FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Hybrid CMOS/Memristor Circuits Emulator

André Filipe Ávila Silva

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Advisor: Manuel Cândido Duarte dos Santos

Co-advisor: Catarina José Loureiro da Silva Dias

October 31, 2020

Hybrid CMOS/Memristor Circuits Emulator

André Filipe Ávila Silva

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Approved by:

President: Vítor Manuel Grade Tavares
Referee: João Manuel Torres Caldinhas Simões Vaz

Faculdade de Engenharia da Universidade do Porto

October 31, 2020

Resumo

Em 1971, Leon Chua reparou que faltava uma ligação entre os 4 elementos fundamentais do eletromagnetismo: tensão, corrente, carga e fluxo magnético. Chua completou essas as ligações através da idealização matemática de um novo dispositivo, o memristor. Este dispositivo, considerado por Chua um elemento fundamental como a resistência, condensador e indutor, tem propriedades não lineares e de memória não volátil: a resistência do dispositivo é dependente de uma tensão que foi aplicada no passado. De forma muito simplificada, o memristor é uma resistência com memória. Este dispositivo só viria a ser implementado fisicamente pela HP Labs em 2008.

Desde então que este dispositivo tem sido explorado em diferentes áreas pelas suas características peculiares, principalmente nas áreas de computação neuromórfica e de memória não volátil. Com o aumento drástico das limitações na tecnologia CMOS devido à sua abrupta diminuição, os memristors apresentam ser uma tecnologia ideal para implementação híbrida com CMOS, contornando assim o que seria o fim da Lei de Moore.

Esta dissertação explora o comportamento do memristor, as aplicações que foram feitas até ao momento, a projeção de um circuito com tecnologia hibrida CMOS/memristor e demonstra os resultados do uso de ambas tecnologias.

Abstract

In 1971, Leon Chua noticed that it was missing a link between the 4 fundamental elements from eletromagnetism: voltage, current, charge and magnetic flux. Chua found the missing link with a mathematical approach of a new device: the memristor. Chua considered this device as fundamental, like the resistor, capacitor and inductor, and it shows properties of non volatile memory and non linearity: its resistance depends on a voltage that was applied to it in the past. In a very simple and crude way, the memristor is a resistor with memory. This device would only be implemented physically in 2008 by HP Labs.

Since then, this device has been a target for investigation for its unique behavior, mostly in neuromorphic computing area and non volatile memory. With the fast grow of the CMOS limitations due to its extreme size reduction, memristors show to be an ideal technology to implement in hybrid technology with CMOS, contouring what would be the end of Moore's Law.

This dissertation studies the behaviour of the memristor, as well as its state of art, the project of circuits with hybrid CMOS/memristor technologies and presents the results of using both technologies.

Agradecimentos

Em primeiro lugar, quero agradecer aos meus orientadores Doutor Cândido e Doutora Catarina por todo o apoio dado durante este percurso. Agradeço também ao Doutor Vítor Tavares e Doutor João Vaz pela presença na apresentação do trabalho final e pela sua avaliação.

Em segundo lugar, quero agradecer ao CEO da *Knowm*, Alex Nugent, que, em plena pandemia, teve a amabilidade de fornecer gratuitamente 4 chips de memristors para fim de realização desta dissertação. Agradeço também à *Keithley*, por me ter emprestado um SMU durante o periodo de um mês e meio.

Em terceiro lugar, quero agradecer ao Pedro Alves que me ajudou a soldar todos os componentes na PCB.

Em quarto lugar, quero agradecer a todos os meus amigos, colegas e familia que, durante todo este percurso académico, ajudaram a tornar-me na pessoa que sou hoje, uma pessoa mais dedicada, mais consciente e mais solidária.

Em quinto lugar, quero a minha avó e minha irmã que, nestes tempos difíceis, têm sido um forte pilar na minha vida.

Em sexto lugar, quero agradecer à minha maravilhosa namorada por ter-me acompanhado neste percurso em todos os momentos, bons ou maus. Agradeço-lhe pela sua enorme paciência para aturar as minhas taras parvas e por todo o amor e apoio que ela me tem dado. Sem ela, sem dúvida que seria uma pessoa bem menos feliz.

Em sétimo e último lugar, quero agradecer a minha mãe. Há quase um ano que partiste do nada. Ainda hoje nem acredito que partiste. Só queria que estivesses cá para me acompanhares neste percurso que tanto lutaste. Sem ti não estaria onde estou hoje. És o meu exemplo de vida, de vontade de viver e, hoje, pego em todas as tuas forças e entrego esta dissertação. Estejas onde estiveres, mãe, esta dissertação é para ti. Obrigado por tudo.

André Filipe Ávila Silva

"Imagination is more important than knowledge."

Albert Einstein

Contents

List of Figures

List of Tables

Abbreviations

ASIC	Source Measure Unit
BE	Bottom Electrode
CLB	Configurable Logic Blocks
CMOS	Complementary Metal-Oxide Semiconductor
FPGA	Field Programmable Gate Arrays
HDL	Hardware Description Language
I-V	Current versus Voltage
IC	Integrated Circuit
LUT	Look-Up Table
MIM	Metal-Insulator-Metal
PCB	Printed Circuit Board
SDC	Self Direct Channel
SMU	Source Measure Unit
SRAM	Static Random Access Memory
TE	Top Electrode
VLSI	Very Large Scale Integration

Chapter 1

Introduction

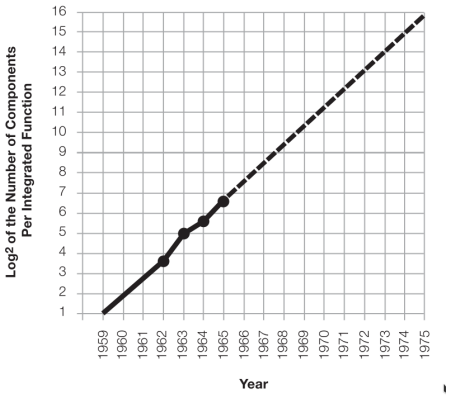
The electronics industry is mainly driven by its capability to create more advanced hardware. This is primarily possible due to the use of Complementary Metal-Oxide Semiconductor (CMOS) technology, which allows to create a considerable amount of building blocks inside a single chip. Since the invention of this technology [1], its use has been fundamentally important in the fabrication of integrated circuits (ICs) due to its very low power consumption - which makes this perfect for very large scale integration (VLSI) - as well as its low signal-to-noise ratio interference from other electrical devices and, the most important, its scalability. The scalability in CMOS consists in dimension and voltage reduction of the MOS transistor by the same factor.

In 1965, Gordon E. Moore, co-founder of Fairchild Semiconductor and former CEO of Intel, published a paper [2] in which he states that the number of transistors in an IC will double about every year for the next 10 years (Figure 1.1a). This statement would be revised by Moore in 1975, creating a new claim saying the number of transistors in a chip would double about not every year, but every two years. This statement became known as Moore's law and it still stands according to real data (Figure 1.1b). The problem of Moore's law is that it has not taken into account the limitations that would occur when shrinking the transistor, like current leakages or limited material for gates or metal conductors.

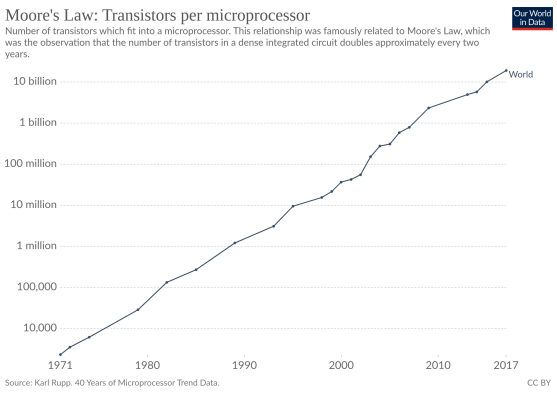
This limitations started to be more noticeable in the last few years, where it can be seen the growth has been reducing [3–5]. The transistor is becoming so small that it introduces quantum tunnelling [6–8], a phenomenon where subatomic particles pass through a potential barrier, making the transistor conduct whatever voltage is applied to its gate, stopping behaving as expected.

Another limitation in Moore's law comes from the nowadays commonly used processor architecture based on von Neumann's architecture [9]. This was proposed by Neumann in 1945, where he describes the system as having a processing unit which manipulates and stores data from a memory unit. The problem appears when the processing unit's speed starts to be much faster than the memory unit's access speed. This makes the processor idle while waiting for a data transfer. Many mechanisms have been propose to overcome this problems, like the addition of a cache or the use of multithreading. This problem is treated as the von Neumann bottleneck and it gets more difficult to solve as newer technologies tend to get faster.

This brings a lot of discussion and research for new ways to counter these problems, such as studying new materials (e.g. calcium fluoride [10], carbon nanotubes [11]) or quantum-mechanical phenomena to perform computation [12,13] or even studying recently backgrounded elements, like the memristor, to achieve new computational architectures, like neuromorphic architectures.



(a) Moore's prediction in 1965.



(b) Number of transistors inside a microprocessor. This data was taken from Our World In Data website.

1.1 The Memristor

The memristor is the fourth fundamental circuit element as proposed by Leon Chua in 1971 [14] (Figure 1.1) that was missing to link two of the electromagnetic variables: charge and flux linkage. This device behaves like a nonlinear resistor with memory: its resistance depends on the previous charge that passed through it. In other words, this device can be seen as a time-varying resistor.

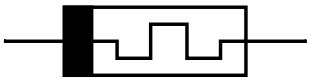


Figure 1.1: Memristor schematic symbol proposed by Chua.

Although Chua proposed the memristor concept in 1971, it wasn't until 2008 that has been annouced a physical implementation. This was done by HP Labs, where they present the results in their 2008 *Nature* paper [15]. They feature a mathematical approach to the physical implementation and the real results from their Titanium Dioxide memristor. From there, HP Labs developed a new technology, the ReRAM (resistive RAM), from the memristor's capability to hold its resistance state.

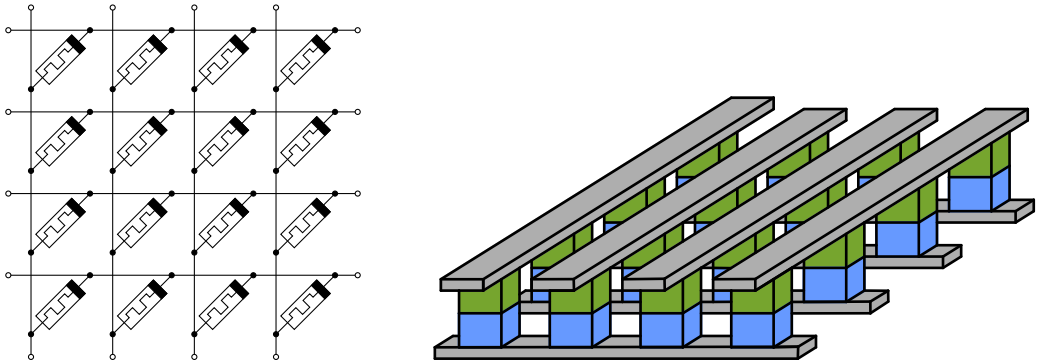
1.2 Motivation

Since the first physical implementation of the memristor by HP Labs, there has been a lot of research on how the device works, what would be the best simulation model , what materials should be used in order to maximize the device's performance and, most important, where the device fits

best, in a sense to maximize performance parameters and solve problems from nowadays technologies.

One of the first implementations made with memristor was to create crossbars, like the ones implemented in HP Labs ReRAM. Memristor crossbars (Figure 1.2) consist of a matrix configuration of the memristors. This configuration allows the selection of a memristor either to change or see its state. This offers many applications, such as hybrid memristor/CMOS circuits [16–21], non-volatile memory [22–25], neuromorphic architectures and brain-inspired computing [26–30], digital computing/logic [16, 19, 20, 25, 31], analog computing [31, 32], and radiofrequency applications and filtering [33–35].

It is important to think of improved solutions for better and more efficient computing architectures, since the exiting ones are coming to an end due to size limitation. Exploring and combining the memristor properties with the existing CMOS technology may give interesting and promising results to solve von Neumann bottleneck.



(a) *Memristor* crossbar schematic. (b) *Memristor* crossbar structure.

Figure 1.2: *Memristor* crossbar.

1.3 Goals

The main goal of this dissertation is to achieve optimal circuits using both CMOS and memristor technologies for parameter optimization. This optimal circuits aim to explore the non-volatile memory capability of the memristor to create more specifically reconfigurable logic. A future goal would be phisically integrating memristors in CMOS micro-fabrication to achieve complete electronic systems that can solve and counter nowadays problems like the end of Moore's Law and the von Neumann bottleneck.

Chapter 2

The Memristor

2.1 Chua's Memristor

In the world of electronics, there are three fundamental passive circuit components: the resistor, which links the voltage and current (Figure 2.1 *); the capacitor, which links the charge and voltage (Figure 2.1 **); and the inductor, which links the magnetic flux and current (Figure 2.1 ***). These three components are the main foundation to any electronic circuit. But what about a device that can relate the magnetic flux and charge (Figure 2.1 ****)?

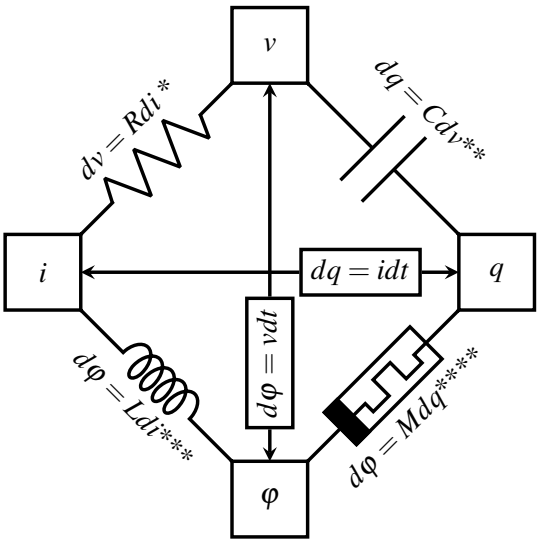


Figure 2.1: Relationship between v, i, φ and q.

The matematical concept of this device was first introduced in 1971 by the well known professor Leon Chua [14] from University of California, Berkeley, in which he calls it the memristor (contraction between memory and resistor). As Chua described in this 1971 paper, the memristor behaves like a nonlinear resistor with memory, which is characterized by flux linkage variation between its two terminals over the charge that passes through it, resulting in what is called the

memresistance, M:

$$M(t) = \frac{d\varphi(t)}{dq(t)} \tag{2.1}$$

The magnetic flux is the integral of voltage over time and the charge is the integral of current over time, resulting in Ohm's law [36,37] (Equation 2.2). If the memresistance is time independent and constant, it is obtained a resistor with resistance R. The memristor explores non linearity and time dependency, giving it a memresistance that depends on the charge that flowed through it in the past:

$$M(t) = \frac{v(t)dt}{i(t)dt} = \frac{v(t)}{i(t)} \tag{2.2}$$

2.2 *HP Labs* Memristor

In 2008, *HP Labs* published an article in the scientific journal *Nature* [15] where they present what would be the first successful implementation of the hipothetical memristor. Its structure represents a metal-oxide-metal structure just like a capacitor: there is one layer of Titanium Dioxide TiO_2 and one layer of Oxigen-poor Titanium Dioxide TiO_{2-x}; outside these layer there are two parallel Platinum plates (Figure 2.2). The vacancies act as mobile charges and drift according to the applied electric field, shifting the dividing line between the two titanium dioxide layers [38, 39]. This happens due to a phenomenon which is called electroforming [40, 41], which creates an high or low resistance path according to the voltage applied, explained in Figure 2.3.

The memristor polarity is defined by the Top Electrode, TE, and the Bottom Electrode, BE.

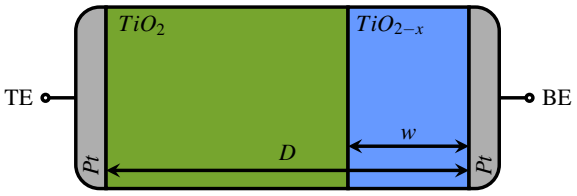
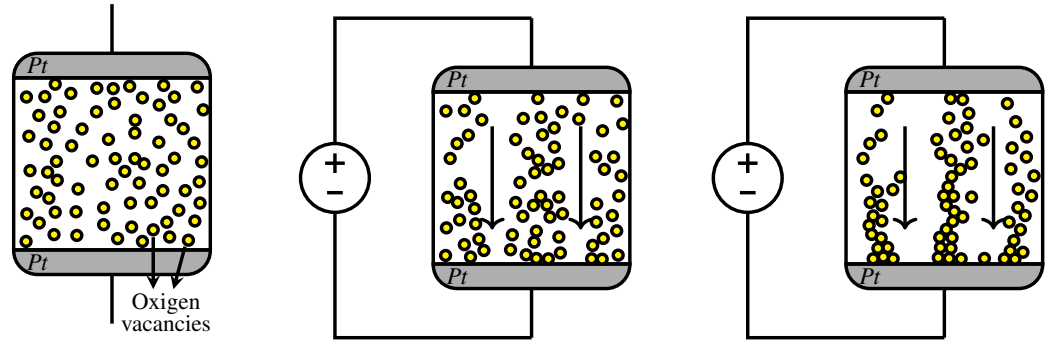


Figure 2.2: *HP Labs* memristor structure.

A basic approach to define mathematically a memristor is to see it as a resistor controlled by an internal state varible w, which corresponds to the width of the doped Titanium Dioxide layer. This width is time dependent of the current that has previously flowed through the memristor (Equation 2.5).

Both the undoped and doped layers have a known resistance, which can be called, respectively, R_{OFF} - due to its low charge carriers concentration, giving it low conductivity (Figure 2.4b) - and R_{ON} - due to its high charge carriers concentration, giving it high conductivity (Figure 2.4a).

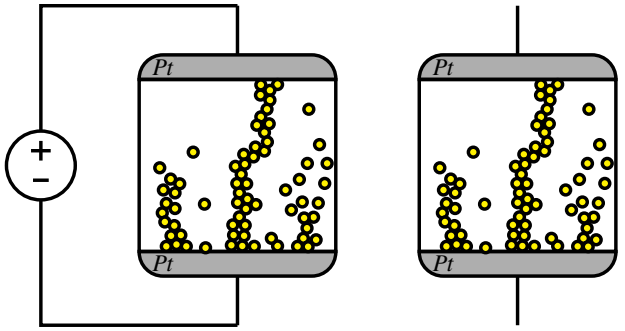
The application of an external bias voltage between the memristor terminals will cause a change in the boundary of the two layers. So, the memristor can be modeled as two variable



(a) Memristor at an initial state, where its conductance is low.

(b) A positive voltage is applied to the memristor. Vacancies start to move in the electric field direction.

(c) Vacancies start to accumulate, creating filaments, thus increasing the memristor's conductivity.



(d) The memristor gets to its high conductivity state, since the vacancies cannot move any further.

(e) Without the need of the voltage supply, the memristor maintains its physical state, therefore maintaining its conductivity state.

Figure 2.3: *HP Labs* memristor structure.

resistors connected in series with both middle contact points connected together, making the w variable to be modulated (Figure 2.4c).

Knowing the length of the memristor, D, a width of doped Titanium Dioxide layer, w, the voltage between its terminals is given by:

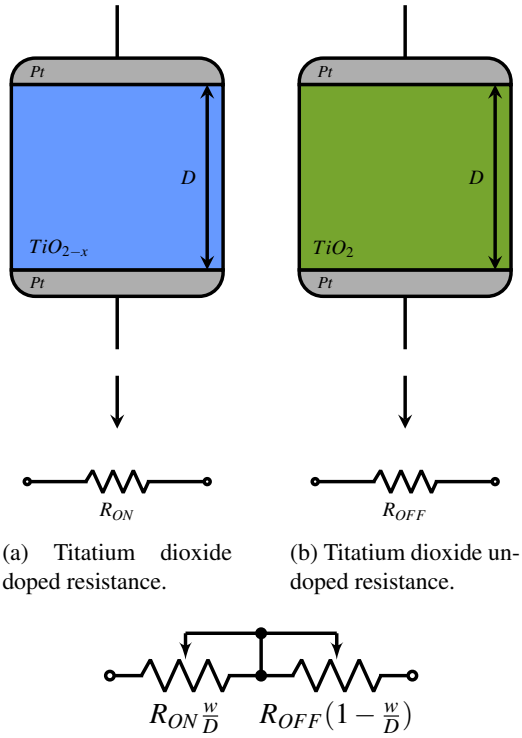
$$v(t) = \left(R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right) \right) i(t) \tag{2.3}$$

With the average ion mobility μ_v, it can be obtained w variance for this situation:

$$\frac{dw}{dt} = \mu_v \frac{R_{ON}}{D} i(t) \tag{2.4}$$

Since the current is charge variation over time, w can be expressed and simplified by:

$$w(t) = \mu_v \frac{R_{ON}}{D} q(t) \tag{2.5}$$



(a) Titatium dioxide doped resistance.

(b) Titatium dioxide undoped resistance.

(c) Memristor equivalent circuit.

Figure 2.4: Memristor structure and equivalent schematic.

Knowing that the doped resistance is much lower than the undoped resistance, M is:

$$M(t) = R_{OFF}\left(1 - \frac{\mu_v R_{ON}}{D^2} q(t)\right) \tag{2.6}$$

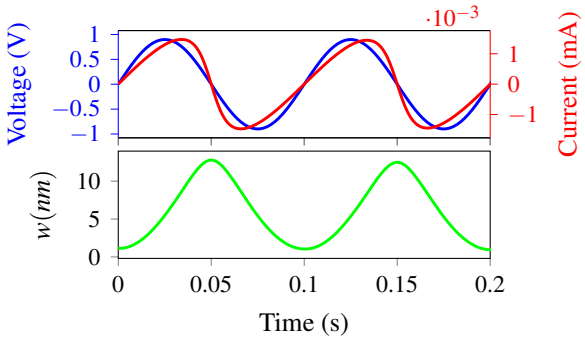
And it is charge dependent, being its crucial contribution. For higher μ_v values and smaller lengths D, the memresistance becomes higher.

To obtain a better understanding of this device properties, a graphical interpretation was made through the simulation of the model (Figure 2.5).

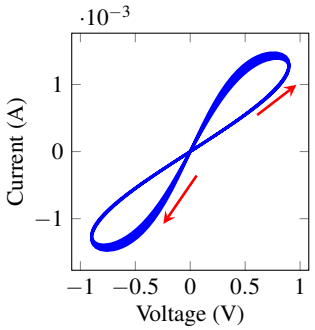
From Figure 2.5b, it can be seen the memristor's I-V curve with hysteresis, proving that its resistance changes over time according to the voltage applied.

2.3 Memristor and CMOS

With the introduction of the first working memristor, the main approach to the use of this device was to implement it in a large scale structure, a crossbar, like the one shown previously in Figure 1.2. This memristor crossbar allows the arranging of x memristors in a matrix configuration of $x = n \times m$. The selection of a specific memristor can be achieved by using the column and row to which it is connected.



(a) Time domain behaviour of the HP Labs memristor model for a sinusoidal voltage with 0.9 V of amplitude and frequency of 10 Hz.

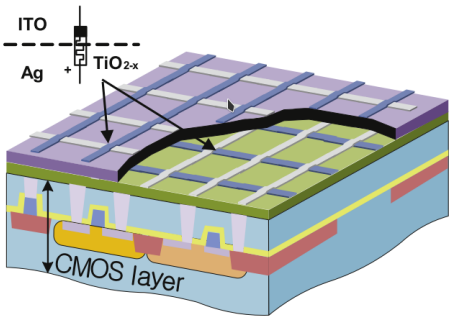


(b) I-V characteristic curve for HP Labs memristor model. The arrows indicate the current flow.

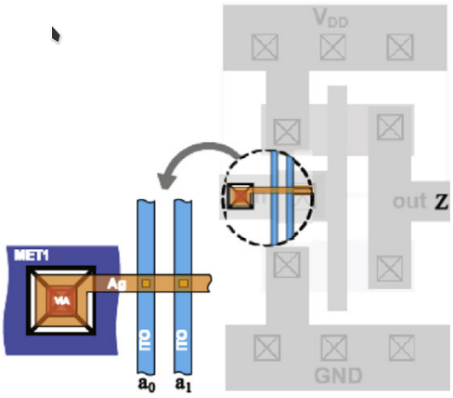
Figure 2.5: HP Labs memristor simulation results. Input voltage: $0.9\sin(2\pi 10t)$ Memristor parameters: $R_{on} = 1k\Omega$, $R_{off} = 10k\Omega$, $w = 0.9\times10^{-9}m$, $D = 12\times10^{-9}m$, $u_v = 2\times10^{-14}m^2s^{-1}V^{-1}$.

A more specific research and development area consists on using these crossbars with CMOS technology [16–21]. The main idea goes through stacking the memristor crossbar on top of the CMOS substrate (Figure 2.6a). The connection between the CMOS layer and the crossbar layer is made through the CMOS vias. These vias connect the CMOS circuits to the crossbars electrodes, selecting specific memristors according to the row-column connection. This introduces a new layout procedure for the CMOS, where vias need to be added for hybrid connection (Figure 2.6b).

In the literature presented in this section, hybrid circuits were designed and it was shown that the hybrid technology stack is physically compatible.



(a) Hybrid CMOS/Memristor structure.



(b) Hybrid CMOS/Memristor layout.

Figure 2.6: Hybrid CMOS/Memristor representation (Figures from [19]).

Chapter 3

Memristor Simulation Models

In this chapter, general memristor simulation models in Spice are studied. Their I-V curves and dynamics are analyzed in order to get a better understanding of how each model works. Most of the simulation models are reviewed by Chis Yakopcic in his 2012 Book chapter [42]. The *Knowm*'s memristor model is also explored and a model parameter fitting is done from real data. A problem with this model is presented and a modification is proposed.

All simulations were done in *LTSpice IV* and all the Spice models can be found in Appendix A.

3.1 Spice Model

The Spice circuit used for modeling a memristor (Figure 3.1) is based on a state variable x (equation 3.1). This variable is used instead of w to normalize the memristor's state, resulting in values between 0 (least conductive state) and 1 (most conductive state). The memristor's top electrode is denoted by TE and the bottom electrode by BE. This indicates the memristor polarity, where a positive voltage bias ($v_{TE-BE} = v_{mem}$) results in an increase in conductivity.

$$x(t) = \frac{w(t)}{D} \tag{3.1}$$

The current that goes through the memristor is given by the voltage across it divided by its memresistance, which is dependent on x. So the memristor can be replaced by a voltage controlled current source, which value corresponds to Equation 3.3.

$$v = M(x)i \tag{3.2}$$

$$i = \frac{v(t)}{M(x)} = \frac{v}{R_{ON}x + R_{OFF}(1-x)} \tag{3.3}$$

From the first *HP Labs* model presented in Chapter 2, the w derivative is given by Equation 2.4 and, to obtain an x value, w is normalized and then the equation is integrated. The integration

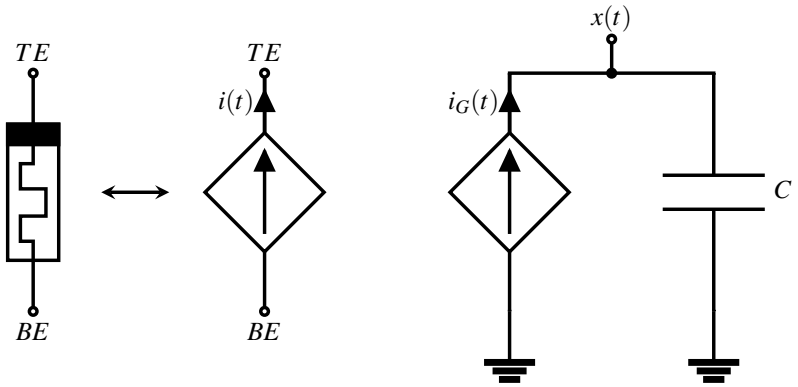


Figure 3.1: Equivalent circuit for memristor simulation model.

is obtained from a voltage across a one Faraday capacitor (Equation 3.4), in which it is applied a current with the value of $\frac{dx}{dt}$ (Equation 3.5).

$$x(t) = V_C = \frac{1}{C}\int_{-\infty}^{t} i_G(t)dt \tag{3.4}$$

$$\frac{dx(t)}{dt} = i_G(t) \tag{3.5}$$

3.2 Window Functions

One of the main problems about the memristor model proposed by *HP Labs* (Section 2.2) is that it assumes a linear change of the internal state variable x (Equation 2.4). It presents high non linearity in its boundaries, where the carriers speed is strongly reduced when x goes towards 0 ($w \rightarrow 0$) or towards 1 ($w \rightarrow D$). This phenomena is known as boundary effect.

Another problem surges with this linear drift model when x reaches the boundaries: there are no limits. If the internal state variable reaches a boundary, it should not surpass it. With *HP Labs* model, x can take values outside the interval between 0 and 1, which is not physically possible and thus the model starts to behave differently from its original concept.

To overcome the above problems, a second function (window function) is added to Equation 2.4 in order to add boundary limits and to add non linearity to the state variable motion (Equation 3.6).

$$\frac{dx(t)}{dt} = \mu_V \frac{R_{ON}}{D^2} i(t) F(x) \tag{3.6}$$

3.2.1 Joglekar Model

In 2009, Yogesh Joglekar et al. proposed a window function [43] where the equation is controlled by a parameter p (Figure 3.5): as p increases, the boundary gets narrower and the effect becomes more pronounced. It can be verified that, towards the boundaries, the window function introduces

non linearity, whereas between the boundaries, the window function behaves approximately linear and constant.

$$F(x) = 1 - 2(x-1)^{2p} \tag{3.7}$$

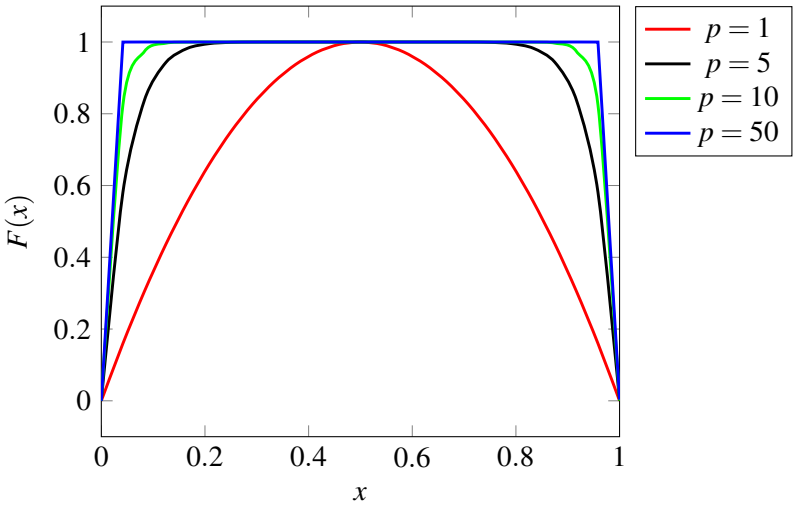


Figure 3.2: Joglekar window function for different p values.

This window function was placed in Equation 3.6, resulting in Equation 3.8, in order to simulate and analyze its behaviour compared to the original *HP Labs* model results (from Figure 2.5).

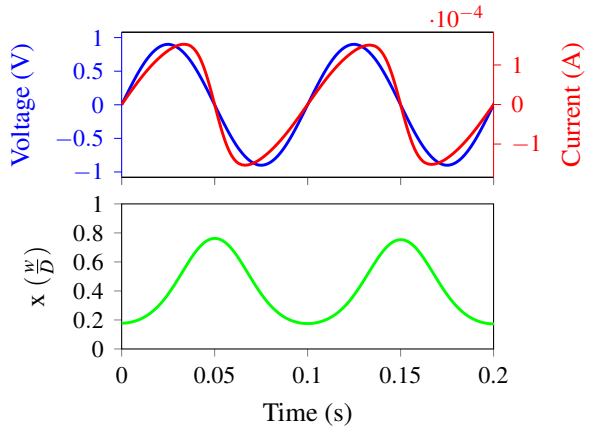
$$\frac{dx(t)}{dt} = \mu_V \frac{R_{ON}}{D^2} i(t)(1 - 2(x-1)^{2p}) \tag{3.8}$$

The simulation results from Joglekar modification from Figure 3.3 show that, when the applied voltage is sufficiently low to make x stay approximately in the window linear region, the I-V curve looks similar to the one obtained from the original model in Figure 2.5. However, as the input voltage amplitude gets higher, x starts to reach the boundaries (Figure 3.4) and the window function introduces non linearity.

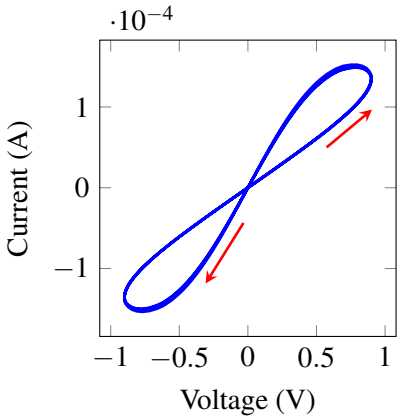
3.2.2 Biolek Model

In 2009, Biolek et al. published a paper proposing a new window function (Figure 3.5) [44]. Biolek points out a problem with Joglekar window: x suffers non linearity whether it goes towards or away from the boundary. This means that, when x is towards a boundary and changes direction to the opposite one, Joglekar window function will still introduce non linearity when it should not. Instead of x motion depend only on its ongoing x value, it should depend also on its direction (Equation 3.9), as porposed by Biolek.

$$F(x) = 1 - (x - u(-i))^{2p} \tag{3.9}$$

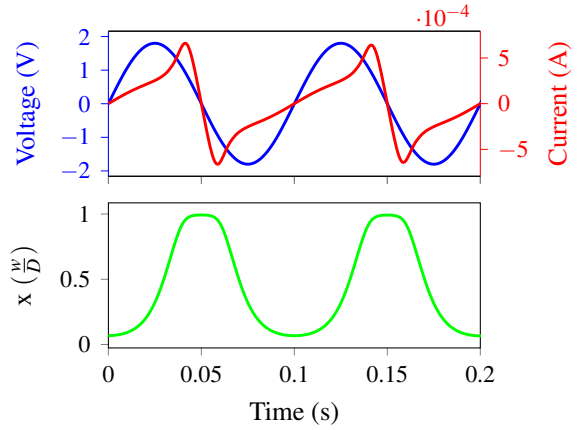


(a) Time domain behaviour of the HP Labs with Joglekar modification memristor model for a sinusoidal voltage with 0.9 V of amplitude and frequency of 10 Hz.

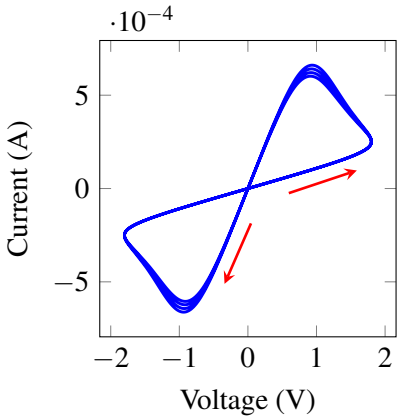


(b) I-V characteristic curve for HP labs with Joglekar modification memristor model. The arrows indicate the current flow.

Figure 3.3: Simulation results for HP Labs model with Joglekar modification. Input voltage: $0.9\sin(2\pi 10t)$; Memristor parameters: $R_{on} = 1k\Omega$, $R_{off} = 10k\Omega$, $x_0 = 0.5$, $D = 12\times10^{-9}m$, $u_v = 2\times10^{-14}m^2s^{-1}V^{-1}$, $p = 1$.



(a) Time domain behaviour of the HP Labs with Biolek modification memristor model for a sinusoidal voltage with 1.8 V of amplitude and frequency of 10 Hz.



(b) I-V characteristic curve for HP Labs with Joglekar modification memristor model. The arrows indicate the current flow.

Figure 3.4: Simulation results for HP Labs model with Joglekar modification. Input voltage: $1.8\sin(2\pi 10t)$ Memristor parameters: $R_{on} = 1k\Omega$, $R_{off} = 10k\Omega$, $x_0 = 0.076$, $D = 12\times10^{-9}m$, $u_v = 2\times10^{-14}m^2s^{-1}V^{-1}$, $p = 1$.

$$\frac{dx}{dt} = \eta\mu_V\frac{R_{ON}}{D^2}i(t)(1-(x-u(-i))^{2p}) \tag{3.10}$$

Like in Joglekar window, when the state variable is within the boundaries, the I-V curve (Figure 3.6) looks similar to the one obtained from *HP Labs* model. However, when x starts to go near the boundaries (3.7), the non linear effect on x motion starts to get noticeable. In comparison with Joglekar results (Figure 3.4), when x changes direction, Biolek window looks more linear, as an

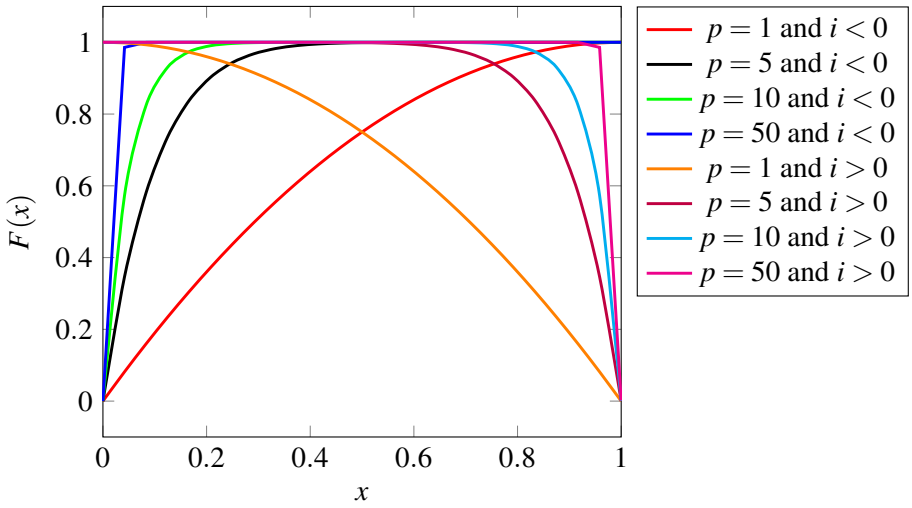
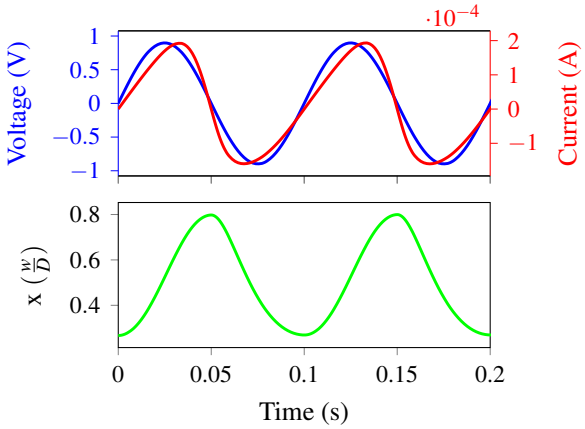
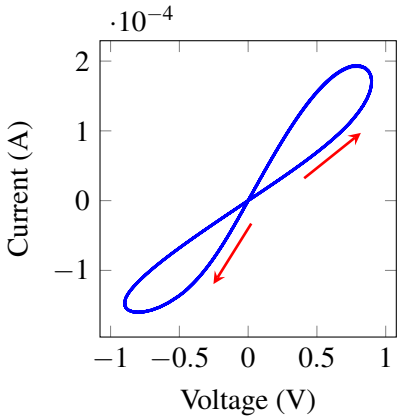


Figure 3.5: Biolek window function for different p values.



(a) Time domain behaviour of the HP Labs with Biolek modification memristor model for a sinusoidal voltage with 0.9 V of amplitude and frequency of 10 Hz.



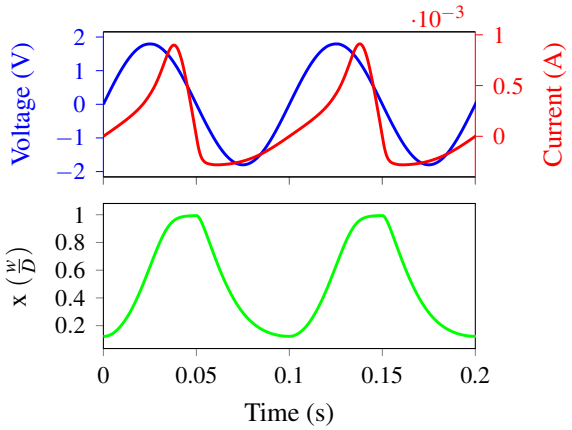
(b) I-V characteristic curve for HP labs with Biolek modification memristor model. The arrows indicate the current flow.

Figure 3.6: Simulation results for HP Labs model with Biolek modification. Input voltage: $0.9\sin(2\pi 10t)$; Memristor parameters: $R_{on} = 1k\Omega$, $R_{off} = 10k\Omega$, $x_0 = 0.5$, $D = 12 \times 10^{-9}m$, $u_v = 2 \times 10^{-14} m^2 s^{-1} V^{-1}$, $p = 1$.

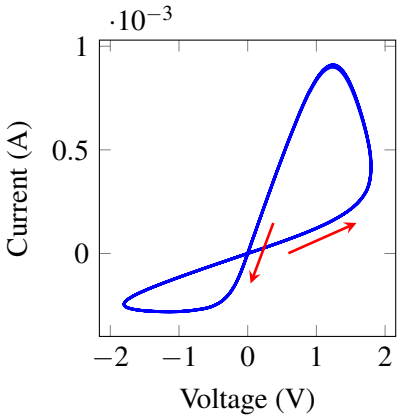
effect of its direction-dependent equation.

3.3 Hyperbolic Sine Models

The vast majority of memristor's structures are based on a Metal electrode, an Insulator layer and another Metal electrode, forming a MIM structure. These type of structures present what is called electron tunneling [38], which are typically characterized by hyperbolic sine functions. This led to the creation of memristor models based on hyperbolic sine functions.



(a) Time domain behaviour of the HP Labs with Biolek modification memristor model for a sinusoidal voltage with 1.8 V of amplitude and frequency of 10 Hz.



(b) I-V characteristic curve for HP Labs with Biolek modification memristor model. The arrows indicate the current flow.

Figure 3.7: Simulation results for HP Labs model with Biolek modification. Input voltage: $1.8\sin(2\pi 10t)$ Memristor parameters: $R_{on} = 1k\Omega$, $R_{off} = 10k\Omega$, $x_0 = 0.076$, $D = 12 \times 10^{-9}m$, $u_v = 2 \times 10^{-14}m^2s^{-1}V^{-1}$, $p = 1$.

3.3.1 General Hyperbolic Sine Model

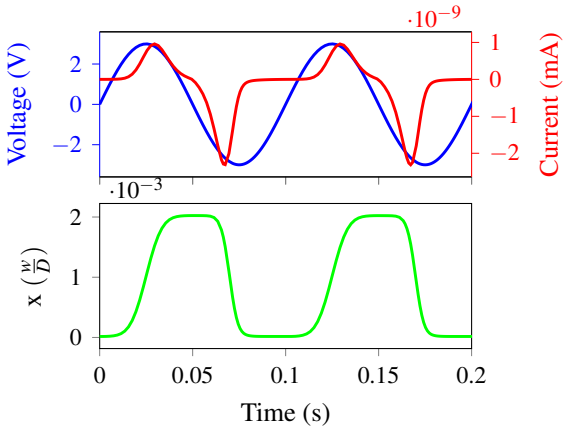
The General Hyperbolic Sine model was proposed Mika Laiho et al. [45] in 2010, where both the current and x motion are based on hyperbolic sine functions (Equations 3.11 and 3.12). The parameters a_1, a_2, b_1 and b_2 are used to adjust the I-V curve, and the parameters c_1, c_2, d_1 and d_2 to ajust the threshold and motion intensity of x. Like *HP Labs* model, this model does not contain limit for x values. To limit the boundaries, it was added a Biolek window function (Figure 3.8).

$$i(t) = \begin{cases} a_1 x(t) \sinh(b_1 v(t)) & v(t) \geq 0 \\ a_2 x(t) \sinh(b_2 v(t)) & v(t) < 0 \end{cases} \tag{3.11}$$

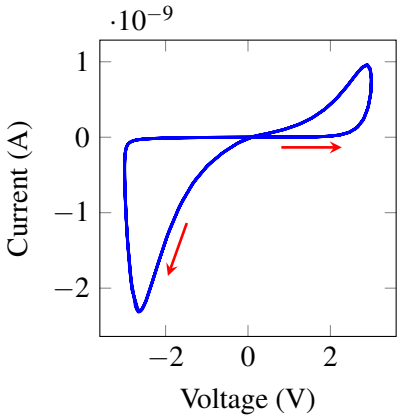
$$\frac{dx}{dt} = \begin{cases} c_1 \sinh(d_1 v(t)) F(x(t)) & v(t) \geq 0 \\ c_2 \sinh(d_2 v(t)) F(x(t)) & v(t) < 0 \end{cases} \tag{3.12}$$

3.3.2 University of Michigan Model

This model was proposed by Ting Chang et al. in 2011 [46]. The memristor's current (Equation 3.13) is a result from two current: one due to the Schottky barrier formed from the metal-insulator layers; and one due the electron tunneling effect from the MIM structure. The x variable no longer represents the length of TiO_2 but the ion migration, which relates to the device's conductivity. α, β, γ and δ are fitting positive parameters for the I-V curve. $\eta 1$, η_2 and λ are used to shape the state variable dynamics (Equation 3.14) and the second term $\frac{x(t)}{\tau}$ was added to include the overlapping



(a) Time domain behaviour of the General Hyperbolic Sine memristor model for a sinusoidal voltage with 3 V of amplitude and frequency of 10 Hz.



(b) I-V characteristic curve for General Hyperbolic Sine memristor model. The arrows indicate the current flow.

Figure 3.8: Simulation results for General Hyperbolic Sine model. Input voltage: $3\sin(2\pi 10t)$. Biolek window parameters: $p = 7$. Model parameters: $a_1 = 4\times10^{-8}$, $b_1 = 1.2$, $a_2 = 1.25\times10^{-7}$, $b_2 = 1.2$, $c_1 = 6\times10^{-4}$, $d_1 = 2$, $c_2 = 6.6\times10^{-4}$, $d_2 = 3.8$, $x_0 = 0.001$

of several hysteresis loop (Figure 3.9).

$$i(t) = (1-x(t))\alpha(1-e^{\beta v(t)}) + x(t)\gamma\sinh(\delta v(t)) \tag{3.13}$$

$$\frac{dx}{dt} = \lambda\left[\eta 1\sinh(\eta_2 v(t)) - \frac{x(t)}{\tau}\right] \tag{3.14}$$

$$f(x,v) = \frac{1+sign(v)}{2}\frac{1+sign(1-x)}{2} + \frac{1+sign(-v)}{2}\frac{1+sign(x)}{2} \tag{3.15}$$

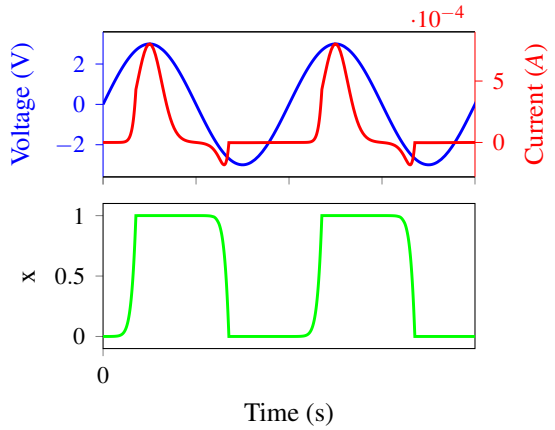
3.3.3 Yacopcic Model

Chris Yakopcic developed this memristor model in 2011 [47]. Like the above models, this one takes advantage of the hyperbolic sine function. The memristor current is similar to the General Hyperbolic Sine model, with a difference in the parameter b, where it is the same whatever voltage polarity:

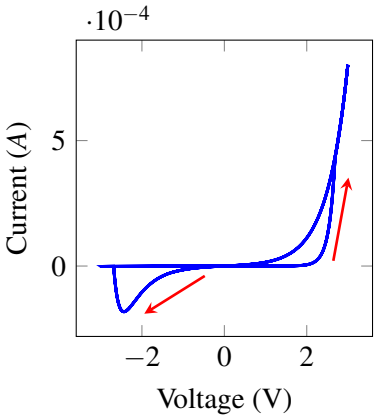
$$i(t) = \begin{cases} a_1 x(t)\sinh(bv(t)) & v(t) \geq 0 \\ a_2 x(t)\sinh(bv(t)) & v(t) < 0 \end{cases} \tag{3.16}$$

The state variable derivative is given by the product of two functions g and f, where η indicates the direction of which the state variable changes. η is typically 1, meaning that a positive voltage bias will increase the memristor's conductivity and a negative voltage bias will decrease it.

$$\frac{dx}{dt} = \eta g(v(t)) f(x(t)) \tag{3.17}$$



(a) Time domain behaviour of the University of Michigan memristor model for a sinusoidal voltage with 3 V of amplitude and frequency of 10 Hz.



(b) I-V characteristic curve for University of Michigan memristor model. The arrows indicate the current flow.

Figure 3.9: Simulation results for University of Michigan model. Input voltage: $3\sin(2\pi 10t)$. Model parameters: $\alpha = 5\times10^{-7}$, $\beta = 0.5$, $\gamma = 4\times10^{-6}$, $\delta = 2$, $x_{min} = 0$, $x_{max} = 1$, $driftbit = 0$, $\lambda = 4.5$, $\eta_1 = 4$, $\eta_2 = 0.004$, $\tau = 10$

g function is used to control the memristor's threshold voltages - using V_p for positive voltage threshold and V_n for positive voltage threshold - and the speed of transitioning from On to Off or vice versa - using A_p for positive x motion speed and A_n for negative x motion speed:

$$g(v(t)) = \begin{cases} A_p(e^{v(t)} - e^{V_p}) & v(t) > V_p \\ A_n(e^{-v(t)} - e^{V_p}) & v(t) < -V_n \\ 0 & -V_n \leq v(t) \geq V_p \end{cases} \tag{3.18}$$

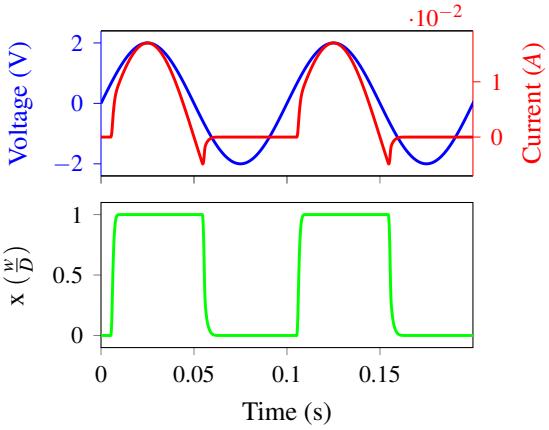
f is a windowing function (Equation 3.19) that reduces the carrier speed when x reaches the boundaries, just like the window functions previously studied in this chapter. The x motion reduction depends on the voltage polarity and the window only takes effect when x reaches the boundaries defined by x_p or x_n. α_p and α_n are parameters used to define the decay of x motion. w_p and w_p are used to ensure that f is 0 when $x = 1$ or $x = 0$ (Equations 3.20 and 3.21).

$$f(x(t)) = \begin{cases} \begin{cases} e^{e^{-\alpha_p(x(t)-x_p)}w_p(x(t),x_p)} & x(t) \geq x_p \\ 1 & x(t) < x_p \end{cases} & \eta v(t) > 0 \\ \begin{cases} e^{e^{-\alpha_n(x(t)+x_n)-1}w_n(x(t),x_n)} & x(t) \leq 1 - x_n \\ 1 & x(t) > 1x_n \end{cases} & \eta v(t) \leq 0 \end{cases} \tag{3.19}$$

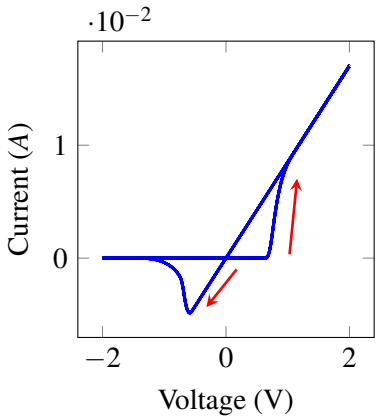
$$w_p(x,x_p) = \frac{x_p - x}{1 - x_p} + 1 \tag{3.20}$$

$$w_n(x,x_n) = \frac{x}{1 - x_n} \tag{3.21}$$

A simulation was done to verify its transient behavior and respective I-V curve (Figure 3.10).



(a) Time domain behaviour of the Yakopcic's memristor model for a sinusoidal voltage with 2 V of amplitude and frequency of 10 Hz.



(b) I-V characteristic curve for Yakopcic's memristor model. The arrows indicate the current flow.

Figure 3.10: Simulation results for University of Michigan model. Input voltage: $3\sin(2\pi 10t)$. Model parameters: $a_1 = 0.17$, $a_2 = 0.17$, $B = 0.05$, $V_p = 0.65$, $V_p = 0.56$, $A_p = 4000$, $A_n = 4000$, $x_p = 0.3$, $x_n = 0.5$, $\alpha_p = 1$, $\alpha_n = 5$, $x_0 = 0.11$, $\eta = 1$

3.4 *Knowm*'s Mean Metastable Switch model

A new type of memristor (Self Directed Channel or SDC) was created by Kris Campbell [48] in 2017 and it is based on MetaStable Switches (MSSs). A MSS is an ideal two state element which changes it state with a probability dependent on the voltage bias and temperature. The memristor is formed by a collection of MSSs that change state over time, which gives them the hysteresis and memory behaviour. As opposed to this previous models studied in this chapter, the MSS model is not deterministic, it is rather probabilistic and stochastic but the final equations rely on hyperbolic sine functions.

The memristor current is defined by Equation 3.22: i_m is the current dependent on the MSSs - the memristor represents a collection of channels which switch to one of the two states with different resistances, thus creating a current according to the voltage bias; i_S is a current formed due to the Schottky barrier formed by the metal-oxide junction; and Φ is the term that gives a relative quantity of each current.

$$i = \Phi i_m(v,t) + (1-\Phi) i_S(v) \tag{3.22}$$

The Schottky barrier current is given by Equation 3.23, where α_f, α_r, β_f and β_r are positive values that shape its exponential behaviour.

$$i_S = \alpha_f e^{\beta_f v} - \alpha_r e^{-\beta_r v} \tag{3.23}$$

The MSS had two states A a B and each state has a probability to move to the other state. The probability of going from B state to A is given by Equation 3.24 and from B to A by Equation 3.25.

$$P_A = \alpha \frac{1}{1+e^{-\beta(v-v_A)}} = \alpha\Gamma(v, v_A) \tag{3.24}$$

$$P_B = \alpha(1-\Gamma(v, v_B)) \tag{3.25}$$

β is the thermal voltage (Equation 3.26) and α is the ratio of time step period in relation to the time scale of the device (Equation 3.27).

It can be seen that P_A and P_B are based on $\frac{1}{1+e^{-x}}$, similar to the hyperbolic sine models studied previously.

$$\beta = \frac{q}{kT} \tag{3.26}$$

$$\alpha = \frac{dt}{\tau} \tag{3.27}$$

The memristor is modeled as a collection of N MSSs, so its conductance is given by the sum of every MSSs. N_A is the number of MSSs in state A, N_B is the number of MSSs in state B and $N = N_A + N_B$:

$$G_m = N_A G_A + N_B G_B = N_B(G_B - G_A)NG_A \tag{3.28}$$

The probability of k out of n MSSs change state is given by a binomial distribution, with p being the state transition probability:

$$P(n,k) = \binom{n}{k} p^k (n-k) \tag{3.29}$$

When n is sufficiently large, $P(n,k)$ can be approximated by a normal distribution, with $\mu = np$ and $\sigma^2 = np(1-p)$:

$$\lim_{x\to\infty} P(n,k) = \frac{e^{\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} = \mathcal{N}(\mu, \sigma^2) \tag{3.30}$$

The update to the memristor conductance is the sum of two random variables:

$$\Delta N_B = \mathcal{N}(N_A P_A, N_A P_A(1-P_A)) - \mathcal{N}(N_B P_B, N_B P_B(1-P_B)) \tag{3.31}$$

The final conductance update is given by:

$$\Delta G_m = \Delta N_B(G_B - G_A) \tag{3.32}$$

The change of the MSS number is:

$$dx = N_{OFF\rightarrow ON} - N_{ON\rightarrow OFF} \tag{3.33}$$

And the number of MSSs switching state:

$$N_{OFF\rightarrow ON} = P_{OFF\rightarrow ON}(1-x) \tag{3.34}$$

$$N_{ON\rightarrow OFF} = P_{ON\rightarrow OFF}x \tag{3.35}$$

Replacing Equations 3.34 and 3.35 in 3.33:

$$dx = P_{OFF\rightarrow ON}(1-x) - N_{ON\rightarrow OFF} = P_{ON\rightarrow OFF}x \tag{3.36}$$

And replacing Equations 3.24 and 3.25:

$$dx = \frac{dt}{\tau}\frac{1}{1+e^{-\beta(v-v_A)}}(1-x) - \frac{dt}{\tau}\left(1 - \frac{1}{1+e^{-\beta(v+v_B)}}\right)x \tag{3.37}$$

$$\frac{dx}{dt} = \frac{1}{\tau}\left[\frac{1}{1+e^{-\beta(v-v_A)}}(1-x) - \left(1 - \frac{1}{1+e^{-\beta(v+v_B)}}\right)x\right] \tag{3.38}$$

The final memristor conductance is dependent on x:

$$G = \frac{x}{R_{ON}} + \frac{1-x}{R_{OFF}} \tag{3.39}$$

And finally it can be applied Ohm's Law:

$$I = G \times V \tag{3.40}$$

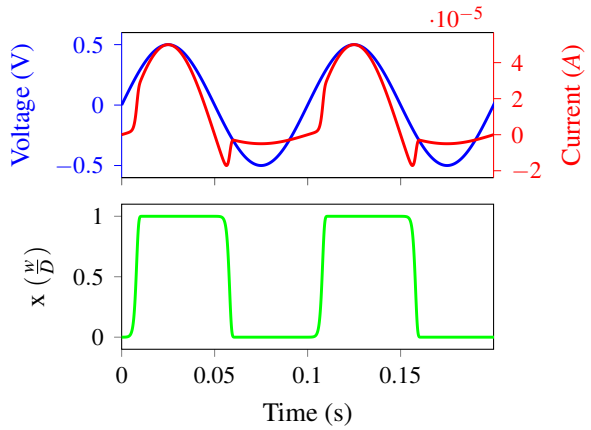
3.4.1 Simulation Model Implementation

In this subsection it is obtained the characteristic I-V curve for 3 *Knowm*'s Tungsten commercial memristors in order to obtain mean model parameters to implement in a simulation model.

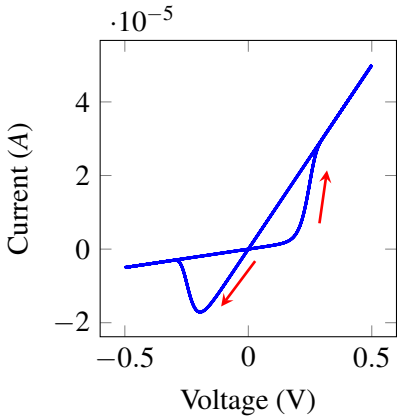
3.4.1.1 Memristor Data

Data from physical memristors were obtained. They should be handled very carefully due to their low current tolerance. In the case of Tungsten memristors, they can handle currents up to 1 mA, according to its datasheet [49].

To obtain the memristors I-V curve, it was used a *Keithley* 2450 Source Measure Unit (SMU) and the circuit consisted on a current limiting resistor (Equation 3.41) in series with the memristor. Circuit current was measured, as well as the input voltage.



(a) Time domain behaviour of the Knowm memristor model for a sinusoidal voltage with 0.5 V of amplitude and frequency of 10 Hz.



(b) I-V characteristic curve for Knowm memristor model. The arrows indicate the current flow.

Figure 3.11: Simulation results for Knowm Mean Metastable Switch model. Input voltage: $0.5\sin(2\pi 10t)$. Model parameters: $R_{on} = 10k\Omega$, $R_{off} = 100k\Omega$, $V_{off} = 0.3V$, $V_{on} = 0.3V$, $T = 298.5K$.

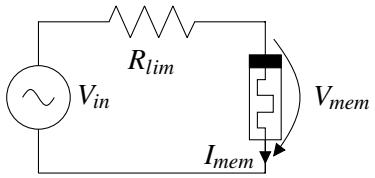


Figure 3.12: Memristor I-V characterization schematic.

The circuit's current is given by:

$$I_{mem} = \frac{V_{in}}{R_{lim} + R_{mem}} \tag{3.41}$$

To calculate R_{lim}, the memristor resistance should be the lowest. Considering $R_{mem} = 0$, $V_{in} = 0.5V_{peak}$ and $I_{mem} = 1mA$:

$$1 \times 10^{-3} = \frac{0.5}{R_{lim} + 0} \tag{3.42}$$

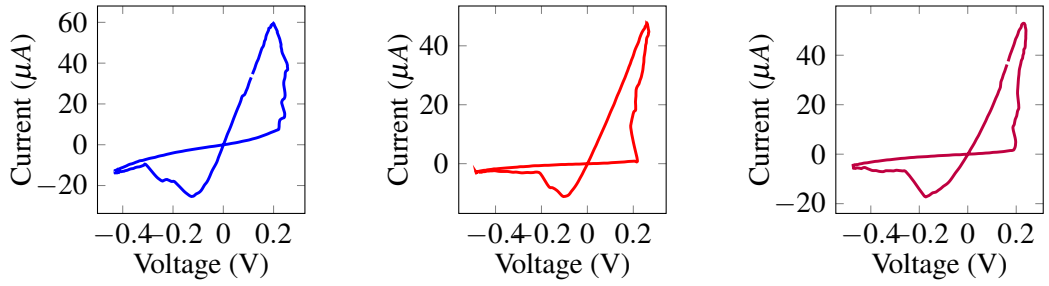
For limiting the maximum current of $1mA$, R_{lim} should be:

$$R_{lim} \geq 500\Omega \tag{3.43}$$

The limiting resistor value chosen was $5k\Omega$ and the input voltage was $0.5\sin(2\pi 100t)$.

The experimental results are in Figure 3.13. To obtain a simulation model, it was made a script in MatLab to obtain R_{on} and R_{off} for each memristor (recurring to linear regression) and then calculate the mean value:

$$R_{on} = 4.419k\Omega; R_{off} = 121.826k\Omega \tag{3.44}$$



(a) Memristor 1 I-V curve. (b) Memristor 2 I-V curve. (c) Memristor 3 I-V curve.

Figure 3.13: I-V curves of three Knowm memristors.

To obtain the other parameters, several simulations of the circuit 3.12 were made to adjust its parameters to get a mean approximation of the 3 memristors I-V curves.

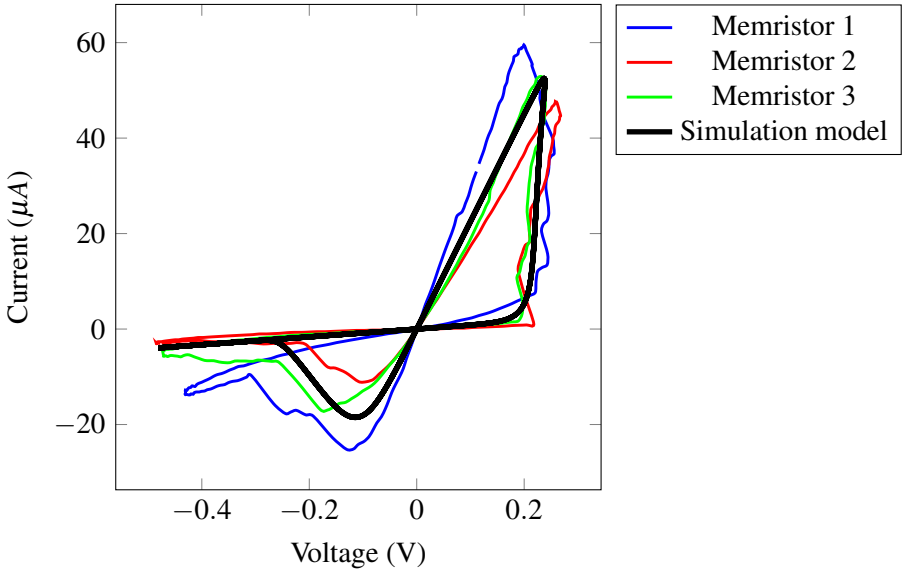


Figure 3.14: I-V curves of three Knowm memristors and simulation model.

The simulation model correlates with the experimental data and the obtained parameters are shown in Table 3.1.

R_{on}	R_{off}	V_{on}	V_{off}	τ	T
$4.419k\Omega$	$121.83k\Omega$	$0.3V$	$0.1777V$	$30\mu s$	$298.5K$

Table 3.1: Model parameters for experimental data fitting.

3.4.1.2 The Drift Problem

This model as a serious issue when it comes to holding it's state. From Equation 3.38, it is possible to notice that when the voltage at the memristors terminals is 0V, the x motion results in a value

different from 0, making the memristor change its state without any voltage bias (EEqaution 3.45).

$$\frac{dx}{dt} = \frac{1}{\tau}\left[\frac{1}{1+e^{\beta v_A}}(1-x) - \left(1 - \frac{1}{1+e^{-\beta v_B}}\right)x\right] \tag{3.45}$$

Assuming that $x = 1$ and $a = \frac{1}{1+e^{\beta v_A}}$ and $b = \frac{1}{1+e^{-\beta v_B}}$:

$$\frac{dx}{dt} = \frac{b-1}{\tau} \tag{3.46}$$

b is a value always greater than 0 and lesser than 1, so the x motion will resulting in a negative number, meaning that x value will decrease when no voltage is applied. The opposite happens when $x = 0$:

$$\frac{dx}{dt} = \frac{a}{\tau} \tag{3.47}$$

Like b, a is a value always greater than 0 and lesser than 1 and, consequently, the x motion will be a positive value, which implies a increase in the x value when there is no voltage bias.

A transient simulation was made with the memristor simulation model obtained in 3.14, with its terminals connected to ground and initial conditions $x = 0$ and $x = 1$ (Figure 3.15).

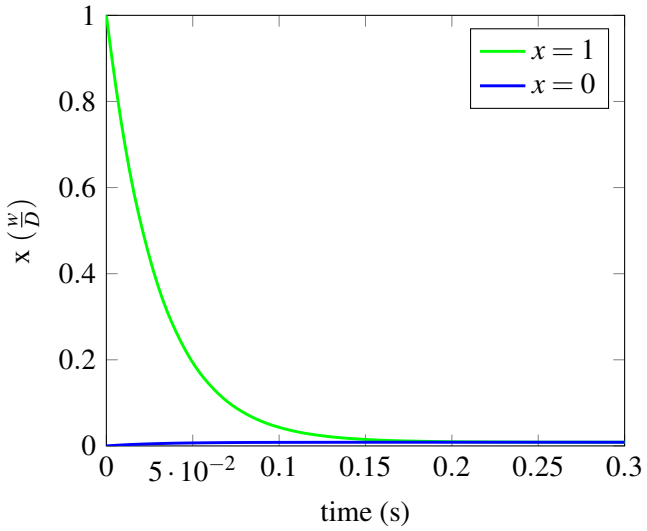


Figure 3.15: Memristor drift with 0V applied in both terminals. Parameters used from Table 3.1.

3.4.1.3 Simulation Model Modification

A solution for this models problem is to add a *PSpice IF* statement to the x motion: if there is a current accross the memristor (meaning it is biased), the x motion will be given by Equation 3.38; else, its value will be 0, assuring that the memristors does not change it's state. This is a simple solution to simulate the memristor's memory capability, which may not comply with the physical behavior of the device, because its state retention may not last forever.

The Spice code with the modification can be found in the Appendix.

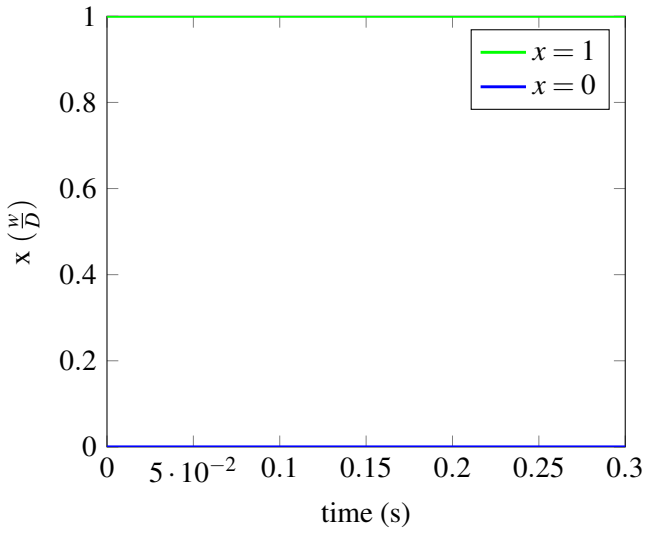
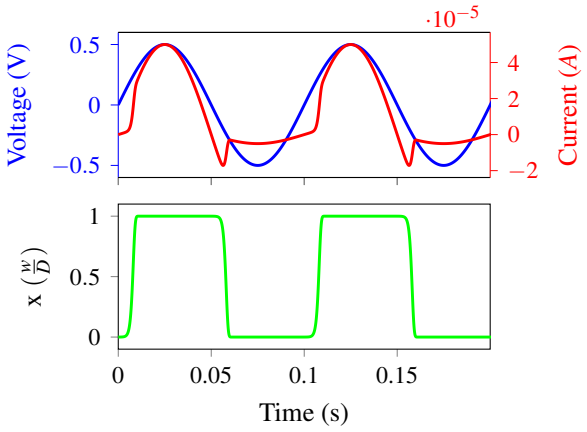


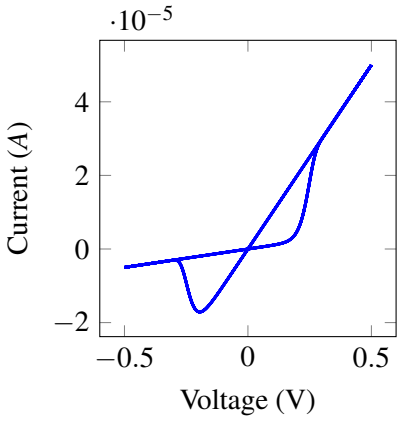
Figure 3.16: Memristor drift with 0V applied in both terminals. Parameters used from Table 3.1.

A transient simulation of the modified model, with its results shown in Figure 3.16. I can be seen than now the memristors has the capability to retain its state.

To confirm if the model still holds its behavior, a simulation was made, using the parameters from 3.11.



(a) Time domain behaviour of the Knowm memristor model with proposed modificiation for a sinusoidal voltage with 0.5 V of amplitude and frequency of 10 Hz.



(b) I-V characteristic curve for the modified Knowm memristor model.

Figure 3.17: Simulation results for Knowm Mean Metastable Switch model with proposed modification. Input voltage: $0.5\sin(2\pi 10t)$. Model parameters: $R_{on} = 10k\Omega$, $R_{off} = 100k\Omega$, $V_{off} = 0.3V$, $V_{on} = 0.3V$, $T = 298.5K$.

Simulation results from Figure 3.17 show that, with the modification, the memristor's hysteretic behaviour remains the same.

Chapter 4

Reconfigurable Logic

In this chapter, memristor dynamics are studied from a digital perspective. A brief introduction to Look-Up tables is presented in order to gain a better outlook on the circuits with this functionality. A circuit proposal is made with has advantages compared to the studied circuits.

All simulations and designs were done in *Cadence* with the *UMC* 180*nm* technology. When not explicitly indicated the opposite, the PMOS substrates are connected to the highest voltage, V_{dd}, and the NMOS substrates are connected to the lowest voltage, ground. All square waveform inputs have 100*ps* rise and fall time.

4.1 Memristor Dynamics from Digital Perspective

In digital circuits, there are only two voltage levels: the low voltage - corresponding to a digital 0, and the high voltage - corresponding to a digital 1. In this perspective, the memristor I-V curve can defined by two straight lines (Figure 4.1), one for memristor state R_{off} (Equation 4.1) and other for R_{on} (Equation 4.2) . The transition between these two lines is given by two threshold voltages, v_{tl} when memristor goes from R_{on} to R_{off} and v_{th} when memristor goes from R_{off} to R_{on}. This means that, when the memristor is in R_{off} state and reaches the threshold voltage v_{th}, the memristor changes to R_{on}. The same happens when memristor is in R_{on}: when it reaches the threshold voltage v_{tl}, the memristor changes to R_{off}.

$$i = \frac{v}{R_{off}} \tag{4.1}$$

$$i = \frac{v}{R_{on}} \tag{4.2}$$

With this approach, a digital 0 can be assign to R_{off} and a digital 1 can be assign to R_{on}, for example. It simplifies the circuit project at an initial stage. In later stages, all other parameters have to be taken into account, like the state transition times and the fact that the memristor changes its resistance whenever a voltage is applied to its terminals, due to x derivative being depend of its voltage/current (Equation 3.38 for *Knowm*'s model).

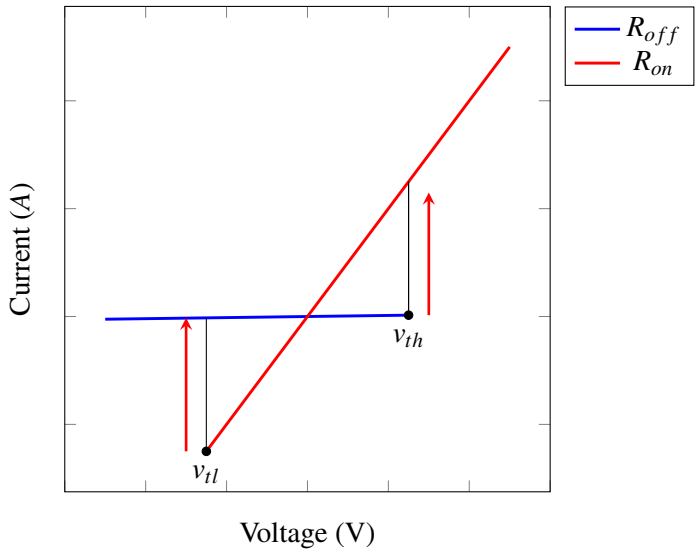


Figure 4.1: Simplified I-V curve for the memristor.

4.2 FPGAs and Look-Up Tables

When it comes to Application Specific Integrated Circuit (ASIC) design flow [50], one of the fundamental steps is the Logic and Functional design. In this step it its defined the hardware (logic gates and interconnections) to be implemented in the ASIC. This hardware is described through an Hardware Description Language (HDL) and is implemented in Field Programmable Gate Array (FPGA) to test its functionality. An FPGA consists on an IC which can be programmed with an array of logic gates and interconnections to achieve sequential and combinatorial circuits.

Most of FPGA are formed by Configurable Logic Blocks (CLB). This CLBs are blocks in which its logic can be programmed, as the name implies. It is basically a programmable logic gate. The logic of this blocks is stored in Look-Up Tables (LUTs), which are arrays of memory cells. For example, implementing a three input XOR gate would require a LUT with $2^3 = 8$ memory cells. The output for all the input combinations are stored in the LUT. This example is shown on Table 4.1.

Gate Inputs			**Expected**	**LUT**	**Value Stored**	**LUT**
A2	**A1**	**A0**	**Output**	**memory cell**		**Output**
0	0	0	0	C0	0	0
0	0	1	1	C1	1	1
0	1	0	1	C2	1	1
0	1	1	0	C3	0	0
1	0	0	1	C4	1	1
1	0	1	0	C5	0	0
1	1	0	0	C6	0	0
1	1	1	0	C7	0	0

Table 4.1: LUT configuration to simulate a 3 input XOR gate.

4.3 Robinson Pino et al. circuit

Robert Pino et al. patented a circuit in 2013 [51] and later published it in Wey Yu's et al. Book chapter [52], which consists on a LUT based on memristors, exploring their memory properties, and CMOS, for handling the memristors.

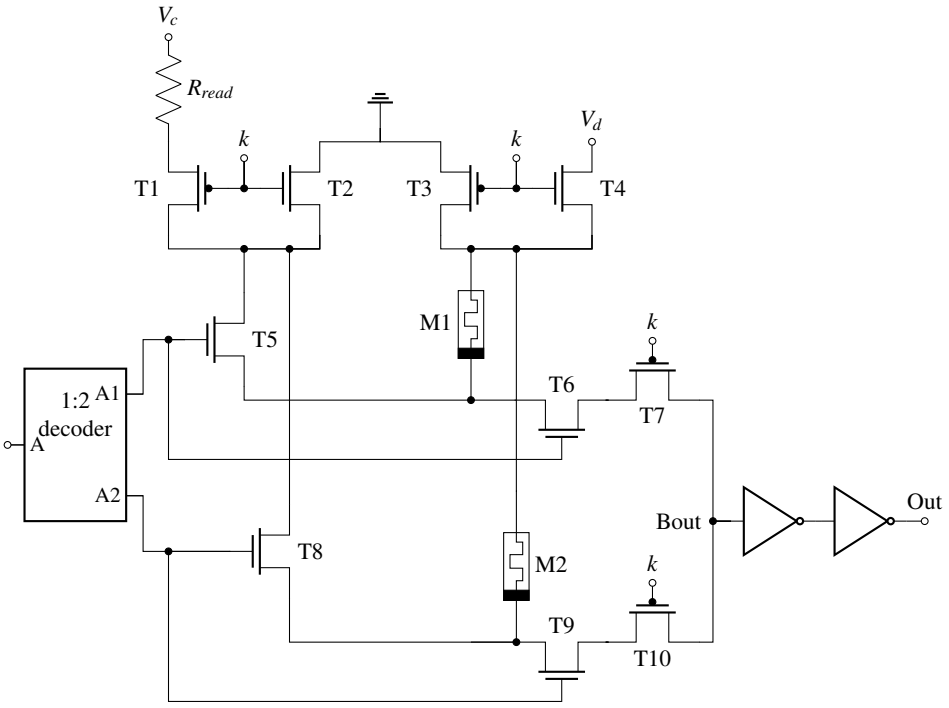


Figure 4.2: Circuit proposed by Pino et al. [51].

In this example (4.22), it is a 1 bit LUT and the circuit operation can be described as follows:

- The first stage of this circuit is to translate an address to a bit, using a decoder with n inputs and 2^n outputs for a n input LUT. In this case, the decoder should have 1 inputs and 2 outputs. This ensures that only 1 bit is selected;

- The memristors are selected using two NMOS transistors - T5 and T6 for M0 and T8 and T9 for M1. The biasing of the two NMOS allows the respective memristor to be accessed and modified;

- There are two main paths: the one formed by the NMOS T2 and T4, used for changing the selected memristor's resistance - which can be called write path; and the one formed by the PMOS T1 and T3 and the resistor R_1, used to read the selected memristor resistance - which can be called read path. When in write mode, a voltage V_d is applied and it is sufficiently large to change the memristor's state. When in read mode, the resistor R_{read} and the memristor create a voltage divider. A low voltage V_c is applied to the divider in order to prevent the memristor to change its state completely. Neglecting the transistors channel

resistance, the voltage divider is given by Equation 4.3. R_{read} value is chosen to maximize the reading margin (Equation 4.4 and Figure 4.3).

- The node k selects which path will be used: if $k = 0$, read path is selected; if $k = 1$, write path is selected. T7 and T10 are used to isolate the memristors from the buffering stage when in write mode;

- The inverters are used to restore the signal from the voltage divider.

$$V_{Bout} = \begin{cases} V_{M1} = \frac{R_{M1}}{R_{read}+R_{M1}} V_c & \text{if } A1 = 1 \\ V_{M2} = \frac{R_{M2}}{R_{read}+R_{M2}} V_c & \text{if } A2 = 1 \end{cases} \tag{4.3}$$

$$V_{margin} = \frac{R_{mem,max}}{R_{read} + R_{mem,max}} V_c - \frac{R_{mem,min}}{R_{read} + R_{mem,min}} V_c \tag{4.4}$$

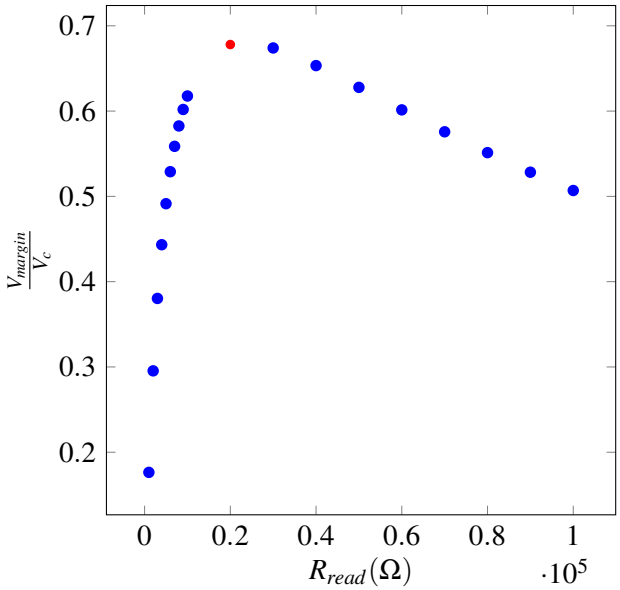


Figure 4.3: Normalized voltage margin function of the memristors resistance. $R_{mem,min}$ and $R_{mem,min}$ where taken from the simulation model 3.44 and are, respectively, $4.419k\Omega$ and $121.83k\Omega$. The R_{read} that offers higher reading margin is $20k\Omega$.

For all input values and memristor states, the circuit output is given by Table 4.2.

In total, the circuit contains 4 transistors for reading and writing the memristors and 3 transistors per memristor:

$$T_{total} = 4 + 3M \tag{4.5}$$

Input Value	Decoder Output Selection	Memristor State Configuration		Output Value
		M1	M2	
0	A1	R_{off}	R_{off}	0
0	A1	R_{off}	R_{on}	0
0	A1	R_{on}	R_{off}	1
0	A1	R_{on}	R_{on}	1
1	A2	R_{off}	R_{off}	0
1	A2	R_{off}	R_{on}	1
1	A2	R_{on}	R_{off}	0
1	A2	R_{on}	R_{on}	1

Table 4.2: Memristor configuration and respective output.

4.4 Robinson Pino et al. Circuit Modification

In 2017, Xiaoping Wang et al. [53] proposed a modification (Figure 4.4) to Pino's circuit 4.22: the decoupling transistors T7 and T10 can be removed from the circuit because only one memristor is being selected at any given point in time, so it will not impact any of the unselected memristors. The circuit operation remains the same.

The proposed circuit contains a total of 4 transistors for reading and writing the memristors and 2 transistors per memristor, one less than the previous circuit:

$$T_{total} = 4 + 2M \tag{4.6}$$

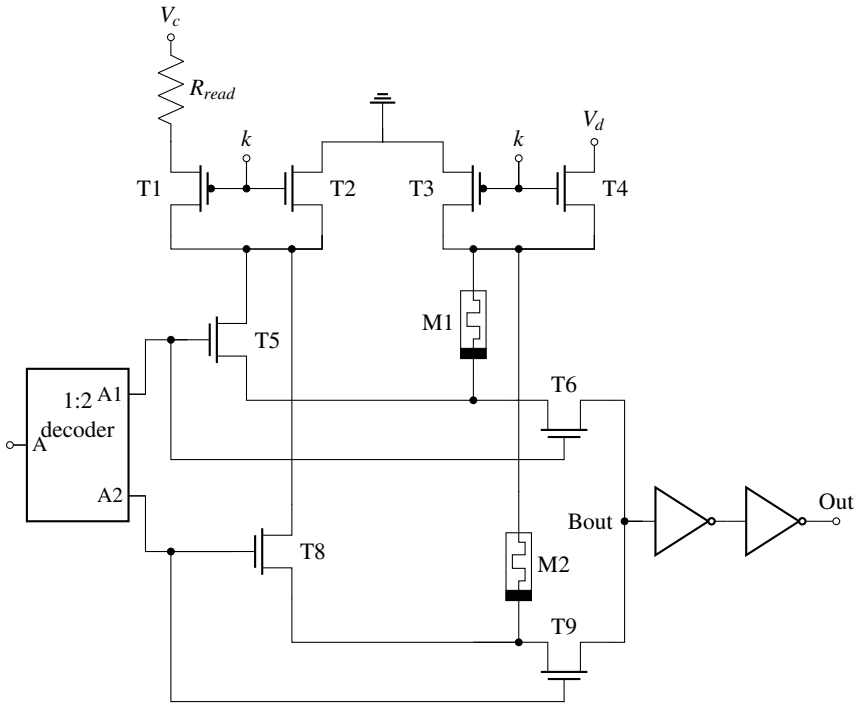


Figure 4.4: Circuit proposed by Wang et al. [53].

4.5 Proposed Circuit

Although the previous circuits have promising results when it comes to transistor count and area overhead in comparison to a conventional SRAM LUT, no control circuit was projected to manipulate the LUT functionalities and, with the increase of the LUT's size, the number of transistors will have a multiplication factor of 3 (for Pino's proposal) or 2 (for Wang's proposal) for each memristor. This can be optimized by using a strategic memristor reading point, B_{out}, using only one transistor per memristor (Equation 4.7). A proposal was made in this dissertation, the schematic is presented in Figure 4.5. The circuit contains a total given by Equation 4.7

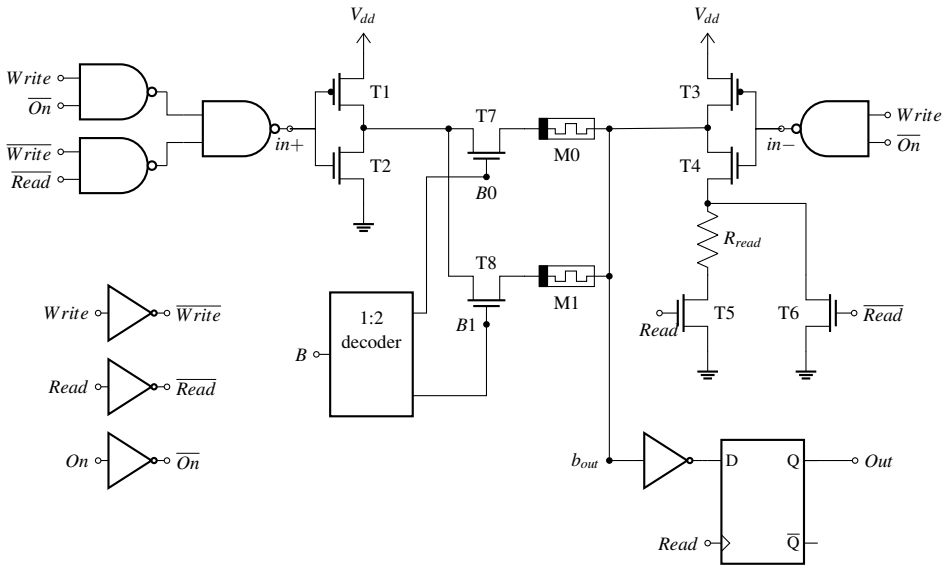


Figure 4.5: Proposed circuit for a 1 bit LUT.

$$T_{total} = 40 + M \tag{4.7}$$

4.5.1 LUT Signals

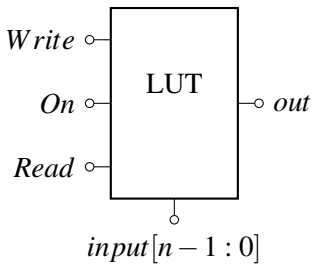


Figure 4.6: Equivalent block for the circuit in 4.5.

The LUT signals (Figure 4.6) are described as follows:

- $Write$: this signal is used to write a bit in the LUT. When 1, the value in On is written to the selected $input$. When no write operation is needed, it should be 0;
- On: this signal is the value to be saved to the selected bit in the LUT. This signal is used only when $Write$ is 1. In this situation, when On is 1, it changes the selected memristor to R_{on}, when On is 0, it changes the selected memristor to R_{off};
- $Read$: this signal is used to read a LUT value selected by $input$. This signal is only used when $Write$ is 0. When a read is needed, this signal gives a short pulse to 1, enough to read the selected memristor and presenting it's value in the output;
- $input[2^n : 0]$: this signal is used to selected a memristor in the LUT to perform a read or write operation;
- out: this signal is used to output the value of the read memristor. This signal is valid only when a read operation is done.

4.5.2 MOS Sizing

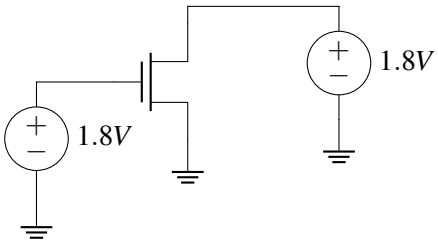
Simulations for NMOS (Figure 4.7a) and PMOS (Figure 4.7b) were made in order to get the transistor sizing. The width of the transistors was sweept to check which value the current is greater or equal than the maximum memristor current. From the memristor simulation model, the maximum current is given by:

$$i_{mem,max} = \frac{v_{mem}}{R_{mem,min}} \tag{4.8}$$

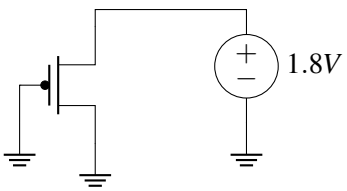
Assuming $v_{mem} = 1.8V$:

$$i_{mem,max} = \frac{1.8}{4419} \approx 408\mu A \tag{4.9}$$

The resulting width are shown in Table 4.3, values relative to $w_{min} = 240nm$.



(a) NMOS sizing circuit.



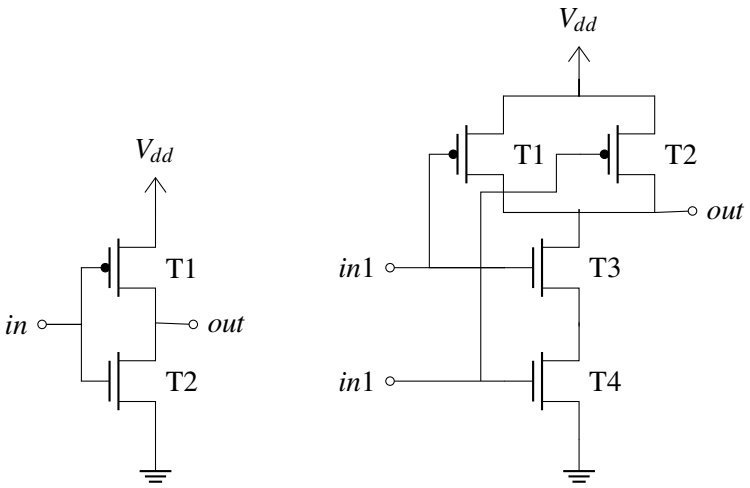
(b) PMOS sizing circuit.

Transistor	T1	T2	T3	T4	T5	T6	T7	T8	Logic Gates PMOS	Logic Gates NMOS
Width	$7w_{min}$	$3w_{min}$	$7w_{min}$	$3w_{min}$	$3w_{min}$	$3w_{min}$	$3w_{min}$	$3w_{min}$	w_{min}	w_{min}

Table 4.3: Resulting transistor width sizes.

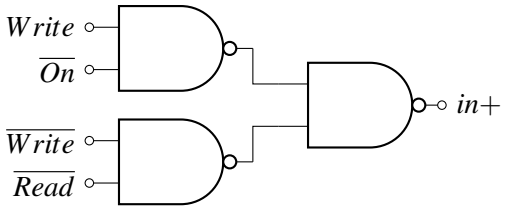
4.5.3 Logic Gates

For the NOT gates, the circuit used is in Figure 4.7a and the NAND gates in Figure 4.7b. These gates are used to control the LUT. The Table 4.4 contains the truth table of the control circuit and the respective functionality for each value.

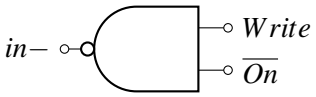


(a) NOT gate circuit.

(b) NAND gate.



(a) Logic circuit used for controling $in+$.



(b) Logic circuit used for controling $in-$.

Figure 4.7: Control gates.

$Write$	On	$Read$	$in+$	$in-$	**functionality**
1	0	X	0	1	Write R_{off} to memristor
0	1	X	1	0	Write R_{on} to memristor
0	X	0	1	1	Standby mode (no operation is applied)
0	X	1	0	1	Read memristor state

Table 4.4: Logic table for control circuit. X represents a *Don't Care* bit.

4.5.4 Minimum Write Time

The minimum write time (t_{write}) is the time required to make the memristor change from one state to another, R_{off} to R_{on} (t_{off-on}) or R_{on} to R_{off} (t_{on-off}). The highest time will define the minimum writing time, in order to ensure that the memristor changes its state.

To obtain the memristor writing times, a transient simulation was made with the circuit from Figure 4.8.

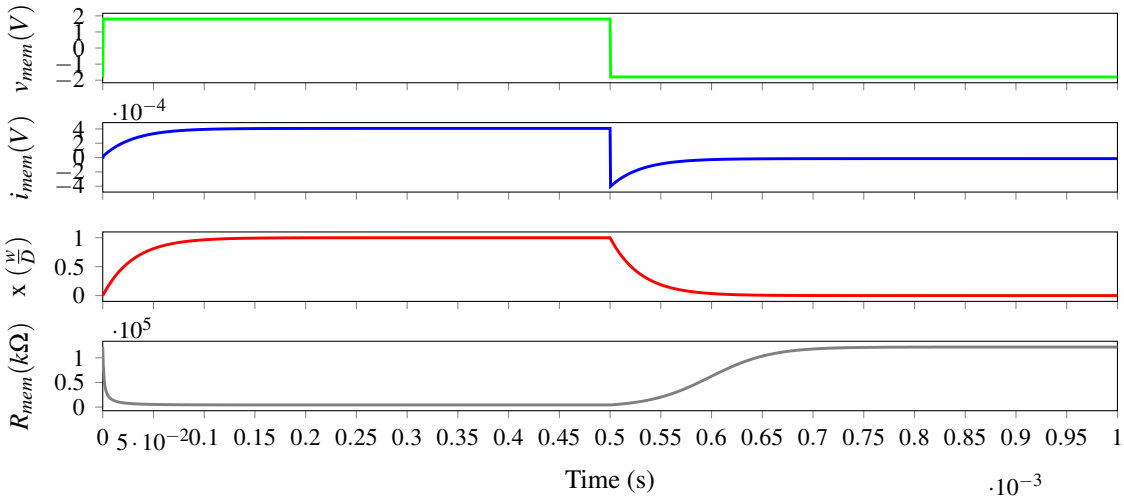


Figure 4.8: Memristor writing times.

The writing times of both transitions are obtained by counting the time from the initial resistance value to a resistance value with 1% tolerance of the transition interval - $(R_{low} - R_{high}) \times 1\%$. This tolerance is used to avoid the exponential behavior of x: as x goes towards a boundary, its derivative tends to get very low and, consequently, it takes a long time to achieve the final boundary value. Reaching a very close value will take much less time and it will pratically not affect the saved value tolerance.

The writing times obtained are:

$$t_{off-on} = 45\mu s; t_{on-off} = 239\mu s \tag{4.10}$$

Resulting in a minimum writing time of:

$$t_{write} = 239\mu s \tag{4.11}$$

To ensure the writing is done, the circuit uses a $2kHz$ ($500\mu s$) impulse for its writing mechanism.

4.5.5 Memory Cell

Each memory cell is composed by a transistor and a memristor. The transistor acts as a switch: when $B = 1$, T is biased and a current will flow through the memristor, when $B = 1$, T is cutoff and the memristor maintains its state.

To test the memory cell's behavior, a $0V$ to $1.8V$ square voltage with $1kHz$ was applied to v_p and a similiar voltage, with a 180° phase shift, was applied to v_m. This allows to alternate the memory cell's applied voltage polarity, $v_p - v_m$, between $-1.8V$ and $1.8V$ and test the current in

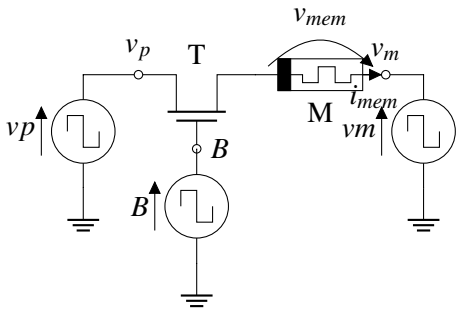


Figure 4.9: Simulation circuit for the memory cell.

both directions. A $0V$ to $1.8V$ square voltage with $500Hz$ was applied to B to test the memristor selection.

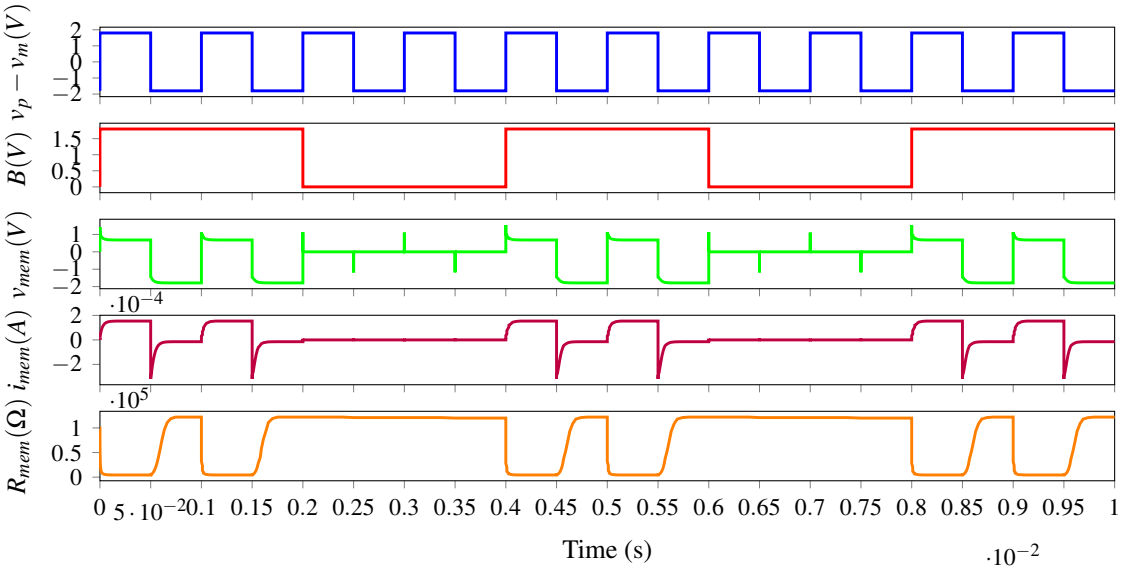


Figure 4.10: Memory cell transient simulation.

From the simulation results (Figure 4.10), it can be seen that, when $B = 1.8V$ (thus T0 is biased), a voltage is applied to the memristor, thus changing is conductivity according to the voltage polarity. When $B = 0V$ (thus T0 is in the cutoff region), the memristor is high impedance state, meaning that its voltage is 0, causing its state to be maintained. Voltage spikes can be seen in v_{mem} when $B = 0$ and a voltage transition in v_p or v_m occurs, which are caused by the NMOS parasitic capacitances. These spikes not affect the memristor's state, since their energy is considerably low.

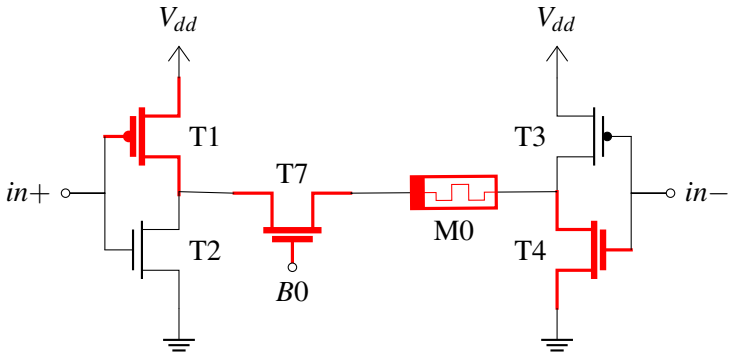
4.5.6 Bridge Stage

To write and read the memristor's state, an H bridge, formed by T1-4, is used to apply a voltage with the polarity desired. T5 and T6 are used to alternate shorting T4 Source to ground (for writing), or adding a resistor (for reading).

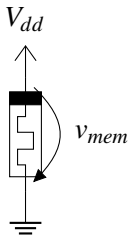
4.5.6.1 Bridge Write

When in a write operation, the *Write* is pulled to 1 and *Read* pulled to 0. This puts T5 in the cutoff region and biases T6, shorting the T4 Source, resulting in the circuits in Figure 4.11.

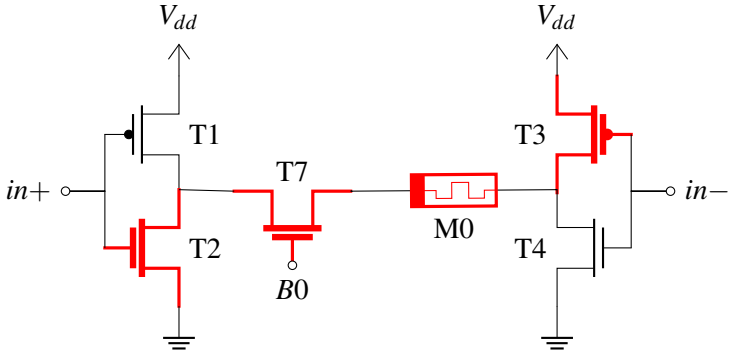
To lower the memristor's resistance, T1 and T4 are connected in order to get a positive voltage from accross the memristor (Figure 4.11a). According to the memristor model obtained in Figure 3.14, the voltage applied is higher than the positive threshold voltage, making the memristor change it's resistance to R_{on}. The same process to increase the memristor's resistance (Figure 4.11c): T2 and T3 are biased and a negative voltage is applied. Consequently, this voltage is low enough to change the memristor resistance to R_{off}.



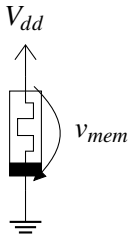
(a) H brigde when changing memristor state to R_{on}. Red transistors indicate the current flow.



(b) Equivalent schematic of 4.11a.



(c) H brigde when changing memristor state to R_{off}. Red transistors indicate the current flow.



(d) Equivalent schematic of 4.11c.

Figure 4.11: Schematics for H bridge.

4.5.6.2 Bridge Read

To read the memristor, a voltage divider is used. The H bridge is used with T1, T4 and T5 biased (Figure 4.13a). It was applied a $500MHz$ square wave to *Read* in order to obtain $1ns$ reading intervals.

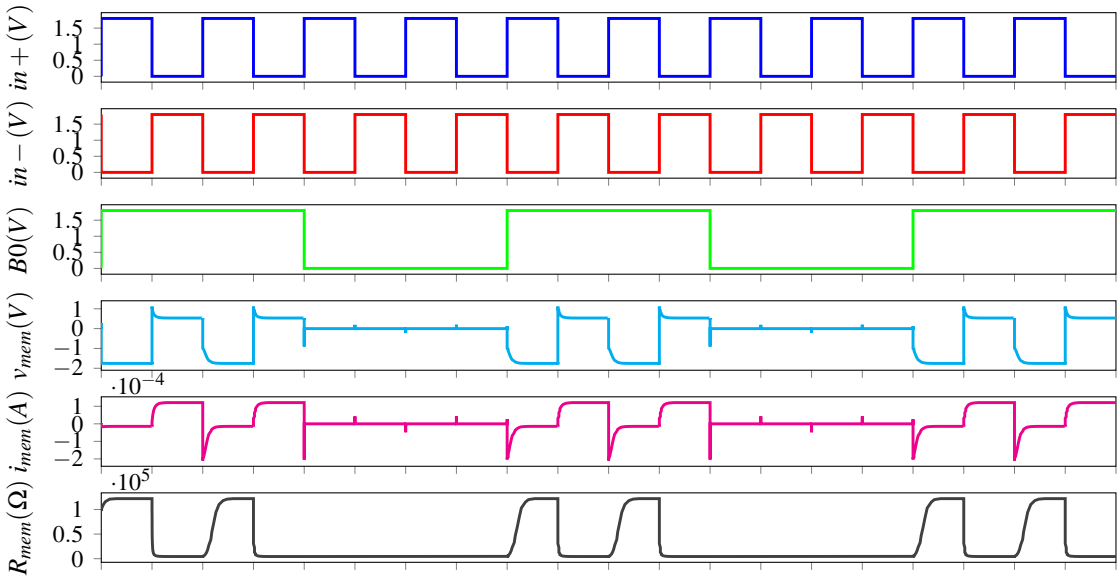
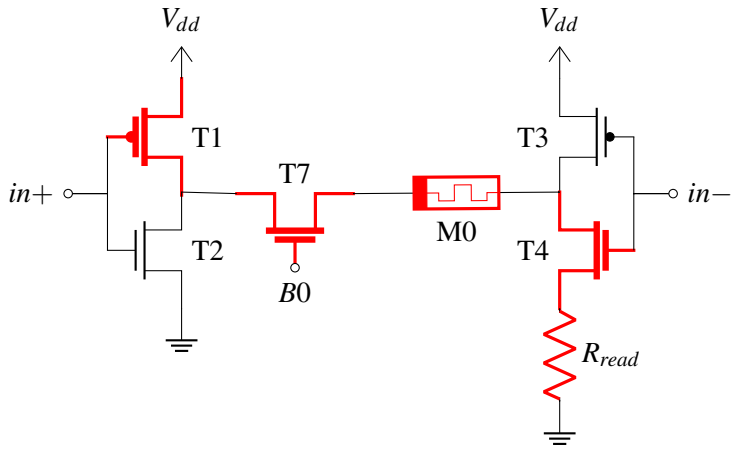
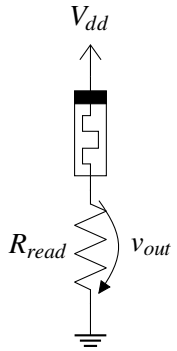


Figure 4.12: Transient simulation for testing the H bridge writing function.



(a) Schematic of the H brigde for reading the memristor state. Red components indicate current path.



(b) Simplified schematic of 4.13a.

From Figure 4.13b and R_{read} value obtained from Figure 4.3, the output of the voltage divider can be calculated in order to check how well the circuit is performing. For $R_{mem} = 99k\Omega$, $v_{out} \approx 0.2V$, according to the simulations in Figure 4.13.

$$V_{out} = \frac{R_{read}}{R_{read} + R_{mem}} V_{dd} \tag{4.12}$$

$$V_{out} = \frac{20k}{20k + 99k} 1.8 \approx 0.3V \tag{4.13}$$

The simulation value for V_{out} is really close to the calculated value, the difference comes to the fact that the transistor channel resistances were initially ignored to simplify calculations.

One thing important to notice is that, with each *Read* impulse, the memristor decreases its

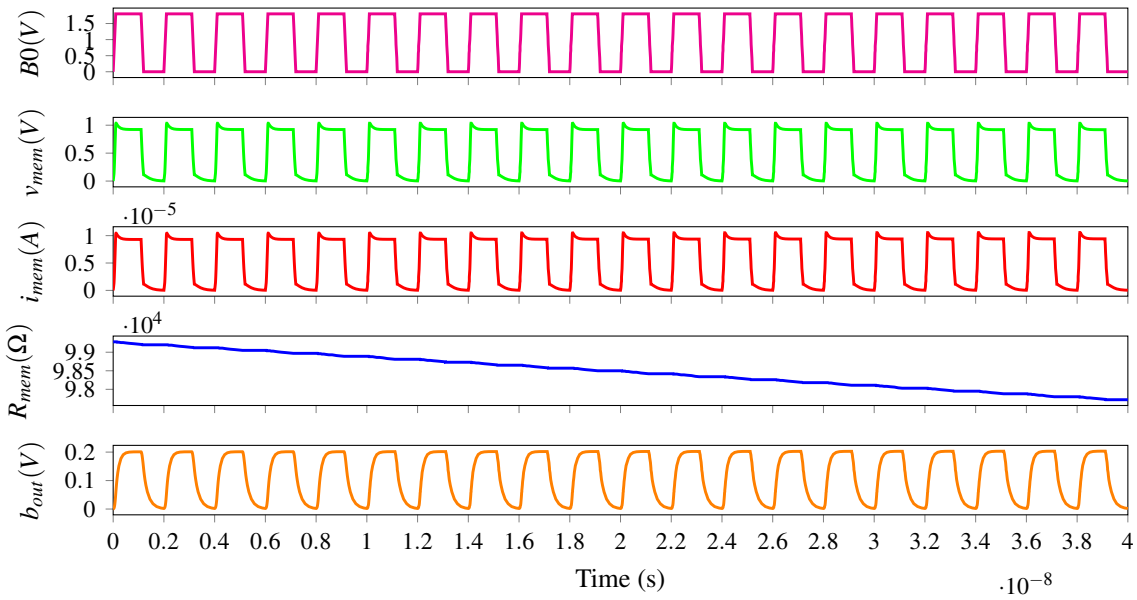


Figure 4.13: Simulation for the H bridge in read mode.

resistance due to being biased with a positive voltage. This problem only happens when the memristor is in high resistance state, the only situation where the memristor's resistance can decrease.

The simulations results showed that, after 3806 readings, the memristor state would not be reliable, meaning that reading its state might not output the correct value.

4.5.6.3 Sampling Stage

Sampling stage (Figure 4.14) is used to save the read memristor state. It is composed by an inverter, used for restoring the signal from the voltage divider, and a level triggered flip flip, to save the inverter's output.

In this case, only one inverter is used, meaning that when reading the memristor with state R_{off}, the output will be 1 and, for R_{on}, output will be 0, according the the voltage divider 4.12. This indicates which state to write the memristor in order to write the desired digital value.

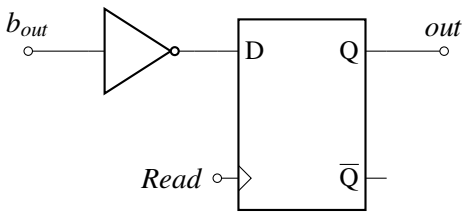


Figure 4.14: Sample stage.

A simulation was made to check the sampling stage behavior and the delay from *Read* positive edge to the *out*'s ascending/descending edge (Figure 4.16).

The output delays taken from the simulations can be found in Table 4.5.

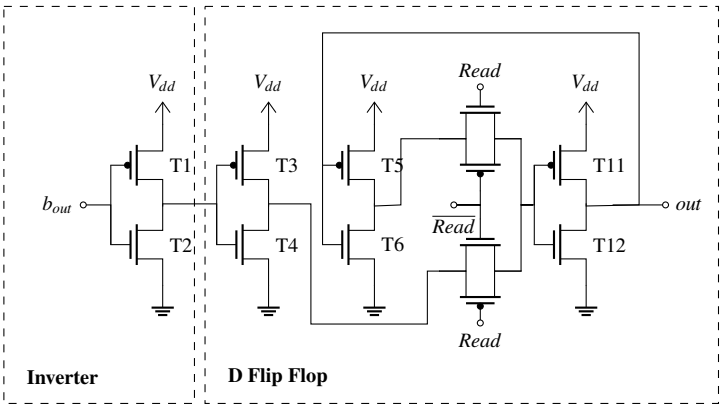


Figure 4.15: Sampling circuit.

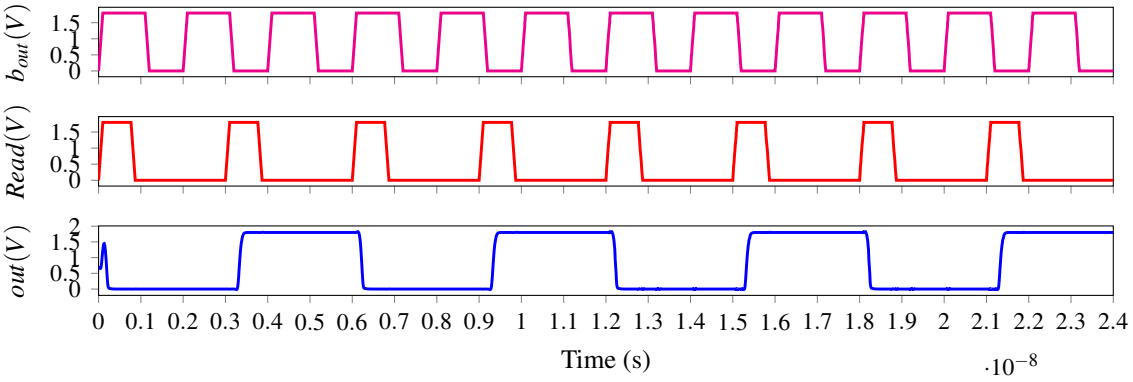


Figure 4.16: Simulation for the sample stage.

Read	out	delay
↑	↑	330ps
↑	↓	213ps

Table 4.5: Delay times for the sample stage.

4.5.6.4 Complete Circuit

To test the complete circuit, two simulation consisted on changing one memristor resistance to R_{on} (Figure 4.17) or R_{off} (Figure 4.18) and then reading it's state through a set of read impulses. With a complete circuit consisting in 2 bit LUT, simulations were made for an AND (Figure 4.19), an OR (Figure 4.20) and a XOR (Figure 4.21) gates which respective results can be found on Table 4.6, 4.7 and 4.8. A 5 bit LUT was also simulated with a complex gate. Its boolean equation is given by 4.14, which results in the logic circuit in Figure 4.22. The simulations results can be found in Figure 4.23 and the respective truth table results in Table 4.9.

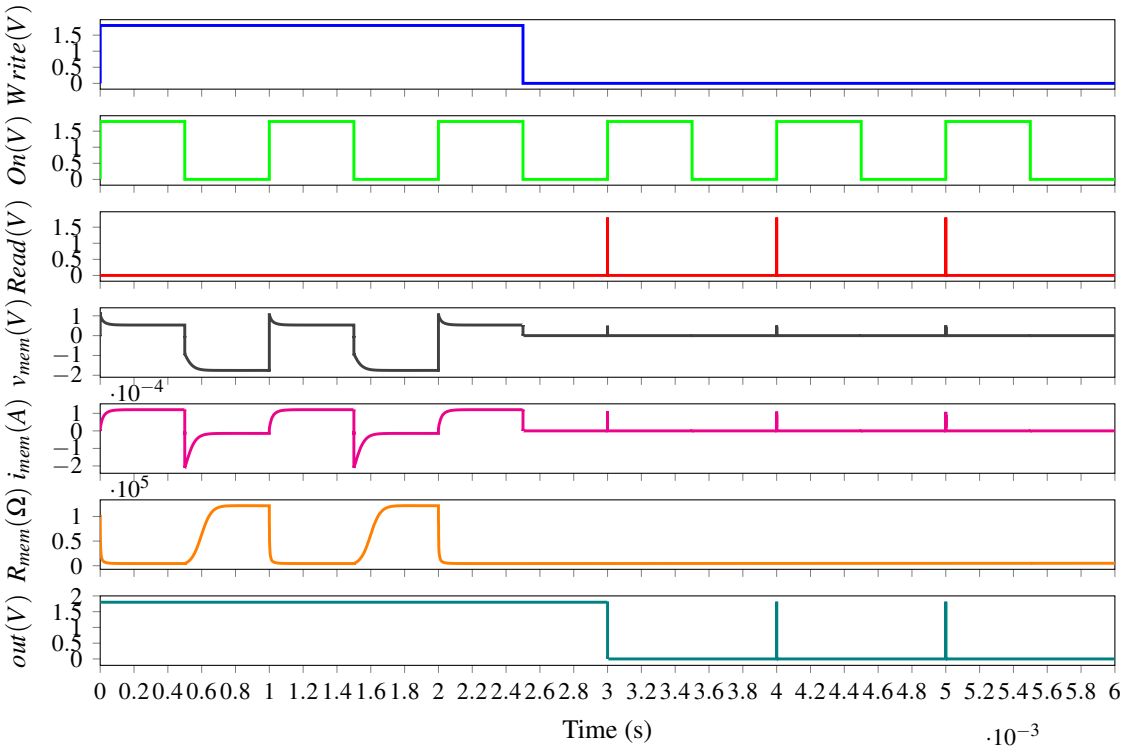


Figure 4.17: Simulation for writing R_{on} to the memristor and then reading its value.

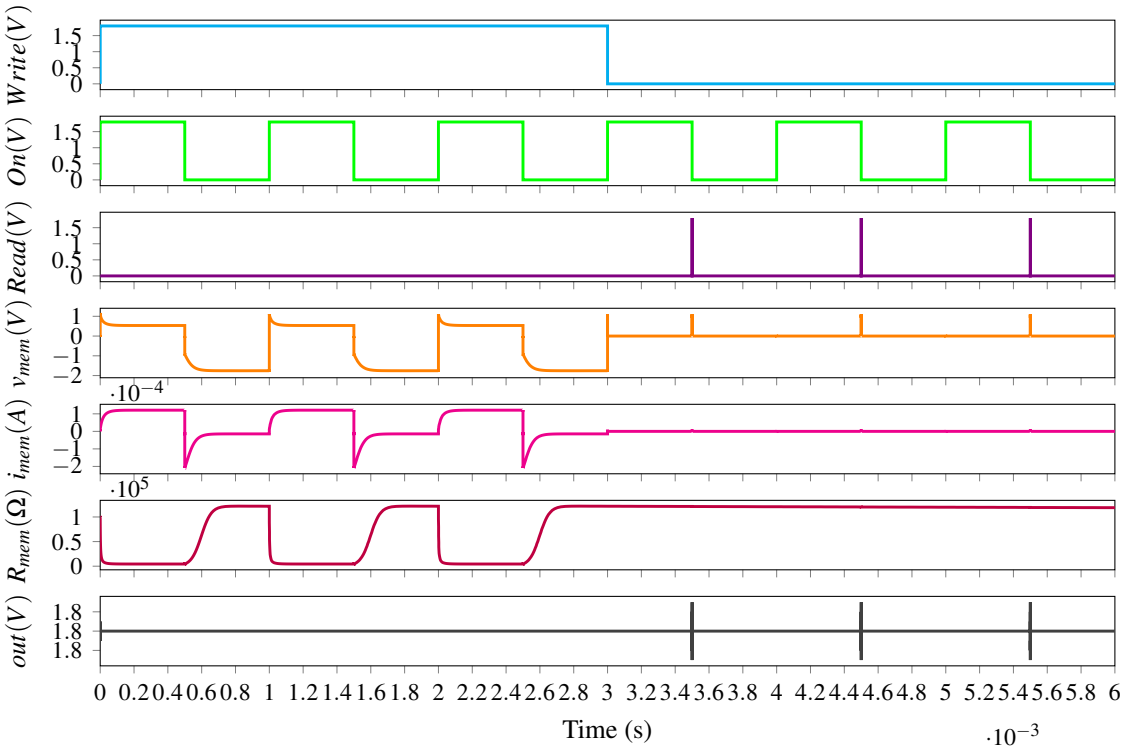


Figure 4.18: Simulation for writing R_{off} to the memristor and then reading its value.

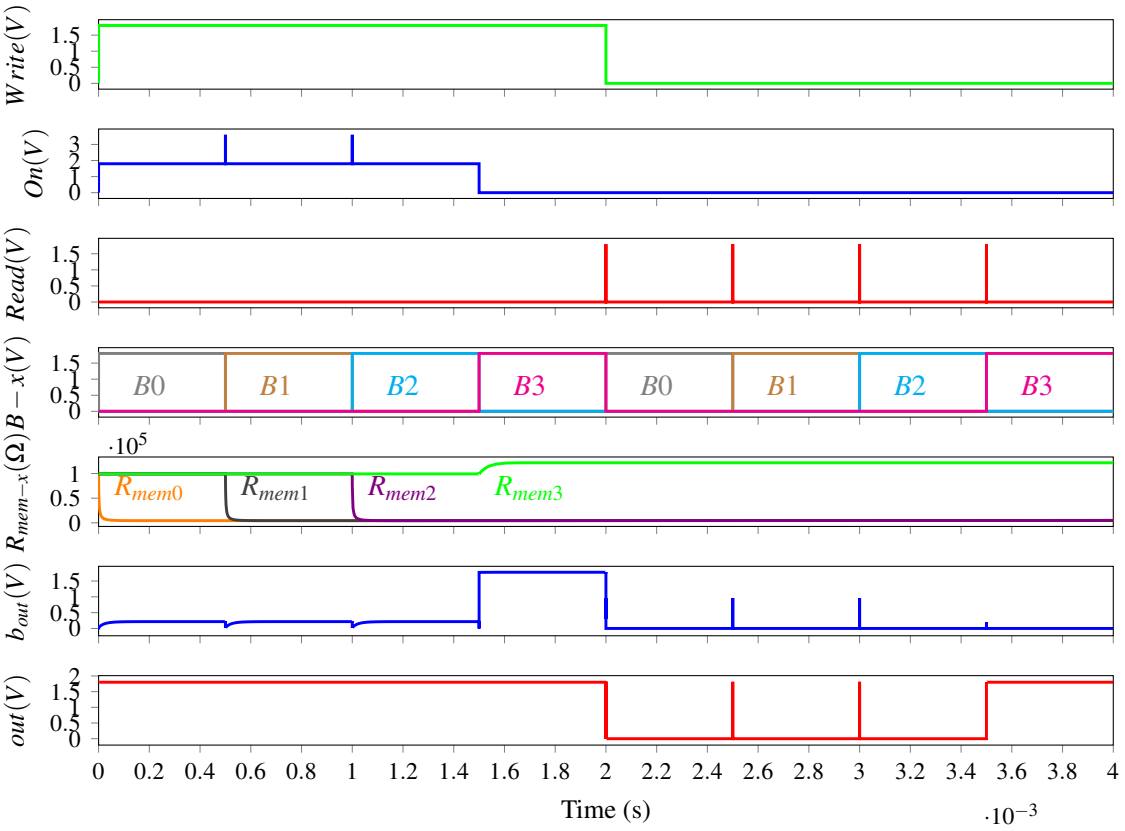


Figure 4.19: Transient simulation for write a 2 input AND gate and testing.

Gate Inputs		Memristor	Memristor	Expected	Simulation
A1	**A2**	**Select**	**State**	**Output**	**Output**
0	0	B0	R_{on}	0	0
0	1	B1	R_{on}	0	0
1	0	B2	R_{on}	0	0
1	1	B3	R_{off}	1	1

Table 4.6: AND gate truth table and simulation results.

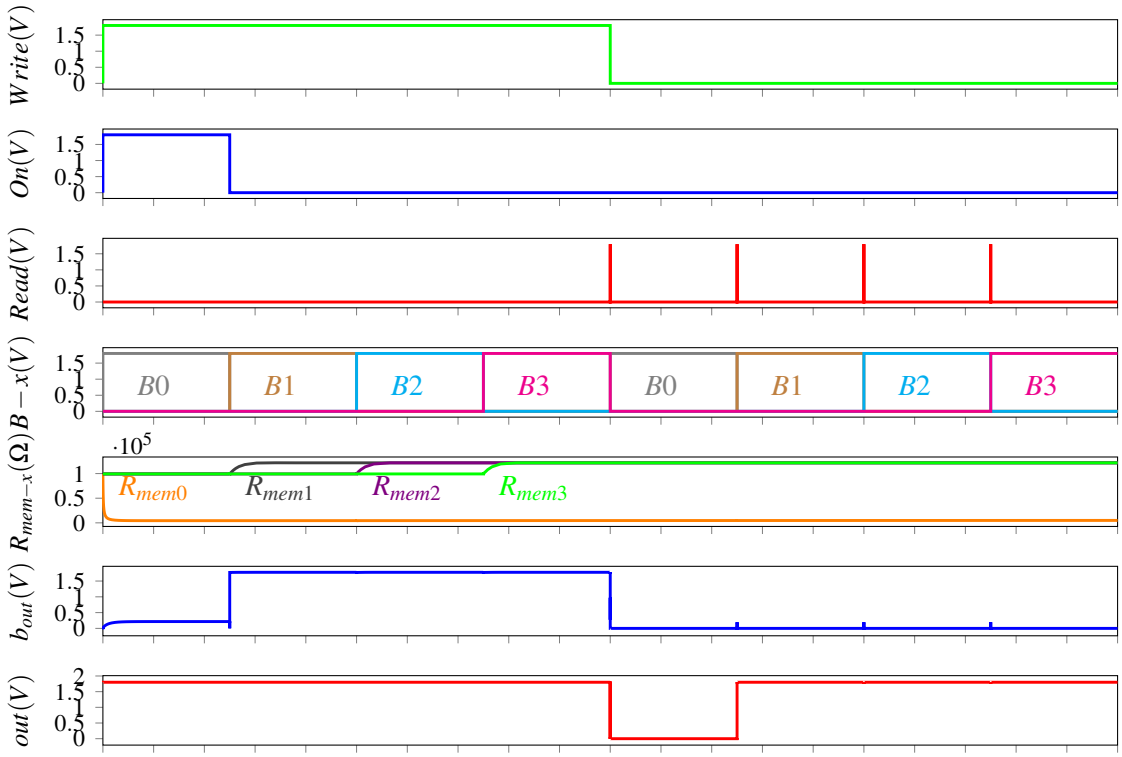


Figure 4.20: Transient simulation for write a 2 input OR gate and testing.

Gate Inputs		Memristor	Memristor	Expected	Simulation
A1	A2	Select	State	Output	Output
0	0	B0	R_{off}	0	0
0	1	B1	R_{off}	1	1
1	0	B2	R_{off}	1	1
1	1	B3	R_{on}	1	1

Table 4.7: OR gate truth table and simulation results.

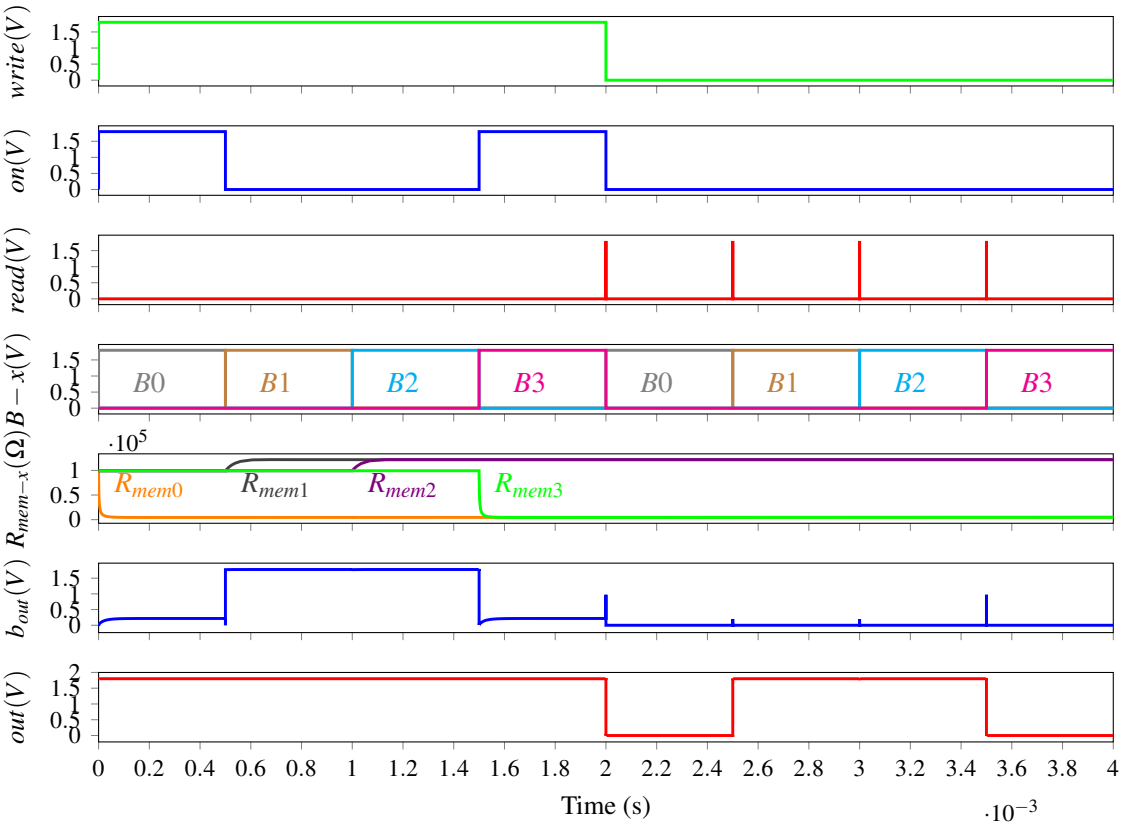


Figure 4.21: Transient simulation for write a 2 input XOR gate and testing.

Gate Inputs		Memristor	Memristor	Expected	Simulation
A1	A2	Select	State	Output	Output
0	0	B0	R_{on}	0	0
0	1	B1	R_{off}	1	1
1	0	B2	R_{off}	1	1
1	1	B3	R_{on}	0	0

Table 4.8: XOR gate truth table and simulation results.

$$Y = (A.\overline{B}) + (\overline{A}.\overline{C}) + (\overline{A}.D.\overline{E}) \tag{4.14}$$

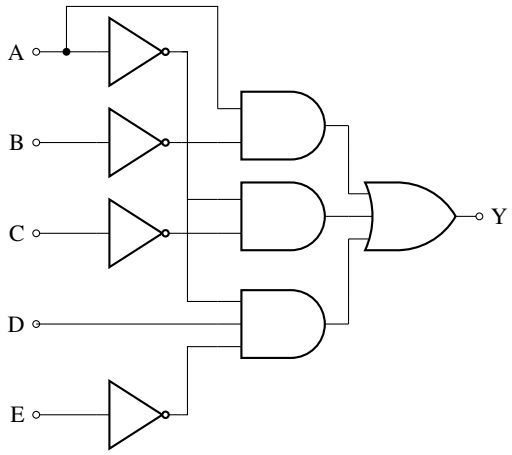


Figure 4.22: 5 input complex logic gate schematic to be implemented by the LUT.

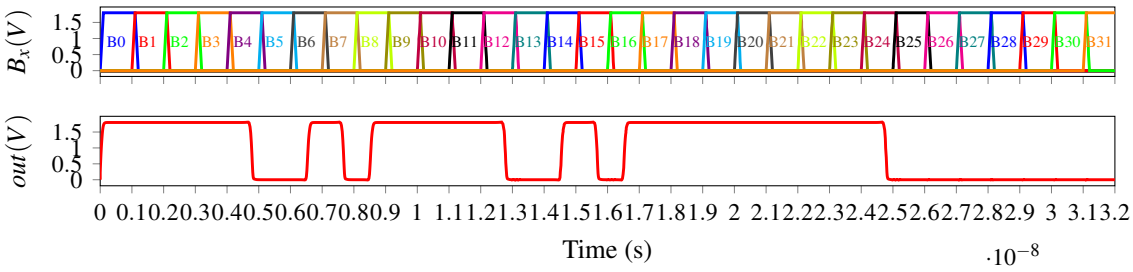


Figure 4.23: Transient simulation for write a 5 input complex gate and testing.

4.5.6.5 Read Speed

The delay times for reading in final circuit are shown in Table 4.10. The biggest delay will define the minimum reading time $t_{read-min}$ (Equation 4.15) and the respective maximum reading frequency (Equation 4.16).

$$t_{read-min} = 527.21ps \tag{4.15}$$

$$f_{read-max} = \frac{1}{t_{read-min}} \approx 1.89GHz \tag{4.16}$$

Read	out	delay
↑	↑	$319ps$
↑	↓	$527.21ps$

Table 4.10: Read delay time for the complete circuit.

Gate Input					Memristor Select	Expected Output	Memristor State	Simulation Output
A	**B**	**C**	**D**	**E**				
0	0	0	0	0	B0	1	R_{off}	1
0	0	0	0	1	B1	1	R_{off}	1
0	0	0	1	0	B2	1	R_{off}	1
0	0	0	1	1	B3	1	R_{off}	1
0	0	1	0	0	B4	0	R_{on}	0
0	0	1	0	1	B5	0	R_{on}	0
0	0	1	1	0	B6	1	R_{off}	1
0	0	1	1	1	B7	0	R_{on}	0
0	1	0	0	0	B8	1	R_{off}	1
0	1	0	0	1	B9	1	R_{off}	1
0	1	0	1	0	B10	1	R_{off}	1
0	1	0	1	1	B11	1	R_{off}	1
0	1	1	0	0	B12	0	R_{on}	0
0	1	1	0	1	B13	0	R_{on}	0
0	1	1	1	0	B14	1	R_{off}	1
0	1	1	1	1	B15	0	R_{on}	0
1	0	0	0	0	B16	1	R_{off}	1
1	0	0	0	1	B17	1	R_{off}	1
1	0	0	1	0	B18	1	R_{off}	1
1	0	0	1	1	B19	1	R_{off}	1
1	0	1	0	0	B20	1	R_{off}	1
1	0	1	0	1	B21	1	R_{off}	1
1	0	1	1	0	B22	1	R_{off}	1
1	0	1	1	1	B23	1	R_{off}	1
1	1	0	0	0	B24	0	R_{on}	0
1	1	0	0	1	B25	0	R_{on}	0
1	1	0	1	0	B26	0	R_{on}	0
1	1	0	1	1	B27	0	R_{on}	0
1	1	1	0	0	B28	0	R_{on}	0
1	1	1	0	1	B29	0	R_{on}	0
1	1	1	1	0	B30	0	R_{on}	0
1	1	1	1	1	B31	0	R_{on}	0

Table 4.9: 5 input complex gate truth table and simulation results.

4.5.6.6 Energy

The energy spent on the circuit was obtained for both read and write operations, as well as for only the CMOS, only the memristors and the total (Table 4.11).

		Energy		
		Memristor	**CMOS**	**Total**
Write	$R_{on} \rightarrow R_{off}$	$23.28nJ$	$2.28nJ$	$25.56nJ$
	$R_{off} \rightarrow R_{on}$	$31.55nJ$	$75.15nJ$	$106.7nJ$
Read	R_{on}	$14.57fJ$	$288.1fJ$	$302.67fJ$
	R_{off}	$10.16fJ$	$97.58fJ$	$107.74fJ$

Table 4.11: Table with circuit energy.

4.5.7 Conclusions

The proposed LUT transistor count is shown in Table 4.12 for different LUT sizes, as well as for the previous circuits explored in this Chapter and for a conventional LUT.

Taking for example a 5 bit LUT from the Spartan 6 FPGA configurable Logic Block Guide [54], the proposed LUT, without any control circuit just like Pino's and Wang's circuits, has a reduction of 62 transistors (62%) compared to Pino's circuit and 30 transistors (44%) compared to Wang's circuit. The proposed LUT with control circuit has a reduction of 142 transistors (74%) compared to a conventional LUT. In this terms, the proposed circuit takes the biggest advantage when it comes to the number of transistors and, consequently, the area overhead. Furthermore, the proposed LUT works within the expectations, taking advantage o the hybrid technology to create an optimal circuit. The use of memristor gives this LUT a non volatile property, in which the LUT data is maintained after powering off, unlike conventional LUTs.

The main problem with the proposed circuit, as well as Pino's and Wang's proposals, is that, with each memristor reading, its state will change, even if it is a very small. The assumption that the memristor only changes its state when it reaches a threshold voltage is a crude assumption and should be taken for analysis simplification. After a several number of readings, the memristor will be in the opposite state, which means the data in that bit was lost and the circuit will not function as expected. This situation only happens when the memristor is in the R_{off} state, because the reading causes a positive voltage bias that translates into a state change going for R_{on}.

This problem could be solved by adding a more complex control circuit to the proposed LUT in order to read the memristor state, save it in the D flip flop and, according to the read value, apply a signal that will restore the memristor's previous state.

Bits	**Memristors**	**Number of transistors**				
		Conventional LUT (6M)	**Pino's LUT (4 + 3M)**	**Wang's LUT (4 + 2M)**	**Proposed LUT w/o control (6+M)**	**Proposed LUT w/ control and output stage (40+M)**
1	2	12	10	8	8	20
2	4	24	16	12	10	22
3	8	48	28	20	14	26
4	16	96	52	36	22	34
5	32	192	100	68	38	50
6	64	384	196	132	70	82
7	128	768	388	260	134	146
8	256	1536	772	516	262	274

Table 4.12: Comparison of diferent LUT topologies.

Chapter 5

PCB Design and Experimental testing

In this Chapter, a PCB with LUT functionality is designed. It is presented simulations for its behavior and respective experimental results.

5.1 The Circuit

To test the functionality of the proposed circuit in Chapter 4, a PCB was designed (Figure 5.1) with *EasyEDA* software, to ease the ordering process from *JLCPCB*. Whole circuit without the control logic, because this logic could be simulated by the PCB controller. Although the circuit was implemented with sampling stage, this would not work due to an project error in the flip flop, which is discussed in conclusions.

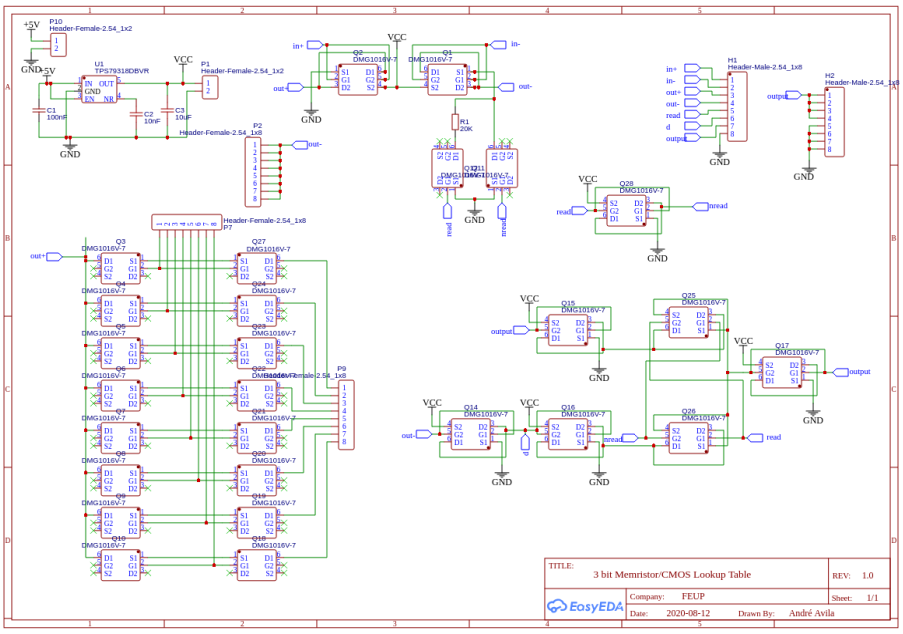
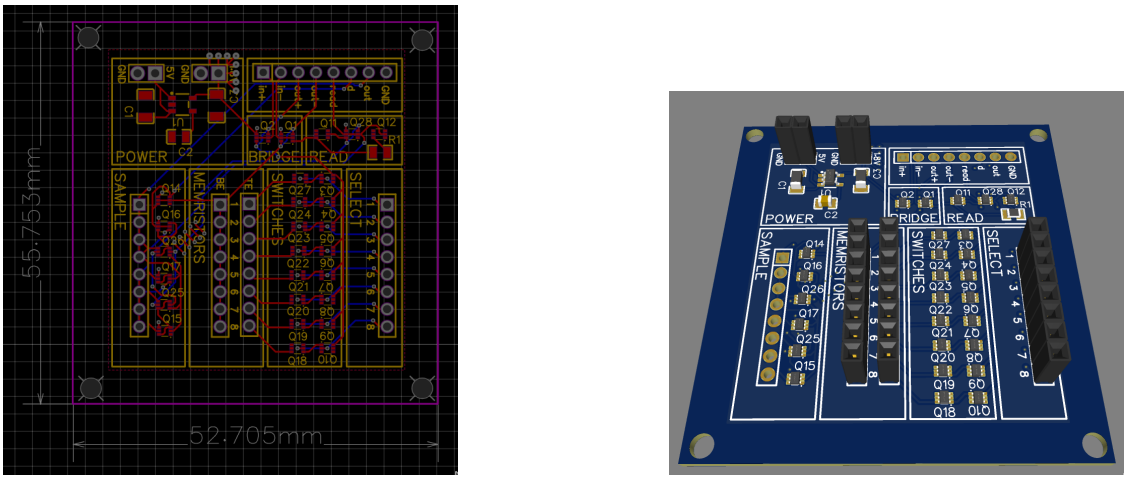


Figure 5.1: PCB circuit schematic.



(a) PCB design result.

(b) 3D model of the PCB.

Figure 5.2: PCB circuit schematic.

5.2 Voltage Regulator

The voltage regulator used to obtain the operating the LUT's 1.8V was a *TPS79318DBVR* from *Texas Instruments*. It was chosen due to its applications in fast RF and its high stability, which complies with the need for operating the LUT with fast signals. The schematic for this voltage regulator was taken from its datasheet, where there are added 3 capacitors: one for filtering the input, one for filtering the output and another to stabilize the regulator.

5.3 Discrete Complementary Transistors

The transistors model used for this porpose was DMG1016V-7 from *Diodes Incorporated*. Each IC contains a complementary pair NMOS and PMOS, both of them having a threshold voltage below 1V, which is good for simulating a smaller scale CMOS circuit and to operate at 1.8V. Besides that, these transistors have fast switching speeds (maximum of 26.7ns).

Because these transistors have their bulk connected to the source, an antiparallel diode is formed between the Drain and Source. This is problematic when selecting the memristors, in which the polarity changes according to the operation applied, because the transistors corresponding to the unselected memristors will only cutoff in one direction, the reverse direction will cause a current flow through the body diode. To solve this, another transistor is added in series to each selecting transistor in order to get two body diodes in series with oposite polarities (Figure 5.3). This ensures that when a diode is forward biased, the other diode is reverse biased, resulting in a complete cutoff.

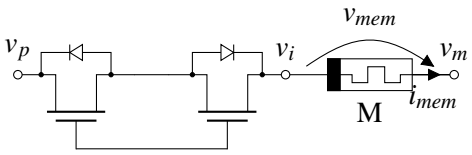


Figure 5.3: Memory cell for PCB design.

5.4 Simulations

5.4.1 Write

To simulate this circuit in write mode, B and write where pulled to 1 and read pulled to 0. The *on* pin signal consisted on a square voltage with a $1kHz$ frequency.

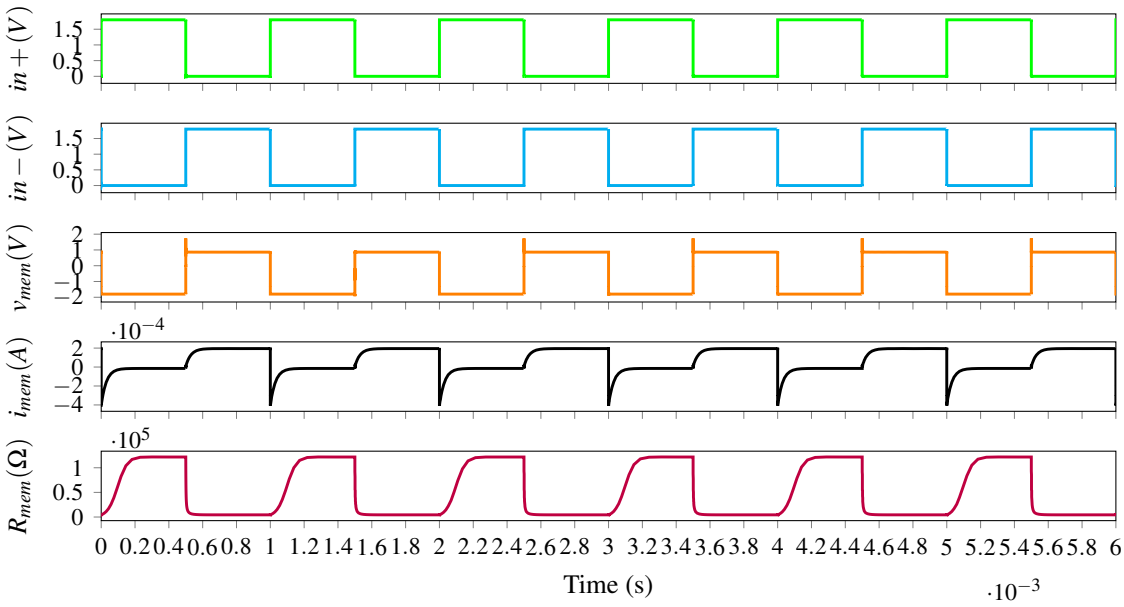


Figure 5.4: Simulation for writing memristor.

From the simulation results (Figure 5.4), it is possible to observe the memristor changing its resistance state according to the on signal, just as expected.

5.4.2 Read

To simulate the read mode, the memristor state was changed to R_{off} and then read and was changed to R_{on} and then read (Figure 5.5). As the PCB sample stage is not working, the output will be in out-, the output of the voltage divider. To read the memristor state, a pulse with $500kHz$ ($2\mu s$).

5.5 Experimental Results

In order to test the LUT PCB, control signals must be applied. It was achieved by programming the microcontroller ESP32, where all the signals were generated. This microcontroller pins outputs

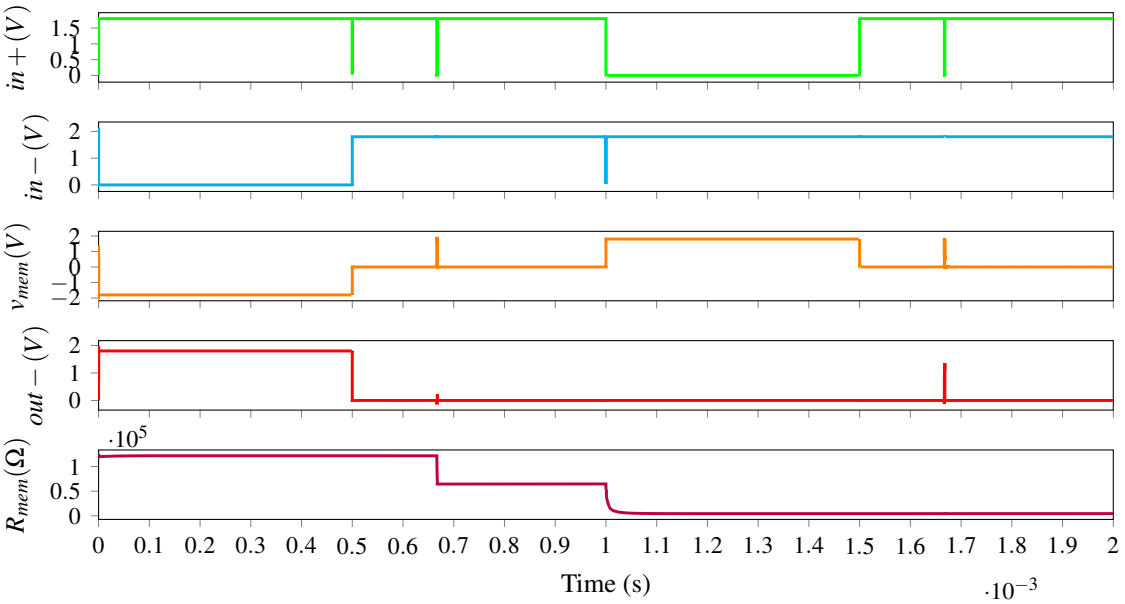


Figure 5.5: Simulation for writing R_{off} and reading its value, writing R_{on} and reading. For R_{off}, the output voltage is $1.353V$ and for R_{on}, the output voltage is $276mV$.

3.3V and were directly connected to the PCB pins.

The experimental tests were done only for one memristor.

5.5.1 Write

For testing the writing mode, the ESP32 was programmed to generate a 25kHz square wave, where in pin $in+$ there was no phase shift and in pin $in-$ there was a 180° phase shift. The writing frequency is much higher than in 4.5.6.1 due to the memristor being faster than what the model described. So, in order to observe the state transitioning, the writing frequency was increased.The read signal was pulled low and the write and bit select B pulled high.

The oscilloscope measurements were done by placing (from Figure 5.3) the common ground in v_i, Channel 1 probe in v_m - to obtain memristor voltage, and Channel 2 probe in v_p - to obtain the memristor current, assuming a constant transistor channel resistance (Figure 5.6).

The results obtained from Figure 5.6 are really close to the curves of v_{mem} and i_{mem} from Figure 5.4, which indicates the memristor is changing its state according to the voltage applied.

5.5.2 Read

For reading mode, the common ground was set on the PCB's ground, the Channel 1 to $out+$ and Channel 2 to $out-$. The measurement of these two nodes are used to see the voltage polarity, as well as what value was given by the voltage divider (Figure 5.7).

The first peak voltage, corresponding to the reading of a memristor in R_{off} state, is approximately $0,76V$ and the second peak, corresponding to the reading of a memristor in R_{off} state, is

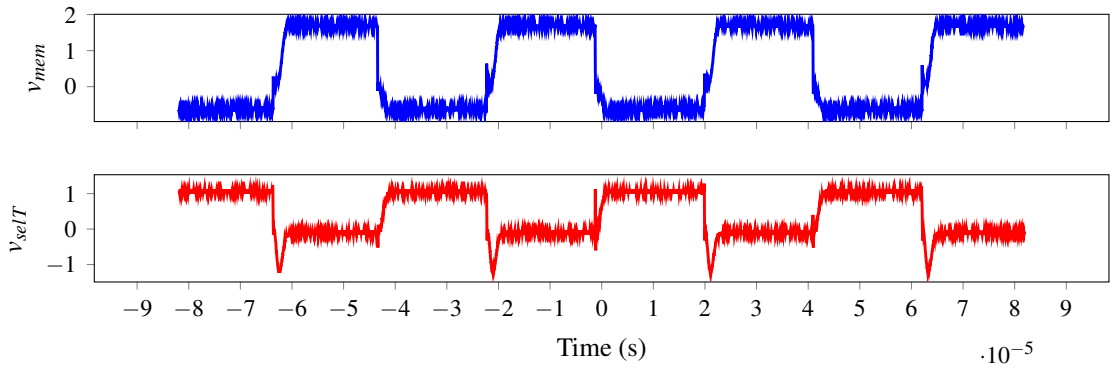


Figure 5.6: Experimental results for write mode.

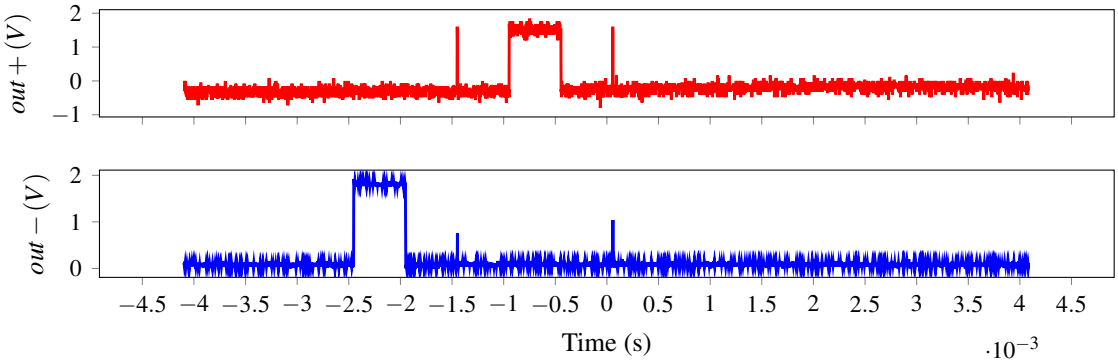


Figure 5.7: Experimental results for reading mode.

approximately 1.04. According to the voltage divider, the memristor resistance values are given by:

$$0.76 = \frac{R_{read}}{R_{read} + R_{off}} 1.8 \tag{5.1}$$

$$1.04 = \frac{R_{read}}{R_{read} + R_{on}} 1.8 \tag{5.2}$$

$$R_{off} \approx 27.4k\Omega \tag{5.3}$$

$$R_{on} \approx 14.6k\Omega \tag{5.4}$$

Both resistance values are too close to each other, meaning the memristor didn't have enough time to change it's state. This results due to the transistors having high channel resistance, which will affect how fast the memristors will change state, as well as the memristor state reading.

5.6 Conclusions

The PCB project did not go as expected. Due to the current pandemic from COVID-19, the PCB only arrived 2 weeks before this dissertation delivery, as it was needed to solder everything in place and then do the testing. In the testing phase it occurred a problem in the sampling stage: like the situation explained in 5.3, the transmission gates from the flip flop conduct whenever there is a positive bias voltage for its body diode. This created a short circuit from V_{dd} to ground, making the regulator's voltage drop drastically and making its temperature rise abruptly. In the end, the flip flop circuitry would take off the whole circuit's functionality. The solution was to do all the testing without the sampling stage.

The write mode results where positive, as it could be seen similarities from the simulations.

The reading mode mode got neutral results, meaning that a final conclusion cannot be taken due to both calculated resistance being relatively close to each other. With more time, the sampling stage problem could be resolved and the PCB circuit could be tested work as a full LUT.

Chapter 6

Conclusions

6.1 Work Done

The aim of this dissertation was to achieve an optimal circuit using hybrid CMOS/memristor technology. This would be introduced in the first Chapter with a general explanation of both technologies, the problems that have arisen with the CMOS evolution and how the memristor properties would be interesting to achieve optimal and better mixed circuits.

Chapter 2 creates an understanding of the mathematical concept of the memristor is given, in order to comprehend how this complex device relates to some simple mathematical formulas. Both Chua's and HP Labs memristors are presented, as they were the main marks in the memristors short history. A brief explanation of hybrid technology is presented and how it is physically implemented

In Chapter 3, some of the most known memristor Spice models were simulated to study their behavior and physical approach to the mathematical models. All simulations were done in *LT Spice IV*. The last studied model was the Knowm's model to which a model fitting was done using experimental data. This data was tobtained from commercially avaible memristor offered by Knowm. A problem was detected in this last models, so a solution was proposed. This final modified model is used in the next Chapters.

Afterwards, in Chapter 4, an introduction is made to the LUTs, which are fundamental reconfigurable logic blocks in FPGAs. A simplified approach to the memristor I-V curve behaviour was done to get a better comprehension on the digital side of the memristor. Then, two similar LUT topologies were studied in order to try achieving a better circuit. A circuit was proposed, achieving better results in terms of area overhead. The proposed circuit is explained piece by piece with complementar simulations for a full understanding of the circuit working principle. A study of the used energy and speed is done to describe the circuit for future comparisons.

Finally, in Chapter 5, a PCB was designed with the circuit proposed in Chapter 4. The circuit was simulated and compared to the experimental results. A problem with the circuit made it not work completely and, due to the pandemic, little time was left to test the PCB and to take better solving actions.

6.2 Final Conclusions

The deep study done in this dissertation for the memristor concept, its story and working principles aimed to achieve optimal applications for this device, allied with CMOS. The memristor capability of saving its state was explored for digital application, more specifically reconfigurable logic. With a literature review of hybrid circuits for this applications, a better circuit was proposed and simulated, proofing that hybrid CMOS/Memristor technology is reliable and presents excellent advantages when it comes to all the limitations explored at the beginning of this dissertation. The proposed circuits proofs to be a great starting point for hybrid technology implementation.

With all the obstacles the current pandemic brought during the course of this dissertation, from the impossibility to access a laboratory to all the delays, the main goals of this dissertation were positively accomplished. Self dedication and proactivity were the main keys to counter this obstacles, by finding was to work from home, with the help of Knowm and *Keithley* who provided material for experimental testing, or to work with the right software tools, with the help of the advisors.

6.3 Future Work

As for this dissertation work, a set of tasks are proposed for future work:

- For the proposed circuit, further work can be done by implementing additional logic to make the memristor retrieve its previous state after being read, because for each reading the memristor changes its state;
- For the proposed circuit, a layout can be done in order to get a final working chip to be implemented in FPGAs with hybrid technology;
- For the projected PCB, a new PCB can be designed to solve the sample stage problem and improve the discrete transistor to a more suitable model;
- After creating a totally functional LUT PCB, a physical memristor crossbar can be simulated to get closer to what would be a chip implementation.

Appendix A

Appendix

A.1 Spice Code

Listing A.1: HP Labs Spice model with Joglekar modification

```
* HP Memristor SPICE Model

* Connections:
* TE:   Top electrode
* BE:   Bottom electrode
* W:    Width of TiO2

.SUBCKT memristor_hp TE BE W

* Ron:    Minimum resistance
* Roff:   Maximum resistance
* D:      Width of memristor thin film
* uv:     Dopant mobility
* w0:     Initial width value

.params Ron=100 Roff=1K w0=5N D=10N uv=50F

* Memristor I-V Relationship
.func IVRel(V1,V2) = V1/(Ron*V2/D + Roff*(1-V2/D))

* Circuit to determine state variable
Gx 0 W value={I(Gmem)*Ron*uv/D}
Cx W 0 {1}
.ic V(W) = w0
```

```
* Current source representing memristor
Gmem TE BE value={IVRel(V(TE,BE),V(XSV,0))}

.ENDS memristor_hp
```

Listing A.2: HP Labs Spice model with Joglekar modification

```
* HP Memristor SPICE Model Using Joglekar Window

* Connections:
* TE:  Top electrode
* BE:  Bottom electrode
* XSV: External connection to plot state variable
*      that is not used otherwise

.SUBCKT memristor_joglekar TE BE XSV

* Ron: Minimum device resistance
* Roff: Maximum device resistance
* D: Width of the thin film
* uv: Dopant mobility
* p: Parameter for window function
* x0: State variable initial value

.params Ron=100 Roff=10K x0=.5 D=10N uv=50F p=1


* Joglekar Window Function
.func f(V1) = 1-pow((2*V1-1),(2*p))

* Memristor I-V Relationship
.func IVRel(V1,V2) = V1/(Ron*V2 + Roff*(1-V2))

* Circuit to determine state variable
Gx 0 XSV value={ I(Gmem)*Ron*uv*f(V(XSV,0))/pow(D,2) }
Cx XSV 0 {1}
.ic V(XSV) = x0

* Current source representing memristor
```

```
Gmem TE BE value={IVRel(V(TE,BE),V(XSV,0))}

.ENDS memristor_joglekar
```

Listing A.3: HP Labs Spice model with Biolek modification

```
* HP Memristor SPICE Model Using Biolek Window

* Connections:
* TE:  Top electrode
* BE:  Bottom electrode
* XSV: External connection to plot state variable
*      that is not used otherwise

.SUBCKT memristor_biolek TE BE XSV

.params Ron=100 Roff=1K x0=.076 D=16N uv=40F p=7

* Biolek Window Function
.func f(V1,I1)={1-pow((V1-stp(-I1)),(2*p))}

* Memristor I-V Relationship
.func IVRel(V1,V2) = V1/(Ron*V2 + Roff*(1-V2))

* Circuit to determine state variable
Gx 0 XSV value={I(Gmem)*Ron*uv*f(V(XSV,0),I(Gmem))/pow(D,2)}
Cx XSV 0 {1}
.ic V(XSV) = x0

* Current source representing memristor
Gmem TE BE value={IVRel(V(TE,BE),V(XSV,0))}

.ENDS memristor_biolek
```

Listing A.4: General Hyperbolic Sine model

```
* Memristor Model proposed by Lahio et al with Biolek window for
    boundaries

* Connections:
* TE: Top electrode
* BE: Bottom electrode
* XSV: External connection to plot state variable that is not
   used otherwise

.SUBCKT memristor_hyperbolic TE BE XSV

.param a1=4e-8 b1=1.2 a2=1.25e-7 b2=1.2 c1=6e-4 d1=2 c2=6.6e-4
   d2=3.8 x0=0.001 p=7

* Hyperbolic sine IV relationship
.func IVRel(V1,V2) = IF(V1 >= 0, a1*V2*sinh(b1*V1), a2*V2*sinh(
   b2*V1))

* Equation for state variable
.func SV(V1) = IF(V1 >= 0, c1*sinh(d1*V1), c2*sinh(d2*V1))

* Biolek Window Function
.func f(V1,I1)={1-pow((V1-stp(-I1)),(2*p))}

* Current source representing memristor
Gmem TE BE value = {IVRel(V(TE,BE),V(XSV,0))}

* Circuit to determine value of state variable
Gxsv 0 XSV value = {SV(V(TE,BE))*f(V(XSV,0),I(Gmem))}
Cx XSV 0 {1}
.ic V(XSV) = x0

.ENDS memristor_hyperbolic
```

Listing A.5: University of Michigan model

```
* Memristor subcircuit developed by Chang et al.

* Connections:
```

```
* TE:  Top electrode
* BE:  Bottom electrode
* XSV: External connection to plot state variable
*      that is not used otherwise

.SUBCKT memristor_umich TE BE XSV

* Parameters:
* alpha:       Prefactor for Schottky barrier
* beta:        Exponent for Schottky barrier
* gamma:       Prefactor for tunneling
* delta:       Exponent for tunneling
* xmax:        Maximum value of state variable
* xmin:        Minimum value of state variable
* drift_bit:   Binary value to switch the ionic drift in (1)
*              or out (0) of the equation
* lambda:      State variable multiplier
* eta1, eta2:  State variable exponential rates
* tau:         Diffusion coefficient

.param alpha=0.5e-6 beta=0.5 gamma=4e-6 delta=2 xmax=1 xmin=0
+drift_bit = 0 lambda=4.5 eta1=0.004 eta2=4 tau=10

.param cp={1}
Cpvar XSV 0 {cp}

* Rate equation for state variable
Gx 0 XSV value={ trunc(V(TE,BE),cp*V(XSV))*lambda*(eta1*sinh(
    eta2*V(TE,BE))-
+drift_bit*cp*V(XSV)/tau) }

.ic V(XSV) = 0.0

* Auxiliary functions to limit the range of x
.func sign2(var) {(sgn(var)+1)/2}
.func trunc(var1,var2) {sign2(var1)*sign2(xmax-var2)+sign2(-
+var1)*sign2(var2-xmin)}

* Memristor IV Relationship
Gm TE BE value={(1-cp*V(XSV))*alpha*(1-exp(-
```

```
+beta*V(TE,BE)))+(cp*V(XSV))*gamma*sinh(delta*V(TE,BE))}

.ENDS memristor_umich
```

Listing A.6: Yakopcic's model

```
* SPICE model for memristive devices
* Created by Chris Yakopcic
* Last Update: 8/9/2011
*
* Connections:
* TE - top electrode
* BE - bottom electrode
* XSV - External connection to plot state variable
* that is not used otherwise

.subckt memristor_yakopcic TE BE XSV

* Fitting parameters to model different devices
* a1, a2, b:       Parameters for IV relationship
* Vp, Vn:          Pos. and neg. voltage thresholds
* Ap, An:          Multiplier for SV motion intensity
* xp, xn:          Points where SV motion is reduced
* alphap, alphan: Rate at which SV motion decays
* xo:              Initial value of SV
* eta:             SV direction relative to voltage

*.params a1=0.17 a2=0.17 b=0.05 Vp=0.16 Vn=0.15
*+Ap=4000 An=4000 xp=0.3 xn=0.5 alphap=1 alphan=5
*+xo=0.11 eta=1

.params a1=.17 a2=.17 b=0.05 Vp=.65 Vn=0.56 Ap=4000
+An=4000 xp=0.3 xn=0.5 alphap=1 alphan=5 xo=0.11 eta=1

* Multiplicitive functions to ensure zero state
* variable motion at memristor boundaries
.func wp(V) = xp/(1-xp) - V/(1-xp) + 1
.func wn(V) = V/(1-xn)
```

```
* Function G(V(t)) - Describes the device threshold
.func G(V) = IF(V <= Vp, IF(V >= -Vn, 0, -An*(exp(-
+V)-exp(Vn))), Ap*(exp(V)-exp(Vp)))

* Function F(V(t),x(t)) - Describes the SV motion
.func F(V1,V2) = IF(eta*V1 >= 0, IF(V2 >= xp, exp(-
+alphap*(V2-xp))*wp(V2) ,1), IF(V2 <= (1-xn),
+exp(alphan*(V2+xn-1))*wn(V2) ,1))

* IV Response - Hyperbolic sine due to MIM structure
.func IVRel(V1,V2) = IF(V1 >= 0, a1*V2*sinh(b*V1),
+a2*V2*sinh(b*V1) )

* Circuit to determine state variable
* dx/dt = F(V(t),x(t))*G(V(t))
Cx XSV 0 {1}
.ic V(XSV) = xo
Gx 0 XSV value={eta*F(V(TE,BE),V(XSV,0))*G(V(TE,BE))}
* Current source for memristor IV response
Gm TE BE value = {IVRel(V(TE,BE),V(XSV,0))}

.ends memristor_yakopcic
```

Listing A.7: Knowm Mean Metastable Switch model

```
* Knowm Mean Metastable Switch Memristor SPICE Model

* Copyright Tim Molter Knowm Inc. 2017

* Connections:
* TE:  Top electrode
* BE:  Bottom electrode
* XSV: External connection to plot state variable
*      that is not used otherwise

.SUBCKT memristor_knowm TE BE XSV

* Ron:  Minimum device resistance
* Roff: Maximum device resistance
```

```
* Von:  Threshold voltage to turn device on
* Voff: Threshold voltage to turn device off
* TAU:  Time constant
* T:    Temperature

.params Ron=10k Roff=100k Voff=0.3 Von=0.3 TAU=0.0001 T=298.15
    x0=0

* Function G(V(t)) - Describes the device threshold
.func G(V) = V/Ron+(1-V)/Roff

* Function F(V(t),x(t)) - Describes the SV motion
.func F(V1,V2) = (1/TAU)*((1/(1+exp(-1/(T*boltz/echarge)*(V1-Von
    ))))*(1-V2)-(1-(1/(1+exp(-1/(T*boltz/echarge)*(V1+Voff)))))*
    V2

* Memristor I-V Relationship
.func IVRel(V1,V2) = V1*G(V2)

* Circuit to determine state variable
* dx/dt = F(V(t),x(t))*G(V(t))
Cx XSV 0 {1}
.ic V(XSV) = x0
Gx 0 XSV value={F(V(TE,BE),V(XSV,0))}

* Current source for memristor IV response
Gmem TE BE value={IVRel(V(TE,BE),V(XSV,0))}

.ENDS memristor_knowm
```

Listing A.8: list:my-model

```
* Knowm Mean Metastable Switch Memristor SPICE Model Modified

* Copyright Andre Silva @ FEUP 2020

* Connections:
* TE:  Top electrode
* BE:  Bottom electrode
```

```
* XSV: External connection to plot state variable
*      that is not used otherwise

.SUBCKT memristor_knowm TE BE XSV

* Ron:  Minimum device resistance
* Roff: Maximum device resistance
* Von:  Threshold voltage to turn device on
* Voff: Threshold voltage to turn device off
* TAU:  Time constant
* T:    Temperature

.params Ron=4419 Roff=121830 Voff=0.1777 Von=0.3 TAU=0.00003 T
   =298.5 x0=0

* Function G(V(t)) - Describes the device threshold
.func G(V) = V/Ron+(1-V)/Roff



* Function F(V(t),x(t)) - Describes the SV motion
.func F(V1,V2) = IF(ABS(I(Gmem)) > 0, (1/TAU)*((1/(1+exp((-
   echarge/(T*boltz))*(V1-Von))))*(1-V2)-(1-(1/(1+exp((-echarge
   /(T*boltz))*(V1+Voff)))))*V2), 0)

* Memristor I-V Relationship
.func IVRel(V1,V2) = V1*G(V2)

* Circuit to determine state variable
* dx/dt = F(V(t),x(t))*G(V(t))
Cx XSV 0 {1}
.ic V(XSV) = x0
Gx 0 XSV value={F(V(TE,BE),V(XSV,0))}

* Current source for memristor IV response
Gmem TE BE value={IVRel(V(TE,BE),V(XSV,0))}

.ENDS memristor_knowm
```

References

[1] Nanowatt Logic and Using Field-effect Metal-oxide Semiconductor. Nanowatt Logic Using Field-Effect Metal-Oxide Semiconductor Triodes. pages 50–51, 1963.

[2] G M Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114, 1965. URL: https://newsroom.intel.com/wp-content/uploads/sites/11/2018/05/moores-law-electronics.pdf.

[3] Ferdinand Peper. The End of Moore's Law: Opportunities for Natural Computing? *New Generation Computing*, 35(3):253–269, 2017. doi:10.1007/s00354-017-0020-4.

[4] Kelin J. Kuhn. Moore's law past 32nm: Future challenges in device scaling. *Proceedings - 2009 13th International Workshop on Computational Electronics, IWCE 2009*, pages 1–4, 2009. doi:10.1109/IWCE.2009.5091124.

[5] George Strawn, Candace Strawn, and Gordon Moore. Moore's Law at Fifty. pages 69–72, 1970.

[6] Walter A. Harrison. Tunneling from an independent-particle point of view. *Physical Review*, 123(1):85–89, 1961. doi:10.1103/PhysRev.123.85.

[7] Juan C. Ranuárez, M. J. Deen, and Chih Hung Chen. A review of gate tunneling current in MOS devices. *Microelectronics Reliability*, 46(12):1939–1956, 2006. doi:10.1016/j.microrel.2005.12.006.

[8] Ali Ovg. Quantum Tunneling and Hartman Effect. (June), 2015. doi:10.13140/RG.2.1.2979.3449.

[9] John Von Neumann and Michael D. Godfrey. 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.

[10] Yury Yu Illarionov, Alexander G. Banshchikov, Dmitry K. Polyushkin, Stefan Wachter, Theresia Knobloch, Mischa Thesberg, Lukas Mennel, Matthias Paur, Michael Stöger-Pollach, Andreas Steiger-Thirsfeld, Mikhail I. Vexler, Michael Waltl, Nikolai S. Sokolov, Thomas Mueller, and Tibor Grasser. Ultrathin calcium fluoride insulators for two-dimensional field-effect transistors. *Nature Electronics*, 2(6):230–235, 2019. URL: http://dx.doi.org/10.1038/s41928-019-0256-8, doi:10.1038/s41928-019-0256-8.

[11] B. Radisavljevic, A. Radenovic, J. Brivio, V. Giacometti, and A. Kis. Single-layer MoS2 transistors. *Nature Nanotechnology*, 6(3):147–150, 2011. doi:10.1038/nnano.2010.279.

[12] Tzvetan S. Metodi, Arvin I. Faruque, and Frederic T. Chong. *Quantum computing for computer architects, second edition*, volume 13. 2011. doi:10.2200/S00331ED1V01Y201101CAC013.

[13] C. G. Almudever, N. Khammassi, L. Hutin, M. Vinet, M. Babaie, F. Sebastiano, E. Charbon, and K. Bertels. Towards a scalable quantum computer. *Proceedings - 2018 13th IEEE International Conference on Design and Technology of Integrated Systems In Nanoscale Era, DTIS 2018*, 53(1):1, 2018. doi:10.1109/DTIS.2018.8368579.

[14] Leon Chua. Memristor - The Missing Circuit Element. *IEEE Transactions on Circuit Theory*, C(5):507–519, 1971.

[15] Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart, and R. Stanley Williams. The missing memristor found. *Nature*, 453(7191):80–83, 2008. doi:10.1038/nature06932.

[16] Qiangfei Xia, Warren Robinett, Michael W. Cumbie, Neel Banerjee, Thomas J. Cardinali, J. Joshua Yang, Wei Wu, Xuema Li, William M. Tong, Dmitri B. Strukov, Gregory S. Snider, Gilberto Medeiros-Ribeiro, and R. Stanley Williams. Memristor-CMOS hybrid integrated circuits for reconfigurable logic. *Nano Letters*, 9(10):3640–3645, 2009. doi:10.1021/nl901874j.

[17] D. B. Strukov, D. R. Stewart, J. Borghetti, X. Li, M. Pickett, G. Medeiros Ribeiro, W. Robinett, G. Snider, J. P. Strachan, W. Wu, Q. Xia, J. Joshua Yang, and R. S. Williams. Hybrid CMOS/memristor circuits. *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, (May):1967–1970, 2010. doi:10.1109/ISCAS.2010.5537020.

[18] Kwang Ting Cheng and Dmitri B. Strukov. 3D CMOS-memristor hybrid circuits: Devices, integration, architecture, and applications. *Proceedings of the International Symposium on Physical Design*, pages 33–40, 2012. doi:10.1145/2160916.2160925.

[19] Kyoungrok Cho, Sang Jin Lee, and Kamran Eshraghian. Memristor-CMOS logic and digital computational components. *Microelectronics Journal*, 46(3):214–220, 2015. URL: http://dx.doi.org/10.1016/j.mejo.2014.12.006, doi:10.1016/j.mejo.2014.12.006.

[20] Mehri Teimoory, Amirali Amirsoleimani, Arash Ahmadi, and Majid Ahmadi. A hybrid memristor-CMOS multiplier design based on memristive universal logic gates. *Midwest Symposium on Circuits and Systems*, 2017-Augus:1422–1425, 2017. doi:10.1109/MWSCAS.2017.8053199.

[21] Vishal Saxena. A compact CMOS memristor emulator circuit and its applications, 2019. arXiv:1711.06819, doi:10.1109/MWSCAS.2018.8624008.

[22] Myoung Jae Lee, Chang Bum Lee, Dongsoo Lee, Seung Ryul Lee, Man Chang, Ji Hyun Hur, Young Bae Kim, Chang Jung Kim, David H. Seo, Sunae Seo, U. In Chung, In Kyeong Yoo, and Kinam Kim. A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta2O5-xx/TaO2-xbilayer structures. *Nature Materials*, 10(8):625–630, 2011. URL: http://dx.doi.org/10.1038/nmat3070, doi:10.1038/nmat3070.

[23] Kuk Hwan Kim, Siddharth Gaba, Dana Wheeler, Jose M. Cruz-Albrecht, Tahir Hussain, Narayan Srinivasa, and Wei Lu. A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano Letters*, 12(1):389–395, 2012. doi:10.1021/nl203687n.

[24] F. Pan, S. Gao, C. Chen, C. Song, and F. Zeng. Recent progress in resistive random access memories: Materials, switching mechanisms, and performance. *Materials Science and Engineering R: Reports*, 83(1):1–59, 2014. URL: http://dx.doi.org/10.1016/j.mser.2014.06.002, doi:10.1016/j.mser.2014.06.002.

[25] Georgios Papandroulidakis, Ioannis Vourkas, Angel Abusleme, Georgios Ch Sirakoulis, and Antonio Rubio. Crossbar-Based Memristive Logic-in-Memory Architecture. *IEEE Transactions on Nanotechnology*, 16(3):491–501, 2017. doi:10.1109/TNANO.2017.2691713.

[26] Jose M. Cruz-Albrecht, Timothy Derosier, and Narayan Srinivasa. A scalable neural chip with synaptic electronics using CMOS integrated memristors. *Nanotechnology*, 24(38), 2013. doi:10.1088/0957-4484/24/38/384011.

[27] Melika Payvand, Justin Rofeh, Avantika Sodhi, and Luke Theogarajan. A CMOS-memristive self-learning neural network for pattern classification applications. *Proceedings of the 2014 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2014*, (1):92–97, 2014. doi:10.1109/NANOARCH.2014.6880486.

[28] Isha Gupta, Alexantrou Serb, Ali Khiat, Ralf Zeitler, Stefano Vassanelli, and Themistoklis Prodromakis. Real-time encoding and compression of neuronal spikes by metal-oxide memristors. *Nature Communications*, 7:1–9, 2016. URL: http://dx.doi.org/10.1038/ncomms12805, doi:10.1038/ncomms12805.

[29] Ya Nan Zhong, Tian Wang, Xu Gao, Jian Long Xu, and Sui Dong Wang. Synapse-Like Organic Thin Film Memristors. *Advanced Functional Materials*, 28(22):1–6, 2018. doi:10.1002/adfm.201800854.

[30] Qiangfei Xia and J. Joshua Yang. Memristive crossbar arrays for brain-inspired computing. *Nature Materials*, 18(4):309–323, 2019. URL: http://dx.doi.org/10.1038/s41563-019-0291-x, doi:10.1038/s41563-019-0291-x.

[31] Xinjie Guo, Farnood Merrikh-Bayat, Ligang Gao, Brian D. Hoskins, Fabien Alibart, Bernabe Linares-Barranco, Luke Theogarajan, Christof Teuscher, and Dmitri B. Strukov. Modeling and experimental demonstration of a hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits. *Frontiers in Neuroscience*, 9(DEC):1–8, 2015. doi:10.3389/fnins.2015.00488.

[32] Ahmad Fuad Adzmi, Azman Nasrudin, Wan Fazlida Hanim Abdullah, and Sukreen Hana Herman. Memristor Spice model for designing analog circuit. *SCOReD 2012 - 2012 IEEE Student Conference on Research and Development*, pages 78–83, 2012. doi:10.1109/SCOReD.2012.6518615.

[33] Shuang Pi, Mohammad Ghadiri-Sadrabadi, Joseph C. Bardin, and Qiangfei Xia. Nanoscale memristive radiofrequency switches. *Nature Communications*, 6(May):1–9, 2015. URL: http://dx.doi.org/10.1038/ncomms8519, doi:10.1038/ncomms8519.

[34] Myungsoo Kim, Ruijing Ge, Xiaohan Wu, Xing Lan, Jesse Tice, Jack C. Lee, and Deji Akinwande. Zero-static power radio-frequency switches based on MoS2 atomristors. *Nature Communications*, 9(1):1–7, 2018. URL: http://dx.doi.org/10.1038/s41467-018-04934-x, doi:10.1038/s41467-018-04934-x.

[35] Naheem Olakunle Adesina and Ashok Srivastava. Memristor-based loop filter design for phase locked loop. *Journal of Low Power Electronics and Applications*, 9(3), 2019. doi:10.3390/jlpea9030024.

[36] Giovanni E. Pazienza and Jordi Albo-Canals. Teaching memristors to EE undergraduate students. *IEEE Circuits and Systems Magazine*, 11(4):36–44, 2011. doi:10.1109/MCAS.2011.942752.

[37] Shibing Wang, Yashuai Yang, and Jing Wang. Introducing Memristor to Electrical and Electronic Engineering Undergraduates. (Icesame):107–112, 2016. doi:10.2991/icesame-16.2016.23.

[38] J. Joshua Yang, Matthew D. Pickett, Xuema Li, Douglas A.A. Ohlberg, Duncan R. Stewart, and R. Stanley Williams. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature Nanotechnology*, 3(7):429–433, 2008. doi:10.1038/nnano.2008.160.

[39] 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), 2009. doi:10.1063/1.3236506.

[40] J. Joshua Yang, Feng Miao, Matthew D. Pickett, Douglas A.A. Ohlberg, Duncan R. Stewart, Chun Ning Lau, and R. Stanley Williams. The mechanism of electroforming of metal oxide memristive switches. *Nanotechnology*, 20(21), 2009. doi:10.1088/0957-4484/20/21/215201.

[41] Feng Miao, J. Joshua Yang, Julien Borghetti, Gilberto Medeiros-Ribeiro, and R. Stanley Williams. Observation of two resistance switching modes in TiO2 memristive devices electroformed at low current. *Nanotechnology*, 22(25), 2011. doi:10.1088/0957-4484/22/25/254007.

[42] John G Taylor. *Advances in Neuromorphic Memristor Science and Applications Springer Series in Cognitive and Neural Systems*.

[43] Yogesh N. Joglekar and Stephen J. Wolf. The elusive memristor: Properties of basic electrical circuits. *European Journal of Physics*, 30(4):661–675, 2009. arXiv:0807.3994, doi:10.1088/0143-0807/30/4/001.

[44] Zdeněk Biolek, Dalibor Biolek, and Viera Biolková. SPICE model of memristor with nonlinear dopant drift. *Radioengineering*, 18(2):210–214, 2009.

[45] Mika Laiho, Eero Lehtonen, Alex Russell, and Piotr Dudek. Memristive synapses are becoming reality. *The Neuromorphic Engineer*, (January):10–12, 2010. URL: http://awareness-mag.eu/pdf/003396/003396.pdf, doi:10.2417/1201011.003396.

[46] Ting Chang, Sung Hyun Jo, Kuk Hwan Kim, Patrick Sheridan, Siddharth Gaba, and Wei Lu. Synaptic behaviors and modeling of a metal oxide memristive device. *Applied Physics A: Materials Science and Processing*, 102(4):857–863, 2011. doi:10.1007/s00339-011-6296-1.

[47] 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. doi:10.1109/LED.2011.2163292.

[48] Kristy A. Campbell. Self-directed channel memristor for high temperature operation. *Microelectronics Journal*, 59(August 2016):10–14, 2017. URL: http://dx.doi.org/10.1016/j.mejo.2016.11.006, doi:10.1016/j.mejo.2016.11.006.

[49] Knowm.com. Self Directed Channel Memristors. pages 1–16, 2019. URL: https://knowm.org/downloads/Knowm_Memristors.pdf.

[50] Joachim Neves Rodrigues, Matthias Kamuf, Hugo Hedberg, and Viktor Öwall. A manual on ASIC front to back end design flow. *Proceedings - 2005 IEEE International Conference on Microelectronic Systems Education, MSE '05 - Promoting Excellence and Innovation in Microelectronic Systems Education*, 2005:75–76, 2005. doi:10.1109/MSE.2005.9.

[51] Robinson E. Pino and Youngok K. Pino, April 2013. US Patent 8.427.203 B2.

[52] Wei Yu, Linqiang Ge, Guobin Xu, and Xinwen Fu. *Cybersecurity Systems for Human Cognition Augmentation*, volume 61. 2014. URL: http://link.springer.com/10.1007/978-3-319-10374-7, doi:10.1007/978-3-319-10374-7.

[53] Yanwen Guo, Xiaoping Wang, and Zhigang Zeng. A Compact Memristor-CMOS Hybrid Look-Up-Table Design and Potential Application in FPGA. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(12):2144–2148, 2017. doi:10.1109/TCAD.2017.2681079.

[54] Xilinx Inc. Spartan-6 FPGA Configuration User Guide. *Ug380*, 380(2.3):1–58, 2011. URL: https://www.xilinx.com/support/documentation/user_guides/ug384.pdf.