Memory circuits for hardware security applications

By

Jesús Miguel Germán Martínez

Thesis submitted as a partial requirement for the degree of

Master in Science with specialty in Electronics

at the

Instituto Nacional de Astrofísica, Óptica y Electrónica
December 2020
San Andrés Cholula, Puebla

Adviser:

Dr. Librado Arturo Sarmiento Reyes, INAOE

Memory circuits for hardware security applications

Master's Thesis

By:

Jesús Miguel Germán Martínez

Adviser:

Dr. Librado Arturo Sarmiento Reyes

Instituto Nacional de Astrofísica Óptica y Electrónica
Electronics Department

San Andrés Cholula, Puebla. January 20, 2021

Acknowledgments

I thank my mother Laura Martínez, my grandmother Ma. Oralia Quintero and my brothers Gustavo Germán and Christian Germán for their immense support today and always. To Dr. Arturo Sarmiento for his time and dedication that he invested during the development of this thesis. Finally, to all my friends I met and lived with during the master's program who showed their support and important contributions to the development of this work.

Dedicatory

To my future readers. I hope you will enjoy reading this work as much as I did when writing it and that it will serves as a guide for future work.

Abstract

In this work, a novel proposal for a memristor-based hardware security scheme has been developed.

The proposal aims at generating physical unclonable functions (PUFs) by the combined use of ring oscillators and current mirrors that randomly select memristors embedded in a nanocrossbar array.

The memristors of the array are described by a model consisting in a charge-controlled branch relationship, which speeds up the electric simulation and allows a straightforward assignment of the device parameters that establishes the aleatory behavior of the hardware security scheme.

In addition, the most commonly used metrics have been calculated in order to determine the quality of the proposal, namely uniformity, uniqueness and bit-aliasing.

Resumen

En este trabajo se ha desarrollado una nueva propuesta de *Hardware Security* (HS) basada en el uso de memristores.

La propuesta se centra en incluir al memristor en la generación de funciones físicas no clonables a través del diseño combinado de osciladores de anillo con espejos de corrientes que seleccionan aleatoriamente a memristores colocados en un arreglo de barras cruzadas (nanocrossbar array).

Los memristores del arreglo están descritos por una función de rama controlada por carga expresada en forma totalmente analítica, lo que añade facilidad de uso para la simulación eléctrica del sistema y para la asignación de los parámetros que establecen el comportamiento aleatorio del esquema de HS.

Además, las métricas más frecuentemente utilizadas para evaluar el comportamiento de esquemas de HS han sido aplicadas, es decir, uniformidad, unicidad y porcentaje de enmascaramiento de bits.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent developments in the topics related to memristor and memristive circuits and systems represent a thrust in novel applications of the fourth fundamental element of Circuit Theory, as stated by L.O. Chua in his seminal work [6].

As security has gained the attention in all aspects of human life, so has been the case in the development of integrated circuits, from inception to actual fabrication. The starting point of the discussion establishes that because safety is a paramount to every being in the planet, it must be also for science and industry.

It clearly results that one cannot overlook the implications that security possesses for new discoveries or technological developments, in the sense that both must be free from any kind of interference or assault.

At software level, there are well-known forms of authentication in order to grant access to users to a particular platform. Secret keys and double- of even triple- gates for legitimizing the access have been set up more recently.

However linking the concept of security at hardware level has gained the attention just in the last years, as a middle of avoiding assaults to IC designs. This is called *Hardware Security* (HS). The memristor can play an important role in this new area because for starting it is a passive element, i.e. its inclusion does not compromise power consumption.

In this work, a HS proposal is introduced. It consists of using the memristor as a key-element for generating *Physical Unclonable Functions* (PUFs).

1.1 Motivation

Modeling of memristors is a topic that has been developed in the last years by scholars and researchers. Initially, the research in this topic ended up with a model that was able to fulfill the most important fingerprints of the memristor.

Nowadays, memristor modeling has evolved to seek for applications of the developed models in specific tasks. Therefore, this work has been motivated by the idea of using memristor models as a promising alternative for HS applications.

1.2 Objective

The main objective of this work is to develop a methodology in order to generate a memristor-based HS scheme that rests its functioning on the dynamics of the memristor.

1.3 Hypothesis of the work

In the literature, there are several HS schemes that consist of ring oscillators based PUFs and memristors that are modeled in a very simple way. In the new proposal, we will try to use the time-varying resistance of the memristor that is recast as a charge-controlled branch function, in a HS scheme that has an embedded nanocrossbar array.

1.4 Methodology

The methodology is outlined as follows:

- Carry out a brief description of the state-of-the-art of memristive HS.
- Introduce the memristor model to be used in the current proposal.
- Analyze thoroughly a pair of previous works with the aim of establishing their pros and cons.
- Introduce the HS proposal as a systemic view and bring it to circuit level.
- Determine the metrics of the proposal and achieve a comparison with previous works.

1.5 Organization of the thesis

The manuscript is further organized as follows: in Chapter 2, the HS fundamentals are introduced and the use of the memristor in HS is highlighted; in Chapter 3, the new proposal is presented and developed by following a top-to-bottom design approach; in Chapter 4, the results of the proposed system are presented; and finally, some conclusions are drawn and a group of research lines for future work are mentioned.

Chapter 2

Basic concepts on Hardware Security

This chapter introduces the most basic concepts and definitions of HS. In particular, special attention is given to the most commonly used metrics that are used to establish the quality of a HS scheme. Due to the fact, this thesis is focussed on the use on the memristor as a fundamental building block for the development of the proposed HS scheme, a brief description of the model that describes the memristor used in the proposal is also given.

2.1 A glimpse on HS

Hardware has long been regarded as a reliable tool that supports the entire computer system. Therefore, Hardware-related Security research has been associated to hardware implementations of cryptographic algorithms, where the cryptographic algorithms are used to improve computational performance and efficiency for cryptographic applications [7]. However, in recent years, the evolution of Hardware Security research has moved away from the classical Hardware Trojan Detection [8, 9, 10] and now leans towards trustworthy hardware development for the construction of the root-of-trust [11, 12].

A breakthrough in developing HS schemes has been the use of the intrinsic properties of hardware devices. One leading example is the development of PUFs which rely on device process variations to generate chip-specific fingerprints in the format of challenge-response pairs. Looking beyond MOSFETs, researchers are investigating the use of emerging technologies, such as the spin-transfer torque device, the memris-

tor, and spintronic domain wall, leveraging their special properties for HS applications [13, 14, 15].

2.2 Physical Unclonable Functions

PUFs are hardware-based security primitives introduced in 2007 [16]. PUFs use the intrinsic manufacturing variations in a device to generate a fingerprint of the hardware that offers the valuable advantage of unclonability. This means that the device cannot be cloned even when a hacker has physical access to the device. Therefore, PUFs are unique to their device and can be used as a security primitive to enable device-based identification, authentication, and secret key generation.

In recent years, researchers have proposed versatile security solutions using PUFs as an alternative root-of-trust to conventional cryptographic solution using black-box models [17]. For example, PUFs are used in device identification and authentication, binding software to hardware platforms, secure storage of cryptographic secrets, and secure protocol designs [18, 19].

Because, the functioning of the PUF itself rests on variations of distinct nature, it clearly results that the data derived from PUFs is often highly sensitive to environmental changes and the physical conditions where the device is being tested. Therefore, different types of PUFs have been used for the purpose of identification and authentication of circuits, where a certain margin of error rate is tolerable. However, even a small amount of variation in the PUFs responses under different conditions can prevent them from being utilized in key generation because the keys used for encryption needs to be perfectly reproducible to decrypt the messages.

The reading of a PUF to given input is denoted as *the response*, while the input itself constitutes *the challenge*. Both signals form a *challenge-response pair* (CRP).

By considering the number of of possible CRPs, PUFs can be generally classified into two broad categories: *strong* and *weak*. Weak PUFs leverage the manufacturing variability and allow digitization of some fingerprints of the hardware device. The number of responses in a weak PUF is directly proportional to the number of components in the device used for generation of CRPs [13]. This fact results in a small number of CRPs with stable responses which are usually robust to environmental conditions. Weak PUFs are generally used for secret key generation because the responses are more stable and hence easily reproducible. Strong PUFs can support a

large number of CRPs. Ideally, if the number of unique CRPs is high, even though an attacker gets temporary accesses to the system, he/she will not be able to apply all the responses (brute force attack) and get access to the system. Hence, strong PUFs are generally used for authentication [13]. However, a large set of PUF responses may offer stronger cryptographic strength as it leads to longer cryptographic keys [20]. Figure 2.1 depicts a taxonomy of PUFs. Further information on PUF taxonomy can be found in [1].

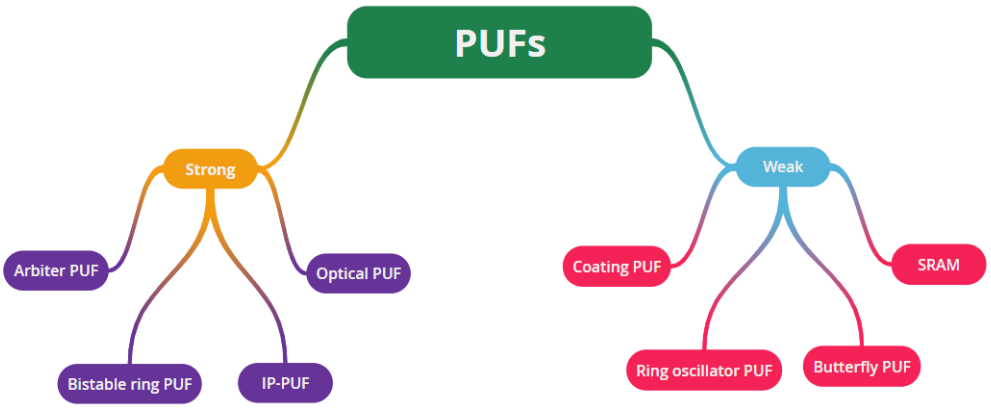


Figure 2.1: Classification of PUFs

2.3 Metrics

Independently of the type of a PUF, the quality of a PUF is determined by metrics that can be the result of its variability on a specific application. Since different types of applications have different sets of requirements, not all metrics are of equal importance [1]. Figure 2.2 shows a graphical classification of the most commonly used metrics for PUFs.

In order to evaluate the performance of PUFs, it is necessary to apply certain criteria to measure their quality; the most commonly used metrics are described [21].

Figure 2.2: PUF metrics [1]

2.3.1 Uniformity

This metric measures the ratio of "1" and "0" bits in a response bit string. That is, it ensures the randomness of the response of a PUF instance. Let $r_{i,j}$ be the j-th bit of the i-th response (R_i), then the uniformity of the i-th PUF instance is given by:

$$Uniformity = \frac{1}{n}\sum_{l=1}^{n} r_{i,j} \times 100\% \tag{2.1}$$

when the challenge remains constant. The ideal value for uniformity is 50%, which implies a balanced ratio of "1" and "0" in a particular response bit-string. The average Uniformity value is obtained simply by averaging all the values of all PUF instances.

2.3.2 Uniqueness

This metric measures the average inter-chip Hamming distance (HD) of the response obtained from a group of chips. The HD of two strings of bits is simply the number of bits in which the strings differ. It quantifies how different one chip is from another. An ideal PUF has a uniqueness value of 50%.

$$Uniqueness = \frac{1}{\binom{x}{2}}\sum_{i=1}^{x-1}\sum_{j=i+1}^{x}\frac{HD(R_i, R_j)}{n} \times 100\% \tag{2.2}$$

2.3.3 Bit-aliasing

This metric measures the tendency of different PUF instances to produce nearly identical responses, which is an unwanted result. Another definition of bit-aliasing is the affinity of a response bit towards "1" or "0", the ideal value for it is 50%. Let $r_{i,j}$ be the j-th bit of the i-th response (R_i), then the bit alias of the position of the j-th bit is given by:

$$\textit{Bit-aliasing} = \frac{1}{x}\sum_{i=1}^{x} r_{i,j} \times 100\% \tag{2.3}$$

2.4 Memristor in Hardware Security

The discovery of the actual memristor has pushed forward the research on several fields such as modeling and fabrication, as well as applications of memristive circuits.

With the recent advances on memristors as potential building blocks for future hardware design, it becomes an important and timely topic to study the role that memristors may play in hardware security. The main idea of incorporating the memristor into a HS scheme is to use the device intrinsic fingerprints to produce security primitives, i.e. a PUF that uses the parameter variations of the memristor as new guidelines for PUF generation. In the following sections, two memristor models for HS applications are described.

2.4.1 A voltage-controlled memristor model

The first model is defined as a threshold-type switching model [2] of a *voltage-controlled* memristive device that attributes the switching effect to the modulation of tunneling.

The memristive behavior is given by:

$$I(t) = G(L,t)V_M(t) \tag{2.4}$$

and

$$\dot{L} = f(V_M, t) \tag{2.5}$$

Herein, L is defined as the tunnel barrier width which is given as a time-function of the applied voltage V_M, G is the device conductance and I the device current.

Besides, the memristance (the device tunneling resitance R_t) is given as:

$$Rt(L_{V_M,t}) = f_0 \cdot \frac{e^{2L_{V_M,t}}}{L_{V_M,t}} \tag{2.6}$$

and

$$L(V_M, t) = L_0 \cdot (1 - \frac{m}{r(V_M, t)}) \tag{2.7}$$

where f_0 is a fitting parameter, L_0 is the maximum value that $L(V_M, t)$ can attain and m is a fitting parameter that determines the boundaries of the barrier width.

The resulting model is recast into the function of memristance derivative:

$$\dot{r}(V_M, t) = \begin{cases} a \cdot \frac{V_M + V_{th}}{c + |V_M + V_{th}|}, & V_M \epsilon [-V_0, V_{RESET}] \\ b \cdot V_M, & V_M \epsilon [V_{RESET}, V_{SET}] \\ a \cdot \frac{V_M - V_{th}}{c + |V_M - V_{th}|}, & V_M \epsilon [V_{SET}, +V_0] \end{cases} \tag{2.8}$$

Herein, the parameters a, b, and c are fitting constants that are used to shape the rate of memristance change.

The memristive system defined by Equations 2.4–2.8 is implemented as a circuit equivalent by combining two current sources, G_{pm} and G_r, an integrating capacitor C_r (modelling the memory effect of the memristor) and a resistor R_{aux}. The current source G_r generates a current based on Equation 2.8. The voltage across the capacitor C_r defines the value of parameter $r(V_M, t)$ and the plus and minus terminals of the controlled-current source G_{pm}, represent the top and bottom electrodes of the device.

The model is implemented as a SPICE subcircuit, and the corresponding values of the fitting parameters of the model can be passed on to subcircuit as arguments.

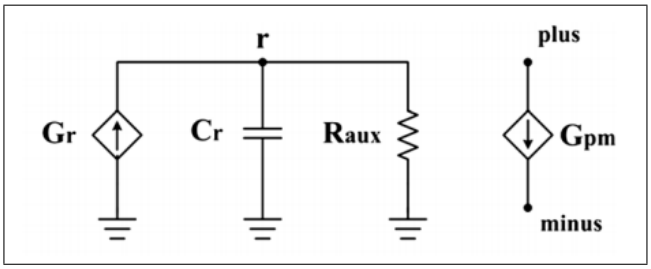
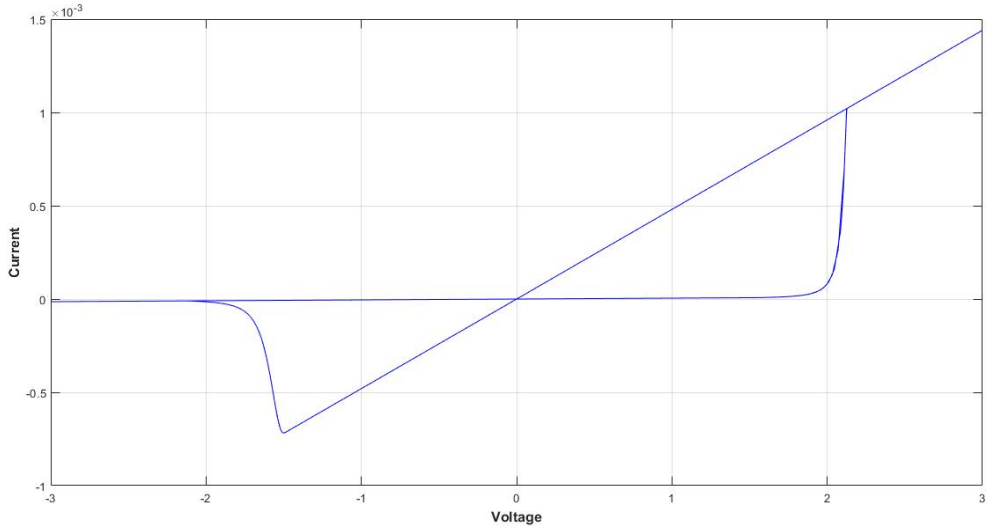


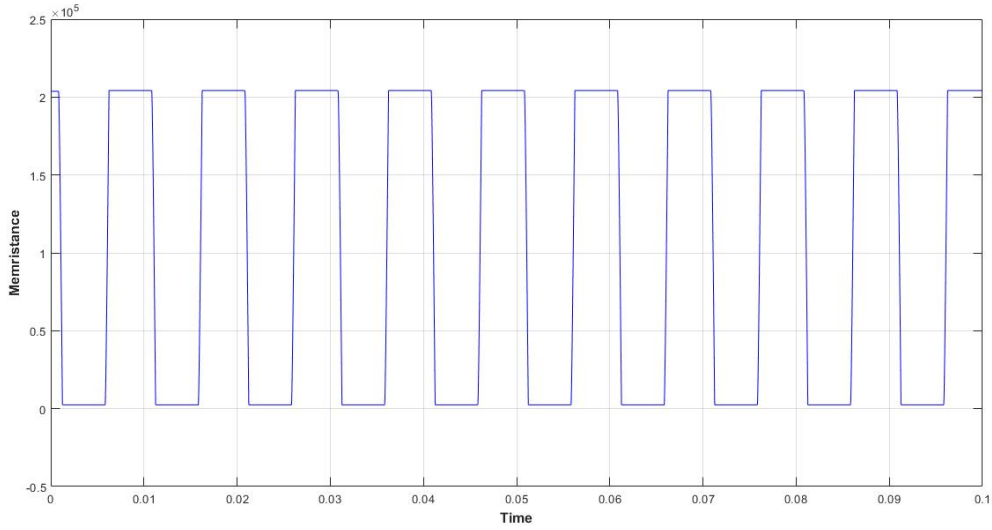
Figure 2.3: Macromodel of the memristor [2]

On the one hand, the model has been simulated in Matlab and the reproduction of the graphs presented in [2] is shown in Figure 2.4. From Figure 2.4b, it is possible

to observe that the *on-state resistance* (R_{on}) and the *off-state resistance* (R_{off}) are 2 $K\Omega$ and 200 $K\Omega$ respectively.

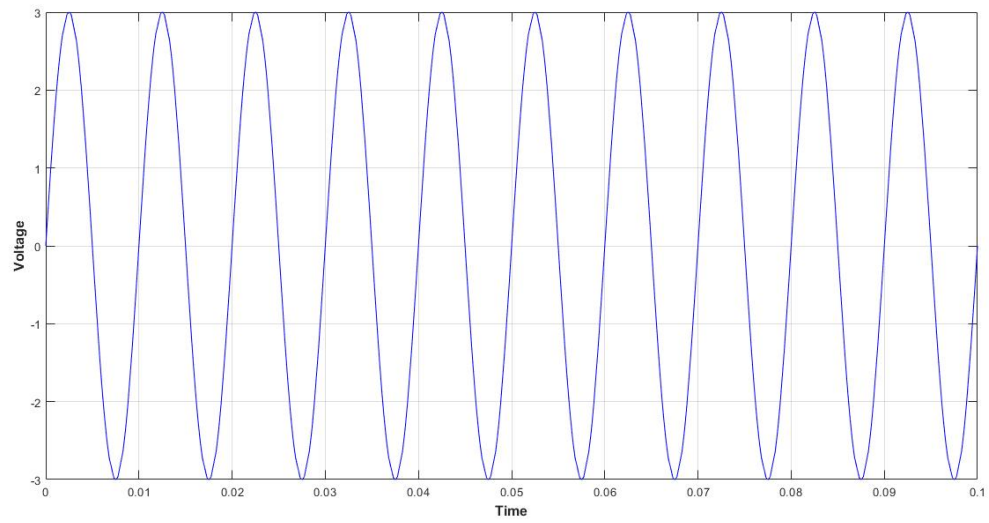


(a)

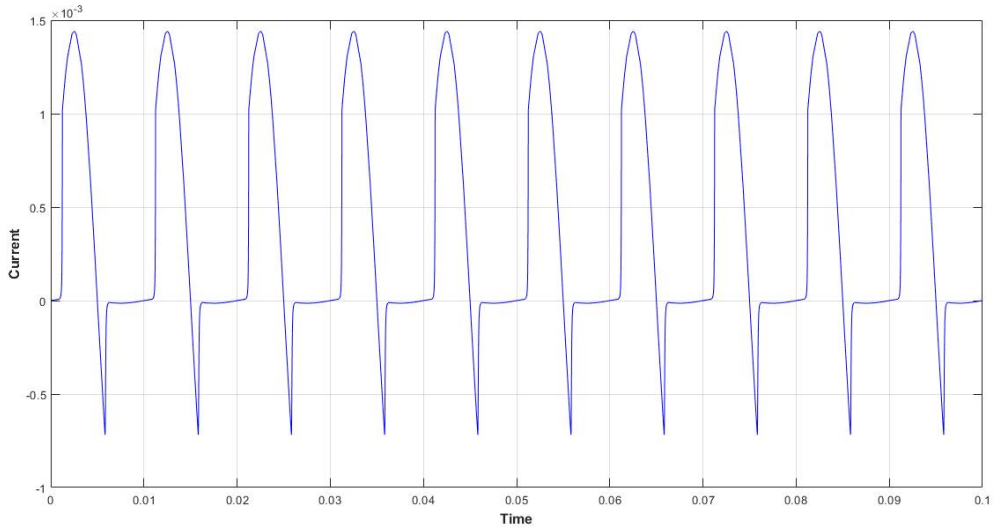


(b)

Figure 2.4: Voltage-controlled memristor model response to a $3V$ and $100Hz$ sinusoidal applied voltage. In (a) the hysteresis loop current-voltage characteristic shows the existence of threshold voltages around $|1.5|V$ and b) presents device memristance switching within the valid value range



(a)



(b)

Figure 2.5: Voltage-controlled memristor model response to a $3V$ and $100Hz$ sinusoidal applied voltage. The plots illustrate the applied voltage and memristor current as a function of time

On the other hand, electric simulation of the macromodel from Figure 2.3 has been achieved with HSPICE. Figures 2.6–2.11 show the pinched hysteresis loops of the memristor for several conditions of the device parameters.

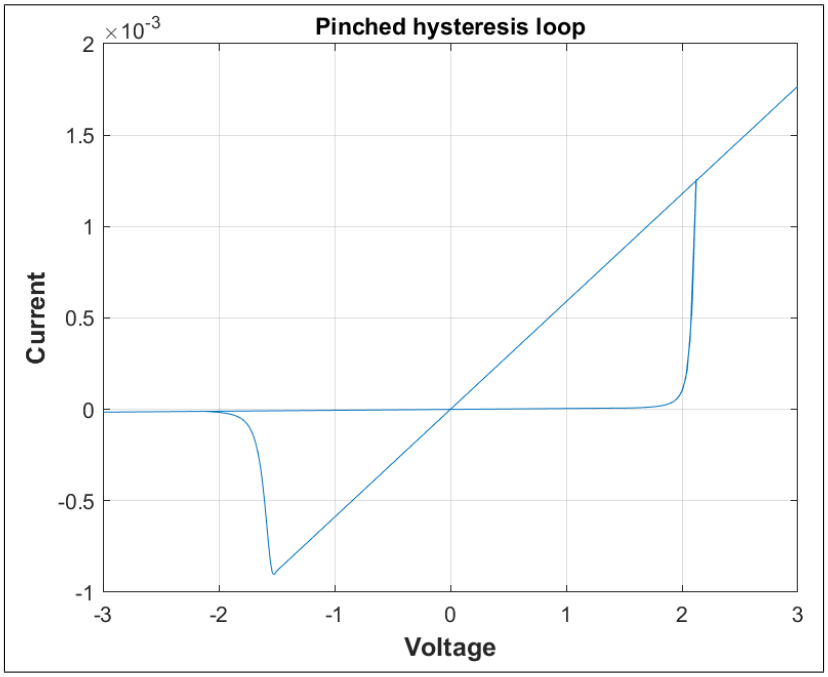


Figure 2.6: Memristor model with R_{min} and R_{max} at 90 and 380 respectively

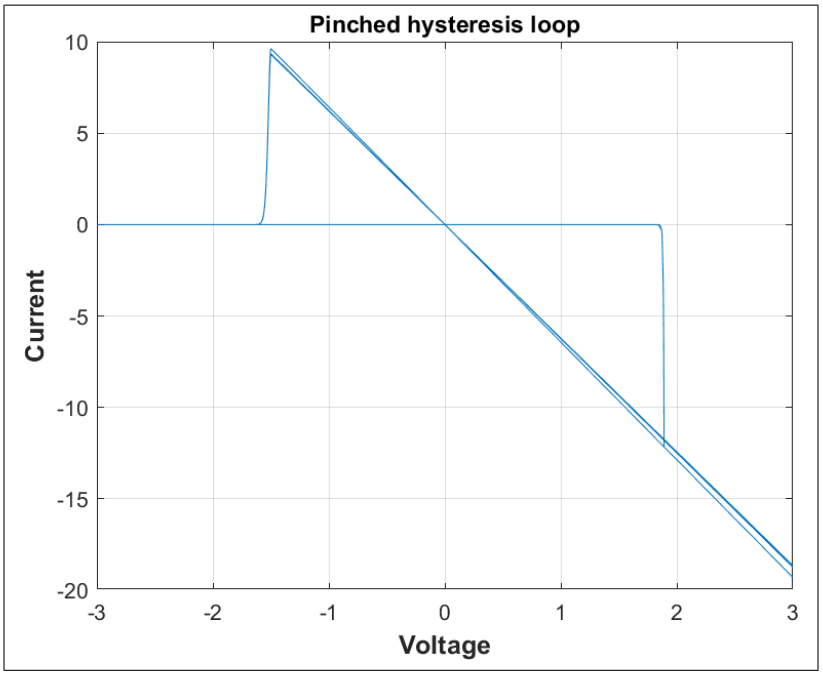


Figure 2.7: Memristor model with R_{min} and R_{max} at 50 and 200 respectively

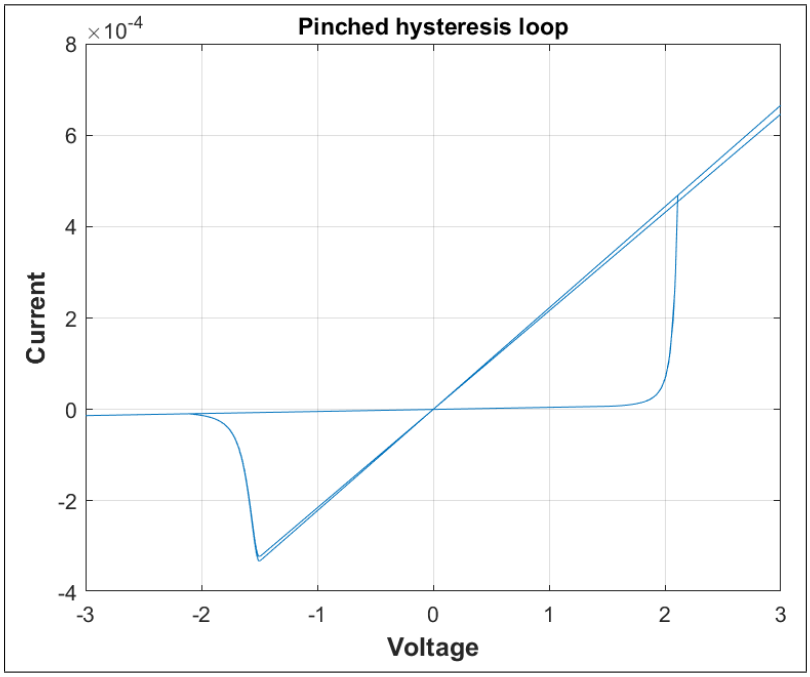


Figure 2.8: Memristor model with R_{min} and R_{max} at 120 and 400 respectively

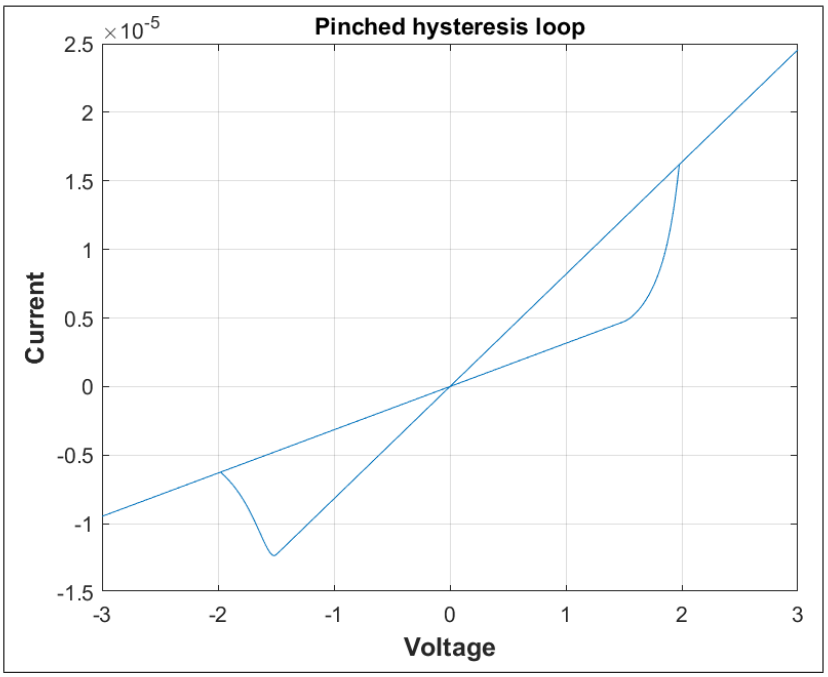


Figure 2.9: Memristor model with R_{min} and R_{max} at 300 and 500 respectively

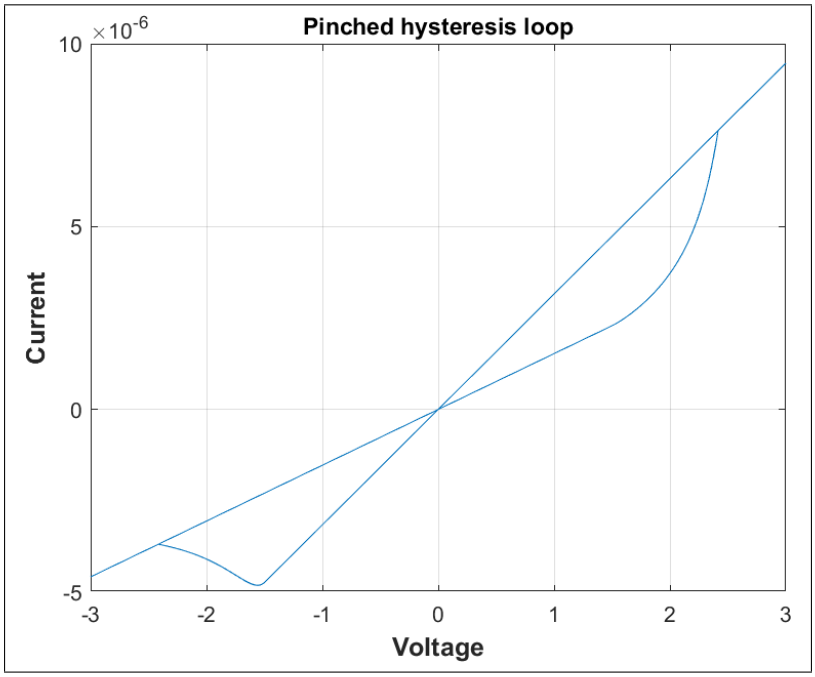


Figure 2.10: Memristor model with R_{min} and R_{max} at 500 and $1k$ respectively

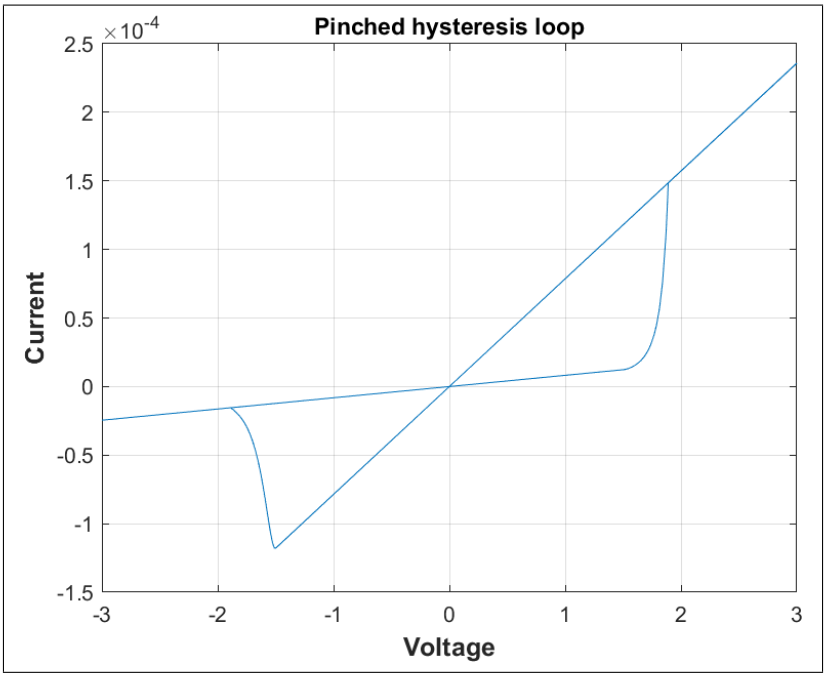


Figure 2.11: Memristor model with R_{min} and R_{max} at 150 and 300 respectively

2.4.2 A charge controlled memristor model

This model has been reported in [22], and it has been recast as a *charge-controlled* memristance analytic function. The methodology to obtain such as function for the memristance can be described as: firstly, the nonlinear drift mechanism is expressed as a function of charge with a state variable that is limited by the window function of Joglekar [23]. Then, a homotopy-perturbation method is used to find a symbolic solution to the nonlinear equation for the normalized state variable $x(q)$; and finally, $x(q)$ is used to generate the charge-controlled memristance by substituting in the coupled resistor equivalent from [4].

In the following, the set of nested equations for order 1 to 3 with $k = 1$ (each model of order greater than one contains the model of previous order). The notation used in the equations below is with the form: $M_{O_j k_i \eta^{sign}}$, where the index i determines the value of k of the Joglekar function and the index j is the order O of homotopy. In addition, the direction of the drift [22] is given by η that takes the value of $\eta = \pm 1$.

Equations for η^-

A first set of model equations is defined for the negative polarity of the drift, namely as $\eta = -1$.

$$M_{O_1,k_1,\eta^-} = \begin{cases} R_d(X_0-1)[(X_0-2)e^{4\kappa q} - (X_0-1)e^{8\kappa q}] + R_{on} & q \leq 0 \\ R_d X_0[X_0 e^{-8\kappa q} - (X_0+1)e^{-4\kappa q}] + R_{off} & q > 0 \end{cases} \tag{2.9}$$

$$M_{O_2,k_1,\eta^-} = M_{O_1,k_1,\eta^-} + R_d \begin{cases} (X_0-1)^3[-e^{4\kappa q} + 2e^{8\kappa q} - e^{12\kappa q} & q \leq 0 \\ X_0^3[-e^{-12\kappa q} + 2e^{-8\kappa q} - e^{-4\kappa q} & q > 0 \end{cases} \tag{2.10}$$

$$M_{O_3,k_1,\eta^-} = M_{O_2,k_1,\eta^-} + R_d \begin{cases} (X_0-1)^4[e^{4\kappa q} - 3e^{8\kappa q} + 3e^{12\kappa q} - e^{16\kappa q}] & q \leq 0 \\ X_0^4[e^{-16\kappa q} - 3e^{-12\kappa q} + 3e^{-8\kappa q} - e^{-4\kappa q}] & q > 0 \end{cases} \tag{2.11}$$

Equations for η^+

A second set of model equations is defined for the positive polarity of the drift, namely as $\eta = +1$.

$$M_{O_1,k_1,\eta^+} = \begin{cases} R_d X_0[X_0 e^{8\kappa q} - (X_0+1)e^{4\kappa q}] + Roff & q \leq 0 \\ R_d(X_0-1)[(X_0-2)e^{-4\kappa q} - (X_0-1)e^{-8\kappa q}] + R_{on} & q > 0 \end{cases} \tag{2.12}$$

$$M_{O_2,k_1,\eta^+} = M_{O_1,\kappa_1,\eta^+} + R_d \begin{cases} X_0^3[-e^{12\kappa q} + 2e^{8\kappa q} - e^{4\kappa q}] & q \leq 0 \\ (X_0-1)^3[-e^{-4\kappa q} + 2e^{-8\kappa q} - e^{-12\kappa q} & q > 0 \end{cases} \tag{2.13}$$

$$M_{O_3,k_1,\eta^+} = M_{O_2,\kappa_1,\eta^+} + R_d \begin{cases} X_0^4[e^{16\kappa q} - 3e^{12\kappa q} + 3e^{8\kappa q} - e^{4\kappa q}] & q \leq 0 \\ (X_0-1)^4[e^{-4\kappa q} - 3e^{-8\kappa q} + 3e^{-12\kappa q} - e^{-16\kappa q}] & q > 0 \end{cases} \tag{2.14}$$

In these equations, X_o represents the initial condition of the state variable and the useful resistance range is given as $R_d = R_{off} - R_{on}$. In addition, κ is an auxiliary variable given as:

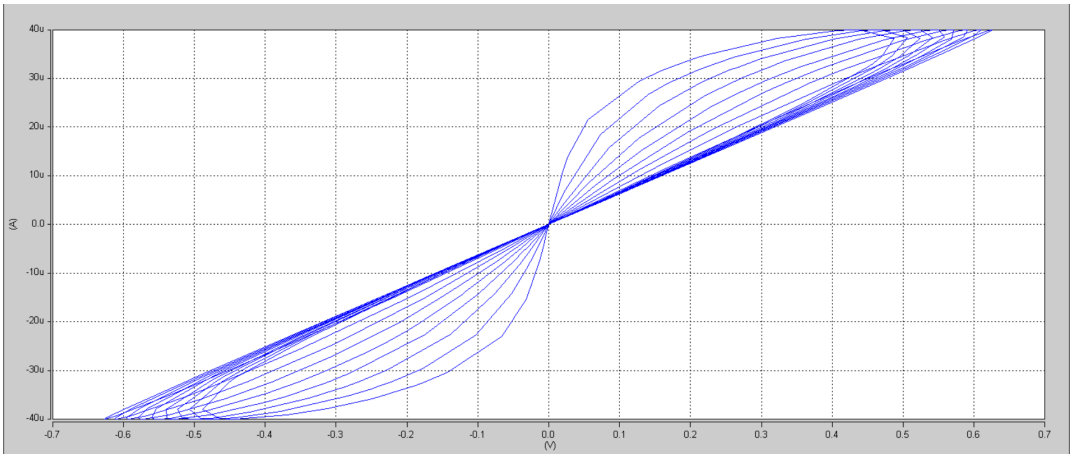
$$\kappa = \frac{\mu_v R_{on}}{\Delta^2} \tag{2.15}$$

Where Δ is the total length of the device and μ_v is the mobility of the charges in the doped region. Table 2.1 shows the typical parameters for the HP memristor.

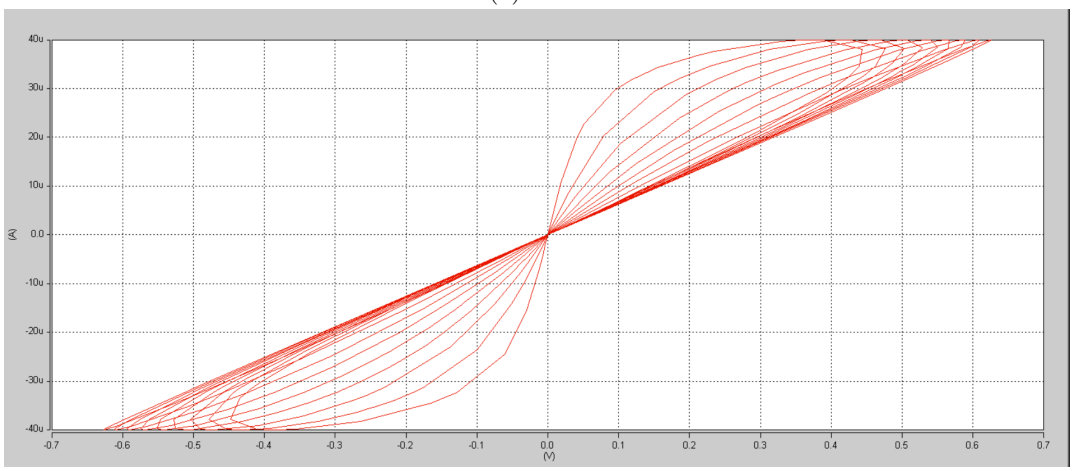
Memristor parameters					
μ_v $[\frac{m^2}{Vs}]$	Δ $[nm]$	κ $[\frac{m}{As}]$	R_{on} $[\Omega]$	R_{off} $[\Omega]$	Ap $[\mu A]$
1×10^{-14}	10	10000	100	16×10^3	40

Table 2.1: HP memristor parameters [4]

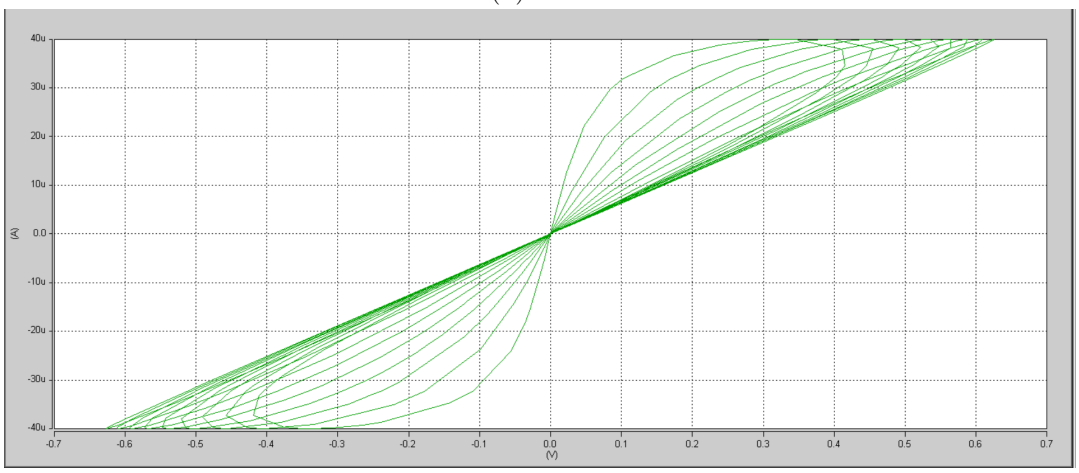
The voltage-current plots of the charge-controlled model generated by HSPICE are presented in Figures 2.12 and 2.13. They fulfil the fingerprint related to the pinched hysteresis loop of the *i-v* characteristics.



(a) O1

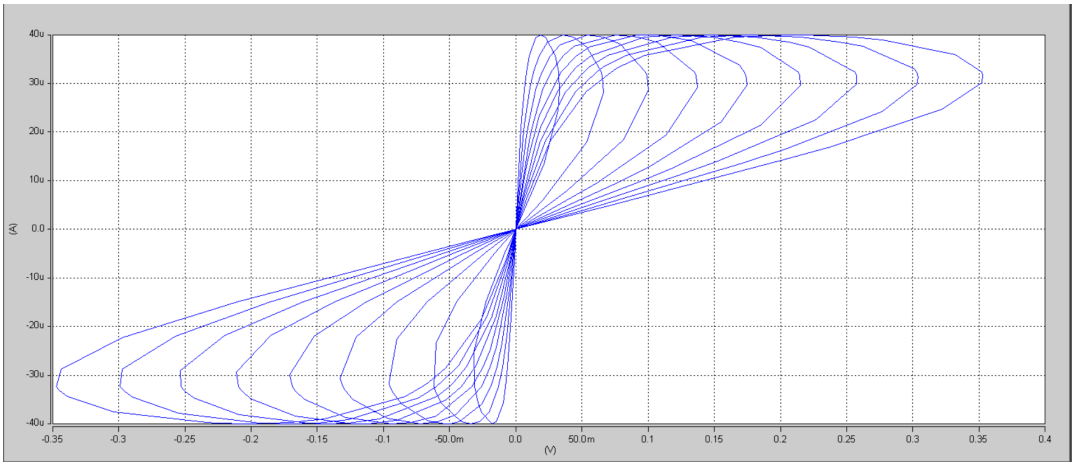


(b) O2

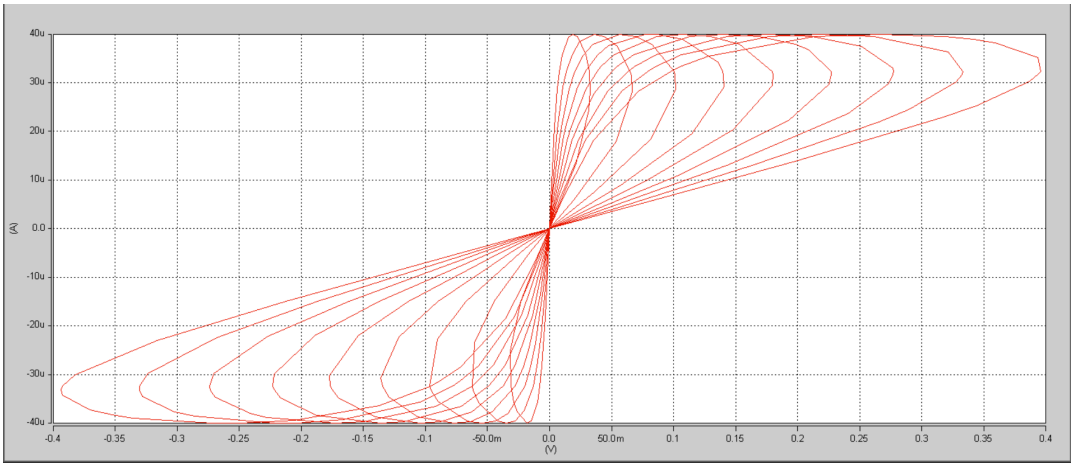


(c) O3

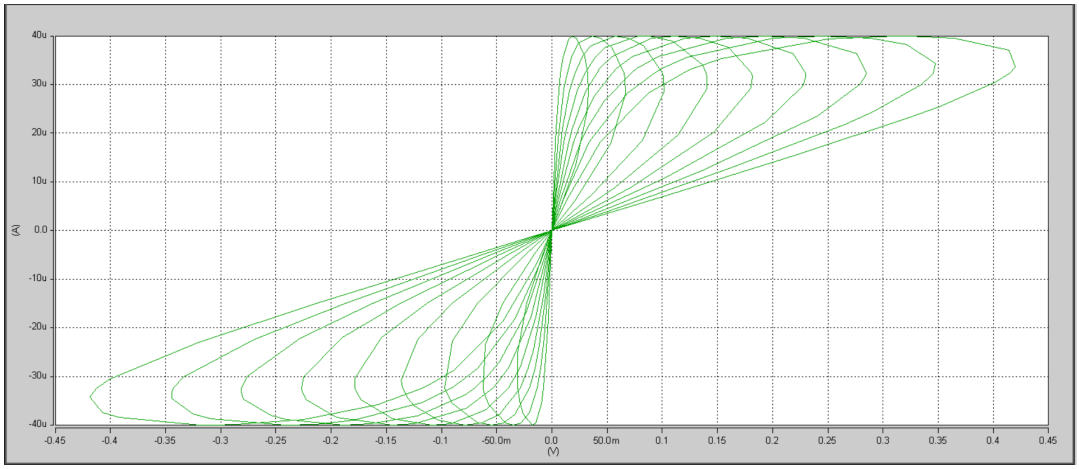
Figure 2.12: Voltage-Current plot for $O1, O2, O3$ with $k = 1, \omega = 1, \eta = -1$ and the sweep for X_o is given as $X_o = 0.1 : 0.9 : 0.1$



(a) O1



(b) O2



(c) O3

Figure 2.13: Voltage-Current plot for $O1, O2, O3$ with $k = 1$, $\omega = 1$, $\eta = +1$ and the sweep for X_o is given as $X_o = 0.1 : 0.9 : 0.1$

From the two expounded models, it is worthy to mention that the voltage-controlled model assumes a piece-wise solution for the nonlinear drift differential equation, while the charge-controlled model results in a continuous time-dependent memristance function.

It is important to notice that the charge-controlled model fulfills all memristor fingerprints and the resulting pinched hysteresis loops posses less discontinuities than the model from [2]. On top of this, the charge-controlled model is recast in a fully analytic form which simplifies all needed evaluations during electric simulation. This allows us to use the charge-controlled model as the fundamental building block in our proposal of a memristor-based HS methodology.

Chapter 3

Memristive ROs in HS schemes

In this chapter, the incorporation of the memristor in Ring Oscillator circuits is introduced with the aim of emphasizing the use of the device in the proposal for a memristive HS scheme.

3.1 Oscillator

This section gives a glimpse on the most common concepts and classification related to oscillator circuits.

An oscillator is defined as an autonomous circuit that converts DC power into a periodic waveform. There are many types of oscillators, and many different circuit configurations that produce a variety of periodic waveforms. However, the most commonly used waveforms are reduced only to two different types: the sinusoidal signals and the pulsed signals. Usually, sinusoidal oscillations are used in analog applications while pulse oscillators are mainly present in digital systems.

In plain words, an oscillator can be represented as a feedback system as depicted in the block diagram in Figure 3.1. The closed-loop gain is given as:

$$\frac{V_{out}}{V_{in}} = \frac{A(s)}{1 - A(s)\beta(s)} \tag{3.1}$$

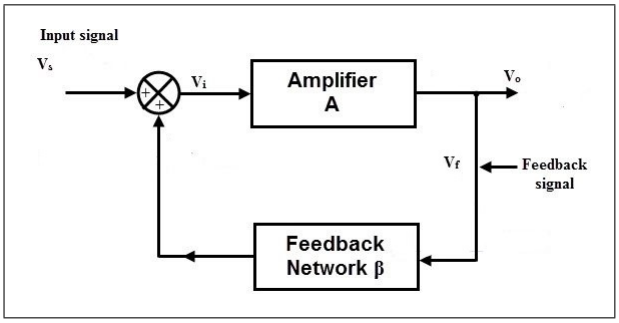


Figure 3.1: Block diagram of a feedback system circuit

Oscillators work on the principle of positive feedback or regenerative feedback [24]. That is, a fraction of the output (from the amplifier) is added back to input with proper magnitude and phase. The output is sustained even though the input is removed. In order to guarantee this, the Barkhausen criterion has to be satisfied [25]:

Condition 1. Magnitude of overall gain around the loop should be unity

$$|\beta A| = 1 \tag{3.2}$$

Condition 2. Overall phase shift around the loop should be either zero or multiple of 360 degrees.

$$\angle \beta A = 2\pi n, n \in \{0, 1, 2, ...\} \tag{3.3}$$

3.1.1 Classification of oscillators

Oscillators can broadly be classified into two main categories [26]: Harmonic Oscillators (also known as Linear Oscillators) and Relaxation Oscillators.

Harmonic Oscillators produce a sinusoidal wave output signal. Ideally, the output signal is of constant amplitude with no variation in frequency. The sinusoidal oscillators may be further sub-divided into:

1 **Tuned circuit oscillator**: These oscillators use a tuned circuit consisting of inductors and capacitors and are used to generate high frequency signals. Such oscillators are Hartley, Colpitts, Clapp oscillators etc.

2 **RC oscillators**: These oscillators use resistors and capacitors and are used to generate low or radio frequency signals, such as phase-shift and Wien bridge oscillators.

3 **Crystal oscillators**: These oscillators use a piezoelectric crystal (commonly a quartz crystal). The crystal mechanically vibrates as a resonator, and its frequency of vibration determines the oscillation frequency. Such are Pierce, Tri-tet and Butler oscillators.

4 **Negative-resistance oscillator**: These oscillators use a resonant circuit, such as an LC circuit, crystal or cavity resonator, which is connected across a device with a negative differential resistance, and a DC bias voltage is applied to supply energy. A resonant circuit by itself is "almost" an oscillator; it can store energy in the form of electronic oscillations if excited, but because it has electrical resistance and other losses the oscillations are damped and decay to zero. The negative resistance of the active device cancels the (positive) internal loss resistance in the resonator, in effect creating a resonator with no damping, which generates spontaneous continuous oscillations at its resonant frequency.

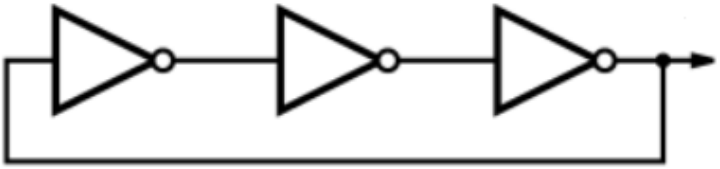
Relaxation Oscillators produce a non-sinusoidal output, such as a square, sawtooth or triangle wave. They consist of an energy-storing element and a nonlinear switching device connected in a feedback loop. The switching device periodically charges and discharges the energy stored in the storage element thus causing abrupt changes in the output waveform. Some of the more common Relaxation oscillators are listed:

- Multivibrator
- Ring oscillator
- Delay-line oscillator
- Pearson-Anson oscillator
- Function generation

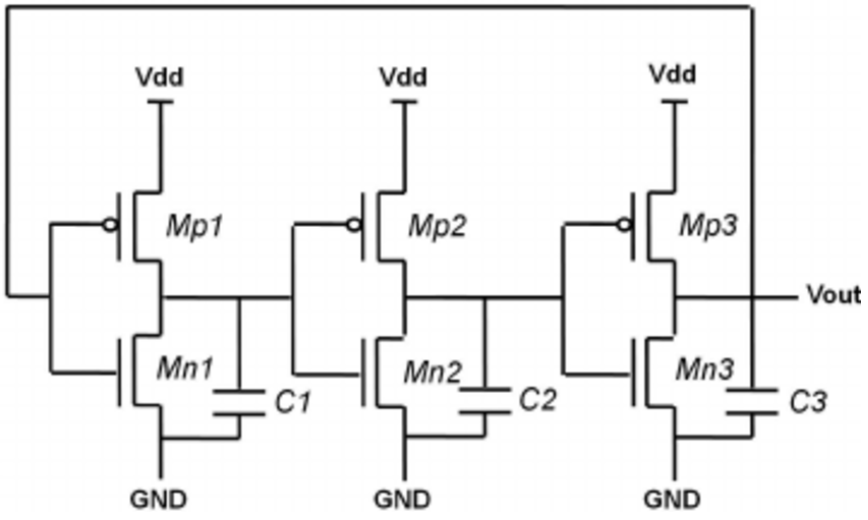
In our case, we will get focus on ring oscillators.

3.2 Ring oscillators

A ring oscillator is made of a cascade of inverter stages in a feedback configuration. The simplest feasible implementation of a ring oscillator consists of 3 inverter stages, as depicted in Figure 3.5a. Theoretically, ring oscillators are composed of an odd number of inverting stages; in the case of a single stage, no oscillation occur because the second condition of the Barkhausen criterion is not satisfied. If it were an even number of stages then, the first condition is not fulfilled because there is a negative feedback. A MOS realization of a 3 stage RO is shown in Figure 3.5b.



(a) Block diagram



(b) Schematic

Figure 3.2: Three-stages ring oscillator with inverter gates

It is well-known that the main feature of any type of oscillator is the oscillating frequency, for ROs there have been several efforts [27, 28, 29, 30] to establish suitable models that lead to accurate definitions of that frequency. Usually, all the stages are identical, therefore the frequency is only dependent on the stage delay and the number of stages. Adding more inverters in cascade generates more robustness to process variations but increases the power consumption which is an undesired characteristic.

The most commonly used expression for determining the oscillation frequency is given by [31]:

$$f_o = \frac{1}{2Nt_d} \tag{3.4}$$

Where N is the number of stages and t_d is the time delay of each stage. Then it clearly results that the accuracy on the calculation of the oscillation frequency depends on accuracy of the time delay.

3.3 Memristors in RO-PUF applications

Due to the scaling of technology to a nano-metric regime, the memristor has become an interesting option to protect integrated circuits. Memristors are attractive components to be used in PUF design because they are compatible with CMOS manufacturing standards and their sensitivity to process variations can be controlled. One of the first schemes is presented in Figure 3.3, this circuit creates its physical function by leveraging the variations in the writing time to generate a bit of information. This circuit proposal in [3] has also been used to generate bit-strings, denoted as BS, of N bits in the output and the memristor fulfills the same function of producing the physical function of the PUF.

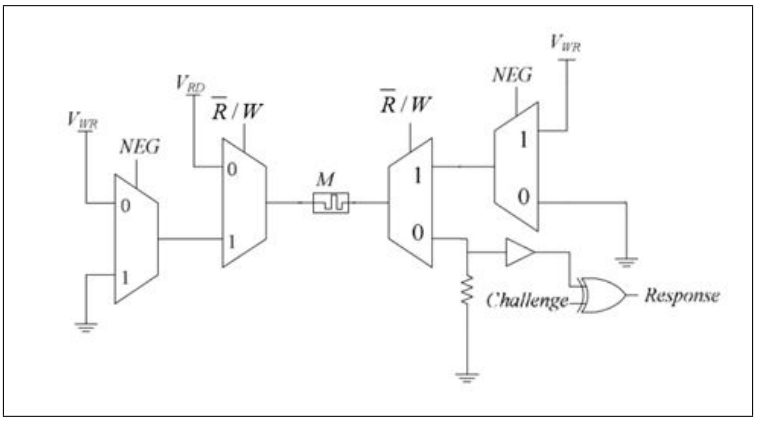


Figure 3.3: A 1-bit memristive memory-based PUF cell [3]

3.4 RO-PUF background

In 2007, Suh et al. proposed the Ring Oscillator RO-PUF [16] which is based on the delay difference among ROs to generate random bit-strings. An RO is a simple circuit of a set of inverters connected in a loop, as shown in Figure 3.4, that oscillates with a particular frequency. The simplest form of PUF generates the output logic-0 or logic-1 by comparing the frequencies of a pair of oscillator circuits. The presented configuration [16], consists of N ring oscillators with two $k - to - 1$ multiplexers which select a pair of ring oscillators (RO_i and RO_j which at output generates two frequencies (f_i and f_j respectively), two counters, and a comparator.

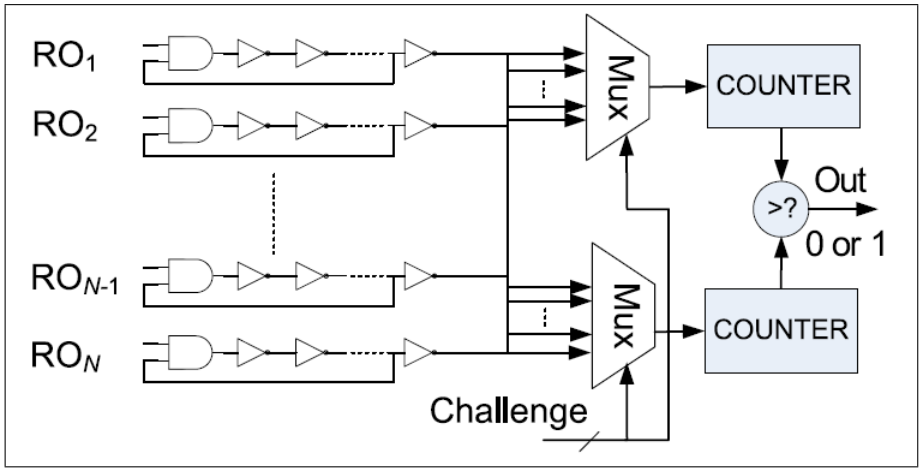


Figure 3.4: Basic structure of a ring oscillator PUF

In the next paragraphs, the operation of the RO is briefly explained.

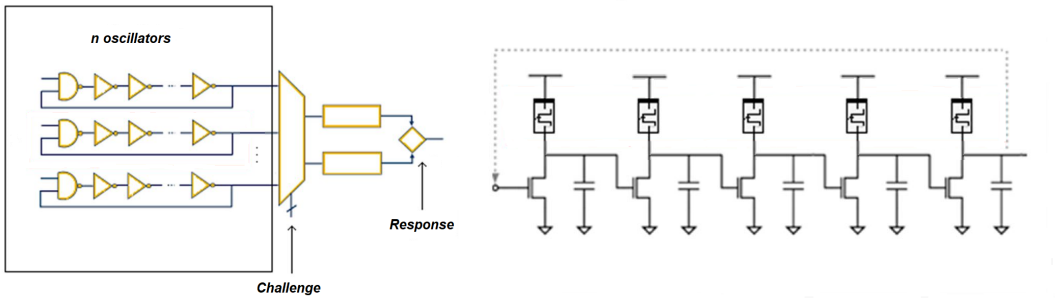
A $\binom{N}{2}$ multiplexer receives the challenge and selects 2 oscillators (RO_i and RO_j), which generate oscillations with frequencies f_i and f_j respectively. The frequencies of both signals are assessed by peak counters that act during a sufficiently large time window for determining the difference of the frequencies. A comparator proceeds to compare both counters and generates the bit response. It becomes a 0 if $f_i < f_j$, otherwise it yields a 1. This process is repeated for another selection of a pair of oscillators to generate an N-bit string.

Although RO-PUFs are better than other PUF structures in terms of robustness, a 100% error-free output is still very difficult to achieve [16].

However, the main drawbacks of RO-PUFs structures are their high energy consumption and speed limitation.

3.5 Charge controlled memristive RO-PUF

In a previous work [5], the charge-controlled memristor model introduced in the previous chapter was used in a RO-PUF. It consists of a number of memristive ROs that were randomly challenged with the use of a MUX that selected and deselected ROs in a given combinatorial sequence – as shown in Figure 3.5a. When implementing the ROs, the memristors appeared as dynamic loads of the inverter stages, as shown in Figure 3.5b. The main idea herein was to reach the steady state of the whole system which implied that every memristor became a resistive load with values that depend on the variations of the memristor model. Two forms of generating PUFs are developed. A first mapping is obtained by resorting to a permutation of n by taking 2 at a time, i.e. a total of $\binom{n}{2}$. For this mapping two variants are included in the study, namely with $n = 8$ and $n = 11$. A second mapping is obtained by taking unrepeated pairs or member of them, i.e. a total of $\frac{n}{2}$ PUFs. A summary of the resulting metrics for these mappings is given in Table 3.1.



(a) RO-PUF scheme [5] (b) Five stage RO with memristor as dynamic load

Figure 3.5: RO-PUF scheme for HS with inverter gates in common source using a memristor as load

Results					
	n	Uniformity %	BitAliasing %	Uniqueness %	Meanbit
n(n-1)/2	8	55%	52.8571%	50.7143%	14.2
n(n-1)/2	11	50.1818%	48.9091%	50.0606%	27.5333
n/2	50	51.60%	49.20%	51.4667%	12.8667
Ideal	-	50%	50%	50%	-

Table 3.1: Table of metrics for different pairing scheme [5]

Chapter 4

Proposal of a charge controlled memristive PUF

In this chapter the proposal of an HS scheme is given. The first section justifies why it is necessary to place the ideas promoted by the proposal in the broader context of the systems studied in the previous chapter. The following sections detail the structure and design of the proposed HS scheme. Finally, a section of results is given.

4.1 Systemic view of the proposal

If one may mention the pros and cons of the systems presented in the previous chapter, it could be pointed out the HS methodology from [5] resorts to a randomly tunning of the steady state memristance that constitute the dynamic loads of the RO-PUFs. The memristor model used herein comes from the solution of the nonlinear drift ordinary differential equation (ODE) which yields a fully symbolic expression for the memristance that can be recast in a Verilog-A behavioural description. The other methodology [21] is focussed on a more complex memristive system, namely, a nanocrossbar array, but in counter-position to the already mentioned work, it uses a model [2] that arises from the solution of the nonlinear drift ODE which is recast as a macromodel. From a general perspective, the methodologies used in [21, 5] are shown in Figure 4.1.

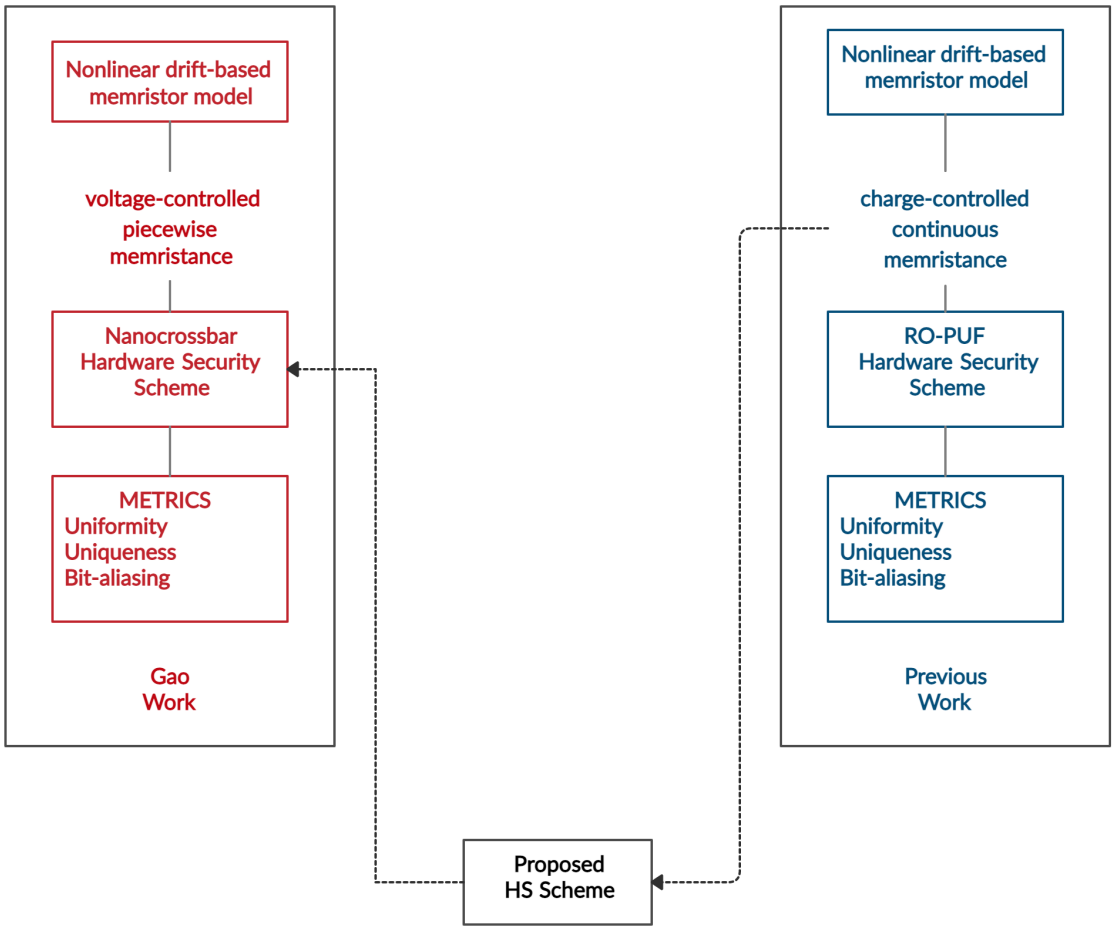


Figure 4.1: Context of the proposal

In summary, the work of [21] aims to a more complex HS array, while using a simple device model recast in a macromodel [2] that posses difficulties when evaluating during circuit simulation loops. The work of [5] has less complexity while using a more complete memristor model [22] that is recast in a very simple form.

From a systemic viewpoint, the proposal can be stated as combining the best features of both systems, i.e. the design of a nanocrossbar array using memristors with the charge-controlled memristor model, as depicted in Figure 4.2.

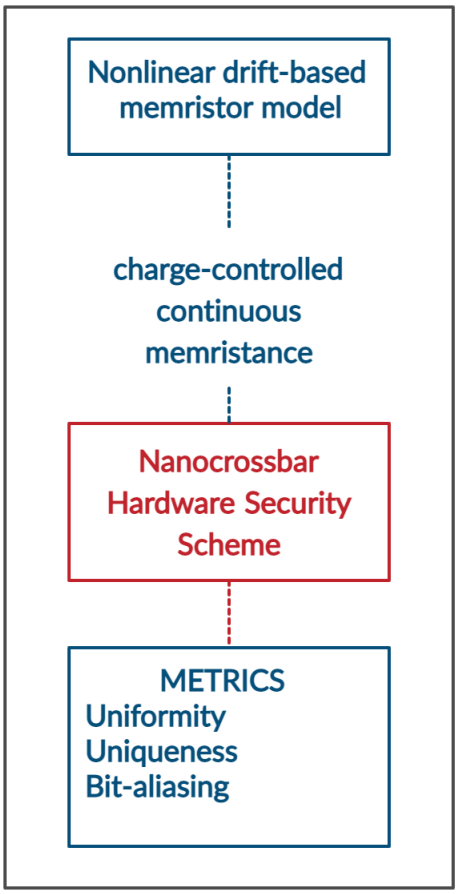


Figure 4.2: Proposed HS Scheme

4.1.1 Functional block: CRP generation of the proposal

The core idea of any HS scheme is how the challenge-response pair is generated. From the conceptual diagram from Figure 4.3, a description at the level of functional block can be devised. The challenge acts as an input signal to the selection block, which is a kind of enable switch that addresses a set of memristors in the nanocrossbar block, which contains the memristors. From the nanocrossbar, two RO blocks are activated and their outputs are compared in order to generate the response signal. In the following, each block is briefly described as a system component.

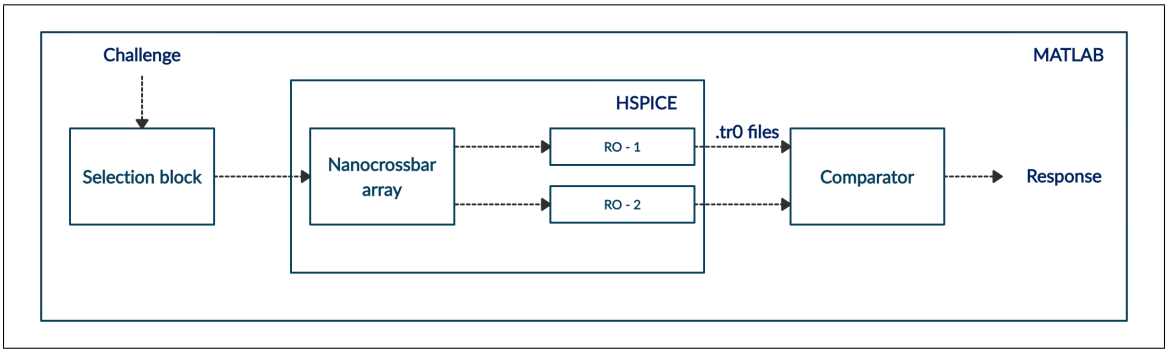


Figure 4.3: HS scheme for generating CRPs

Selection block

The selection block consists of establishing the direction of the signal, controlling the selectivity and activation of the memristors located in the nanocrossbar array. The selection is controlled by a random number generator (as shown in Figure 4.4) and uses a mapping with a permutation of $\binom{N}{2} \times M$.

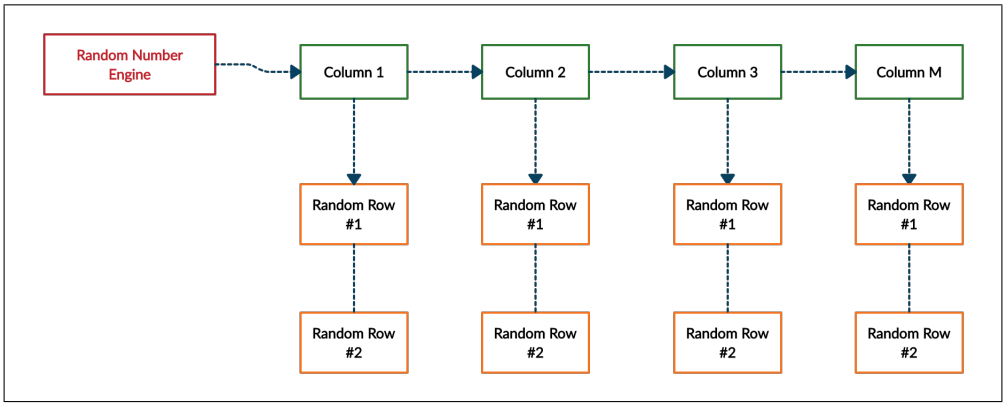


Figure 4.4: Process for randomly selecting two rows of every column from the nanocrossbar array

Nanocrossbar array

The nanocrossbar array as seen in Figure 4.5 consists of parallel and horizontal wires that are used to address a particular memristor. These memristors in the array are modeled by the charge-controlled branch relationship from [5].

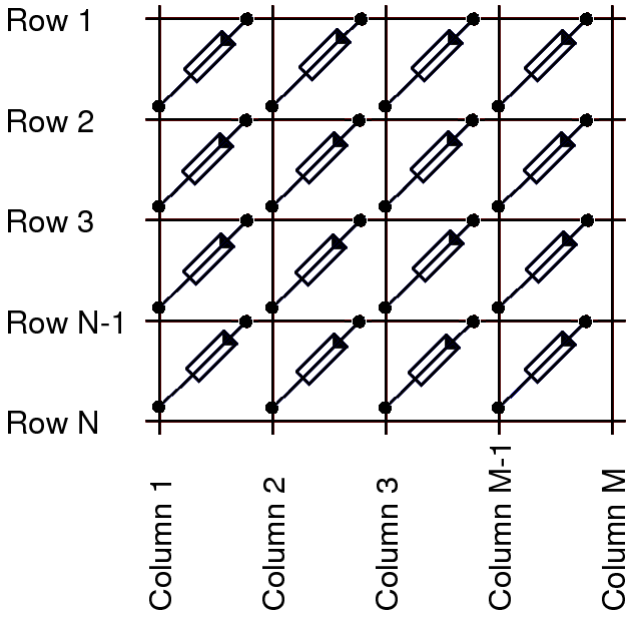


Figure 4.5: A set of memristors in a nanocrossbar array

A 10×10 nanocrossbar was implemented in this proposal. Every memristor is unique and unrepeatable by randomly setting the maximum and minimum resistances (R_{max} and R_{min}) within the ranges given as:

$$1k\Omega < R_{on} < 10k\Omega \tag{4.1}$$

and

$$5k\Omega < R_{off} < 25k\Omega \tag{4.2}$$

From the selection block, a selected pair of memristors is used within a corresponding pair of ROs defining their oscillation frequencies.

Ring Oscillator

The RO block is formed by combining 5 inverting stages and a current mirror, denoted as a CM-RO. The whole circuit (including the memristor model) is simulated in HSPICE in a transient simulation and the outcome of the simulation is a set of `.tr0` files that contains the behaviour of ROs for all memristors in the array.

Comparator

The comparator counts the number of peaks of the oscillations from the RO block, i.e. a way of determining the frequencies of a the selected pair of ROs, namely f_1 and f_2. In order to sweep the whole nanocrossbar array, this process is repeated $\binom{N}{2} \times M$. It yields a 0 if $f_1 < f_2$, otherwise 1. Comparison takes place in Matlab.

4.1.2 System description

The proposed PUF scheme is presented in Figure 4.6 and it comprises the blocks previously outlined. In this description, the system is more closely regarded at circuit level.

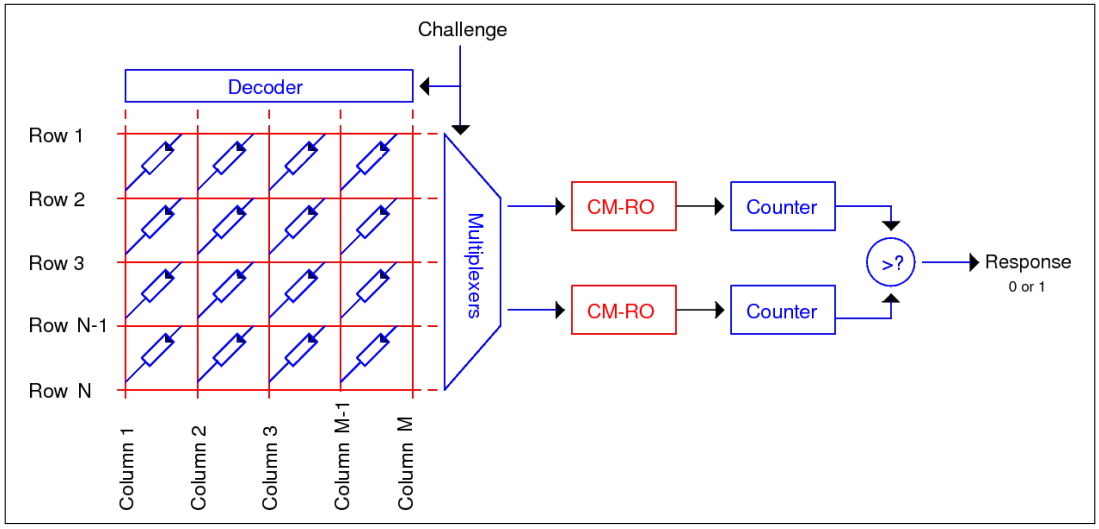


Figure 4.6: Proposed memristive RO-PUF HS Scheme

4.2 Design considerations

In this section, the design of the proposal at circuit level is outlined. The design process follows the structure of the systemic view.

4.2.1 Ring Oscillator

The importance of the analysis of a RO comes from the fact that the oscillation frequency must be determined as a key value for the correct design of the HS proposal. As given in Equation 3.4, this frequency is directly linked to the number of inverting

stages and the time delay. The time delay t_d is defined as the average of t_{dHL} and t_{dLH} which depends on the output high voltage which is labeled V_{OH}, and the output low voltage which is labeled V_{OL}. Which in turn are defined as the time required for the output to fall from V_{OH} to $(V_{OH}+V_{OL})/2$ and the time required for the output to rise from V_{OL} to $(V_{OH}+V_{OL})/2$ respectively, so on the inverters are operating between two voltages. In formula, T_d is determined by:

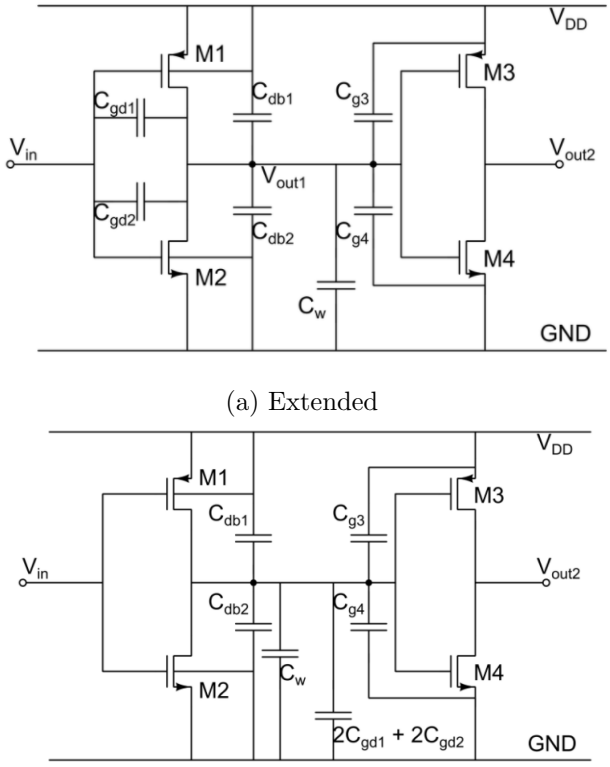
$$T_d = \frac{t_{dHL} + t_{dLH}}{2} \tag{4.3}$$

The parasitic capacitance present in the overall CMOS inverter circuit manifests as the Capacitive Load (C_L). The parasitic capacitance from both the current stage inverter and the next stage inverter is a cause of this load capacitor (C_L). Thus, for better speed, we must keep the parasitic capacitances as low as possible. This one is just the sum of all the parasitic capacitances in the inverter and the capacitive elements present in the wiring used to connect the devices together. Figure 4.7a shows the involved parasitic capacitances in the RO diagram.

The expression for the load parasitic capacitance is given as:

$$C_L = C_{g3} + C_{g4} + C_{db1} + C_{db2} + (2C_{gd1} + 2C_{gd2} + C_w) \tag{4.4}$$

This expression is the result of the simplification of the parasitics capacitances shown in Figure 4.7.



(a) Extended

(b) Simplified

Figure 4.7: Components of the load capacitor

Once the capacitance is known, it is necessary to find the value of the resistance that produces the time-constant given in Equation 3.4 to determine the oscillation frequency. This resistance is the equivalent resistance in N- and P- MOS transistors.

The process conditions for simulation are listed in Table 4.1.

HSPICE conditions						
Process	V_{DD}	V_{in}	W_{NMOS}	L_{NMOS}	W_{PMOS}	L_{PMOS}
180nm	1.8V	0.9V	$1\mu m$	$0.18\mu m$	$4\mu m$	$0.18\mu m$

Table 4.1: Conditions for simulation

Taking the considerations mentioned in Table 4.1, the following values were obtained using HSPICE for the simulation of the circuit of interest (inverter 1 comprised between M1 and M2), this, due to the fact that inverter 2 comprising M3 and M4 is part of a load element and, by means of the connection shown in Figure 4.7b. The

behavior of inverter 1 can be known through the file generated at the simulation output, considering that the inverters must remain in a state of saturation.

The transistors involved must remain in saturation, so by implementing the circuit in Figure 4.7b, we determine time delay, parasitic capacitances, transconductance and resistance, as it can be seen in Table 4.2.

Simulation data					
gds PMOS	gm PMOS	rds PMOS	gds NMOS	gm PMOS	rds NMOS
72.2994μ	547.9740μ	13.8313kΩ	21.7870μ	466.3339μ	45.8989kΩ

Table 4.2: Conditions

Results from data				
T_d	$CL_{calculated}$	$CL_{simulated}$	$f_{osc-calculated}$	$f_{osc-simulated}$
45.4p	5.2fF	4.3fF	2.2GHz	2.3GHz

Table 4.3: Results

After obtaining the results in Table 4.3, it was determined that the parasitic capacitance (CL) will be $5fF$ and therefore, the f_{osc} at the time of this design was kept at $2.3GHz$; In addition, the values of W and L proposed in Table 4.1 were maintained (V_{in} is not considered since the feedback loop is closed and is not necessary for the oscillator).

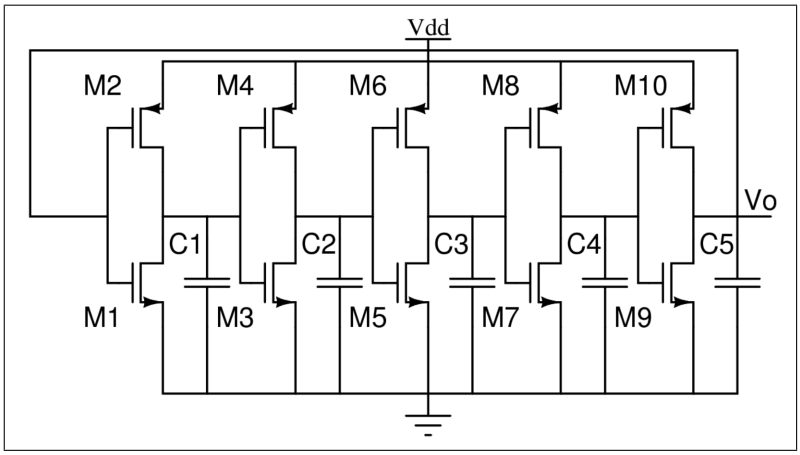


Figure 4.8: Five stage ring oscillator

4.2.2 Current Mirror

The second part of the oscillator design is the addition of current mirrors where it is located the nanocrossbar array and, with this addition, RO is renamed as CM-RO (current mirror-controlled ring oscillator [21]). The current mirror is used to configure the inverters in the RO structure by selecting a specific M_i memristor from the nanocrossbar array. Although, variations in the oscillation frequency of each RO are slightly influenced by the threshold voltage variations in the CMOS transistors comprising the starved inverter and current mirror structures; the overall variation in the oscillation frequency is primarily determined by the variations in memristance of M_i if the supply voltage, VDD, is kept constant.

Each selected memristor is then used to control the current in the current mirror structure. As a result, the oscillation frequency depends on the specific selected memristor.

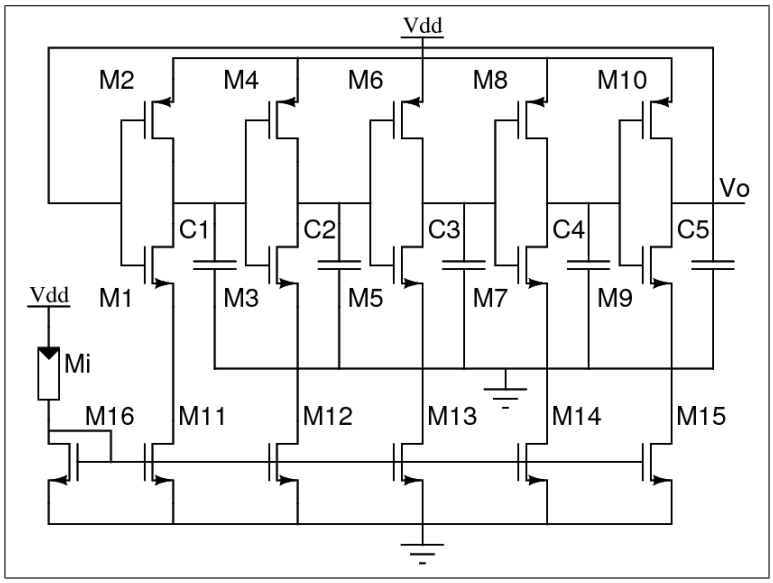


Figure 4.9: CM-RO structure proposed in 180nm process

4.3 Experiments

An experiment of one hundred runs was performed. The generated bit-string (BS) for each run has a length of $N \times \binom{M}{2}$, i.e. the length is 450. The main statistics values of the experiment are given in Table 4.4.

	Mean (μ)	Standard Deviation (σ)	Mode
Uniformity	49.9711%	4.4956	51.11%
Uniqueness	46.4389%	1.7613	45.68%
Bit-aliasing	49.9711%	4.4956	45.55%

Table 4.4: Stadistical data obtained from the one hundred runs

In addition, the behavior of the metrics in the one hundred runs are reported in Figures 4.10, 4.11 and 4.12.

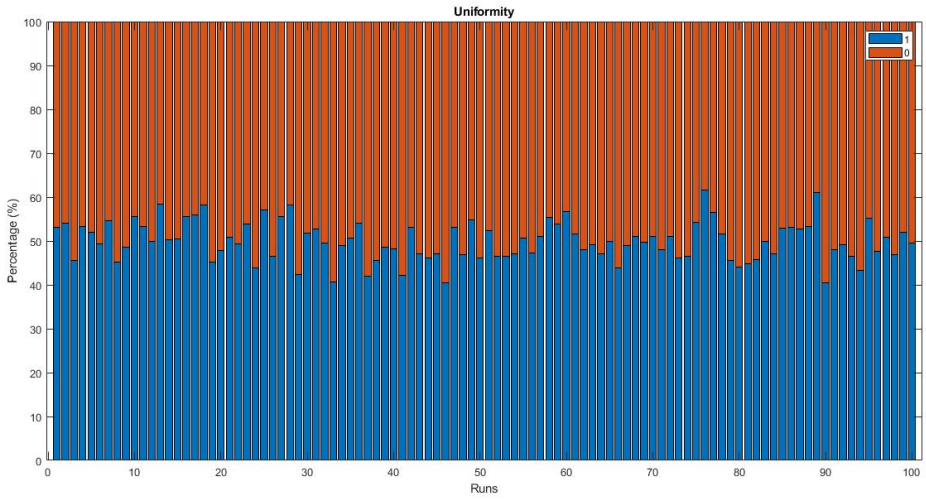


Figure 4.10: Uniformity run distribution

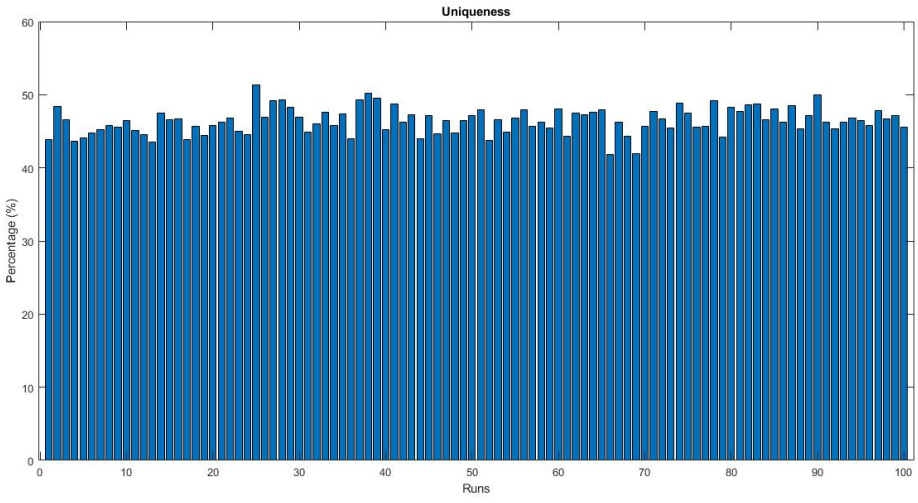


Figure 4.11: Uniqueness run distribution

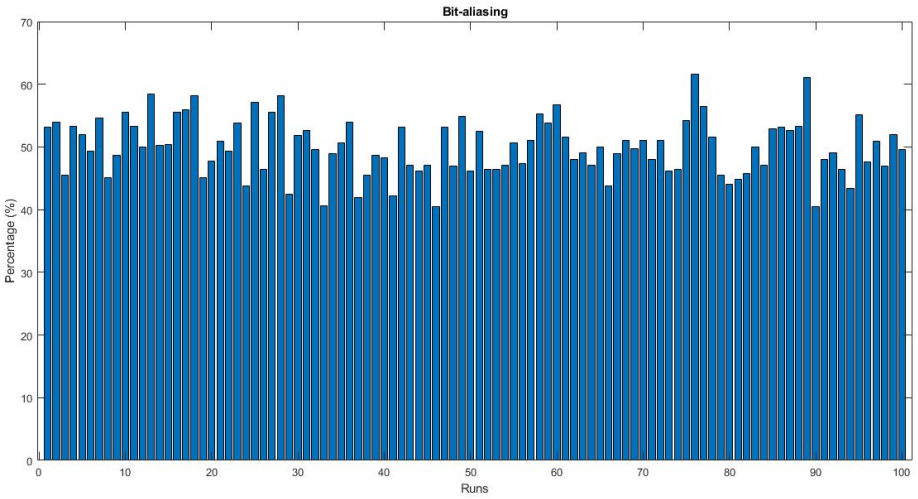


Figure 4.12: Bit-aliasing run distribution

In order to visualize how the metrics are related to the statistical values in Table 4.4, Figures 4.13, 4.14 and 4.15 show the histograms of uniformity, uniqueness and bit-aliasing. These histograms are centered at the mean values for each metric.

The histogram for uniformity shows a mean value that is very close to the ideal but, it has a rather large value for the standard deviation, nearly 4.5%. The central bar of the histogram contains nearly 36% of the results. The histogram shows a good symmetry.

The histogram for uniqueness has also a good symmetry but, the central bar contains nearly 52% of the results. The average is less closer to the ideal but, the standard deviation is also lower than the case of the uniformity.

The histogram for bit-aliasing is very similar to the histogram for uniformity.

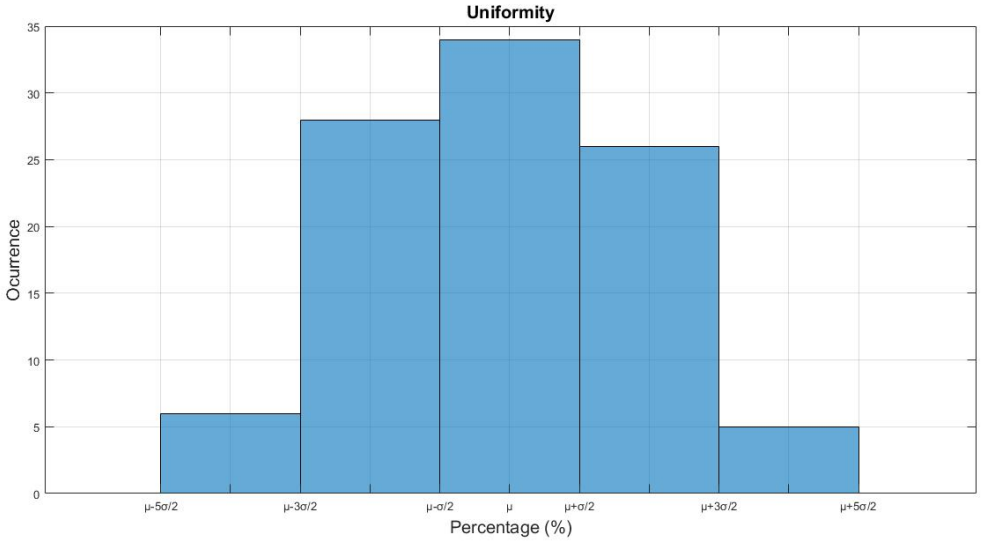


Figure 4.13: Uniformity histogram centered at Mean with $\mu = 49.97$ and $\sigma = 4.4956$

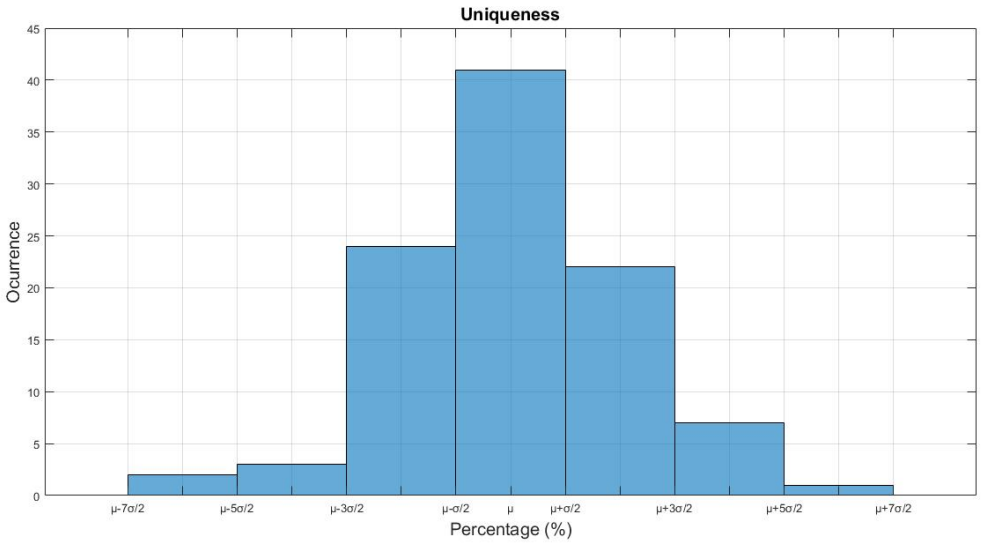


Figure 4.14: Uniqueness histogram centered at Mean with $\mu = 46.44$ and $\sigma = 1.7613$

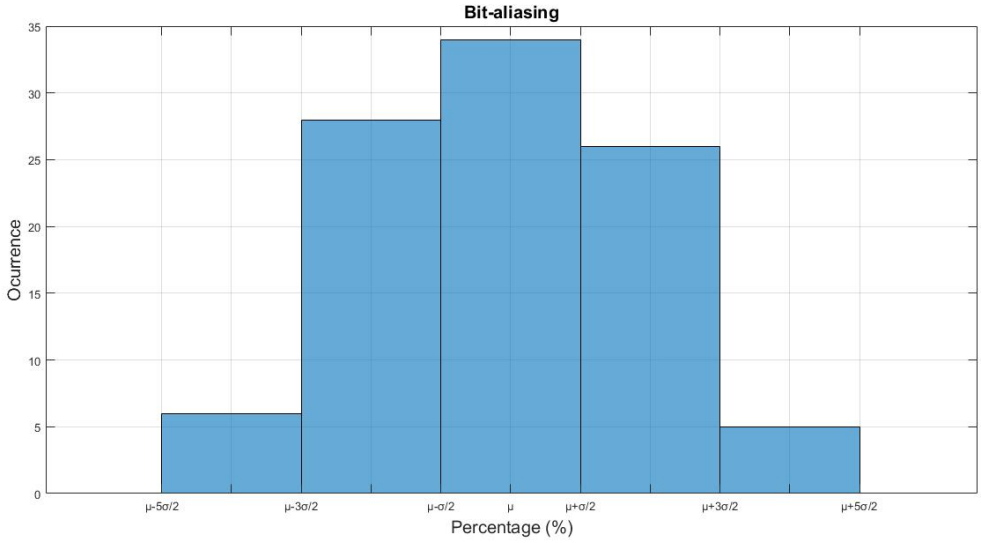


Figure 4.15: Bit-aliasing histogram centered at Mean with $\mu = 49.97$ and $\sigma = 4.4956$

The metrics are also reported centered at the ideal value (50%) in Figures 4.16, 4.17 and 4.18.

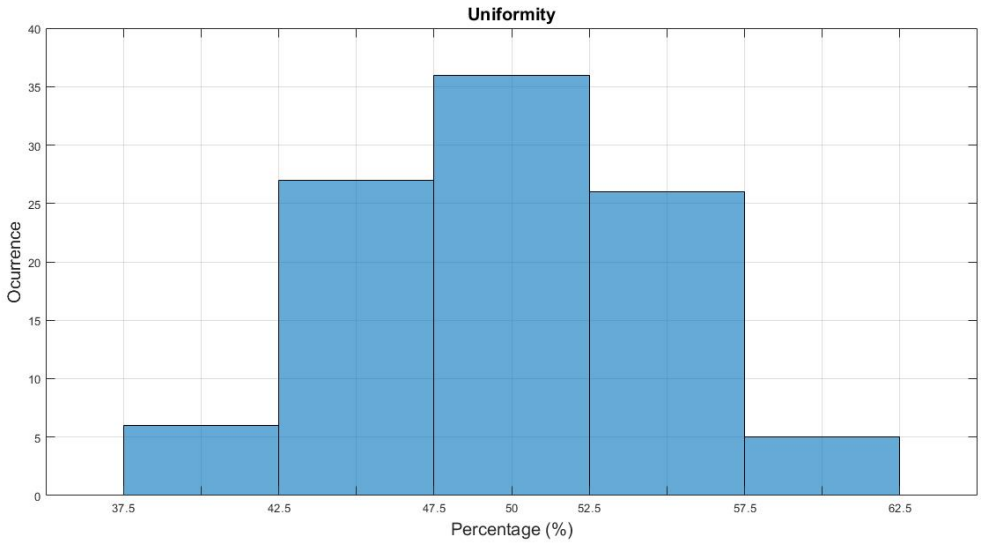


Figure 4.16: Uniformity histogram centered at 50%

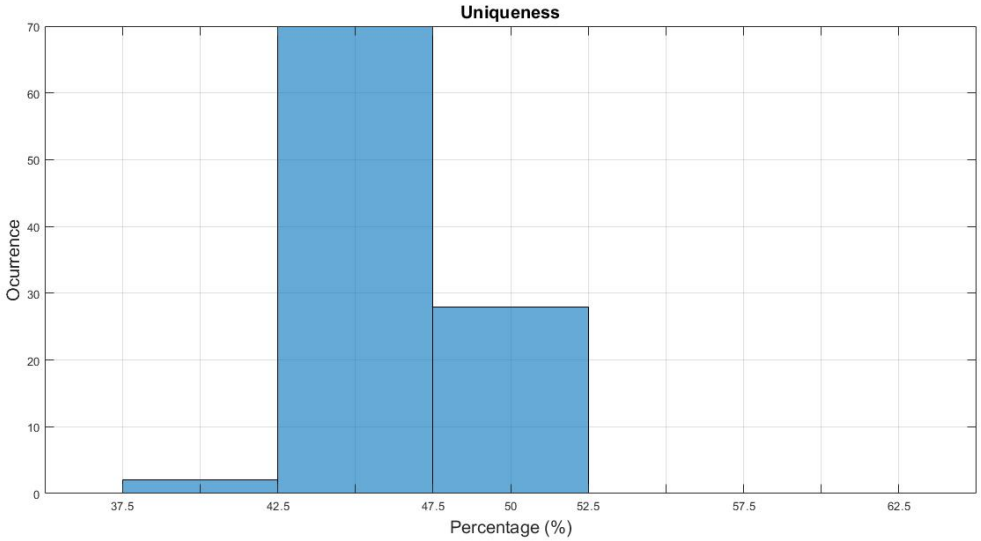


Figure 4.17: Uniqueness histogram centered at 50%

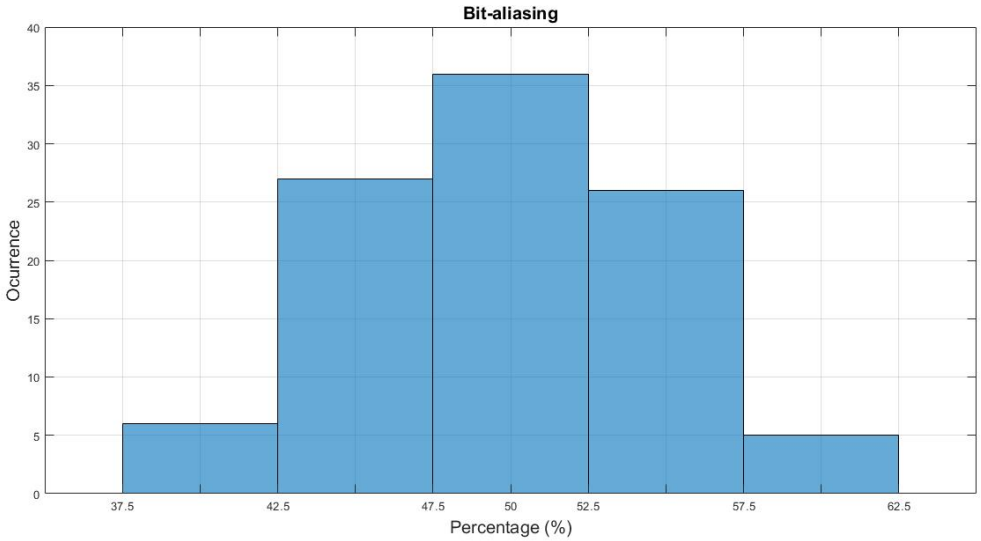


Figure 4.18: Bit-aliasing histogram centered at 50%

4.3.1 Comparison with other works

Table 4.5 shows the main features of the proposal of this thesis in comparison with other works.

	[21]	[32]	[3]	[5]	[33]	This Work
Uniformity	49.66%	NA	$\approx 50\%$	50.18%	51.43%	49.97%
Uniqueness	50.17%	$\approx 50\%$	$\approx 50\%$	50.06%	48.57%	46.44%
Bit-aliasing	NA	NA	NA	48.91%	51.43%	49.97%
Crossbar	40×40	15×10	No	No	No	10×10
CRPs	31200	150	100	55	NA	450
BS length	15	NA	NA	55	NA	450

Table 4.5: Comparison chart

An important feature of the proposal rests in the fact that the BS length is noticeable larger with respect to the other works shown in this table. Unlike the articles [21, 5] where they have a BS of 15 and 55 respectively, we handle a response length of 450 so that its analysis is more precise and we avoid the elimination of worst or best cases, resulting in a metrics with values closer to a real implementation. The data presents acceptable results and therefore, it can be considered an interesting alternative in the field of Hardware Security.

Chapter 5
Conclusions

In this work, a new memristor-based Hardware Security scheme has been presented. The scheme consists of a general system for generating PUFs that uses ROs in combination with current mirrors that randomly access a memristor nanocrossbar array. The research has demonstrated the feasibility of a charge-controlled memristor model for the description of the memristors within the array for the development of the complete HS scheme. A top-to-bottom approach was applied for the design of the scheme. Starting from a systemic view down to device-level realization.

The aleatory feature needed for the generation of the challenge-response pairs is defined by the variability of memristor parameters. Since the memristor model is given as a fully analytic charge-controlled branch relationship, the parameter X_o was used as the random key in the process.

The system has been developed using a combination of HSPICE and Matlab. On the one side, HSPICE achieves the electric circuit simulation where the memristor model has been recast as a subcircuit block. On the other side, Matlab has been used for the implementation at system level of the random selection of the memristors in the array and for the generation of the metrics and the statistics.

The most commonly used metrics, namely Uniformity, Uniqueness and Bit-aliasing, have been used to evaluate the quality of the proposal. The performance of the new scheme has been compared with other memristor-based HS systems. In addition a statistical analysis was achieved for a set of numerical experiments.

Future work

Some lines of further research can be mentioned:

- Develop other types of models for the memristors to be embedded in the nanocrossbar array.
- Develop different arrangements for the random access to a pair of memristors in the array with the aim of reducing the bit-string length.
- Carry out an analysis for different values on the parameters of the current proposal, for instance the size of the memristive array, and the number of inverters in the ROs.
- Develop a front-end interface for a more efficient handling and communication of the tools used in the proposal, i.e. HSPICE and Matlab.

Bibliography

[1] Shital Joshi, Saraju P Mohanty, and Elias Kougianos. Everything you wanted to know about pufs. *IEEE Potentials*, 36(6):38–46, 2017.

[2] Ioannis Vourkas, Athanasios Batsos, and Georgios Ch Sirakoulis. Spice modeling of nonlinear memristive behavior. *International Journal of Circuit Theory and Applications*, 43(5):553–565, 2015.

[3] 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.

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

[5] Joseph Herbert Mitchell Moreno. Memristive modeling for hardware security applications. 2019.

[6] Leon Chua. Memristor-the missing circuit element. *IEEE Transactions on circuit theory*, 18(5):507–519, 1971.

[7] Bart Preneel and Tsuyoshi Takagi. *Cryptographic Hardware and Embedded Systems–CHES 2011: 13th International Workshop, Nara, Japan, September 28–October 1, 2011, Proceedings*, volume 6917. Springer, 2011.

[8] Samuel T King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. Designing and implementing malicious hardware. *Leet*, 8:1–8, 2008.

[9] Yier Jin, Nathan Kupp, and Yiorgos Makris. Experiences in hardware trojan design and implementation. In *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 50–57. IEEE, 2009.

[10] Cynthia Sturton, Matthew Hicks, David Wagner, and Samuel T King. Defeating uci: Building stealthy and malicious hardware. In *2011 IEEE Symposium on Security and Privacy*, pages 64–77. IEEE, 2011.

[11] Eric Love, Yier Jin, and Yiorgos Makris. Proof-carrying hardware intellectual property: A pathway to trusted module acquisition. *IEEE Transactions on Information Forensics and Security*, 7(1):25–40, 2011.

[12] Yier Jin, Bo Yang, and Yiorgos Makris. Cycle-accurate information assurance by proof-carrying based signal sensitivity tracing. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 99–106. IEEE, 2013.

[13] Md Tanvir Arafin, Carson Dunbar, Gang Qu, N McDonald, and L Yan. A survey on memristor modeling and security applications. In *Sixteenth International Symposium on Quality Electronic Design*, pages 440–447. IEEE, 2015.

[14] Darya Almasi. *Enhancing Hardware Security Using Spin Transfer Torque Logic*. PhD thesis, San Francisco State University, 2015.

[15] Yong Shim, Abhronil Sengupta, and Kaushik Roy. Low-power approximate convolution computing unit with domain-wall motion based" spin-memristor" for image processing applications. In *Proceedings of the 53rd Annual Design Automation Conference*, pages 1–6, 2016.

[16] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *2007 44th ACM/IEEE Design Automation Conference*, pages 9–14. IEEE, 2007.

[17] Yunxi Guo. Cryptographic application of physical unclonable functions (pufs). 2018.

[18] Yansong Gao, Yang Su, Wei Yang, Shiping Chen, Surya Nepal, and Damith C Ranasinghe. Building secure sram puf key generators on resource constrained

devices. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 912–917. IEEE, 2019.

[19] Ilze Eichhorn, Patrick Koeberl, and Vincent van der Leest. Logically reconfigurable pufs: Memory-based secure key storage. In *Proceedings of the sixth ACM workshop on Scalable trusted computing*, pages 59–64, 2011.

[20] Abhranil Maiti, Inyoung Kim, and Patrick Schaumont. A robust physical unclonable function with enhanced challenge-response set. *IEEE Transactions on Information Forensics and Security*, 7(1):333–345, 2011.

[21] Yansong Gao, Damith C Ranasinghe, Said F Al-Sarawi, Omid Kavehei, and Derek Abbott. mrpuf: A novel memristive device based physical unclonable function. In *International Conference on Applied Cryptography and Network Security*, pages 595–615. Springer, 2015.

[22] Yojanes Andrés Rodríguez Velásquez. *Development of an analytical model for a charge-controlled memristor and its applications*. PhD thesis, Master's thesis. Puebla, Mexico: National Institute for Astrophysics, Optics ..., 2017.

[23] Yogesh N Joglekar and Stephen J Wolf. The elusive memristor: properties of basic electrical circuits. *European Journal of physics*, 30(4):661, 2009.

[24] JR Wersta, CJM Verhoeven, and AHM van Roermund. Oscillators and oscillator systems: Classification, analysis and synthesis. kluwer acod. pub. 1999.

[25] H Barkhausen. Lehrbuch der elektronenroehren, 3. band: Rueckkopplung. *Verlag S. Hizrel*, 1935.

[26] D Chattopadhyay. *Electronics (fundamentals and applications)*. New Age International, 2006.

[27] Stephen Docking and Manoj Sachdev. A method to derive an equation for the oscillation frequency of a ring oscillator. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 50(2):259–264, 2003.

[28] Utku Seckin and Chih-Kong Ken Yang. A comprehensive delay model for cmos cml circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(9):2608–2618, 2008.

[29] G Jovanovic, M Stojcev, and Zoran Stamenkovic. A cmos voltage controlled ring oscillator with improved frequency stability. *Scientific Publications of the State University of Novi Pazar, Series A: Applied Mathematics, Informatics and mechanics*, 2(1):1–9, 2010.

[30] Abbas Ramazani, Sadegh Biabani, and Gholamreza Hadidi. Cmos ring oscillator with combined delay stages. *AEU-International Journal of Electronics and Communications*, 68(6):515–519, 2014.

[31] MK Mandal and Bishnu Charan Sarkar. Ring oscillators: Characteristics and applications. 2010.

[32] Patrick Koeberl, Ünal Kocabaş, and Ahmad-Reza 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. IEEE, 2013.

[33] Julius Teo Han Loong, Noor Alia Nor Hashim, Muhammad Saiful Hamid, and Fazrena Azlee Hamid. Performance analysis of cmos-memristor hybrid ring oscillator physically unclonable function (ro-puf). In *2016 IEEE International Conference on Semiconductor Electronics (ICSE)*, pages 304–307. IEEE, 2016.