text{on}} = 500\Omega$, $R_{\text{off}} = 80k\Omega$, $C_p = 20f$F and $L_p = 10n$H, we run the simulation under 50MHz. The outputs from the node x and node y are illustrated in Fig. 4.8. The blue signal refer to the output from the first stage x and the green one refer to the output from the fifth stage y. The propagation time delay from the first stage to the fifth stage is approximately 0.519ns. Hence, our 3T-4M XOR structure provides about $0.519/5 = 0.104ns$ time

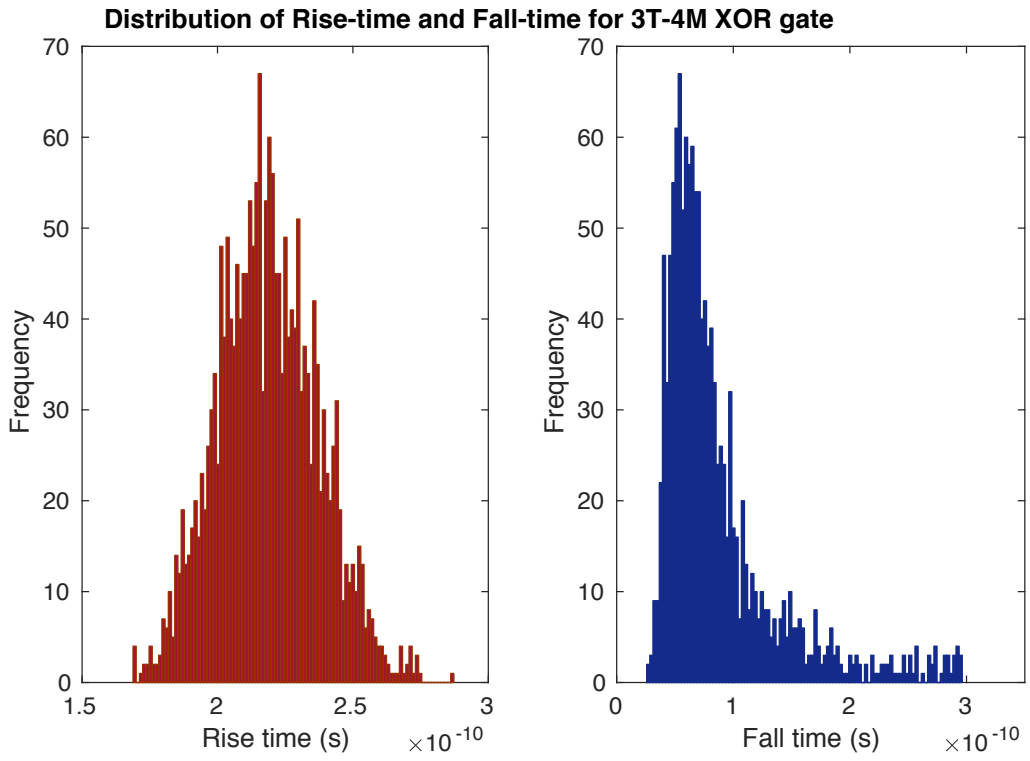


Figure 4.6: Rise and fall propagation delay distribution for the 3T-4M XOR gate.

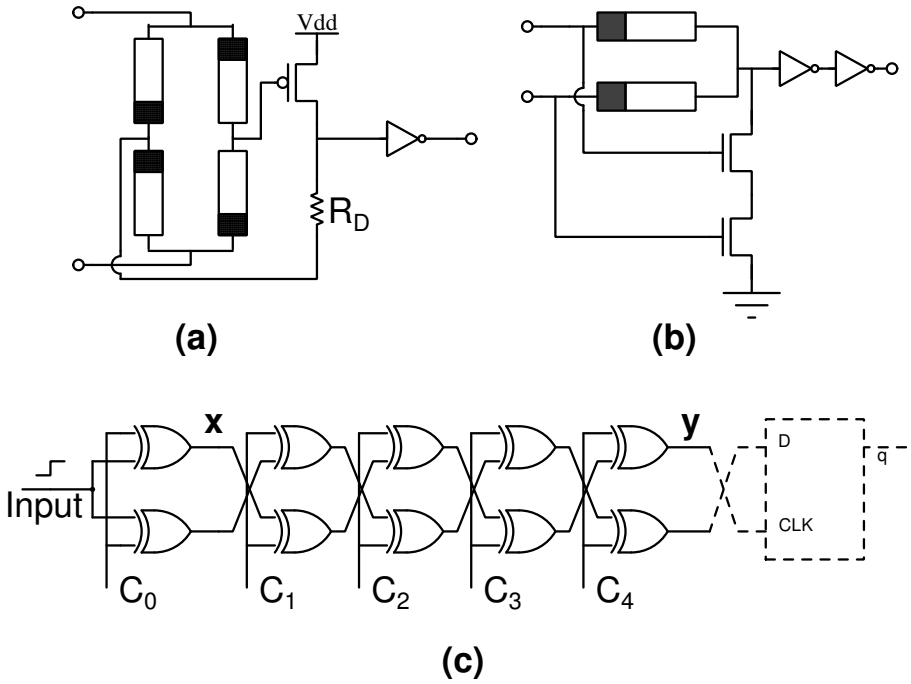


Figure 4.7: Cascaded Memristive XOR gates. (a) 3T-4M buffered XOR architecture; (b) Existing 6T-2M XOR architecture [39]; (c) 5-stage cascaded memristive XOR gates.

delay at each stage.

Fig. 4.8 also demonstrates that our proposed 3T-4M buffered XOR architecture still maintains the signal integrity without being distorted even the signal travels through multiple stages. To compare our architecture with other techniques, we apply the 6T-2M XOR architecture [11] which is shown in Fig. 4.7(b) as the same fashion as the Fig. 4.7(c). According to our experiments in Chapter 3, the 6T-2M XOR design required R_{on} to be 30kΩ to bias the pair of pull-down NMOS transistors. Therefore, under the influence of parasitic components, 6T-2M techniques results in the amount of glitches due to the large RC time constants. Hence, it can significantly reduce the reliability of the XOR gate and produce the wrong output signals before the arbiter operation. Fig. 4.9 presents the output of 5-stage cascaded XOR gates for both 6T-2M and 3T-4M techniques. From the results, our proposed 3T-4M buffered

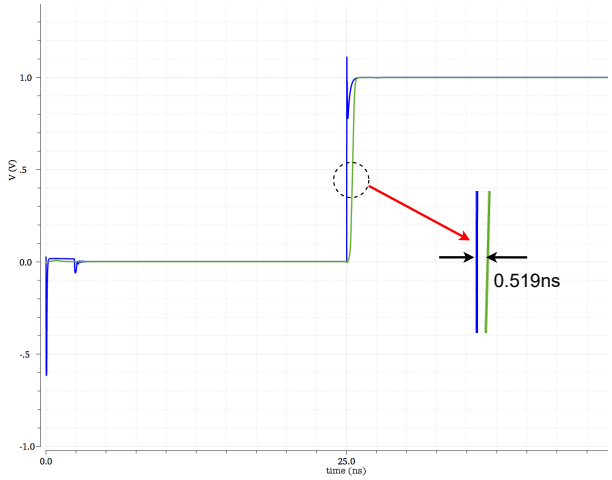


Figure 4.8: The output of the memristive XOR gates between the x and y. The blue signal is the output from the node x and the green signal from the node y.

XOR cascaded architecture provides more reliable performance by comparing with the 6T-2M memristive technique.

4.3.2 The Structure of Basic Memristive Arbiter PUF

Techniques introduced in [40] shows that memristor can be leveraged for security along with its particular characteristics including non-volatility, nonlinear I-V relationship, formation process, memristance drift and radiation-hardness which are dependent on the material property. Therefore, in this section, we use the proposed 3T-4M memristive XOR logic architecture as a fundamental element to build a delay based arbiter PUF.

Fig. 4.10 illustrates the block diagram of the design. The memristor applied for each memristive hybrid XOR gate in this design is the generic model which is shown in Fig. 4.1(b). The design consists of multiple cascaded XOR gates, two AND gates, and an arbiter.. Here, the arbiter is comprised of two D-type flip-flops and a multiplexer (MUX), and it is utilised as a double edge triggered flip-flop as shown in Fig. 4.11. The two AND gates are used to enable the arbiter. In Fig. 4.10, the input stimulus

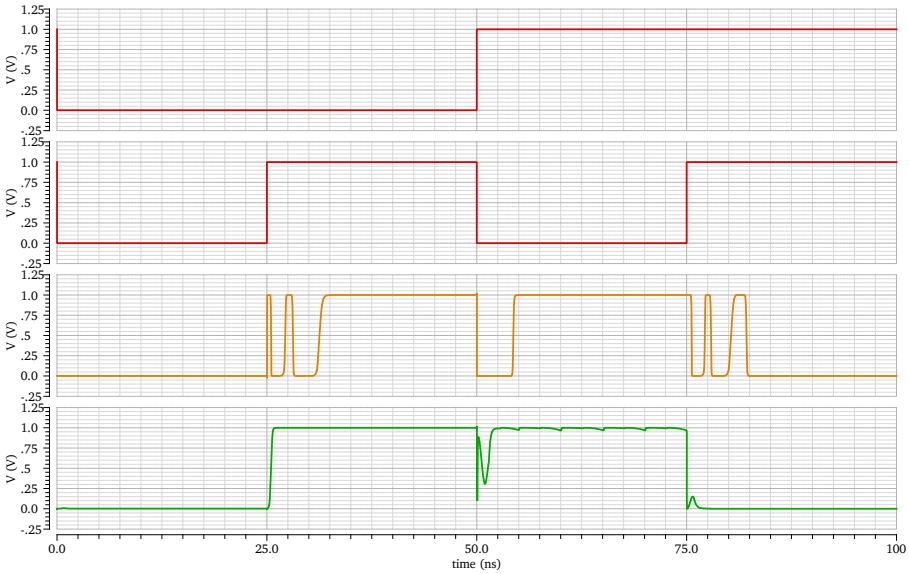


Figure 4.9: Outputs of the cascaded Memristive XOR gates: the top two signals are the inputs; the third signal is the output of the 5-stage cascaded 6T-2M XOR architecture [11]; the fourth signal is the output of the 5-stage cascaded 3T-4M XOR architecture.

splits and feeds into two chains of the cascaded memristive XOR gates. The inputs C_0-C_n are the challenges which control the exact path for the two propagating signals from Path-1 and Path-2. Since the manufacturing variation leads to the propagation delay of each logic gate varied, the two signals race each other by accumulating the delay difference from each gate. This results that the time for each signal arrived at flip-flop is also different. Hence, the final output response which is determined by the arbiter operation is random and unpredictable (see Fig. 4.11). The n-bit response arbiter PUF as shown in Fig. 4.12 can be obtained by simply paralleling the single bit PUF which is shown in Fig. 4.10. Here, we mainly focus on the 8-bit, 16-bit and 32-bit designs. The next section shows some comparisons between our proposed arbiter PUF and other existing similar PUFs in terms of the hardware overhead and performances. Fig. 4.13, Fig. 4.14 and Fig. 4.15 illustrate the delay and delay difference distributions along the Path-1 and Path-2 of the proposed 8-bit, 16-bit and 32-bit arbiter PUFs respectively. The plots in Fig. 4.13 and Fig. 4.14 are obtained through 20000 Monte

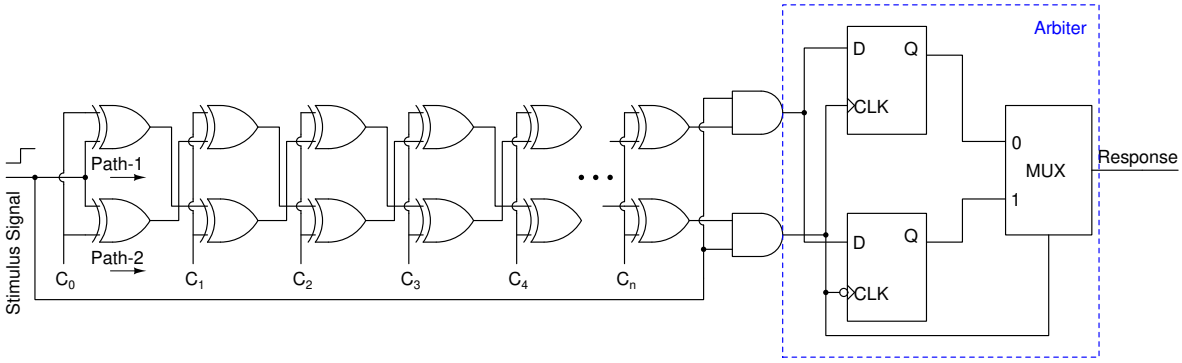


Figure 4.10: The basic structure of 1-bit response arbiter PUF.

Carlo simulations (the parameters for simulations are shown in Table 4.1). The delay distributions as shown in Fig. 4.15 are obtained by 2000 Monte Carlo simulations because of limitation of the storage and simulation time. In general, the simulation time for each arbiter PUF (e.g. 8-bit, 16-bit and 32-bit) is around 1.5 weeks.

4.3.3 Experimental Results & Discussion

All the circuits in this chapter are designed and simulated in Cadence Virtuoso. The supply voltage is $V_{\rm DD} = 1$V and the transistors are based on 32nm Predictive Technology Model (PTM) [117]. The memristor model used here is the same as the model used in Chapter 3. For the Monte Carlo simulations, we consider the process variation effects of the memristor and the transistor. The memristor parameters for Monte Carlo simulations are indicated in Table 4.1. We also set the $\pm 10\%$ variation in transistor threshold voltage, V_{th}. Here, D is the width of the memristor; $V_{\rm on}$ and $V_{\rm off}$ are the threshold voltages of the memristor. In order to avoid the metastability of the arbiter [118], the circuit is simulated at 20MHz.

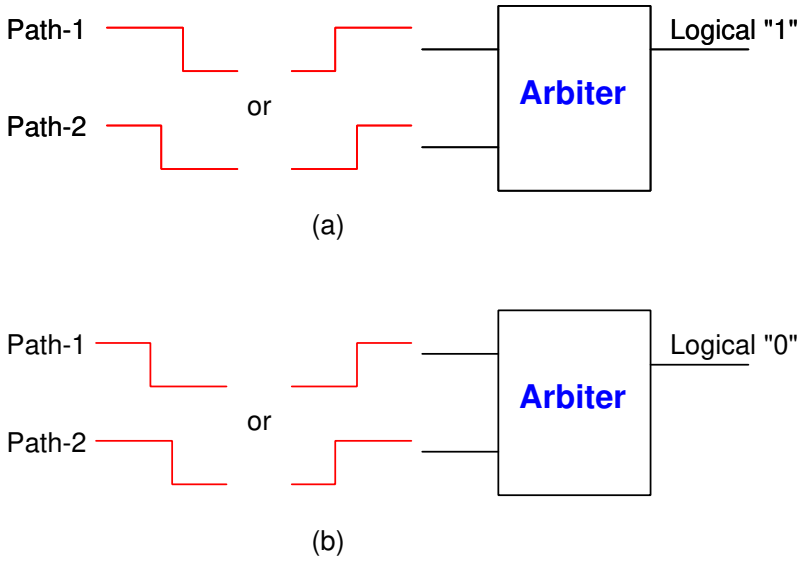


Figure 4.11: Arbiter functionality: The output is logical '1' when the signal of path-1 is faster than path-2, otherwise the output is logical '0'. (a) Signal of Path-1 is faster than Path-2; (b) Signal of Path-2 is faster than Path-1.

Table 4.1: Memristor parameters & Monte-Carlo Setup

Memristor Parameters	Nominal Value	Variation(%)
D	3nm	$\pm 5\%$
V_{on}	-0.2V	$\pm 5\%$
V_{off}	0.02V	$\pm 5\%$
C_p	20fF	$\pm 5\%$
L_p	10nH	$\pm 5\%$

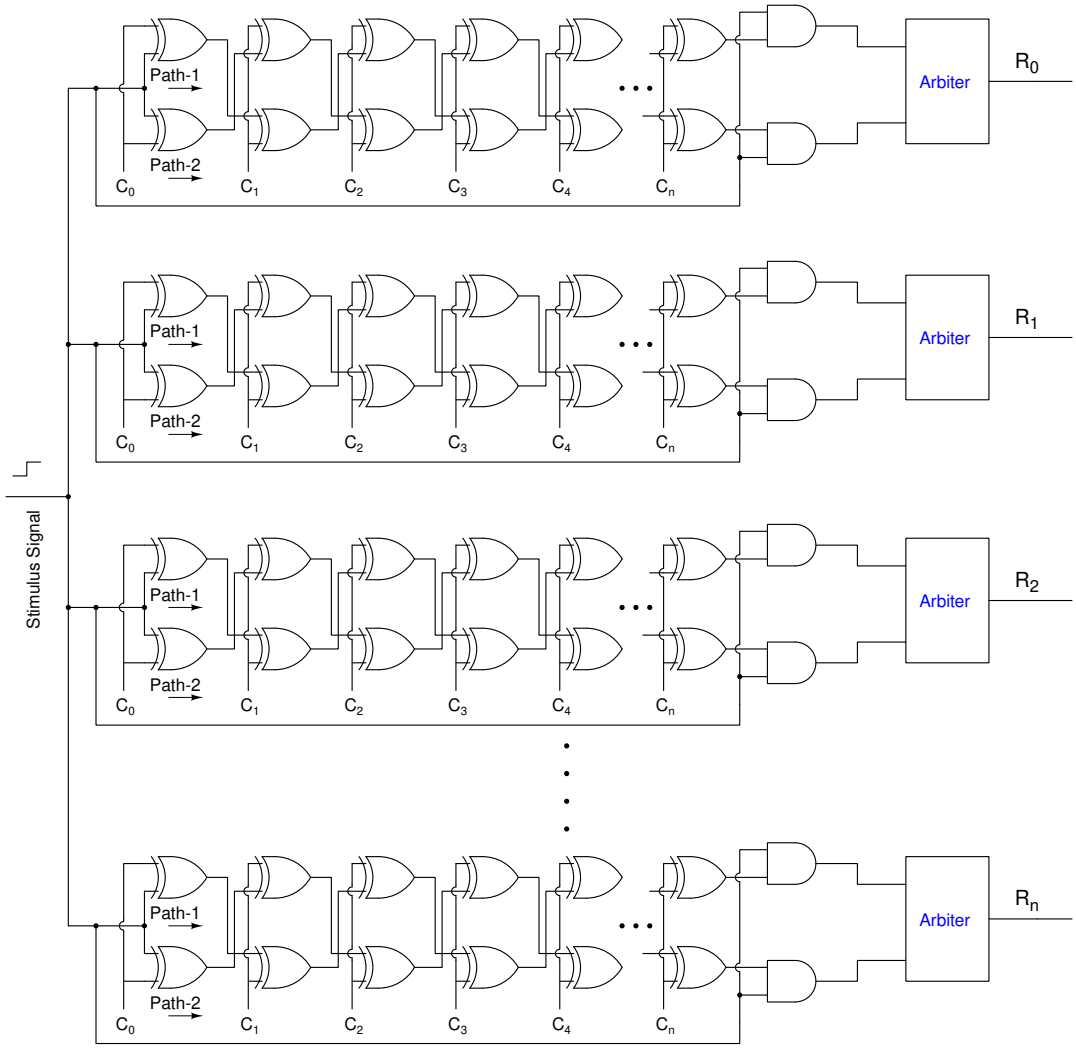


Figure 4.12: n-bit delay-based arbiter PUF.

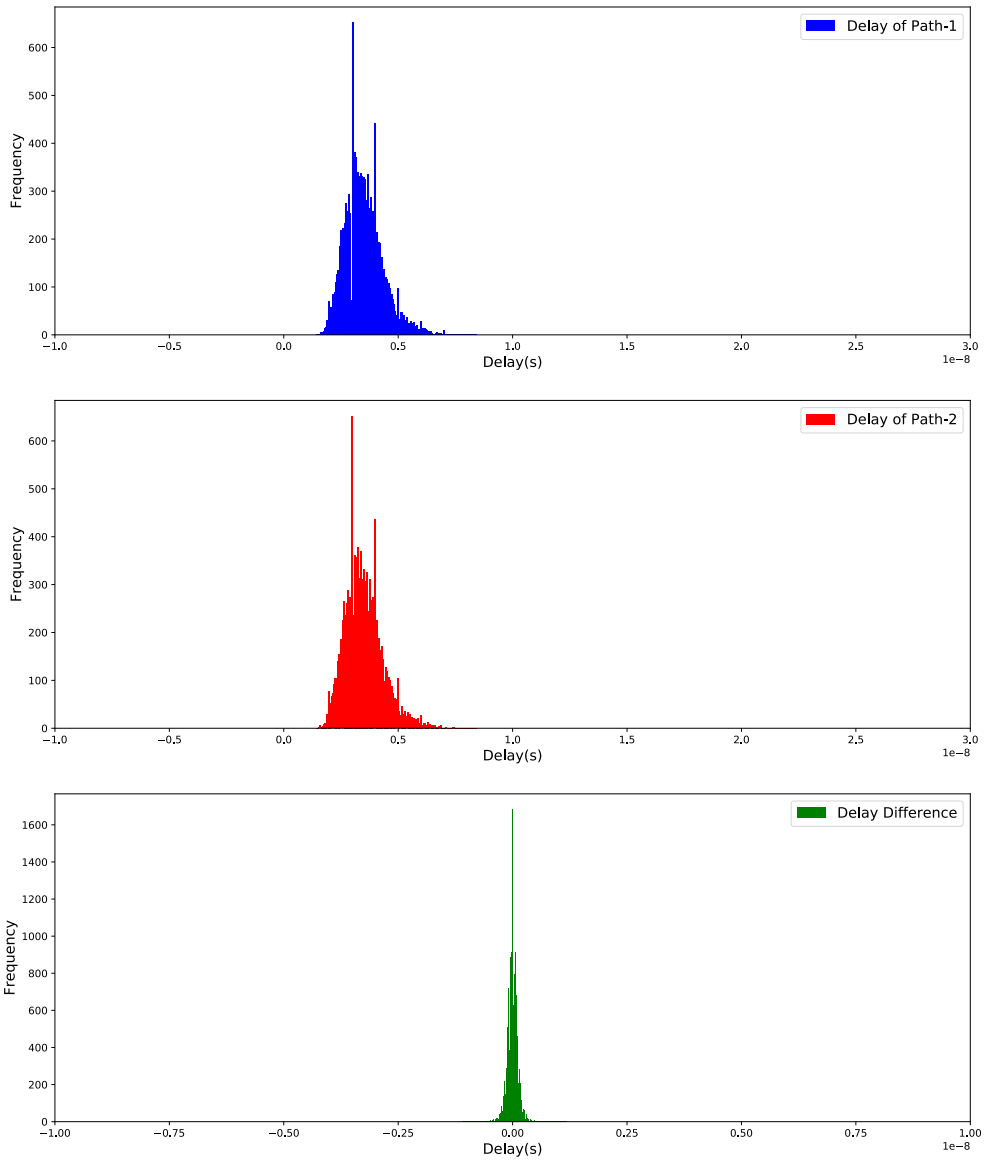


Figure 4.13: Delay distributions obtained by 20000 Monte Carlo simulations for 8-bit arbiter PUF: blue plot is distributions for Path-1 delays; red plot is distributions for Path-2 delays; green plot is distribution for delay difference between Path-1 and path-2.

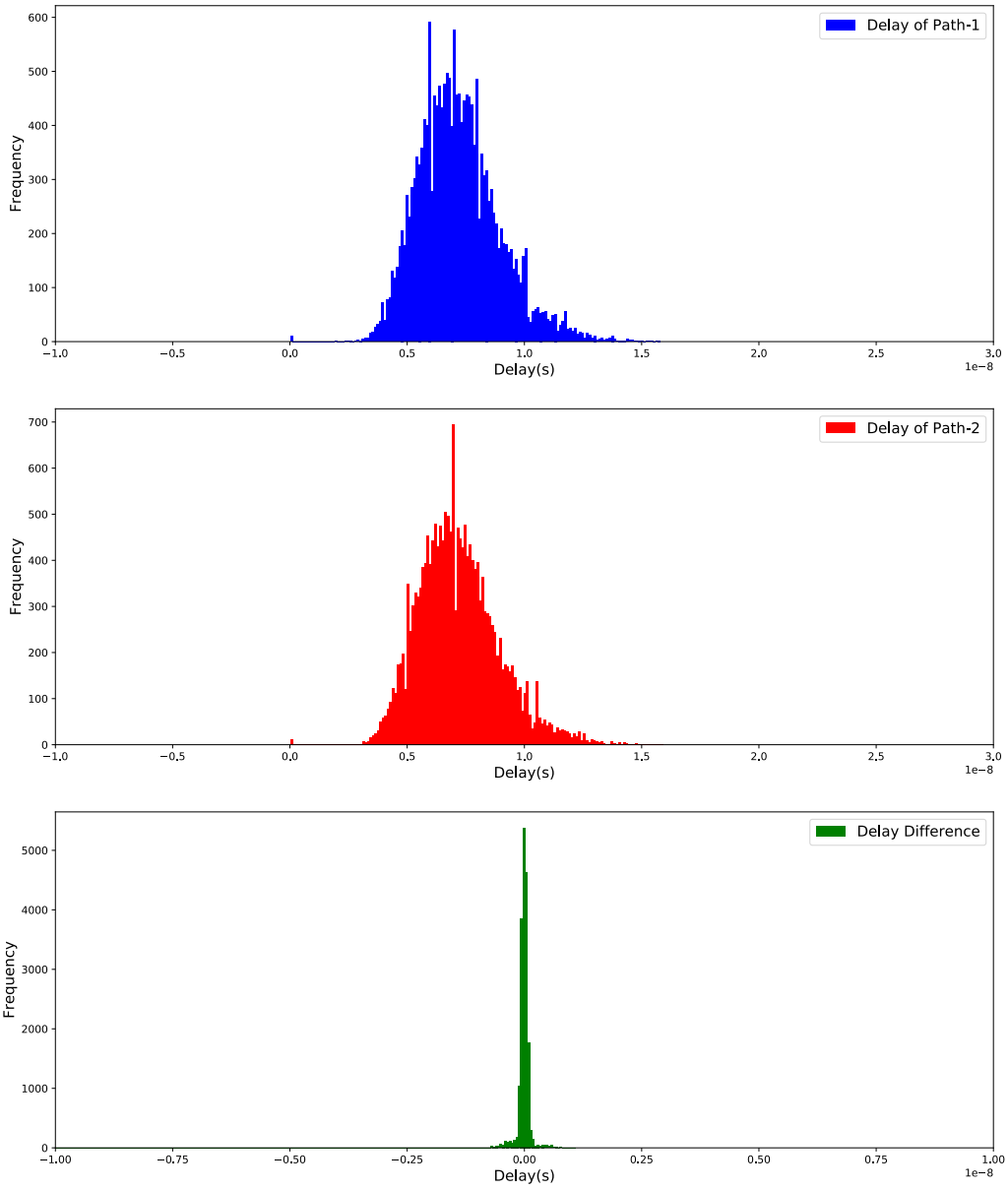


Figure 4.14: Delay distributions obtained by 20000 Monte Carlo simulations for 16-bit arbiter PUF: blue plot is distributions for Path-1 delays; red plot is distributions for Path-2 delays; green plot is distribution for delay difference between Path-1 and path-2.

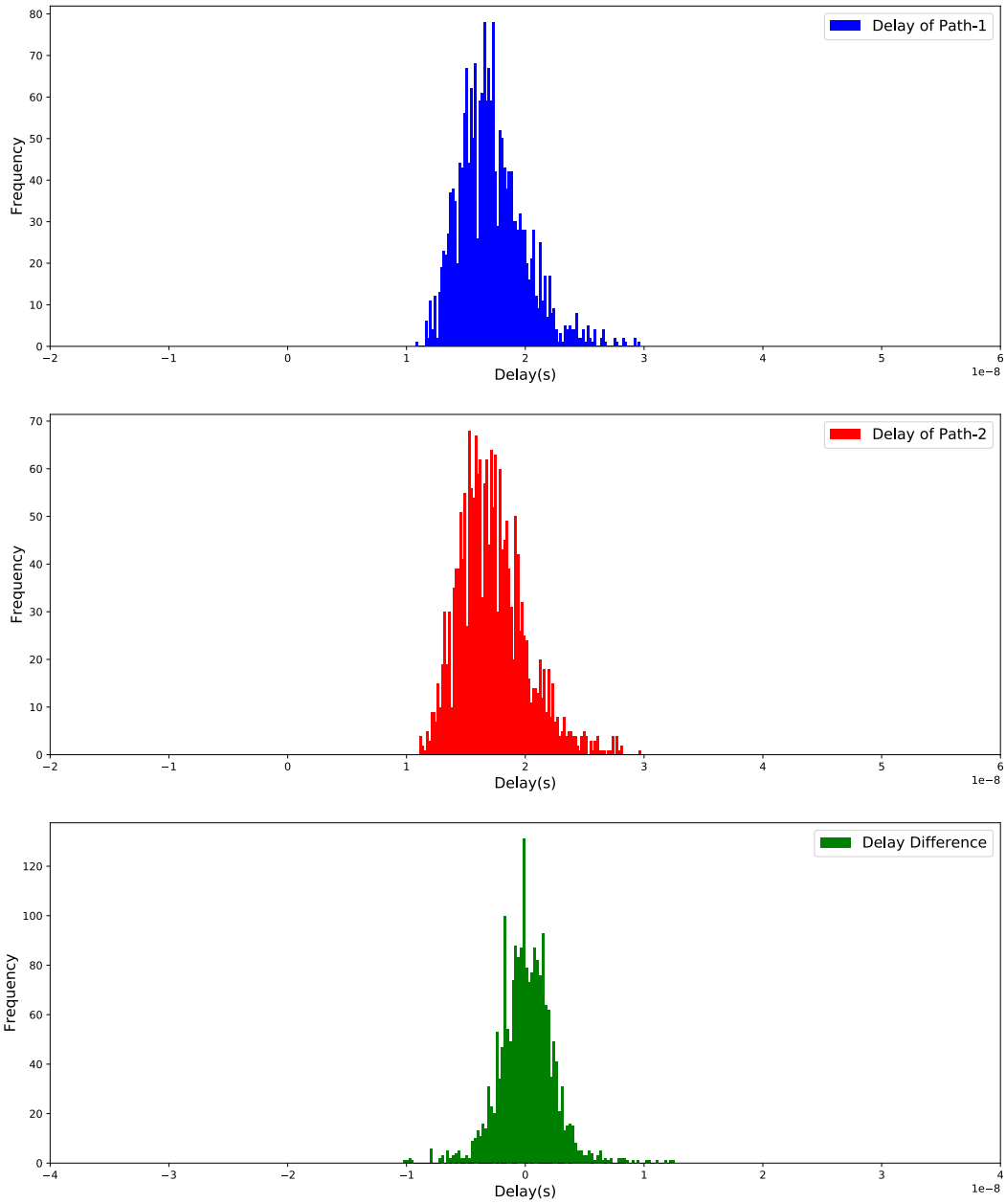


Figure 4.15: Delay distributions obtained by 2000 Monte Carlo simulations for 32-bit arbiter PUF: blue plot is distributions for Path-1 delays; red plot is distributions for Path-2 delays; green plot is distribution for delay difference between Path-1 and path-2.

4.3.3.1 Hardware Overhead

Table 4.2 shows the hardware overhead comparison for proposed arbiter PUF architecture and existing memristor-based PUFs [11, 119–121]. Table 4.2 is originally extracted from [11], which exclude the timing and control circuit for all of the techniques. From the results, the hardware requirement in [119–121] are much higher compared to [11] and the proposed PUF design since a large number of the multiplexers and amplifiers are implemented. The main difference between the proposed PUF and [11] is that the proposed design incorporates fewer transistors but more memristors. Recent research in [122] demonstrates that the critical dimension of the memristor can be scaled down to 2nm. However, the latest transistors designed by Taiwan Semiconductor Manufacturing Company (TSMC) are reaching to 5nm which is almost close to the limit. Hence, using fewer transistors still implies less silicon area. Moreover, the memristors can be seamlessly integrated with transistors and stacked in 3D crossbar array to form a high density memristive architecture which can reduce the area consumption even more. Additionally, in Chapter 5, the author also proposes another technique to realise a light weight memristive PUF architecture.

4.3.3.2 Systematic Measurement of PUF Performance

To examine the effectiveness of the proposed design, we run 5000 points Monte Carlo simulations for the 8-bit and 16-bit PUFs and 2000 points simulations for the 32-bit PUF because of the storage limitation. Then, the equations of Eq. (2.21), Eq. (2.22) and Eq. (2.23) are applied to the simulation results. Table 4.4 and Table 4.3 show the performance proposed design of different sizes compared to existing techniques [11, 39]. Fig. 4.16, Fig. 4.17 and Fig. 4.18 illustrate the bit-aliasing values for all the bit positions. From Table 4.4 and Table 4.3, it is apparent that the proposed arbiter PUF reveals calculated values close to 50% which is ideal value.

Table 4.2: Hardware overhead comparison for memristor-based PUFs.

PUF Architecture	Ref. [119]	Ref. [120]	Ref. [121]	Ref. [11]	Proposed Arbiter PUF
Challenge size (bits)	n	n	n	n	n
Response size (bits)	r	r	r	r	r
Memristors	$n \times r$	$n \times r$	$2n$	$4n \times r$	$8n \times r$
MOSFET switches	–	–	–	$4n \times r$	$2n \times r$
MUXes(2:1)	$3n + r$	–	$3n$	r	r
Resistors	–	–	$2n$	–	$2n \times r$
Amplifiers	r	r	–	–	–
Inverters	–	–	$2n$	$2n \times r$	$2n \times r$
n-bit decoder	–	1	–	–	–
Flip-flops	–	–	–	$2 \times r$	$2 \times r$
Capacitors	–	–	–	–	–
Comparator	–	–	–	–	–

Table 4.3: Proposed 8-bit, 16-bit & 32-bit memristive Arbiter PUF (APUF) Performance (II).

	Uniqueness			**Uniformity**		
	Ref. [11]	Ref. [39]	Proposed APUF	Ref. [11]	Ref. [39]	Proposed APUF
8-bit	49.57%	–	49.94%	47.27%	–	50.59%
16-bit	49.81%	50.04%	49.86%	53.80%	52.3%	50.03%
32-bit	50.06%	50.04%	49.9%	50.10%	50.7%	50.01%
Ideal Value	50%	50%	50%	50%	50%	50%

Table 4.4: proposed 8-bit, 16-bit & 32-bit memristive Arbiter PUF (APUF) Performance.

	Bit-aliasing			**Reliability (Voltage)**		
	Ref. [11]	Ref. [39]	Proposed APUF	Ref. [11]	Ref. [39]	Proposed APUF
8-bit	52.35%	–	50.06%	–	–	99.92%
16-bit	53.40%	50.70%	50.76%	–	96%	99.99%
32-bit	51.40%	49.30%	50.10%	–	97.75%	99.96%
Ideal Value	50%	50%	50%	100%	100%	100%

For reliability characterisation, the hamming distance between the environmental variations of the same PUF is measured as interpreted in Eq. (2.24). Owing to the fact that the temperature parameter is not be specified in the memristor model, we only consider the voltage variations in this thesis. Here, we assess the reliability measurements under the condition of the supply voltage variation. The nominal value of the supply voltage is assumed to be 1V, and the voltage variation is $\pm 15\%$. From Table 4.3, the results demonstrate that the voltage reliability value of the proposed PUF is close to the ideal value. In addition, we also compare the proposed arbiter PUF with some CMOS PUFs. For a fair comparison, we chose a similar PUF architecture - CMOS based arbiter PUF [62] as it evaluated the performance by applying the same measurement technique. Table 4.5 demonstrates that the conventional CMOS-based arbiter PUF manifestly fall behind the proposed memristor-based arbiter PUF in terms of uniqueness, uniformity as well as bit-aliasing.

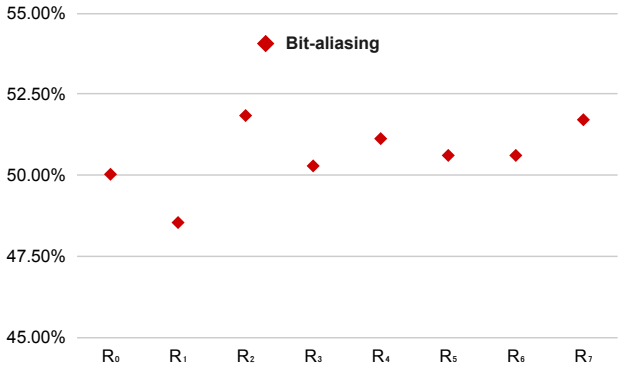


Figure 4.16: The bit-aliasing values for all the bit positions of proposed 8-bit PUF.

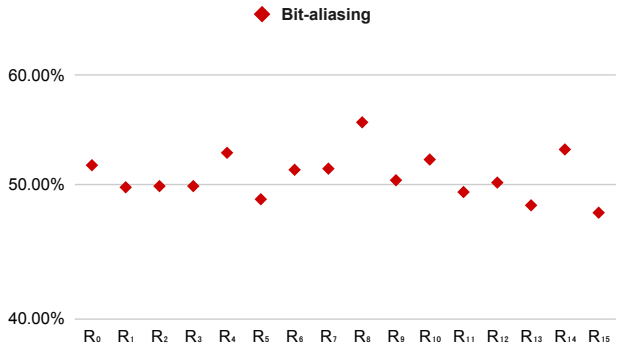


Figure 4.17: The bit-aliasing values for all the bit positions of proposed 16-bit PUF.

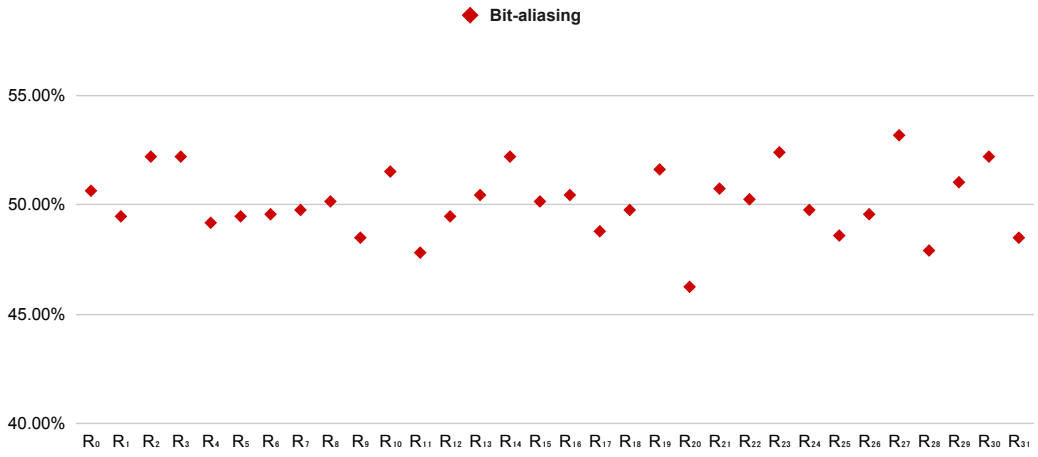


Figure 4.18: The bit-aliasing values for all the bit positions of proposed 32-bit PUF.

Table 4.5: Performance comparison of CMOS arbiter PUF (APUF) [62] and the proposed arbiter PUF

	Uniqueness	**Uniformity**	**Bit-aliasing**	**Reliability**
CMOS APUF	7.2%	55.69%	19.5%	99.76%
Proposed APUF	49.9%	50.01%	50.10%	99.96%
Ideal Value	50%	50%	50%	100%

4.4 Summary

This chapter presented the memristor parasitic effects on both the existing and the proposed memristive logic architecture. The parasitic components L_p and C_p are considered with the pure memristor model to form the RLC circuit, which generates the decaying oscillation especially when the input logic state changes simultaneously. Most of these parasitic effects are being canceled out by our 1T-4M XOR structure as the voltage difference is obtained from the output. In this chapter, we also demonstrated that our 3T-4M buffered XOR gate produces more variation and propagation delay with C_p and L_p. These can be taken as benefits for creating a PUF with high degree of randomness. Hence, we proposed a delay based memristive hybrid arbiter PUF by utilising 3T-4M buffered memristive XOR architecture as a fundamental component. Because that the parasitic components C_p and L_p of the memristor leverage the degree of the randomness. The measured parameters including uniqueness, uniformity, bit-aliasing and reliability of the proposed arbiter PUF demonstrated values close to the ideal values. The experimental results also reveal that the proposed design performs better than the conventional CMOS based arbiter PUF.

However, the general problem of the arbiter PUF is the low resistance to machine learning attacks. The response is determined by the delay difference which is accumulated from the propagation delay of each stage of the PUF. Hence the challenges and the responses are showing a certain linear relationship which is vulnerable to machine learning based modeling attacks. This problem is discussed and addressed

in Chapter 5. We also propose a low overhead and highly effective PUF architecture that is much more immune to these modeling attacks in that chapter.

5

Nonlinear Encoding/Decoding and its Application in Physical Unclonability

5.1 Introduction

As we mentioned earlier in Chapter 2, there has been significant interest in exploiting memristor in the design of high density non-volatile memory, neuromorphic systems, logic design, and most recently in sensors and solar cells [123, 124]. There are several existing techniques for tuning a memristor's conductivity to a predetermined value [123, 125]. Most of these techniques require complex circuitry, while some require external processing and others are unable to perform well for devices with high OFF- to ON-resistance ratios. To this end, we propose an accurate and efficient lightweight architecture for replicating the resistance of a source memristor into a destination memristor by repeatedly applying programming pulses, but without much of the drawbacks of the existing techniques. Such a circuit can be crucial for backing up analogue data, e.g. from a memristor sensor, before or during conversion to digital form. We show that the proposed architecture is extremely versatile and can be used not only for replicating memristors, but also for generating non-linear digital code and decoding the code back to the source memristance/voltage (within quantisation limits). Owing to this non-linear encoding, the architecture also provides a certain level of inherent security features. Our experimental results also demonstrate that it offers physical uncloneability, which can be critical in PUF applications.

The main criticism and drawback of the existing PUFs, such as delay based arbiter PUFs, is that the mapping from challenges to responses shows a certain degree of linearity, which complex machine learning algorithms can figure out. For example, for 64-bit arbiter PUFs machine learning can achieve 99% prediction rate. Additionally, for these architectures many CRPs lack uniqueness. To mitigate these usually a large number of challenge/response bits with many stages of iterative hardware, at the cost of significant overhead and power consumption, is required for a proper PUF. In contrast, we propose a novel architecture that leverages the non-linear behaviour of

the memristors for realising high performance low overhead PUFs. It is based on a low overhead memristive digital to analogue conversion (DAC) circuit in conjunction with a memristive non-linear encoder. The idea is to use the DAC circuit to non-linearly generate unique analogue values from different digital challenges and then use a memristive non-linear encoder (MNE) to non-linearly convert these analogues values back to unique digital codes (responses), which requires much lower overhead and provides attack resistance. Hence, the proposed PUF architecture can be very useful in applications such as chip tagging/identification as well as for preventing unauthorised fabrications and authentication, e.g. via CRPs.

5.2 A Highly Versatile Architecture for Replications and Encoding/Decoding

In this section, we first demonstrate a lightweight and highly versatile architecture for replicating the resistance of a source memristor into a destination memristor. The architecture is also able to produce non-linear digital codes during the replication process for added security by taking advantage of the non-linear behaviour of memristors. The generated codes can also be used to retrieve the analogue value within acceptable conversion errors, with circuit elements already built into the replicator. Finally, we show that the architecture demonstrates physical unclonable properties.

5.2.1 Read/Write Non-linear Memristive Devices

Most fabricated memristive devices manifest non-linear behaviour since their resistance depends on a tunnelling effect. In other words, any change in the tunnel barrier width alters its memristance exponentially. As we discussed in Chapter 2, the instantaneous resistance, R_{M}, is given by Eq. (2.15) . R_{M} shifts non-linearly towards R_{off} when a write voltage $V_{\mathrm{write}} > V_{\mathrm{off}}$ is applied across a memristor and it shifts

non-linearly towards R_{on} when $V_{\text{write}} < V_{\text{on}}$ is applied across it, where V_{off} and V_{on} are the threshold voltages. The state of the memristor remains unchanged for any voltage applied in between V_{on} and V_{off}. Hence, the state of a memristor can be read by applying a read-voltage, V_{read}, such that $V_{\text{on}} \leq V_{\text{read}} \leq V_{\text{off}}$ and $V_{\text{read}} \neq 0$.

5.2.2 Proposed Architecture

Fig. 5.1 shows our proposed architecture for replicating a source memristor M_{S} to a destination memristor M_{D} within certain quantisation limits. M_{S}, M_{D} and the load resistors, R_{SL} and R_{DL}, form two voltage dividers. R_{SL} and R_{DL} are assumed to be closely matched. The output of the voltage dividers V_{INS} and V_{IND} are continually compared by the comparator.

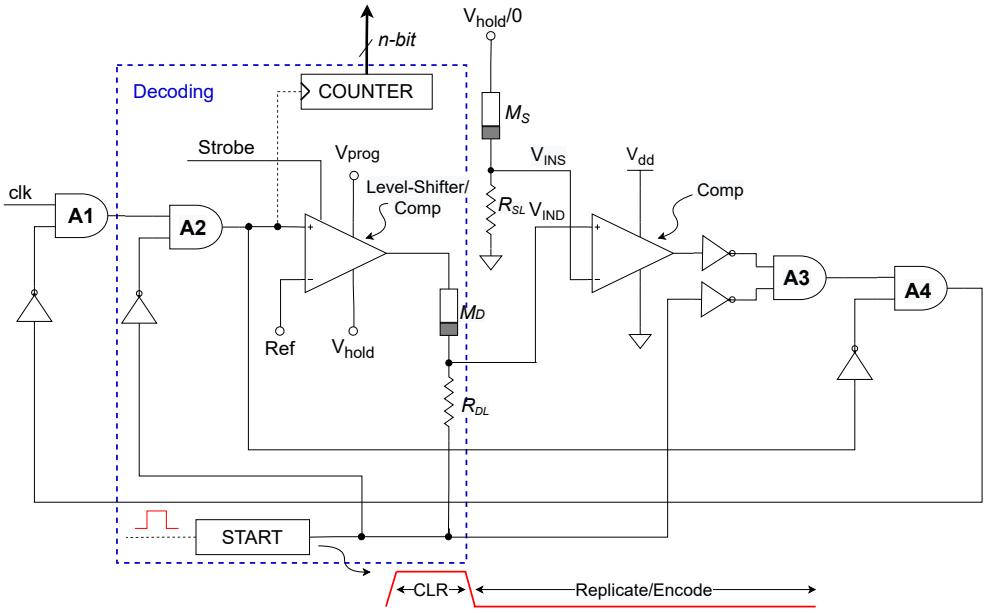


Figure 5.1: Proposed memristive replicator.

Replication and Encoding: Let f_{rep} be the replication frequency of the clock (clk) and the clock cycle can be written as below:

$$T_{\text{rep}} = T_{\text{prog}} + T_{\text{hold}} = 1/f_{\text{rep}}. \tag{5.1}$$

Voltages V_{prog} and V_{hold} are alternatively applied during T_{prog} and T_{hold} as shown in Fig. 5.2. V_{prog} is adjusted to be sufficiently high so that V_{write} appears across M_{D} only even with the load R_{DL}. Similarly, V_{hold} is adjusted to be sufficiently high such that a V_{read} appears across both M_{S} and M_{D} simultaneously. Hence resistance of M_{S} is copied to M_{D} by repeatedly applying these pulses based on Eq. (2.18). During each T_{prog}, V_{prog} shifts the barrier of M_{D} from the R_{on} region towards the R_{off} region by a small amount, but non-linearly, and during the following T_{hold}, M_{D} and M_{S} are compared. The counter at the top counts the number of clock pulses required to replicate. The replication starts by first resetting the counter and a 'CLR' pulse

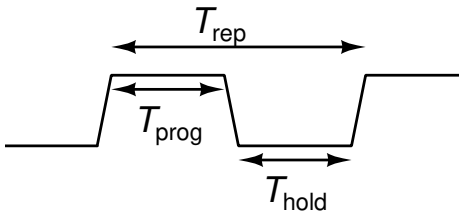


Figure 5.2: Programme pulses for the replicating and encoding operations are generated by level-shifter.

applied to $M_{\text{D}}$from START. This pulse is of sufficient amplitude such that a very short duration resets M_{D} to R_{on}. The first stage of the counter is used to generate the pulse and once CLR is complete the counter is reused for encoding. During CLR: the level-shifter is disabled with the strobe input (Here, the strobe signal is synchronised with 'CLR' pulse); AND gates A_2 and A_3 block the inputs and produce a constant zero. Hence, clk is prevented from reaching the level shifter. As a result, the level-shifter produces 0 at the N-terminal of M_{D} and the high CLR pulse at its P-terminal resets

M_D to R_{on}.

When CLR returns to 0, the level-shifter and the AND gates A_2 and A_3 are enabled and the replication starts. A_2 passes clk to the level-shifter which switches between V_{prog} and V_{hold} in the same cycle (T_{rep}). V_{prog} shifts the barrier of M_D during T_{prog}, and the resulting voltage is compared with that across M_S during T_{hold} by the comparator. During T_{hold} the comparator keeps producing a 1 until the voltage across M_D exceeds M_S. This forces A_3 and A_4 to produce a 0, which lets A_1 pass clk through to the level-shifter. As a result the voltage across M_D gradually increases during each clock cycle, until it exceeds that across M_S at which point the comparator produces a 0 during T_{hold}. This forces A_3 and A_4 to produce a 1 and the level-shifter is disabled via the strobe. This stops A_1 from letting clk through thereby indicating an end of replication. Essentially the circuit enters a 'locked' state which is controlled by the voltage across M_D. During replication, the counter counts the number of clock cycles required to replicate, which is the encoded digital value for analogue voltage (resistance) of M_S. Hence, the proposed architecture performs non-linear encoding of the analogue voltage/resistance while the replication takes place. It is assumed that V_{INS} remains steady during the conversion process. However, if V_{INS} increases during the conversion, then V_{IND} will follow V_{INS}. If this is undesirable then adding a single latch between A_4 and A_1 will ensure that the conversion stops the first time V_{IND} matches V_{INS}. The whole replicating process is demonstrated in Fig. 5.3.

After this, a second phase of replication can start by the CLR pulse, which resets M_D thereby bringing the system out of the locked state. The accuracy depends on the width of T_{prog} for a fixed V_{prog}, and lower T_{prog} results in higher accuracy, but at the cost of increased replication/conversion time. Table 5.1 shows the results as M_S was varied from 10KΩ to 90KΩ. Clearly, the encoded value is non-linear in nature and maintains low percentage error in copying the resistance to the destination memristor M_D. Additionally, this code can also be used for reproducing the analogue values

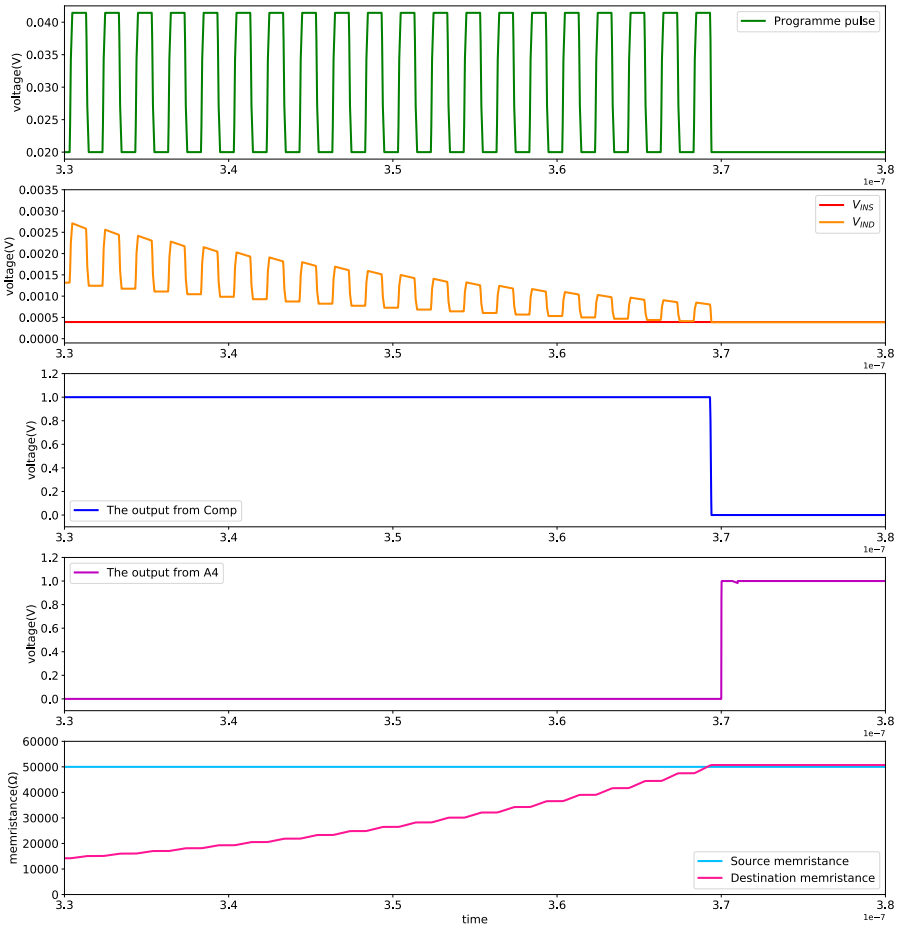


Figure 5.3: Replicating process when *CLR='0'*: As the programme pulses are applied to M_{D}, the voltage across M_{D} starts to increase until V_{IND} exceeds V_{INS}, then the output of Comp becomes a 0. This forces the output from A4 to be 1, which stops the clk passing through A1. Therefore, the replicating process is finished. In this case, M_{S} is 50KΩ, the replication process is stopped when the M_{D} is higher than 50KΩ. (V_{prog}= 41mV, V_{hold}= 20mV, T_{prog}= 2.5ns, R_{SL}= R_{DL}= 1KΩ, R_{on}=500Ω, R_{off}=100KΩ.)

which will be shown in following paragraph, but within the non-linear quantisation limits.

Table 5.1: Replication/Encoding process at 200MHz ($R_{\text{on}} = 500\Omega$, $R_{\text{off}} = 100\text{K}\Omega$).

V_{prog}=41mV,V_{hold}=20mV,T_{prog} = 2.5ns,$R_{\text{SL}} = R_{\text{DL}} = 1\text{K}\Omega$				
M_{S} KΩ	M_{D} KΩ	Time μsec	% Err	Enc Val
10	10.0555	51.6685	0.555	10334
20	20.0525	57.2435	0.263	11449
30	30.051	60.3387	0.170	12068
40	40.0749	62.4987	0.187	12500
50	50.070	64.1536	0.140	12831
60	60.0617	65.4987	0.103	13100
70	70.0653	66.6337	0.0933	13327
80	80.0566	67.6136	0.0708	13523
90	90.068	68.4787	0.0756	13696

What needs to be mentioned for Fig. 5.1 is that the clock pulses reached to A4 may have propagation delays (depending on a specific manufacturer's datasheet) which can cause a glitch. In a simulation environment, this can be solved by adjusting the delay of each logic gate directly. However, we can add buffers or delay elements to synchronise the signals on a real circuit.

Decoding: Let us assume that while replicating, the counter registered digital value C. Decoding is achieved by first clearing M_{D} with CLR and by 'programming' it with the same V_{prog} and V_{hold} at the same frequency $f_{\text{rep}} = 1/(T_{\text{prog}} + T_{\text{hold}})$ for C number of cycles. A part of the decoder logic appears in the blue boundary in Fig. 5.1. Here, we show the basic structure for that decoding application in Fig. 5.4. A down counter, initialised to C, is used to count the number of clock cycles. After decoding,

$V_{\text{hold}} - V_{\text{DL}}$, where V_{DL} is the voltage drop across R_{DL}, divided by the current gives the corresponding encoded resistance within quantisation limits.

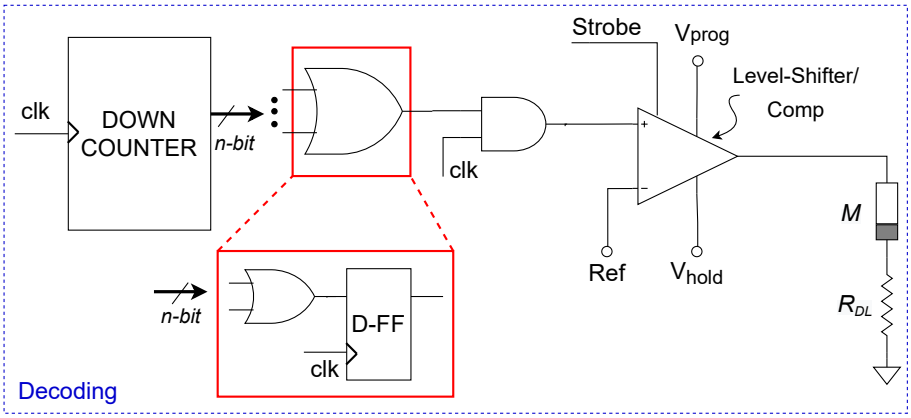


Figure 5.4: Proposed architecture for application on decoding. We use OR gate to combine values from a counter which may get glitches at the output of the OR gate. Eliminating glitches can be achieved by adding an extra D-FF after the OR gate as shown in red block. The input of the AND gate will be glitch free.

Security and Physical Uncloneability: The proposed architecture provides a certain level of inherent security by virtue of non-linear encoding. The encoded value C is a function of V_{write}, T_{prog}, and M_{D} itself. Hence, it is extremely challenging to guess what resistance or voltage C represents without having full knowledge of these quantities. Fig. 5.5, Fig. 5.6 and Fig. 5.7 show the results of varying V_{prog} and T_{prog} respectively while keeping the other parameters fixed. As we see the behaviour of the proposed architecture is non-linear through out, i.e. it is non-linear for specific V_{prog} or T_{prog} and also for their differences.

Additionally, access to an almost exact matching memristor to M_{D} provides further challenge and difficulty. Therefore, the architecture also provides physical unclonability [126, 127] by virtue of non-linearity as well as its sensitivity to process and parametric variations. As revealed by our experimental result, the non-linear code depends heavily on the physical parameters of M_{D}, e.g. D. Any small variations in

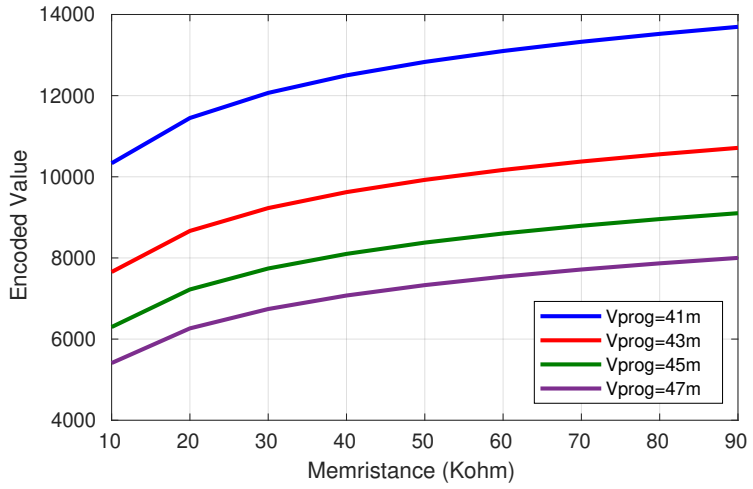


Figure 5.5: Effect of different V_{prog} on the encrypted encoded value vs resistance (V_{hold}=20mV, T_{prog}=2.5ns, R_{SL}=R_{DL}=1KΩ, R_{on}=500Ω, R_{off}=100KΩ and D=3n).

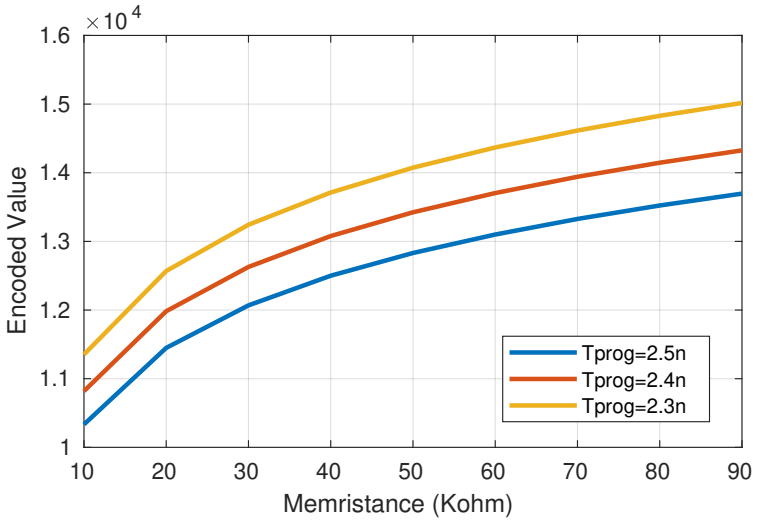


Figure 5.6: Effect of different T_{prog} on the encrypted encoded value vs resistance (V_{prog}=41mV, V_{hold}=20mV, R_{SL}=R_{DL}=1KΩ, R_{on}=500Ω, R_{off}=100KΩ and D=3n).

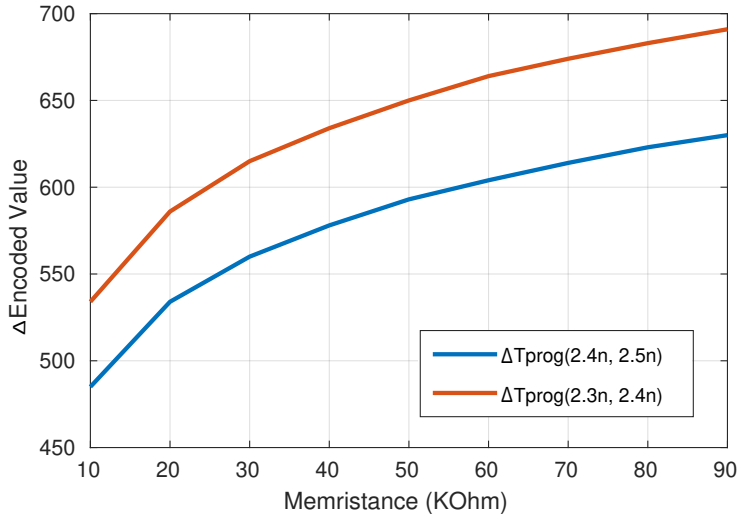


Figure 5.7: Effect of different T_{prog} on the encrypted encoded value vs resistance (V_{prog}=41mV, V_{hold}=20mV, R_{SL}=R_{DL}=1KΩ, R_{on}=500Ω, R_{off}=100KΩ and D=3n).

these are amplified by the counting based encoding mechanism and results in different codes (Fig. 5.8). Hence, any two fabricated chips are likely to produce different codes for the same input voltage/resistance thereby making it very hard to clone.

While this architecture is geared towards replicating memristors, it can also be used for non-linear encoding and for challenge-responses-pair (CRP) based authentication. Instead of M_{S}, in Fig. 5.1 V_{INS} can be an analogue input voltage serving as challenge. After encoding, the contents of the counter can serve as a unique non-linear response. Furthermore, this response will vary from chip-to-chip owing to its physical unclone-ability thereby also offering provisions for chip identification. An analogue voltage at V_{INS} can be obtained from non-linear digital to analogue conversion circuit.

5.3 Proposed PUF Architecture

In this section, we show that the proposed architecture of Fig. 5.1 finds an application in challenge-response-pair based authentication by designing a novel memristive PUF.

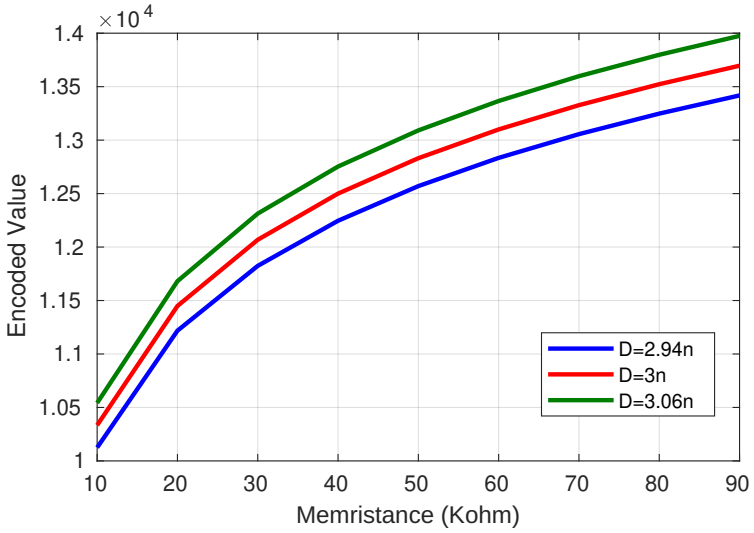


Figure 5.8: Shows the variations in encoded digital output with varying resistance under process/parametric variations. Here the process parameter D was varied by $\pm 2\%$ (V_{prog}=41mV, V_{hold}=20mV, T_{prog}=2.5n, R_{SL}=R_{DL}=1KΩ, R_{on}=500Ω, and R_{off}=100KΩ).

The block diagram of the proposed PUF architecture is shown in Fig. 5.9. The general idea of the architecture is to convert the $n-bit$ digital challenges to analogue signals (DAC), add non-linearity to it, and then convert the result back to $n-bit$ digital responses (ADC). Here, the different chips are assumed to produce different responses for the same challenges owing to process variability. We achieve this with a non-linear memristive digital to analogue conversion circuit in conjunction with Fig. 5.1. We show that, due to unpredictable non-linearity and process variability, the proposed architecture provides unpredictable CRPs as well as good PUF.

5.3.1 Linear/Non-linear Analogue to Digital Conversion

We firstly show the process of analogue to digital conversion. Once the challenges are converted to analogue voltages, we use an encoder to convert the analogue signals back to digital responses. We tested with both Linear encoder as well as with the proposed non-linear encoder and compared the results.

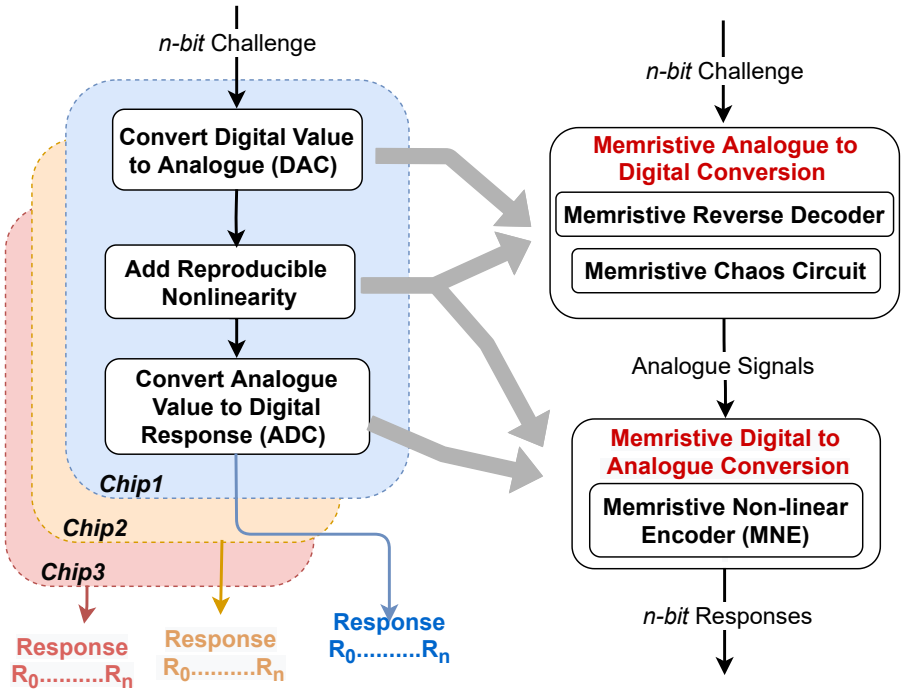


Figure 5.9: Block digram of the proposed architecture.

5.3.1.1 Memristive Non-linear Encoder (MNE)

In this section, we mainly show that the architecture of Fig. 5.1 can be implemented as a memristive non-linear encoder (MNE) to operate the analogue to digital conversion.

Instead of M_{S}, an analogue input voltage connect to V_a as shown in Fig. 5.10. As previously discussed, a memristor can be programmed (tuned) by applying a programming voltage V_{prog} and can be read by applying a hold voltage V_{hold}. The amplitude of the programming voltage V_{prog} and its pulse width T_{prog} determines the shifting of the barrier within the memristor. The process is similar to the replicating process. Let f_{enc} be the encoding frequency of the clock (clk) and $T_{\mathrm{enc}} = T_{\mathrm{prog}} + T_{\mathrm{hold}} = 1/f_{\mathrm{enc}}$ be the clock cycle. Voltages V_{prog} and V_{hold} are alternatively applied during T_{prog} and T_{hold} respectively. During each T_{prog}, V_{prog} shifts the barrier of the memristor from the R_{off} region towards the R_{on} region by a small amount, but non-linearly, and during the following T_{hold}, V_b and V_a are compared. The barrier of the memristor can also be shifted from R_{on} to R_{off} except the fact that the polarity of the memristor needs to be switched (See red boundary of Fig. 5.10). Meanwhile, an m-bit counter counts the number of programming pulses. The programming pulses represent the encoded value, which is non linear, corresponding to V_a. As revealed by Fig. 5.11(a), the encoded value depend heavily on the physical parameters of a memristor such as its width, D. Any small variations in these are amplified by the MNE mechanism and results in variations in responses for the same analogue voltage V_a. The control circuit which includes AND gates and level-shifter disables the programming pulses when V_b exceeds V_a and it is the same as its use of replication.

The counter, and hence the encoded value, can be of any number of bits. For example, a 1-bit counter flips between 0 and 1 during the encoding process and provides a 1-bit encoding of V_a. This outcome depends on the non-linear shifting of the barrier within the device. This flexibility allows us to segment an n-bit response into k smaller bit responses and then combine the results.

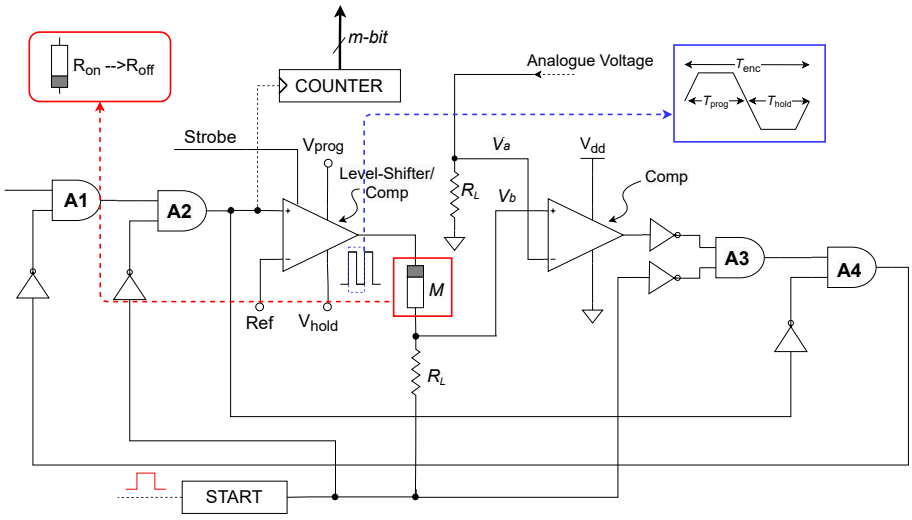


Figure 5.10: Proposed memristive non-linear encoder (MNE).

For the sake of simplicity, in the following discussion, we use the circuit in the blue boundary of Fig. 5.12 to represent the MNE.

5.3.1.2 Linear Encoder (LE)

To compare results with the MNE, we also implement a simple ideal linear encoder (LE) to process the analogue to digital conversion. There is a large body of published work on linear encoder [128, 129]. However, in this thesis, we used the following algorithm to convert analogue values to digital codes. The algorithm is very simple because we only need to observe the resistance of the machine learning attacks under the linear association between the analogue voltages and the digital coders. Let DV represent an array of s analogue values, and EV be an empty array to store the digital values.

1. Let MaxV= maximum value of the decoded voltage. Hence, this is equivalent to the reference voltage in a DAC.

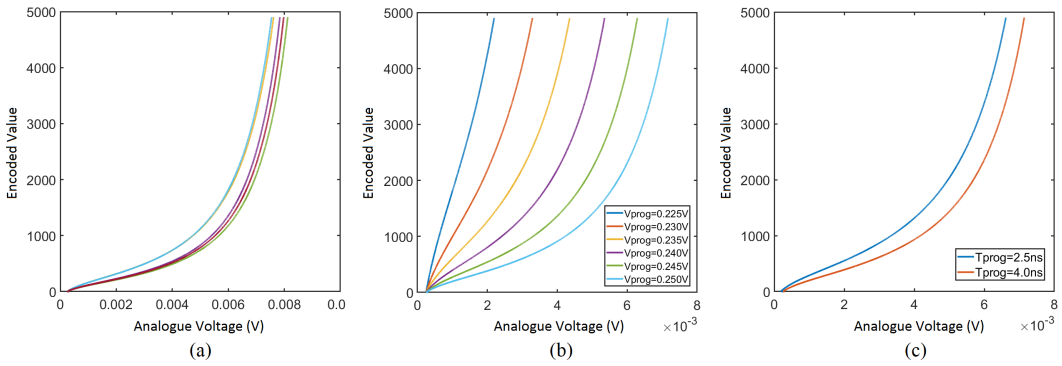


Figure 5.11: Effects of varying (a) process parameter D by 5%; (b) V_{prog}; (c) T_{prog}, while all other parameters are fixed.

2. $\forall i \in \{1, 2, \ldots, s\}, \text{EV}[i] = \lfloor \text{DV}[i] \times 2^m / \text{MaxV} \rfloor$. Here, m is the number of the binary bits.

The digital values stored in EV are treated as the final responses, which are compared with the responses generated from the proposed MNE. The experimental results are illustrated in Table 5.6 and Table 5.7 and will be discussed shortly.

5.3.2 Non-linear Digital to Analogue Conversion

The previous section presents the mechanism of the analogue to digital (response) conversion. Hence, in this section, we show the conversion from the digital inputs (challenges) to analogue voltages. Two different conversion circuits are demonstrated here and will be implemented with the proposed MNE respectively.

5.3.2.1 Memristive Reverse Decoder

Section 5.2.1 presents the use of programme pulses to implement Read/Write operations. Assuming two programme pulses with different amplitude, V_{Prog1} and V_{Prog2} are alternatively applied to a memristor which cause the non-linear movements of the barrier and result in a new instantaneous memristance. Based on this fact, we

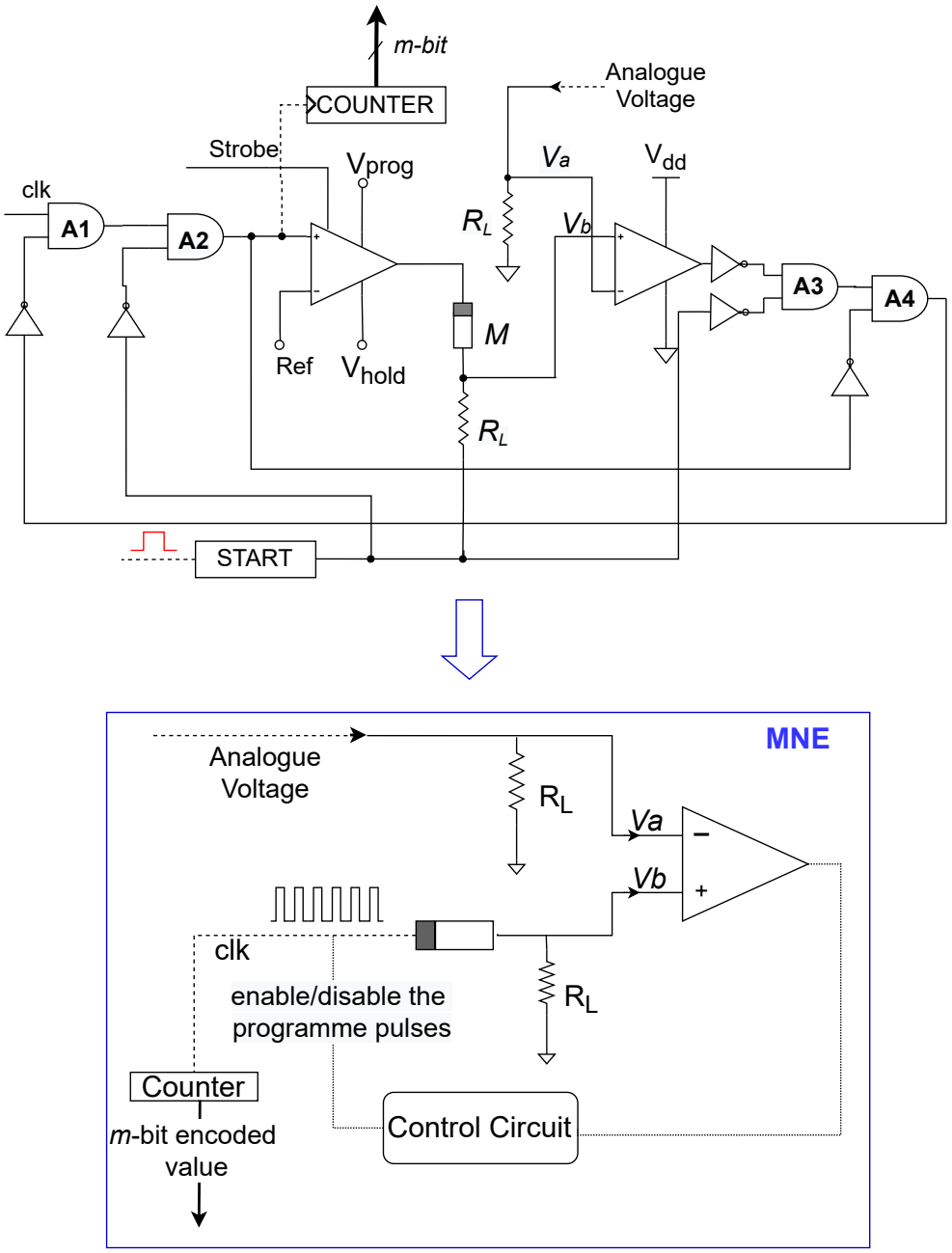


Figure 5.12: Simplified circuit block for the proposed memristive non-linear encoder (MNE).

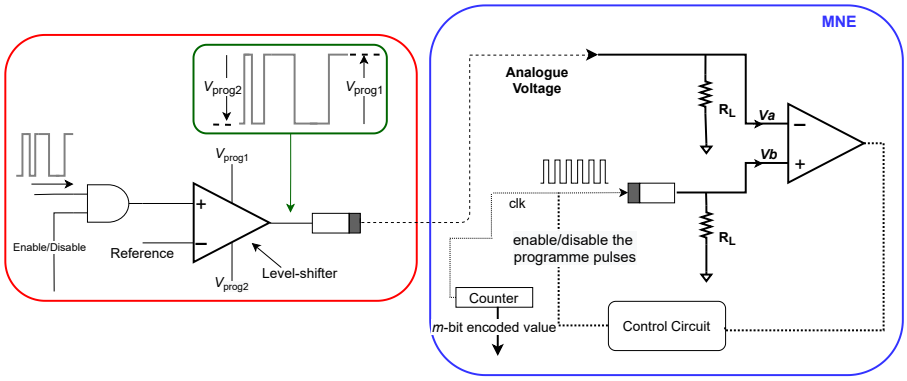


Figure 5.13: The memristive reverse decoder with MNE.

can have a simple circuit to convert the digital inputs to a corresponding analogue information. The basic structure of the circuit demonstrates in the red boundary in Fig. 5.13. Let us assume that a series of binary digits are applied to the AND gate. Meanwhile, an enable signal allows the AND gate pass the binary digits to the level-shifter which switches '1' to V_{Prog1} and '0' to V_{Prog2}. The barrier of the memristor shifts a certain amount from R_{on} towards R_{off} accordingly, thereby providing a different memristance with a new voltage appearing at V_a. Here, we convert a series of binary digits into a corresponding analogue voltage, V_a, but in a non-linear way. Due to asymmetric behaviour of the memristor, V_a can be encode to a different binary digits by simply switching the memristor polarity in MNE. For this reason, this circuit is called a memristive reverse decoder.

5.3.2.2 Memristive Chaotic Decoder

A memristive chaos circuit [130, 131] * is used for coverting an n-bit digital challenge to a unique analogue value. The structure of the chaos circuit is shown in Fig. 5.14 (a). The original chaos circuit constitutes a linear resistor, an inductor, two capacitors,

*The memristive chaos circuit has already been developed. Hence, this thesis directly applied the memristive chaos circuit in conjunction with the Memristive Non-linear Encoder (MNE) to develop a new memristive PUF.

and a nonlinear memristor. Compared to the sizes of all other devices in an integrated circuit, an inductor requires significant area. Owing to this fact on [132], two op-amps can be use to replace the inductor as shown by the equivalent circuit in Fig. 5.14 (b). To realise the chaotic behaviour, we simulate the circuit by applying the random bits in series. The chaotic signals of the circuit are shown in Fig. 5.15. Meanwhile, Fig. 5.16 and Fig. 5.17 demonstrate the double scroll attractor in different dimensions.

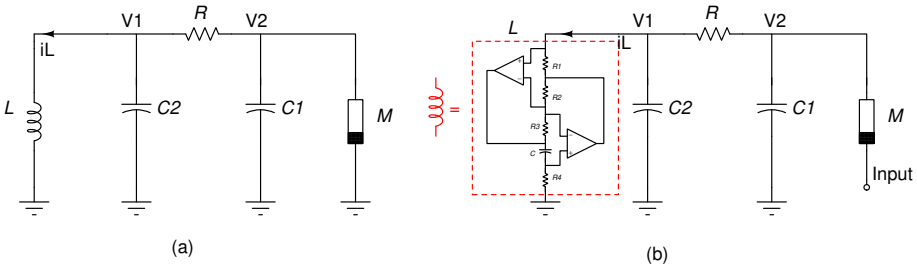


Figure 5.14: nonlinear memristive chaotic circuit.

The idea of adding non-linear memristive chaotic property is to 'pre-scramble' the challenge, but in a reproducible way[†] as long as the initial conditions are satisfied. Any changes in the input alters the behaviour of the circuit completely and this change can be accumulated by applying the challenge bits in series. Hence, the chaotic analogue signal produced by the circuit has much less predictable relationship with the input challenges. This makes it much harder for machine to learn the co-relation between the inputs and the analogue voltage and thus makes it difficult to mount a machine learning based attacks on the CRPs.

5.3.3 Putting It All Together

Inline with Fig. 5.9, Fig. 5.18 shows the proposed architecture for CRP based authentication. It consists of (i) a memristive non-linear digital to analogue converter which

[†]In this thesis, we use the chaotic electronic circuit [130, 131] to emulate chaos behaviour. [133] shows that by concentrating on a particular set of parameters the behaviour of this chaotic circuit can be reproduced by simulation and experiment.

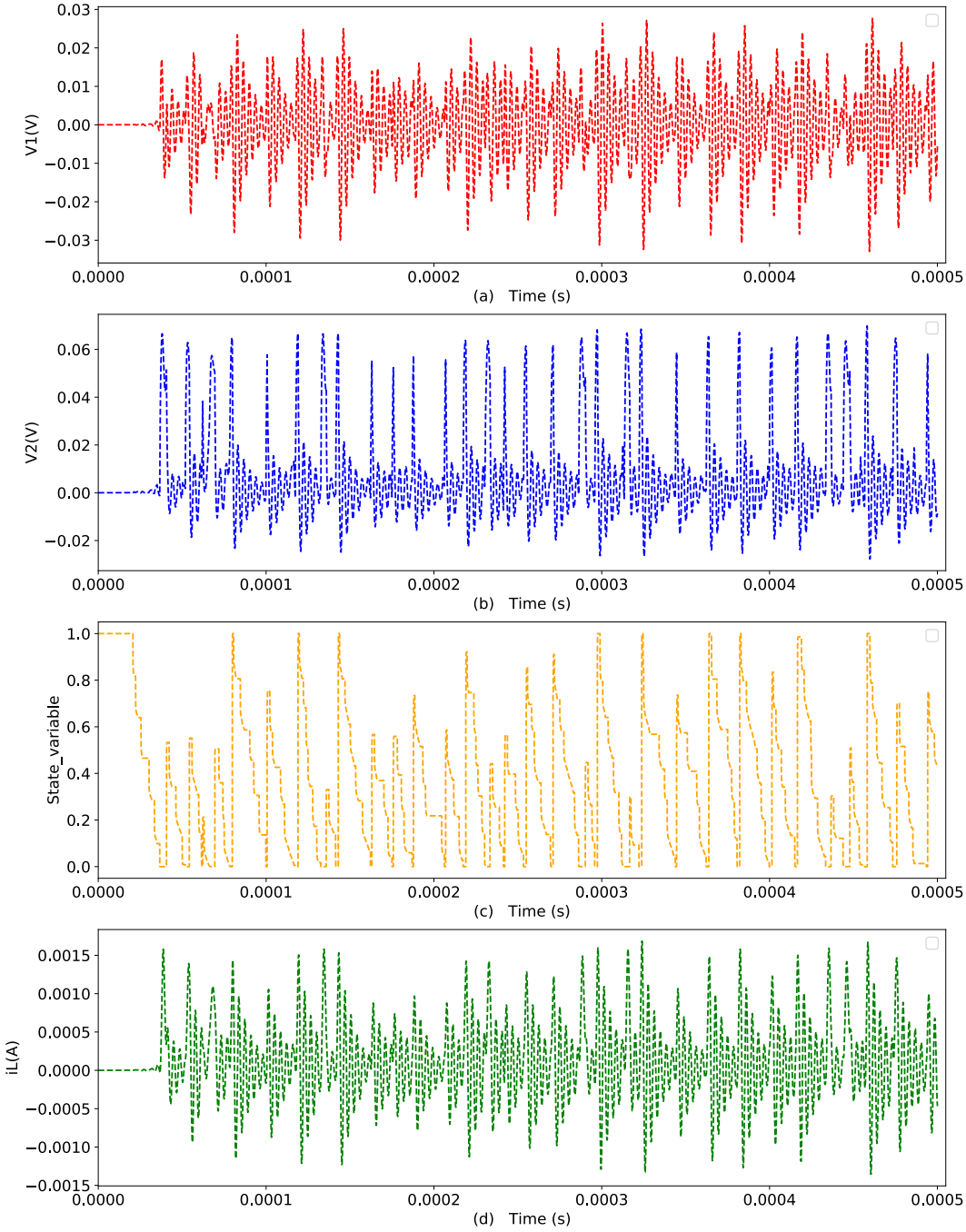


Figure 5.15: The chaotic signals: (a) The chaotic signal of V1; (b) The chaotic signal of V2; (c)The state variable of the memristor; (d) The chaotic signal of iL. (The parameters of the circuits: $R = 60\Omega$, $C1 = 5.75nF$, $C2 = 21.32nF$ and $L = (R1 \cdot R3 \cdot R4 \cdot C)/R2 = 12.5uH$)

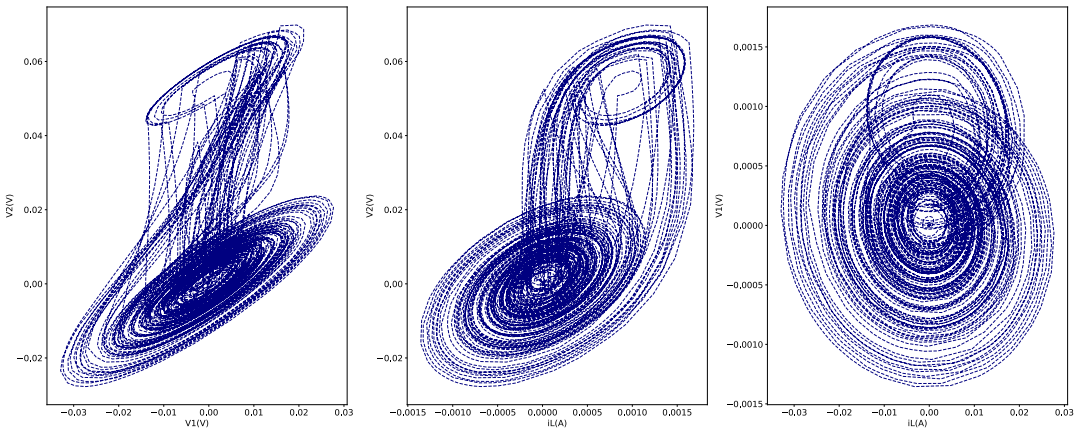


Figure 5.16: The double scroll attractor in dimension of V1-V2, iL-V2 and iL-V1 respectively. (The parameters of the circuits: $R = 60\Omega$, $C1 = 5.75nF$, $C2 = 21.32nF$ and $L = (R1 \cdot R3 \cdot R4 \cdot C)/R2 = 12.5uH$)

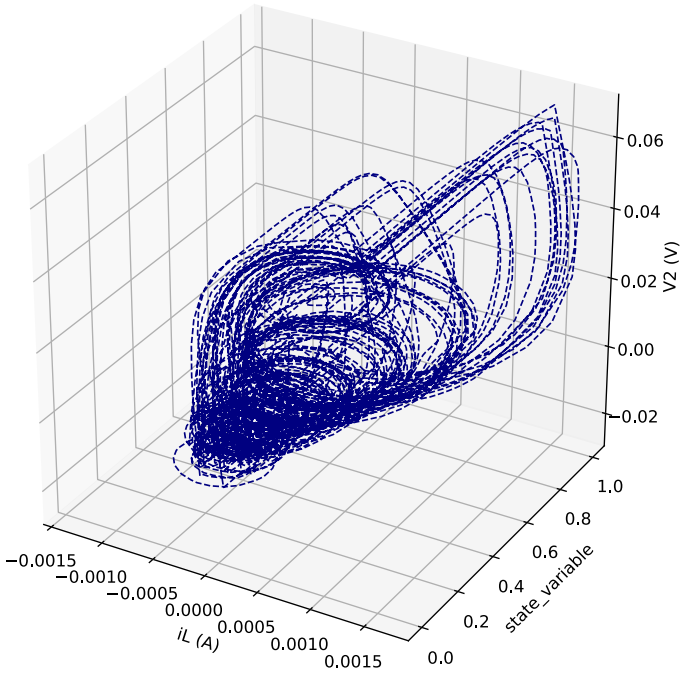


Figure 5.17: The chaotic behaviour in 3D. (The parameters of the circuits: $R = 60\Omega$, $C1 = 5.75nF$, $C2 = 21.32nF$ and $L = (R1 \cdot R3 \cdot R4 \cdot C)/R2 = 12.5uH$)

converts the n-bit challenges to an analogue signals and (ii) the proposed MNE which converts the analogue signals to m-bit unique digital responses.

In this thesis, we use two different ways to implement the analogue to digital conversion, which are the memristive reverse decoder and the memristive chaotic decoder. Then we join them with the proposed MNE respectively as shown as follows:

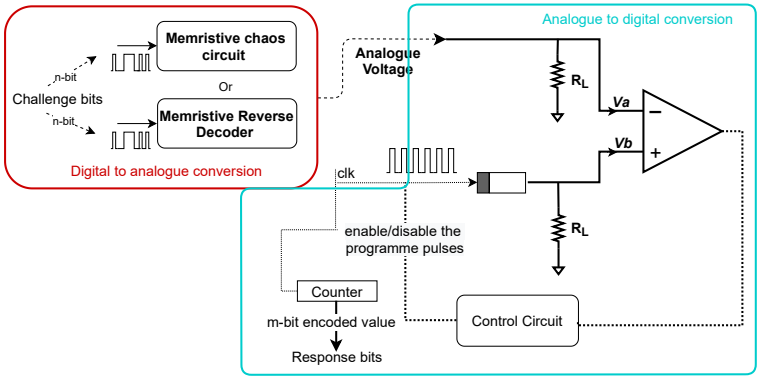


Figure 5.18: Proposed architecture for CRP based authentication.

- Fig. 5.19 shows the proposed memristive PUF which consists of the non-linear memristive reverse decoder and the proposed MNE. Therefore, in this thesis, we name it as the memristive PUF with reverse decoder (RD-PUF).

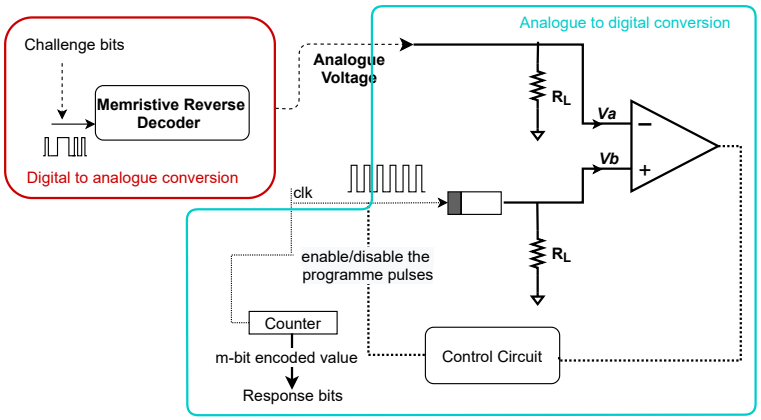


Figure 5.19: Proposed memristive PUF with reverse decoder (RD-PUF).

- Fig. 5.20 illustrates the other proposed memristive PUF architecture, which consists of a memristive chaos circuit and the proposed MNE. Here, we name it as the memristive PUF with chaos circuit (CHAOS-PUF).

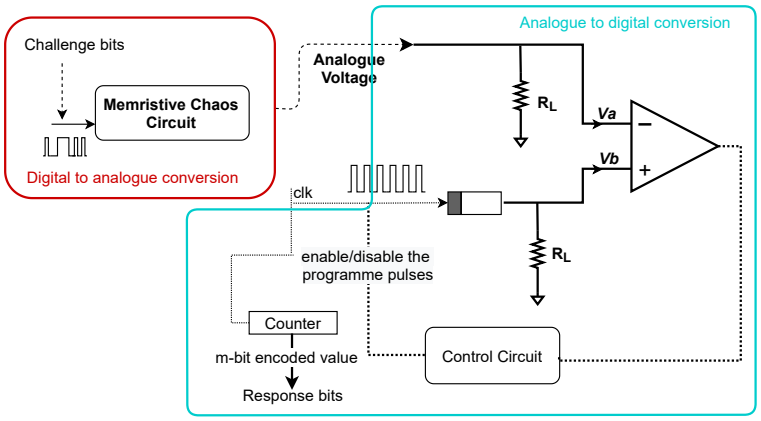


Figure 5.20: Proposed memristive PUF with chaos circuit (CHAOS-PUF).

Both architectures including Fig. 5.19 and Fig. 5.20 generate unpredictably non-linear CRPs owing to the combined effects of the non-linear memristive digital to analogue conversion process (i.e., memristive reverse decoder and memristive chaotic decoder) and the MNE.

An MNE can provide physical unclonability by virtue of its sensitivity to process and parametric variations as shown in Fig. 5.11(a). Fig. 5.21 shows the proposed PUF architecture with multiple MNEs placed in parallel. These paralleled MNEs produce different m-bit responses for the same voltage (challenge). Each m-bit response is concatenated to generate a new unique $k \times m$ bit response. Dividing a single m-bit response into k smaller instances can be useful, e.g. when m is very large.

The proposed architecture also reduces the effects of quantisation error and improves uniqueness. As shown in Fig. 5.22, due to quantisation error some analogue values (challenges) produced by the chaos circuit are mapped to the same encoded value (response) in a single chip. Concatenating responses from different MNEs lowers

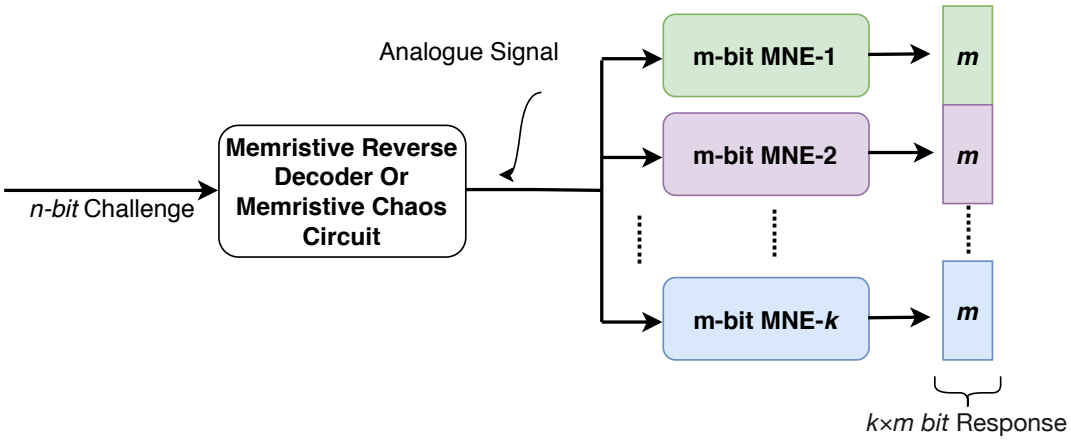


Figure 5.21: Proposed PUF architecture with multiple MNEs.

Table 5.2: The number of unique responses for the proposed PUF architecture with multiple MNEs.

Proposed 16-bit PUF with multiple MNEs (60000 CRPs)		
m**-bit response**	k **MNEs**	**the number of unique responses**
16	1	36381
8	2	39439
4	4	45268
2	8	49142
1	16	52003

this effect. This has also been indicated in Table 5.2. We tested our 16-bit PUF with different number of MNEs. The number of repeated responses significant decreases with the number of MNEs increase. Additionally, this effect can be further reduced by adjusting the initial resistance of the memristor from R_{off} to a specific resistance (the resistance value depends on the linearity of a particular memristor). This can be achieved by applying a certain amount of pulses. For example, we reset the memristor by applying 640 pulses as shown in Fig. 5.23. This causes the resistance of the memristor to decrease and the voltage appearing at V_b increases to 4mV. It also can be observed from Fig. 5.23 that as the number of pulses is less than 620, the probability of quantisation errors increases.

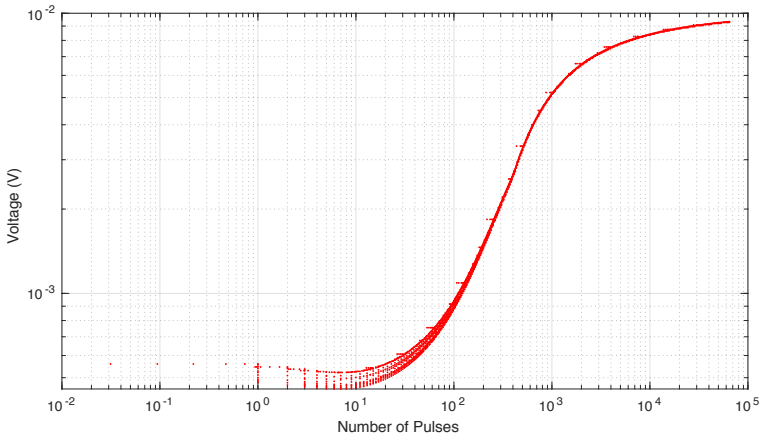


Figure 5.22: Effects of the quantisation error.

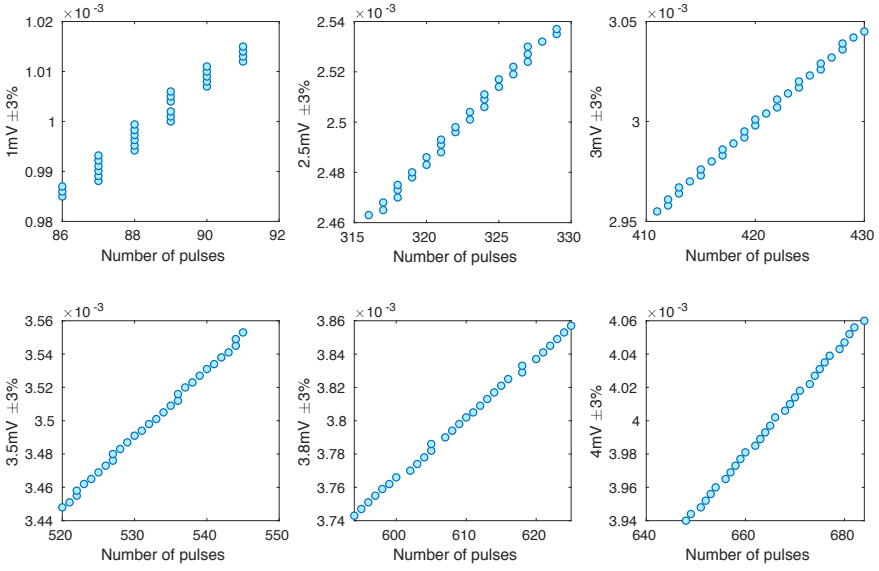


Figure 5.23: Reducing effects of quantisation error.

5.4 Results and Discussions

For the experimental results, the memristors were coded in Verilog-A based on the model given in [6] and the parameters used for the memristors are $V_{\text{on}} = -0.2\text{V}$, $V_{\text{off}} = 0.02\text{V}$, $D = 3\text{nm}$. The systems were designed and simulated in Cadence Virtuoso. We assumed V_{prog}= 250mV, V_{hold}= 50mV, T_{prog}= 2.5ns, $R_{\text{L}} = 1\text{K}\Omega$, R_{on}= 500Ω and R_{off}= 100KΩ. The circuit is simulated at 200MHz.

Non-linear Memristive Encoder: The MNE architecture presented in Fig. 5.10 inherently provides non-linear encoding as show in Fig. 5.11(a). This figure also shows that a small variation in the physical parameters of memristor results in different analogue-to-digital transfer characteristics. Fig. 5.11(b) and Fig. 5.11(c) show the results of varying V_{prog} and T_{prog} respectively while keeping the other parameters fixed. As we see the behaviour of the MNE is non-linear throughout, i.e. it is non-linear for specific V_{prog} or T_{prog} and also for their differences. The MNEs also showed variations in responses for the same challenges under process variations when used as a CRP generator for authentication.

Hardware Overhead: Table 5.3 shows the hardware overhead comparison for proposed PUF architectures and existing memristor-based PUFs [11, 119–121]. Table 5.3 is originally extracted from [11], which excludes the timing and control circuit for all the techniques. From the results, the hardware requirements in [11, 119–121] are much higher compared to the proposed PUF designs which includes non-linear RD-PUF and CHAOS-PUF. The proposed architectures are also flexible in terms of hardware overhead as the value of k and m can be set as per the requirement of the specific design.

Table 5.3: Hardware overhead comparison for memristor-based PUFs.

PUF Architecture	Ref. [119]	Ref. [120]	Ref. [121]	Ref. [11]	Proposed Architecture		
					Arbiter PUF	RD-PUF	CHAOS-PUF
Challenge size (bits)	n	n	n	n	n	n	n
Response size (bits)	r	r	r	r	r	$r = k \times m$	$r = k \times m$
Memristors	$n \times r$	$n \times r$	$2n$	$4n \times r$	$8n \times r$	$k + 1$	$k + 1$
MOSFET switches	–	–	–	$4n \times r$	$2n \times r$	–	–
MUXes(2:1)	$3n + r$	–	$3n$	r	r	–	–
Resistors	–	–	$2n$	–	$2n \times r$	k	$k + 5$
Amplifiers	r	r	–	–	–	–	2
Inverters	–	–	$2n$	$2n \times r$	$2n \times r$	–	–
n-bit decoder	–	1	–	–	–	–	–
Flip-flops	–	–	–	$2 \times r$	$2 \times r$	–	r
Capacitors	–	–	–	–	–	–	3
Comparator	–	–	–	–	–	$k + 1$	k

Performance: To examine the performance of the proposed designs, the systematic method [62] is implemented here. A comprehensive explanation of this method has provided in Chapter 2. Equations Eq. (2.21), Eq. (2.22) and Eq. (2.23) are applied to evaluate uniqueness, uniformity and bit-aliasing respectively. For the proposed non-linear RD-PUF architecture, we tested with 1000 different PUF instances. For the proposed CHAOS-PUF architecture, we tested with 6000 different PUF instances. Table 5.4 shows the performance of the proposed architectures compared to existing architectures [39, 126, 134]. For the reliability, We found the architecture to be reliable to within the quantisation limits of MNE or LE, i.e. as long as the voltage fluctuations from the chaos circuit or the reverse decoder were within this limt MNE or LE was performing reliably. Also, as long as V_{prog} fluctuated to within a few percent and its duration was less than T_{prog} this did not affect the CRPs.

Table 5.4: Performance comparison.

	Uniqueness	Uniformity	Aliasing
Ref. [39]	50.3%	53.8%	49.2%
Ref. [126]	49.8	50.1%	–
Ref. [134]	50%	50.69%	50%
Proposed RD-PUF	49.9%	49.58%	49.3%
Proposed CHAOS-PUF	49.92%	56.33%	49%
Ideal Value	50%	50%	50%

Modelling Attack Resistance: Most existing PUFs are susceptible to machine learning based modelling attacks [127, 135]. Especially for delay based PUFs (e.g. Arbiter PUFs), modelling attack is extremely successful. The CRPs in the existing PUFs are either linearly separable or mathematically differentiable. In contrast, we show that our PUF architectures are capable of offering higher resistance to such

attacks because of the non-linear stochastic properties provided by chaos and MNE circuits.

The modelling attacks we applied in this paper are under the environment of Python 3.6 with the packages Scikit-Learn 0.19.0 and Keras 2.2.4. Logistic Regression (LR), and Support Vector Machine (SVM) are the most widely use techniques for the modelling attacks on PUFs. In our case, to demonstrate the general machine learning attack resistance of the proposed design, we not only use SVM and LR, but also apply another technique, Random Forest (RF) classifier. In SVM attacks, we specifically chose Radial Basis Function (RBF) kernel to fit our nonlinear property [126]. We also tested attack resistance to Artificial Neural Network (ANN) based classifiers. To solve the non-linear problem, we use Multi-layer Perceptron (MLP) feed-forward network structure for classification. ANN based classifiers have been claimed to outperform traditional machine learning algorithms [134].

To improve the machine learning performance, in addition to the CRPs we also considered the voltage V_a (Fig. 5.20) as an additional feature. However, in reality, unlike existing iterative network based PUFs [39, 126], these types of features are much harder to obtain because of the sequential nature of the hardware and accessibility. Here, V_a is dependent on the previous voltage levels, which is much harder to track. However, without this feature the accuracy of all the classifiers drops below 50%.

All classifiers were trained with 60,000 CRPs. Fig. 5.25 and Fig. 5.26 present the trends of accuracy for both of the non-linear PUF architectures, which randomly fluctuate around 50% rather than generally increase as the training set is extended as shown in Fig. 5.24. The figures also indicate that the proposed PUFs are able to effectively resist modelling attacks. We also compare the attack resistance of the proposed designs with the existing PUFs which are subjected to same machine learning attacks. The results are summarised in Table 5.5. In this table, the training sets considered by [39], [126], and [134] are 5,000, 50,000, and 38,000 respectively.

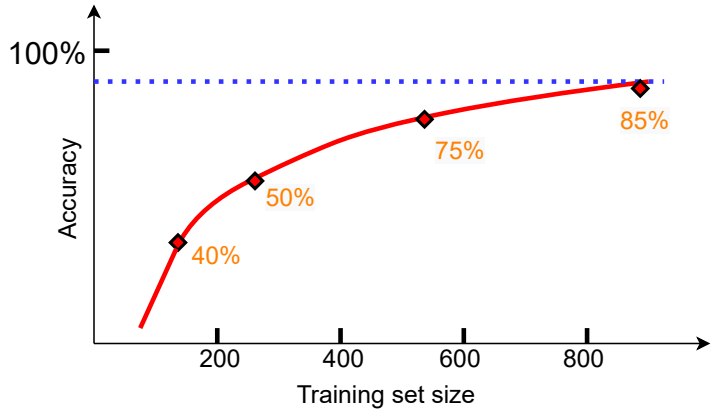


Figure 5.24: Accuracy vs Training Size: increasing the training data always adds information which improve the fit, and increases the accuracy.

Table 5.5: Modelling attack resistance comparisons.

Classification Learner	Ref. [39] %Acc.	Ref. [126] %Acc.	Ref. [134] %Acc.	Proposed Architecture	
				RD- PUF %Acc.	CHAOS-PUF %Acc.
LR	50	–	–	52.68	53.06
SVM	65.67	79	–	55.7	49.81
RF	–	–	–	61.6	49.43
ANN	–	–	≈50	53.35	53.4

The proposed designs provide better resistance to the attacks compared to [39, 126]. It is worth mentioning that for the traditional arbiter PUFs, these attacks can reach almost 99% accuracy [95]. For the ANN classifier, the accuracy of our techniques is around 53.4%. If we consider only the CRPs, without additional features, the accuracy drops below 50%. This is an improvement over [134] also, which did not seem to specify any additional feature for training.

To evaluate the effects of the MNE, we tested attack resistance without it, i.e. with a Linear Encoder (LE) which is described in Section 5.3.1.1.

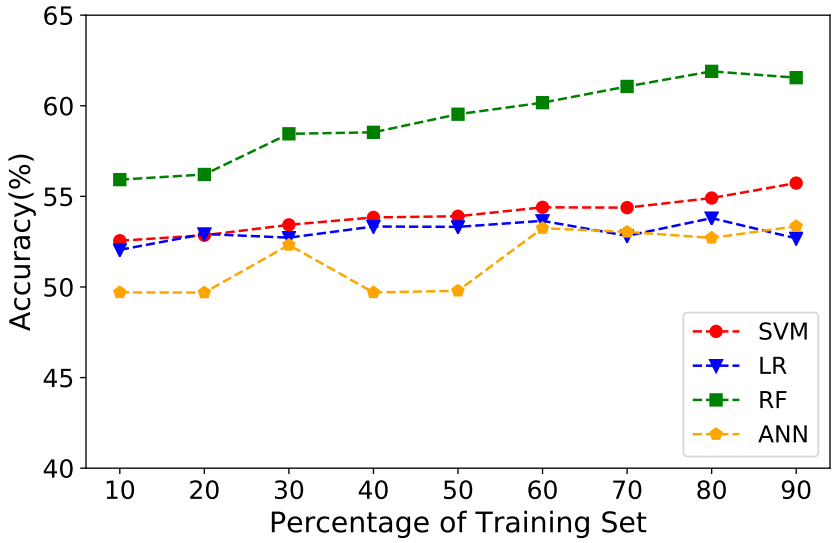


Figure 5.25: Modelling attack results for different size of the training set of proposed RD-PUF.

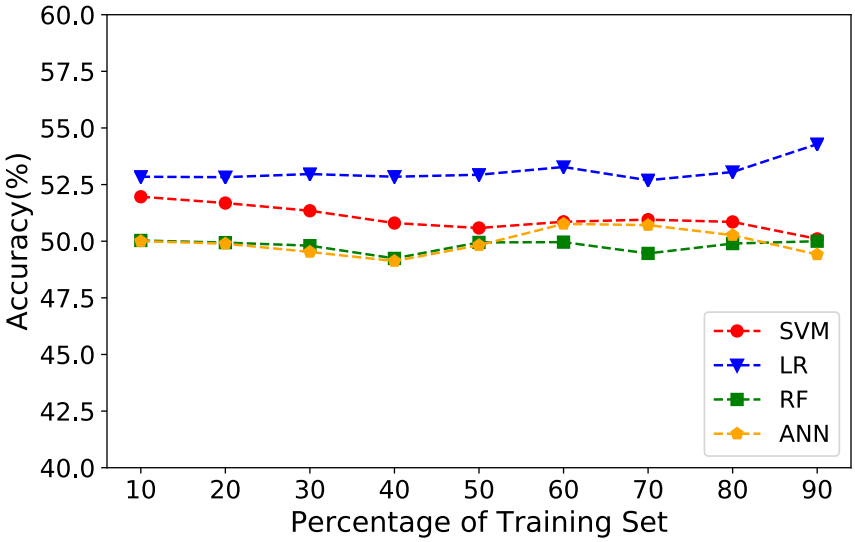


Figure 5.26: Modelling attack results for different size of the training set of proposed CHAOS-PUF.

Table 5.6: Performance of proposed RD-PUF with and without MNE.

Classification Learner	64-bit Challenge with different bit size response (%Accuracy)					
	1-bit		2-bits		4-bits	
	LE	MNE	LE	MNE	LE	MNE
SVM	68.65	55.7	58.95	30.26	53.18	30
LR	68.68	52.68	58.7	30.36	48.69	33
RF	89.6	61.6	83.11	49.15	82.58	48.81
ANN	71.9	53.35	56.85	21.80	59.19	2.9

(a) *Connect LE to the Reverse Decoder:* We also connect the memristive reverse decoder to the LE. The results appear in Table 5.6. Clearly, the accuracy is around 70% without the MNE, and drops to around 55% with the MNE. We also show here the effects of using higher bit counter and training with multi way classifiers. Apart from RF for which the accuracy of using LE is not significantly reduced, the rest of the results are obtained from different classifiers demonstrate that the accuracy of using LE approximately reduces 20%. However, the accuracy of using MNE drops about 50% as the counter size increases. In addition, for ANN classifier, the accuracy of using MNE is reducing further. Hence, for the proposed RD-PUF, the MNE actually provides higher machine learning-based modelling attack resistances in this case.

(b) *Connect LE to the Chaos Circuit:* To compared with the RD-PUF, we connect the memristive chaos circuit to the LE as well. The results illustrate in Table 5.7. Obviously, the accuracy is around 53% for both the MNE and the LE. Hence, the chaos circuit on its own has better machine learning attack resistance than the memristive reverse decoder. Additionally, by adding the MNE, the accuracy is dropping further along with the higher bit counter applied. From Table 5.7,

Table 5.7: Performance of proposed CHAOS-PUF with and without MNE.

Classification Learner	64-bit Challenge with different bit size response (%Accuracy)					
	1-bit		2-bits		4-bits	
	LE	MNE	LE	MNE	LE	MNE
SVM	50.29	49.81	36.01	26.26	31.86	10.02
LR	55.85	53.06	36.32	30.12	30	13.86
RF	50.96	49.43	34.29	24.5	28.45	6.64
ANN	55.53	53.4	38.20	24.1	35.70	4.6

the results show that the accuracy drop nearly 80% for SVM and LR and for RF and ANN, the accuracy are declined below 10%. Therefore, the MNE enhance the machine learning attack resistances for the proposed CHAOS-PUF.

5.5 Summary

In this chapter, we first proposed a highly versatile architecture for replicating a source memristor to a destination memristor, and also for encoding/decoding applications. Then we presented a novel concept on low overhead and high performance PUF realisation based on a non-linear a ADC and DAC scheme. The scheme was tested with a low overhead non-linear memristive reverse decoder and chaos block in conjunction with versatile memristive non-linear encoders respectively. The proposed architectures are sequential in nature and offer much lower overhead compared to existing techniques and higher flexibility in terms of hardware requirements. The combined non-linearity of the digital to analogue conversion circuits (memristive reverse decoder and chaos block) and encoder offer excellent resistance to modelling

attacks. Our results showed that the proposed architectures can outperform existing ones. Owing to their versatility, non-linearity, physical unclonability and lower overhead, we envisage that the proposed architectures will be attractive for diverse security, authentication, and trust applications.

6

Conclusions and Future work

Memristors are considered advantageous over current technologies and have been exploited in high density memory designs. However, memristive devices also have some other potentials beyond their memory applications. The switching and non-linear characteristics of memristors provide opportunities to explore the following application fields, such as logic design and machine learning resistant PUF design. The chapters within this thesis considerably describe memristive logic architectures as well as memristor-based novel PUF circuit designs.

6.1 Summary of this Thesis

The work presented in this thesis explored the memristor-based designs in the following areas:

1. Detailed analysis of memristor-based logic designs and proposal of high-performance single cycle memristive multifunction logic architecture

    - Since most of the existing memristor-based logic designs require multiple clock cycles and complex control circuits. Therefore, this thesis proposed an efficient multifunction logic architecture by using one transistor and four memristors. This architecture can operate in a single clock cycle with fewer components and less power consumption. It also can integrate with the CMOS technology.
    - This thesis presented a technique that shows the reliable performance of the proposed logic design at higher frequencies.
    - The thesis also presented a highly compact full adder design and highly efficient memristive bit parallel multiplier over GF(2^2) and GF(2^4) by implementing the proposed multifunction logic architecture as the primary building block.

2. A application of memristor-based multifunctional logic architecture in delay-based arbiter PUF

    - Most of the memristor models have not taken the parasitic effects into consideration, but this thesis demonstrated effects in the proposed logic architecture when the parasitic components are combined with the memristor model. Due to the specific structure of the proposed logic design, most of the parasitic effects can be taken or canceled out. Additionally, compared with another existing memristor-based logic architecture, the proposed one also maintains reliability.
    - A delay-based arbiter PUF is presented by using the proposed logic architecture. Chapter 4 evaluates the performance of the proposed arbiter PUF architectures (8-, 16-, and 32-bits). The results show that the proposed arbiter PUF can maintain high reliability and provide better performance compared to some existing similar PUF architectures in terms of the uniqueness, uniformity and bit-aliasing.

3. Detailed analysis of the proposed replicating architecture and its applications in encoding/decoding and physical unclonability

    - A highly versatile architecture is proposed for replicating a source memristor to a destination memristor, which can also be used for generating non-linear digital codes for added security, physical unclonability, and also for decoding the analogue value from the digital codes. The proposed architecture is lightweight and relies only on a few logic components, two comparators, and a counter. The simulation results demonstrated its non-linear behaviour and its sensitivity to process and parametric variations.

The latter can be extremely useful for designing physical unclonable functions.

- The main drawback of the existing delay-based PUFs is that the certain linear-relation between the challenges and responses can cause the failure of the modelling attack resistance. Hence, this thesis leverages the non-linear characteristic of memristors to present a novel PUF framework. This framework was tested with a low overhead memristive digital to analogue conversion block in conjunction with the proposed memristive non-linear encoding block. Additionally, it also provides higher flexibility for satisfying some specific hardware requirements. The proposed PUF architectures were also exposed under the modelling attacks (e.g. LR, SVM, RF and ANN). The accuracy of those attacks are around 50% which indicates that the proposed PUF architectures can resist modelling attacks.

4. As an emerging technology, memristor does not have any corresponding built-in models existing like others such as resistors, capacitors and transistors in EDA tools. Hence, Verilog-A models and SPICE models for the memristor were developed to aid simulation with the EDA tools which could possibly make the circuit difficult to converge. Hence, the convergence difficulties of the circuit extremely slow down the simulation process.

In summary, this work contributes to a comprehensive understanding of the applications of memristor-based logic design and authentication. A detailed analysis of the performances for these designs were all covered in this thesis.

6.2 Future Research

Although the proposed techniques in this thesis explores memristor-based applications, there still remain several opportunities to improve and extend the work in this thesis.

1. The memristive full adder and the Galois Field multiplier presented in Chapter 3 are designed by the proposed memristive XOR/AND and XNOR/OR logic gates. Furthermore, other combinational logic circuits (e.g., ALU, MUX, and code converters) can also be implemented by the proposed hybrid logic architectures.

2. So far, a large amount of the memristive applications are based on crossbar arrays to reduce the area consumption and achieve high-density requirements. Hence, the proposed logic systems shown in Chapter 3 can be further implemented on crossbar architecture to realise a highly compact 3D design. Meanwhile, some research work is required on the control circuitry to select the memristors precisely. In this case, the multiple cells select scheme, which has considered the sneak-path and line resistances can be applied accordingly.

3. The thesis presents a novel modelling attack resistant low overhead non-linear memristive PUF in Chapter 5. The performance of the proposed PUF in terms of uniqueness, uniformity, bit-aliasing and reliability are all demonstrated in this chapter. We argued that the architecture can maintain reliability as long as the voltage fluctuations from memristive digital to analogue conversion circuit within the quantisation limits of the MNE or LE. Hence, to ensure the voltage fluctuations within the certain quantisation limits, the further research work related to the precise source generation need to be considered.

4. In this work, the author improved the Verilog-A memristor model without altering the behaviour of the memristor to allow the simulation tool to converge properly. However, the simulation process is still very tedious and slow especially when the Monte Carlo simulations run on the large size circuit. Hence, it requires a large amount of research on the Verilog-A code to accelerate the speed of the simulations. Alternatively, instead of Verilog-A code, a universal memristor model needs to be generated in SPICE model types (e.g. subcircuit model) to make it more compatible with the rest of the circuit. Meanwhile, the universal model can capture all the memristor behaviour including the temperature variations.

References

[1] Frank Schwierz. Graphene transistors. *Nature nanotechnology*, 5(7):487, 2010.

[2] Tejinder Singh. Hybrid memristor-cmos (memos) based logic gates and adder circuits. *arXiv preprint arXiv:1506.06735*, 2015.

[3] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *nature*, 453(7191):80–83, 2008.

[4] J Joshua Yang, Matthew D Pickett, Xuema Li, Douglas AA Ohlberg, Duncan R Stewart, and R Stanley Williams. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature nanotechnology*, 3(7):429–433, 2008.

[5] 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*, 60(1):211–221, Jan 2013. ISSN 1549-8328. doi: 10.1109/TCSI.2012.2215714.

[6] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny. Vteam: A general model for voltage-controlled memristors. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(8):786–790, Aug 2015. ISSN 1549-7747. doi: 10.1109/TCSII.2015.2433536.

[7] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. Memristor-based material implication (imply) logic: Design principles and methodologies. *IEEE Transactions on Very Large Scale Integra-*

tion (VLSI) Systems, 22(10):2054–2066, Oct 2014. ISSN 1063-8210. doi: 10.1109/TVLSI.2013.2282132.

[8] Mohammed Affan Zidan, Hossam Aly Hassan Fahmy, Muhammad Mustafa Hussain, and Khaled Nabil Salama. Memristor-based memory: The sneak paths problem and solutions. *Microelectronics Journal*, 44(2):176–183, 2013.

[9] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser. Magic-memristor-aided logic. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(11):895–899, Nov 2014. ISSN 1549-7747. doi: 10.1109/TCSII.2014.2357292.

[10] Yachuan Pang, Huaqiang Wu, Bin Gao, Dong Wu, An Chen, and He Qian. A novel puf against machine learning attack: Implementation on a 16 mb rram chip. In *2017 IEEE International Electron Devices Meeting (IEDM)*, pages 12–2. IEEE, 2017.

[11] Jimson Mathew, Rajat Subhra Chakraborty, Durga Prasad Sahoo, Yuanfan Yang, and Dhiraj K. Pradhan. A novel memristor based physically unclonable function. *Integration, the VLSI Journal*, 51:37 – 45, 2015.

[12] G. E. Moore. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff. *IEEE Solid-State Circuits Society Newsletter*, 11(3):33–35, Sep. 2006. ISSN 1098-4232. doi: 10.1109/N-SSC.2006.4785860.

[13] IRDS Reports 2020 edition. https://irds.ieee.org/editions/2020. Accessed: 2021-07-20.

[14] H. . P. Wong, L. Wei, and J. Deng. The future of cmos scaling - parasitics engineering and device footprint scaling. In *2008 9th International Conference on Solid-State and Integrated-Circuit Technology*, pages 21–24, 2008.

[15] Digh Hisamoto, Toru Kaga, Yoshifumi Kawamoto, and Eiji Takeda. A fully depleted lean-channel transistor (delta)-a novel vertical ultra thin soi mosfet. In *International Technical Digest on Electron Devices Meeting*, pages 833–836. IEEE, 1989.

[16] George K Celler and Sorin Cristoloveanu. Frontiers of silicon-on-insulator. *Journal of Applied Physics*, 93(9):4955–4978, 2003.

[17] Mathieu Luisier and Gerhard Klimeck. Atomistic full-band design study of inas band-to-band tunneling field-effect transistors. *IEEE Electron Device Letters*, 30(6):602–604, 2009.

[18] SL Miller and PJ McWhorter. Physics of the ferroelectric nonvolatile memory field effect transistor. *Journal of applied physics*, 72(12):5999–6010, 1992.

[19] Shaowen Chen, Zheng Han, Mirza M Elahi, KM Masum Habib, Lei Wang, Bo Wen, Yuanda Gao, Takashi Taniguchi, Kenji Watanabe, James Hone, et al. Electron optics with pn junctions in ballistic graphene. *Science*, 353(6307): 1522–1525, 2016.

[20] Sourav Dutta, Rouhollah Mousavi Iraei, Chenyun Pan, Dmitri E Nikonov, Sasikanth Manipatruni, Ian A Young, and Azad Naeemi. Impact of spintronics transducers on the performance of spin wave logic circuit. In *2016 IEEE 16th International Conference on Nanotechnology (IEEE-NANO)*, pages 990–993. IEEE, 2016.

[21] D. Morris, D. Bromberg, J. Zhu, and L. Pileggi. mlogic: Ultra-low voltage non-volatile logic circuits using stt-mtj devices. In *DAC Design Automation Conference 2012*, pages 486–491, 2012. doi: 10.1145/2228360.2228446.

[22] Behtash Behin-Aein, Deepanjan Datta, Sayeef Salahuddin, and Supriyo Datta. Proposal for an all-spin logic device with built-in memory. *Nature nanotechnology*, 5(4):266–270, 2010.

[23] Hiroyuki Akinaga and Hisashi Shima. Resistive random access memory (reram) based on metal oxides. *Proceedings of the IEEE*, 98(12):2237–2251, 2010.

[24] H-S Philip Wong, Simone Raoux, SangBum Kim, Jiale Liang, John P Reifenberg, Bipin Rajendran, Mehdi Asheghi, and Kenneth E Goodson. Phase change memory. *Proceedings of the IEEE*, 98(12):2201–2227, 2010.

[25] Mark D Stiles and Jacques Miltat. Spin-transfer torque and dynamics. In *Spin dynamics in confined magnetic structures III*, pages 225–308. Springer, 2006.

[26] L. Chua. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, Sep 1971. ISSN 0018-9324. doi: 10.1109/TCT.1971.1083337.

[27] William J Gallagher and Stuart SP Parkin. Development of the magnetic tunnel junction mram at ibm: From first junctions to a 16-mb mram demonstrator chip. *IBM Journal of Research and Development*, 50(1):5–23, 2006.

[28] Shu-Yau Wu. A new ferroelectric memory device, metal-ferroelectric-semiconductor transistor. *IEEE Transactions on Electron Devices*, 21(8):499–504, 1974.

[29] Duncan G Elliott, Michael Stumm, W Martin Snelgrove, Christian Cojocaru, and Robert McKenzie. Computational ram: Implementing processors in memory. *IEEE Design & Test of Computers*, 16(1):32–41, 1999.

[30] John E Kelly. Computing, cognition and the future of knowing. *Whitepaper, IBM Reseach*, 2, 2015.

[31] George Epstein. *Multiple-valued logic design: an introduction*. CRC Press, 1993.

[32] L. O. Chua and Sung Mo Kang. Memristive devices and systems. *Proceedings of the IEEE*, 64(2):209–223, Feb 1976. ISSN 0018-9219. doi: 10.1109/PROC.1976.10092.

[33] L.Wang T.Sadi and A.Asenov. Multi-scale electrothermal simulation and modelling of resistive random access memory devices. In *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2016 26th International Workshop, Bremen*, Sept 2016.

[34] P. Mazumder, S. M. Kang, and R. Waser. Memristors: Devices, models, and applications [scanning the issue]. *Proceedings of the IEEE*, 100(6):1911–1919, June 2012. ISSN 0018-9219. doi: 10.1109/JPROC.2012.2190812.

[35] Y. Ho, G. M. Huang, and P. Li. Dynamical properties and design analysis for nonvolatile memristor memories. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(4):724–736, April 2011. ISSN 1549-8328. doi: 10.1109/TCSI.2010.2078710.

[36] M. Hartmann, P. Raghavan, L. V. D. Perre, P. Agrawal, and W. Dehaene. Memristor-based (reram) data memory architecture in asip design. In *Digital System Design (DSD), 2013 Euromicro Conference on*, pages 795–798, Sept 2013. doi: 10.1109/DSD.2013.138.

[37] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman. Memristor crossbar-based neuromorphic computing system: A case study. *IEEE Transactions on Neural Networks and Learning Systems*, 25(10):1864–1878, Oct 2014. ISSN 2162-237X. doi: 10.1109/TNNLS.2013.2296777.

[38] N. Serafino and M. Zaghloul. Review of nanoscale memristor devices as synapses in neuromorphic systems. In *2013 IEEE 56th International Midwest Symposium*

on Circuits and Systems (MWSCAS), pages 602–603, Aug 2013. doi: 10.1109/MWSCAS.2013.6674720.

[39] Jimson Mathew, Rajat Subhra Chakraborty, Durga Prasad Sahoo, Yuanfan Yang, and Dhiraj K Pradhan. A novel memristor-based hardware security primitive. *ACM Transactions on Embedded Computing Systems (TECS)*, 14(3):60, 2015.

[40] J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. McDonald, G. S. Rose, and B. Wysocki. Nano meets security: Exploring nanoelectronic devices for security applications. *Proceedings of the IEEE*, 103(5):829–849, May 2015. ISSN 0018-9219. doi: 10.1109/JPROC.2014.2387353.

[41] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman. Mrl-memristor ratioed logic. In *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*, pages 1–6, Aug 2012. doi: 10.1109/CNNA.2012.6331426.

[42] Yaxiong Zhou, Yi Li, Lei Xu, Shujing Zhong, Ronggang Xu, and Xiangshui Miao. A hybrid memristor-cmos xor gate for nonvolatile logic computation. *physica status solidi (a)*, 2015.

[43] D. B. Strukov, D. R. Stewart, J. Borghetti, X. Li, M. Pickett, G. M. Ribeiro, W. Robinett, G. Snider, J. P. Strachan, W. Wu, Q. Xia, J. J. Yang, and R. S. Williams. Hybrid cmos/memristor circuits. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 1967–1970, May 2010. doi: 10.1109/ISCAS.2010.5537020.

[44] J. Cong and Bingjun Xiao. mrfpga: A novel fpga architecture with memristor-based reconfiguration. In *2011 IEEE/ACM International Symposium on*

Nanoscale Architectures, pages 1–8, June 2011. doi: 10.1109/NANOARCH.2011.5941476.

[45] K. Sakuma, P. S. Andry, C. K. Tsang, S. L. Wright, B. Dang, C. S. Patel, B. C. Webb, J. Maria, E. J. Sprogis, S. K. Kang, R. J. Polastre, R. R. Horton, and J. U. Knickerbocker. 3d chip-stacking technology with through-silicon vias and low-volume lead-free interconnections. *IBM Journal of Research and Development*, 52(6):611–622, Nov 2008. ISSN 0018-8646. doi: 10.1147/JRD.2008.5388567.

[46] Dmitri B Strukov and R Stanley Williams. Four-dimensional address topology for circuits with stacked multilayer crossbar arrays. *Proceedings of the National Academy of Sciences*, 106(48):20155–20158, 2009.

[47] Kwang-Ting Tim Cheng and Dmitri B. Strukov. 3d cmos-memristor hybrid circuits: Devices, integration, architecture, and applications. In *Proceedings of the 2012 ACM International Symposium on International Symposium on Physical Design*, ISPD '12, pages 33–40, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1167-0. doi: 10.1145/2160916.2160925. URL http://doi.acm.org/10.1145/2160916.2160925.

[48] Dmitri B Strukov and R Stanley Williams. Exponential ionic drift: fast switching and low volatility ofáthin-film memristors. *Applied Physics A*, 94(3):515–519, 2009.

[49] Matthew D Pickett, Dmitri B Strukov, Julien L Borghetti, J Joshua Yang, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. Switching dynamics in titanium dioxide memristive devices. *Journal of Applied Physics*, 106(7):074508, 2009.

[50] Dalibor Biolek, Viera Biolková, and Zdeněk Biolek. Spice model of memristor with nonlinear dopant drift. *Radioengineering*, 2009.

[51] Yogesh Joglekar and Stephen Wolf. Joglekar yn, wolf sj (2009, july)the elusive memristor: properties of basic electrical circuits. eur j phys. *European Journal of Physics*, 30, 07 2008. doi: 10.1088/0143-0807/30/4/001.

[52] T. Prodromakis, B. P. Peh, C. Papavassiliou, and C. Toumazou. A versatile memristor model with nonlinear dopant kinetics. *IEEE Transactions on Electron Devices*, 58(9):3099–3105, Sep. 2011. doi: 10.1109/TED.2011.2158004.

[53] A. A. Adeyemo, J. Mathew, A. M. Jabir, and D. K. Pradhan. Exploring error-tolerant low-power multiple-output read scheme for memristor-based memory arrays. In *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, pages 17–20, Oct 2015. doi: 10.1109/DFT.2015.7315129.

[54] J. von Neumann. First draft of a report on the edvac. *IEEE Annals of the History of Computing*, 15(4):27–75, 1993. doi: 10.1109/85.238389.

[55] H Li, Baoyu Gao, Z Chen, Y Zhao, P Huang, H Ye, L Liu, X Liu, and J Kang. A learnable parallel processing architecture towards unity of memory and computing. *Scientific reports*, 5(1):1–8, 2015.

[56] Shiv Shankar Mishra, Adarsh Kumar Agrawal, and RK Nagaria. A comparative performance analysis of various cmos design techniques for xor and xnor circuits. *International Journal on Emerging Technologies*, 1(1):1–10, 2010.

[57] U. Rührmair and D. E. Holcomb. Pufs at a glance. In *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1–6, March 2014. doi: 10.7873/DATE.2014.360.

[58] R Robbert Berg and Van Den. Entropy analysis of physical unclonable functions. 2012.

[59] Roel Maes and Ingrid Verbauwhede. Physically unclonable functions: A study on the state of the art and future research directions. In *Towards Hardware-Intrinsic Security*, pages 3–37. Springer, 2010.

[60] Yachuan Pang, Bin Gao, Bohan Lin, He Qian, and Huaqiang Wu. Memristors for hardware security applications. *Advanced Electronic Materials*, page 1800872, 2019.

[61] Yansong Gao, Damith C Ranasinghe, Said F Al-Sarawi, Omid Kavehei, and Derek Abbott. Emerging physical unclonable functions with nanotechnology. *IEEE access*, 4:61–80, 2016.

[62] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. A systematic method to evaluate and compare the performance of physical unclonable functions. cryptology eprint archive, report 2011/657, 2011.

[63] A. Maiti, J. Casarona, L. McHale, and P. Schaumont. A large scale characterization of ro-puf. In *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 94–99, June 2010. doi: 10.1109/HST.2010.5513108.

[64] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[65] D. W. Bauder. An anti - counterfeiting concept for currency systems. 1983.

[66] National Research Council et al. *Counterfeit Deterrent Features for the Next-Generation Currency Design*, volume 472. National Academies Press, 1993.

[67] Ghaith Hammouri, Aykutlu Dana, and Berk Sunar. Cds have fingerprints too. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 348–362. Springer, 2009.

[68] Keith Lofstrom, W Robert Daasch, and Donald Taylor. Ic identification circuit using device mismatch. In *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 00CH37056)*, pages 372–373. IEEE, 2000.

[69] Ryan Helinski, Dhruva Acharyya, and Jim Plusquellic. A physical unclonable function defined using power distribution system equivalent resistance variations. In *2009 46th ACM/IEEE Design Automation Conference*, pages 676–681. IEEE, 2009.

[70] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan Van Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 369–383. Springer, 2006.

[71] Jorge Guajardo, Boris Škorić, Pim Tuyls, Sandeep S Kumar, Thijs Bel, Antoon HM Blom, and Geert-Jan Schrijen. Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions. *Information Systems Frontiers*, 11(1):19–41, 2009.

[72] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls. Physical unclonable functions and public-key crypto for fpga ip protection. In *2007 International Conference on Field Programmable Logic and Applications*, pages 189–195, Aug 2007. doi: 10.1109/FPL.2007.4380646.

[73] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. Fpga intrinsic pufs and their use for ip protection. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 63–80, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74735-2.

[74] Sandeep S Kumar, Jorge Guajardo, Roel Maes, Geert-Jan Schrijen, and Pim Tuyls. The butterfly puf protecting ip on every fpga. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 67–70. IEEE, 2008.

[75] Pravin Prabhu, Ameen Akel, Laura M. Grupp, Wing-Kei S. Yu, G. Edward Suh, Edwin Kan, and Steven Swanson. Extracting device fingerprints from flash memory by exploiting physical variations. In *Proceedings of the 4th International Conference on Trust and Trustworthy Computing*, TRUST'11, pages 188–201, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21598-8. URL http://dl.acm.org/citation.cfm?id=2022245.2022264.

[76] S. Rosenblatt, S. Chellappa, A. Cestero, N. Robson, T. Kirihata, and S. S. Iyer. A self-authenticating chip architecture using an intrinsic fingerprint of embedded dram. *IEEE Journal of Solid-State Circuits*, 48(11):2934–2943, Nov 2013. ISSN 0018-9200. doi: 10.1109/JSSC.2013.2282114.

[77] Daihyun Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, Oct 2005. ISSN 1063-8210. doi: 10.1109/TVLSI.2005.859470.

[78] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525)*, pages 176–179, June 2004. doi: 10.1109/VLSIC.2004.1346548.

[79] G. Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th Annual Design Automation Conference*, DAC '07, pages 9–14, New York, NY, USA,

2007. ACM. ISBN 978-1-59593-627-1. doi: 10.1145/1278480.1278484. URL http://doi.acm.org/10.1145/1278480.1278484.

[80] Klaus Kursawe, Ahmad-Reza Sadeghi, Dries Schellekens, Boris Skoric, and Pim Tuyls. Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage. In *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 22–29. IEEE, 2009.

[81] Le Zhang, Zhi Hui Kong, and Chip-Hong Chang. Pckgen: A phase change memory based cryptographic key generator. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pages 1444–1447. IEEE, 2013.

[82] Le Zhang, Zhi Hui Kong, Chip-Hong Chang, Alessandro Cabrini, and Guido Torelli. Exploiting process variations and programming sensitivity of phase change memory for reconfigurable physical unclonable functions. *IEEE Transactions on Information Forensics and Security*, 9(6):921–932, 2014.

[83] Maurizio Aiello, Enrico Cambiaso, Maurizio Mongelli, and Gianluca Papaleo. An on-line intrusion detection approach to identify low-rate dos attacks. In *2014 International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2014.

[84] Takao Marukame, Tetsufumi Tanamoto, and Yuichiro Mitani. Extracting physically unclonable function from spin transfer switching characteristics in magnetic tunnel junctions. *IEEE Transactions on Magnetics*, 50(11):1–4, 2014.

[85] Elena Ioana Vatajelu, Giorgio Di Natale, Marco Indaco, and Paolo Prinetto. Stt mram-based pufs. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 872–875. IEEE, 2015.

[86] Le Zhang, Xuanyao Fong, Chip-Hong Chang, Zhi Hui Kong, and Kaushik Roy. Highly reliable memory-based physical unclonable function using spin-transfer

torque mram. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2169–2172. IEEE, 2014.

[87] Le Zhang, Xuanyao Fong, Chip-Hong Chang, Zhi Hui Kong, and Kaushik Roy. Optimizating emerging nonvolatile memories for dual-mode applications: Data storage and key generator. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(7):1176–1187, 2015.

[88] Le Zhang, Xuanyao Fong, Chip-Hong Chang, Zhi Hui Kong, and Kaushik Roy. Highly reliable spin-transfer torque magnetic ram-based physical unclonable function with multi-response-bits per cell. *IEEE Transactions on Information Forensics and Security*, 10(8):1630–1642, 2015.

[89] Rui Liu, Huaqiang Wu, Yachuan Pang, He Qian, and Shimeng Yu. Experimental characterization of physical unclonable function based on 1 kb resistive random access memory arrays. *IEEE Electron Device Letters*, 36(12):1380–1383, 2015.

[90] A Chen. Reconfigurable physical unclonable function based on probabilistic switching of rram. *Electronics Letters*, 51(8):615–617, 2015.

[91] Po-Hao Tseng, Kai-Chieh Hsu, Yu-Yu Lin, Feng-Min Lee, Ming-Hsiu Lee, Hsiang-Lan Lung, Kuang-Yeu Hsieh, Keh Chung Wang, and Chih-Yuan Lu. Error free physically unclonable function with programmed resistive random access memory using reliable resistance states by specific identification-generation method. *Japanese Journal of Applied Physics*, 57(4S):04FE04, 2018.

[92] Garrett S Rose, Nathan McDonald, Lok-Kwong Yan, and Bryant Wysocki. A write-time based memristive puf for hardware security applications. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 830–833. IEEE, 2013.

[93] P. Koeberl, Ü. Kocabaş, and A. Sadeghi. Memristor pufs: A new generation of memory-based physically unclonable functions. In *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 428–431, March 2013. doi: 10.7873/DATE.2013.096.

[94] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Testing techniques for hardware security. In *2008 IEEE International Test Conference*, pages 1–10, Oct 2008. doi: 10.1109/TEST.2008.4700636.

[95] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 237–249, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0245-6. doi: 10.1145/1866307.1866335. URL http://doi.acm.org/10.1145/1866307.1866335.

[96] U. Rührmair and M. van Dijk. Pufs in security protocols: Attack models and security evaluations. In *2013 IEEE Symposium on Security and Privacy*, pages 286–300, May 2013. doi: 10.1109/SP.2013.27.

[97] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*, 102(8):1126–1141, Aug 2014. ISSN 0018-9219. doi: 10.1109/JPROC.2014.2320516.

[98] Ulrich Rührmair, Jan Sölter, and Frank Sehnke. On the foundations of physical unclonable functions. *IACR Cryptology ePrint Archive*, 2009:277, 2009.

[99] J. Miskelly, C. Gu, Q. Ma, Y. Cui, W. Liu, and M. O'Neill. Modelling attack analysis of configurable ring oscillator (cro) puf designs. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pages 1–5, 2018. doi: 10.1109/ICDSP.2018.8631638.

[100] Leon Chua. Resistance switching memories are memristors. *Applied Physics A*, 102(4):765–783, 2011.

[101] Paul Meuffels and Rohit Soni. Fundamental issues and problems in the realization of memristors. *arXiv preprint arXiv:1207.7319*, 2012.

[102] J Kim, YV Pershin, M Yin, T Datta, and M Di Ventra. An experimental proof that resistance-switching memories are not memristors. *arXiv preprint arXiv:1909.07238*, 2019.

[103] Nafisa Noor and Helena Silva. Phase change memory for physical unclonable functions. In *Applications of Emerging Memory Technology*, pages 59–91. Springer, 2020.

[104] Linggang Zhu, Jian Zhou, Zhonglu Guo, and Zhimei Sun. An overview of materials issues in resistive random access memory. *Journal of Materiomics*, 1 (4):285–295, 2015.

[105] Xu Donglin and Ren Junyan. Study of parasitic capacitance effect on low pass cmos active filters. In *ASIC, 2003. Proceedings. 5th International Conference on*, volume 2, pages 992–995 Vol.2, Oct 2003. doi: 10.1109/ICASIC.2003.1277378.

[106] J. Lacord, G. Ghibaudo, and F. Boeuf. Comprehensive and accurate parasitic capacitance models for two- and three-dimensional cmos device structures. *IEEE Transactions on Electron Devices*, 59(5):1332–1344, May 2012. ISSN 0018-9383. doi: 10.1109/TED.2012.2187454.

[107] Elena Dubrova. Logic synthesis and verification. chapter Multiple-valued Logic Synthesis and Optimization, pages 89–114. Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 0-7923-7606-4. URL http://dl.acm.org/citation.cfm?id=566845.566849.

[108] Abdoreza Pishvaie, Ghassem Jaberipur, and Ali Jahanian. Improved cmos (4; 2) compressor designs for parallel multipliers. *Computers & Electrical Engineering*, 38(6):1703–1716, 2012.

[109] S. Kvatinsky, D. Belousov, S. Liman, N. Wald, and G. Satat. Pure memristive logic gate, September 10 2015. URL http://www.google.com/patents/US20150256178. US Patent App. 14/641,482.

[110] E. Yalon, A. Gavrilov, S. Cohen, D. Mistele, B. Meyler, J. Salzman, and D. Ritter. Resistive switching in $hboxHfO_2$ probed by a metal insulator semiconductor bipolar transistor. *IEEE Electron Device Letters*, 33(1):11–13, Jan 2012. ISSN 0741-3106. doi: 10.1109/LED.2011.2171317.

[111] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and their Applications*. Cambridge University Press, second edition, 1994. ISBN 9781139172769. URL http://dx.doi.org/10.1017/CBO9781139172769. Cambridge Books Online.

[112] A. Reyhani-Masoleh and M. A. Hasan. Low complexity bit parallel architectures for polynomial basis multiplication over gf(2^m). *IEEE Transactions on Computers*, 53(8):945–959, Aug 2004. ISSN 0018-9340. doi: 10.1109/TC.2004.47.

[113] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*, pages 176–179, June 2004. doi: 10.1109/VLSIC.2004.1346548.

[114] Chris Yakopcic, Tarek M Taha, Guru Subramanyam, Robinson E Pino, and Stanley Rogers. A memristor device model. *IEEE electron device letters*, 32 (10):1436–1438, 2011.

[115] Maheshwar Pd Sah, Changju Yang, Hyongsuk Kim, Bharathwaj Muthuswamy, Jovan Jevtic, and Leon Chua. A generic model of memristors with parasitic components. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62 (3):891–898, 2015.

[116] B. Garda and M. J. Ogorzałek. Modelling the generic tio2 memristor with the parasitic components. In *2016 International Conference on Signals and Electronic Systems (ICSES)*, pages 173–176, Sept 2016. doi: 10.1109/ICSES.2016.7593845.

[117] Predictive Technology Model (PTM) for 32nm Metal Gate/High-K MOSFET (v2.0). http://ptm.asu.edu/modelcard/LP/32nm_LP.pm. Accessed: 2016-03-19.

[118] David J Kinniment. *Synchronization and arbitration in digital systems*. John Wiley & Sons, 2008.

[119] J. Rajendran, G. S. Rose, R. Karri, and M. Potkonjak. Nano-ppuf: A memristor-based security primitive. In *2012 IEEE Computer Society Annual Symposium on VLSI*, pages 84–87, 2012.

[120] P. Koeberl, Ü. Kocabaş, and A. Sadeghi. Memristor pufs: A new generation of memory-based physically unclonable functions. In *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 428–431, 2013.

[121] Garrett S Rose, Nathan McDonald, Lok-Kwong Yan, Bryant Wysocki, and Karen Xu. Foundations of memristor based puf architectures. In *2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 52–57. IEEE, 2013.

[122] Shuang Pi, Can Li, Hao Jiang, Weiwei Xia, Huolin Xin, J Joshua Yang, and Qiangfei Xia. Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension. *Nature nanotechnology*, 14(1):35–39, 2019.

[123] Olufemi A Olumodeji, Alessandro P Bramanti, Massimo Gottardi, and Salvatore Iannotta. A memristor-based pixel implementing light-to-resistance conversion. *Optical Engineering*, 55(2):020501, 2016.

[124] Adedotun Adeyemo, Jimson Mathew, Abusaleh Jabir, Corrado Di Natale, Eugenio Martinelli, and Marco Ottavi. Efficient sensing approaches for high-density memristor sensor array. *Journal of Computational Electronics*, 17(3):1285–1296, 2018.

[125] Olufemi A Olumodeji and Massimo Gottardi. Emulating the physical properties of hp memristor using an arduino and a digital potentiometer. In *2016 12th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME)*, pages 1–4. IEEE, 2016.

[126] A. Vijayakumar and S. Kundu. A novel modeling attack resistant puf design based on non-linear voltage transfer characteristics. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 653–658, March 2015. doi: 10.7873/DATE.2015.0522.

[127] A. Vijayakumar, V. C. Patil, C. B. Prado, and S. Kundu. Machine learning resistant strong puf: Possible or a pipe dream? In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 19–24, May 2016. doi: 10.1109/HST.2016.7495550.

[128] Toshikazu Kanaoka and Toshio Ito. Encoding device, decoding device, encoding/decoding device, and recording/reproducing device, April 3 2012. US Patent 8,151,162.

[129] Daniel A Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.

[130] Leon O. Chua. Chua's Circuit: an Overview Ten Years Later. *J Circuit Syst Comp*, 4(2):117–159, 1994.

[131] Leon O. Chua. The Genesis of Chua's Circuit. *AEU-Int J Electron C.*, 46(4): 251–257, 1992.

[132] V Siderskiy. Chua's Circuits. http://www.chuacircuits.com. Accessed: 2019-05-21.

[133] Michael Peter Kennedy. Three steps to chaos. ii. a chua's circuit primer. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40(10):657–674, 1993.

[134] Elena Ioana Vatajelu, Giorgio Di Natale, Mohd Syafiq Mispan, and Basel Halak. On the encryption of the challenge in physically unclonable functions. In *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 115–120. IEEE, 2019.

[135] Lanxiang Chen. A Framework to Enhance Security of Physically Unclonable Functions Using Chaotic Circuits. *Phys. Lett. A*, 382(18):1195–1201, 2018. ISSN 0375-9601.