More than Moore. Experience on Material Implication Computing With an Electromechanical Memristor Emulator

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Stefano Zuin Castillo

In partial fulfilment

of the requirements for the degree in

ELECTRONICS SYSTEMS ENGINEERING

Advisor: Antonio Rubio Sola

Barcelona, September 2016

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

Abstract

The end to the Moore's Law is a fact to have into account. Might be, nanowire FETs could extend transistor's life but not too much time. For that reason, it is necessary to find new logic devices and a new computational paradigm.

Nowadays, Quantum or Neuromorphic computers cannot substitute current computers and just offers specific applications. By the other hand, computing in memory, offers the possibility of store and process data at the same time using memristors and Material Implication Logic.

The Memristor, is a device that offers several interesting characteristics as be a passive two-state device or its non-volatility but, current physical devices offers a high variety in its native parameters.

This project conceives the design and development of an electromechanical memristive emulator circuit and the objective of demonstrate that is possible execute some of the arithmetic-logic instructions from an i4004 using the emulator and IMPLY.

Resum

El final de la llei de Moore es un fet a tindre en compte. Pot set, els Nanowire FETs poden estendre la vida del transistor però no indefinidament. Es per això que es necessari trobar un element lògic a mes de un nou paradigma de computació.

Avui dia, els ordinadors Quàntics i Neuromòrfics no poden substituir el ordinadors actual I nomes ofereixen aplicacions especifiques. No obstant, Computació en Memòria ofereix la possibilitat de desar i processar dades a la vegada utilitzant Memristors i IMPLY.

El Memristor, ofereix diverses característiques interesants com el fet de ser un dispositiu passiu de dos estats o no volatilitat però, els dispositius físics actuals mostren gran variabilitat.

Aquest projecte concep el disseny i desenvolupament d'un circuit electromecànic emulador de memristor i l'objectiu de demostrar que es possible executar algunes de les instruccions aritmètiques i lògiques del i4004 utilitzant l'emulador i IMPLY.

Resumen

El final de la Ley de Moore es un hecho a tener en cuenta. Quizás, los Nanowire FETs pueden extender la vida del transistor pero no indefinidamente. Es por eso que es necesario encontrar un nuevo dispositivo lógico además de un nuevo paradigma de computación.

Hoy en día, los ordenadores Cuánticos i Neuromórficos no pueden substituir los ordenadores actuales i solo ofrecen aplicaciones específicas. No obstante, Computación en Memoria ofrece la posibilidad de procesar y guardar datos a la vez utilizando Memristores e IMPLY.

El Memristor, ofrece diversas características interesantes como el hecho de ser un dispositivo pasivo de dos estados o no volatilidad pero, los dispositivos físicos actuales muestran gran variabilidad.

Este proyecto concibe el diseño i el desarrollo de un circuito electromecánico emulador de memristor i el objetivo de demostrar que es posible ejecutar algunas de las instrucciones aritméticas y lógicas del i4004 utilizando el emulador e IMPLY.

a mi Padre

Acknowledgements

I will express my gratitude to Antonio for be my supervisor in this project, give me the opportunity to conceive it and submit it to an International congress. With the advices of Antonio as well as Manel, has been much easier the project development, thank you very much you too.

Also, I will express thanks to Vicente and Melchor. They are the electronics laboratory technicians who helps me during the design and development of the PCB boards. Again, thank you.

Finally, I will be grateful for support me in every decision to my family, friends and colleagues. Thank you.

Revision history and approval record

Revision	Date	Purpose
0	25/09/2016	Document creation
1	26/09/2016	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Stefano Zuin Castillo	szuincastillo@gmail.com
Antonio Rubio Sola	antonio.rubio@upc.edu

Written by: Stefano Zuin Castillo		Reviewed and approved by: Antonio Rubio Sola	
Date	25/09/2016	Date	26/09/2016
Name	Stefano Zuin Castillo	Name	Antonio Rubio Sola
Position	Project Author	Position	Project Supervisor

Table of contents

Abstract 1
Resum 2
Resumen 3
Acknowledgements 5
Revision history and approval record 6
Table of contents 7
List of Figures 10
List of Tables: 11
1. Introduction 12
1.1. Technology Evolution 12
1.2. Emerging Devices and New Application Fields [4] 14
1.2.1. Memory Devices. Storage Class Memory Concept 14
1.2.1.1. Redox Memory 14
1.2.1.2. Carbon Memory 14
1.2.1.3. Molecular Memory 15
1.2.2. Logic Devices 15
1.2.2.1. Extending MOSFETs to the End of the Roadmap 15
1.2.2.2. Charge-Based Beyond CMOS 16
1.2.2.3. Alternative Information Processing Devices 17
1.2.3. *ITRS*'s benchmarking 17
2. State of the art of the technology used or applied in this thesis: 19
2.1. Memristor concept: Technologies and Models 19
2.1.1. Memristor models 20
2.1.1.1. Linear Ion Drift Model 20
2.1.1.2. Nonlinear Ion Drift Model 20
2.1.1.3. Simmons Tunnel Barrier Model 20
2.1.1.4. ThrEshold Adaptive Memristor (TEAM) Model 21
2.1.1.5. Voltage ThrEshold Adaptive Memristor (VTEAM) Model 21
2.1.2. Memristor Technologies 21
2.1.2.1. TiO_2 HP Labs Memristor 21
2.1.2.2. Knwom Memristor 21
2.2. Memristors and memory 22
2.3. Von Neumann Computing Alternatives Principles 22
2.3.1. Quantum Computers 22

2.3.2. Neuromorphic Computers....23
2.3.3. Computation in Memory (CIM) Architecture....24
2.4. Computing concept by Material Implication....24
2.5. Real Memristor Device Evaluation....26
3. Methodology used in the project development:....27
3.1 Simulation with Memristor Model. Computing Application....27
3.1.1. I-V Characteristic Memristor Curve Simulation....27
3.1.2. Material Implication Simulation....28
3.2. Synthesis of a Memristive Emulator Circuit: Experimental Structure....29
3.2.1. Basic Design....29
3.2.2. Complete Design....30
3.2.3. Connectors....30
3.2.4. PCB Design....30
3.3. Development of a Memristive Circuit able to Compute Arithmetic-logic operations based on IMPLY and Memristors....31
3.3.1. Sequences....31
3.3.2. Required Devices....33
3.3.2.1. Voltage Driver....33
3.3.2.2. Altera DE2 development kit....34
3.3.2.3. Power Supply....34
3.3.3. Interconnection....34
3.3.4. Final Hardware Set....35
3.3.5. Sequence Controller Algorithm....36
3.3.6. Instruction Execution....36
4 Results....37
5. Improvement Opportunities....38
5.1. Memristive Emulator....38
5.2. Final System....38
6 Budget....38
5.3. Staff Costs....38
5.4. Hardware Costs....39
5.4.1. Emulator Circuit Costs....39
5.4.2. Voltage Driver Costs....39
5.4.3. FPGA Costs....40
5.5. Summary....40
7 Conclusions....40

Bibliography.....41
Glossary.....43

List of Figures

Figure 1. Transistors and clock speeds for chip [3] 13
Figure 2. Memristors states using diferents frequencies on input signals 20
Figure 3. I-V curve square signal 20
Figure 4. Linear Ion Drift Model 20
Figure 5. Real memristor performance 20
Figure 6. Knwom Memristor Material Stack [14] 21
Figure 7. Exemple of a crossbar [16] 22
Figure 8. Exemple of a memrestive crossbar memory [16] 22
Figure 9. Memristive IMPLY Crossbar 24
Figure 10. Basic IMPLY Crossbar 24
Figure 11. Knowm Memristor Experience [25] 26
Figure 12. Simulation I-V Curve 28
Figure 13. Simulation Schematic 28
Figure 14. Case 1 IMPLY. Time write 40 ns 28
Figure 15. Case 2 IMPLY 28
Figure 16. Case 3 IMPLY. State drift reduced due to use a time write of 40 ns 28
Figure 17. Case 4 IMPLY 28
Figure 18. Emulator Basic Design 29
Figure 19. Emulator Complete Design 30
Figure 20. Seven-set memristive emulator circuits 31
Figure 21. Voltage Driver, PCB Design 33
Figure 22. FPGA Peripherals 34
Figure 23. Emulator interconnection on Motherboard 35
Figure 24. Final System 35

List of Tables:

Table 1. ITRS's Memory and Logic Devices benchmarking. 18
Table 2. IMPLY's True Table 25
Table 3. Verilog-A Library Parameters 27
Table 4. Table of Sequences, result stored in R1 32
Table 5. Sequence code using shift switches 36
Table 6. Instructions Results 37
Table 7. Staff Costs 38
Table 8. Hardware Costs: Emulator Circuit 39
Table 9. Hardware Cost. Voltage Driver 39
Table 10. Hardware Costs: FPGA 40
Table 11. Budget Summary 40

1. Introduction

With the Moore's Law definition [1], there was specified that the transistor density inside a microcontroller will be double every two years. This has been possible decreasing the transistor's dimension. Nowadays, that law remains active. Nevertheless, the microcontroller's clock speed remains constant since 2000s. In addition, it is expected that with a width length under 10 nm, the transistor will be governed by quantum uncertainties.

Having said that, many research centers guess that a substitute for the transistor, in computing field, and a new computing paradigm will be necessary. A device that may make it possible might be the Memristor, although possibly has a hybrid system with other technologies (including conventional).

This thesis will try to demonstrate that it is possible develop arithmetic-logic operations using the concept of Material Implication with memristors and also, execute some of the instructions from Intel i4004 processor [2] following a nonconventional track. To do that, an exhaustive study of the state of the art of memristor and material implication has been done reading many articles about that fields. In addition, Material Implication simulations have been performed using VTEAM memristor model. With that, it has been possible check the Material Implication behavior. Also, it has been possible have first contact with real memristors and confirm its real behavior. Finally, a memristive ideal emulator circuit design was developed in PCB. Thanks to that, a voltage driver and a FPGA, it has been possible develop a two-register processor able to compute some arithmetic-logic instructions from the Intel i4004 microcontroller.

In this section, it is going to introduce the actual state of the current computing. Firstly, the effects of Moore law nowadays will been explained and, the necessity of a new roadmap for future memories and computation, focusing in emerging devices proposed by *ITRS*. Then, the devices will be explained specifying its strengths and weaknesses. Finally, *ITRS* have done a rating of them, it is interesting to compare their progress.

1.1. Technology Evolution

As predicted Moore's Law [1] in 60s, the transistors density in microcontrollers has been doubling, approximately, each two years. This has been possible decreasing the dimensions of a transistor in a display of technology progress. But, It will arrive a certain point where physically cannot be possible decrease it more, arriving then to a limit of this evolution. And, it seems that this limit is closer [3] [4] [5].

Since 2000s, with the introduction of Intel Pentium IV, it is posible to observe in [*Figure 1*] that the microcontroller's clock speeds remain constant. This is because working at these higher frequencies with silicon made transistors every time smaller, the heat emissions increase and also appears quantum uncertainties [5]. Because of that, semiconductor developers decided to work with multicore processors. Introducing parallelism tools, it is possible to obtain an ideal increase of their clock speeds per the number of cores.

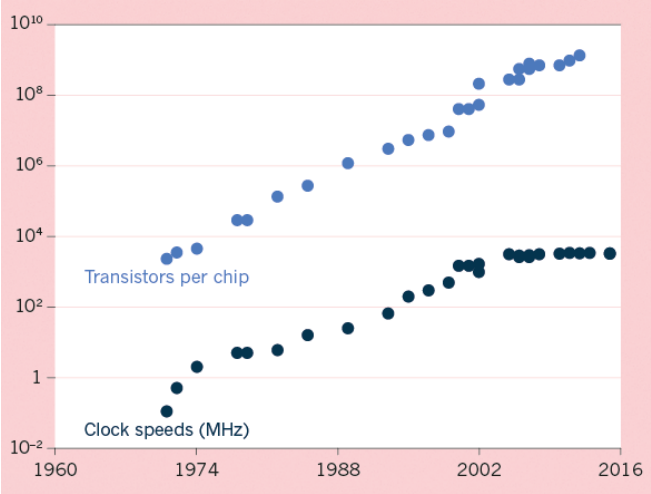


Figure 1. Transistors and clock speeds for chip [3]

According to *ITRS* (International Technology Roadmap for Semiconductors http://www.itrs2.net), it is possible to arrive to the transistors limit in 2020s [4]. Nowadays, magazines as *Nature* or the selfsame *ITRS*, states the necessity of follow a more than Moore R+D plan [3] [4].

Some of these plans are Quantum and Neuromorphic computers, but no one of them have been commercialized out of a laboratory. In addition, these computers are useful for specific applications [3].

Another alternative used is the development of 3D chips. They offered the possibility to package more devices in the same surface. However, the heat problems increase still more [6], for it, nowadays it is useful in memory systems.

There are appearing strategies to unify memory and computing chips in a closer way using the same encapsulation, predicting lower heat emissions (due to the reduction of parasitic capacitances between interconnections) and allowing lower access times.

As well as a powerful microprocessor, it is necessary a device where to store computing data. So a high-capacity non-volatile memory is required in an ideal situation.

Usually, magnetic hard drive disk (HDD) has been used, due to its low cost and high capacity to store data. These devices are still alive and their specifications can evolve in time. Nevertheless, HDD has a high power consumption, a high index of data loss due to mechanical problems and very high access times.

Otherwise, a new technology is competing with HDD. Solid State Drive (SSD), known as flash memories, begin to shows as an alternative. They offers lower access times, lower dimensions, and reduction of the power consumption in comparison with HDD. Unfortunately, they offered important disadvantages as a lifetime of 10^4 – 10^5 erase cycles and increment of the working voltage in the range of 15 V. In addition, as this devices do not include direct data rewriting, it is necessary a sophisticated accumulation and elimination process (rubbish data). So it is necessary to use system resources as microcontroller power and RAM memory space. Is it true that this technology has been reducing its dimensions offering more capacity but, erase, write and read times are still the same. Furthermore, using strategies as “multi-level cell” able to expand the capacity of the memory in a factor of 2 or 4 [7], the lifetime and the data retention values have decreased, which are very important parameters in memory design.

1.2. Emerging Devices and New Application Fields [4]

The scaling limit of the transistor will produce new challenges to overcome related with the invention of new technologies to perform the design of memories and computing systems.

In the memory field, it will be necessary to go beyond the limits set by SRAM and SSD memories.

The MPU capacity to execute programs is limited by the processor and the memory. The current solution is increase the cache memory capacity. Unfortunately the device dimensions also increase. Furthermore, to store the results is necessary a non-volatile memory which offers lower access times in comparison with SSD.

After this, some of the interesting emerging devices according to *ITRS* will be introduced. Firstly, memory devices will be introduced and finally logic devices.

1.2.1. Memory Devices. Storage Class Memory Concept

Storage-Class Memory (SCM) is a roadmap described by *ITRS* which explains a kind of memory that should include the advantages of magnetic Hard Drive Disks and Solid State Drive memories, as high data capacity, low cost, low power consumption and high strength.

That is why a type of nonvolatile memory that meets these requirements is in the crosshairs of research centers and computer companies.

The devices proposed here do not fulfil all these requirements but, in addition they also offer different characteristics than the proposed.

1.2.1.1. Redox Memory

The redox memory operation is based on a change in resistance inside a MIM (Metal-Insulator-Metal) structure caused by an ion migration combined with a redox reaction.

The conduction is normally due to filaments, and hence a one–time formation process is required before the bi-stable switching can be started. If this effect can be controlled, the size of this kind of components can be reduced. According to the *ITRS*, if the active distance over the anions and cations moves (in the <than 10nm regime), the switching time can be reduced to a few nanoseconds. This devices offers lifetimes in the range of 10^6 cycles.

Actually, the details about the switching mechanism of redox memories are still unknown. Another parameter to improve is its device-to-device statistical variation in high and low resistance. Nevertheless, the results obtained are encouraging

1.2.1.2. Carbon Memory

There are several carbon forms and several switching mechanisms of them.

An example of that are the amorphous carbon or diamond-like carbon. In them the switching mechanism is due to a transition in the electronics orbital sp^3 (diamond-like high resistance) and sp^2 (graphite-like low resistance).

Again, this effect is not fully understood because some studies certify that this is produced by a thermal effect, others explain that this occurs in low temperature conditions and applying an electric field.

1.2.1.3. Molecular Memory

In a molecular memory, data are stored by applying an external voltage. That make a changing in conduction transition state of the molecule, there are two different states. Data could be read measuring the resistance of the molecule.

Molecular concept offers an extreme scalability. A bit can be saved in a single molecule. As it is a small device, a very dense circuit could be built. Also, molecules of the same type are identical, so its characteristics are the same too reducing the variability of the device.

Again, there are many questions to understand about molecular memories and several phenomes to be controlled as the high sensitivity of the device due to exterior parameters as contacts and environment. Nowadays, the molecular memory field is a developing long term technology.

1.2.2. Logic Devices

Acording to *ITRS*, with the current technology it is possible arrive to the transistor limit in 2020s, so a new design of the transistor will be the first device to take in account. Silicon transistor's channel is being strained every day more due to higher user requirements. For that reason a new design of transistor technologies and materials are the first options to take in account to extend MOSFETs life.

After that, the necessity to find a new logic switching device, able to work in two defined states, is the future of computing technology. The most important requirement is, at least, work within the same clock speeds as CMOS transistors.

Some of the devices to consider by the *ITRS* could be classified in order to extend MOSFETs to the end of the road map or going beyond CMOS.

1.2.2.1. Extending MOSFETs to the End of the Roadmap

1.2.2.1.1. Carbon Nanotube FETs

In this kind of transistors, drain and collector are connected by single walled carbon nanotubes. Gate terminal is placed around the carbon nanotube.

The advantages offered by Carbon Nanotube FETs are the high mobility of carriers and potential to minimize the subthreshold slope using concentric oxide and metal (gate) around the carbon nanotube. Nevertheless, there are some challenges to realize this advantages as the control of the bandgap energy, reduce the contacts resistance, etc.

Some advances have been the development of a lower than nine nm channel CNT FET without observing short channel effects and the fabrication of a computer composed by 178 CNT FETs.

1.2.2.1.2. Graphene FETs

In these transistors, the cannel is made of graphene. That means that it is possible to obtain higher mobility than carbon nanotubes.

The main limitation of this devices is its zero bandgap energy that makes him a conductor. Several method to open a bandgap have been proposed. The most successful is make a transistor of Graphene nanoribbons.

Graphene nanoribbons are strips of graphene with ultra-thin width. In function of the orientation of its crystal structure and the width is possible to make a semiconducting material with a selectable bandgap.

It is difficult to control the position and direction of nanoribbons which is an obstacle for its fabrication.

1.2.2.1.3. Nanowire Field Effect Transistors (NWFETs)

Nanowire are quasi one-dimensional crystals made of a semiconductor material. There is possible to obtain nanowire with a diameter of less than a nanometer. There are two ways to produce a nanowire: Bottom-up, based on intrinsic properties and produced by VLS (Vapor Liquid Solid technics), and Top-down, based on external tools to develop nanowires as Lithography.

Nanowires are used to produce FETs using them as an alternative to the planar channel in MOSFETs. NWFETs has gate-all-around that produce a reduction of short channel effects.

However, there are problems to solve before commercialization is feasible. The production and uniformity must be improved. Also, treatments to remove impurities like roughness and defects on its surface must be developed.

1.2.2.1.4. Tunnel FETs

Tunnel FETs (TFETs) have a similar structure as MOSFETs but materials are different.

They have a p-type, intrinsic, n-type (p-i-n) junction with the objective to have transitions from on to off state more abrupt than MOSFETs. Reducing this subthreshold, TFETS can reduce its power consumption.

At low gate bias, the width between the p and i energy bands is wide enough to have a very low probability of tunnelling. While the gate voltage increase the band of energy of the intrinsic material are pushed down allowing tunnel effect.

TFETs technology expects to have the same speed than MOSFETs with the same voltage but with lower power. The design of these devices must to be upgraded for using in integrated circuits in order to obtain more compact devices.

1.2.2.2. Charge-Based Beyond CMOS

The following devices are inspired on the transistor's gate which controls the ON and OFF state. However, they don't have the same behaviour.

1.2.2.2.1. Spin FET and Spin MOSFET Transistors

These two technologies are joined in this heading because both transistors have a structures of ferromagnetic source and drain which act has injector and detector respectively.

The main difference between both transistors is in the functionality of the gate. In Spin MOSFET the gate acts as in standard transistors (switching controller). However, in Spin FET the gate controls the direction of the spin.

1.2.2.2.2. NEMS Switch

Micro/Nano-Electro-Mechanical switches are devices that uses electrostatic force to establish a path within two electrodes. There are two properties that are unattainable in MOSFETs: zero off-state leakage and zero subthreshold swing. These devices are able to work in high-temperature conditions and are immune to ionizing radiation.

Nevertheless, it is necessary to implement a four-terminal logic relay to avoid a dependence in the state of a switch from other switches. This is used because to develop many of the logic gates it is necessary to connect many switches in series producing dependences of state in these switches.

1.2.2.3. Alternative Information Processing Devices

These devices have an absolutely different behaviour than CMOS FETs.

1.2.2.3.1. Spin Wave Device (SWD)

It is a type of magnetic logic device that uses collective spin oscillation (spin wave) to transmit information or process it. It's structure is based on four elements: magneto-electric cells (i.e. multiferroic elements) are used to convert voltage pulses in spin wave and vice versa; magnetic waveguides used to propagate the spin wave between the magneto-electricells; magnetic junctions to couple two or more waveguides; and phase shifter to control the phase of the propagated spin wave.

Some advantages between SWD against Si CMOS are:

- It is possible to add voltage amplitude and phase to develop logic functions with a fewer number of elements
- Reduction of the power consumption by using the built-in magnetic non-volatile memory
- It give the possibility to multiplex many channels of information in frequency

But now, SWD must be scaled and here is a challenge for them because is not sure that this device can offer the advantages described before in the range of <100nm.

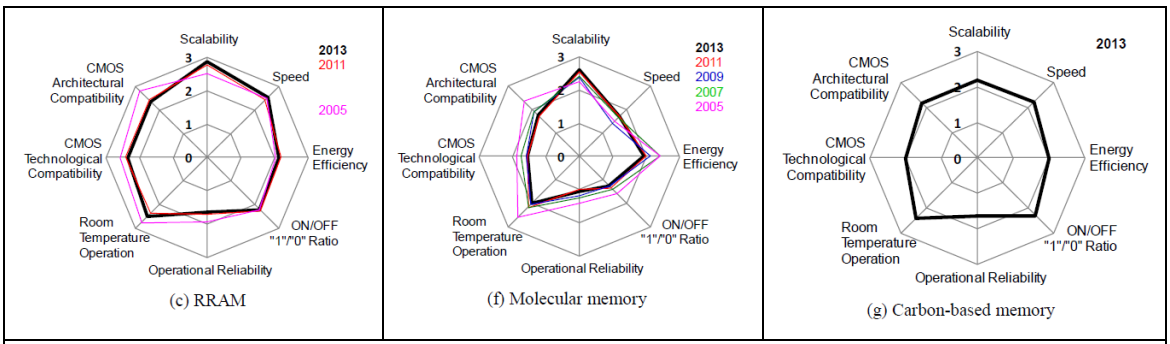
1.2.2.3.2. All Spin Logic (ASL)

This devices uses the blurred concept for distinguish magnetics and spintronics. The demonstration of spin injection into metals and semiconductors from magnetic contacts and the switching of a second magnet by the injected spins makes a closed "ecosystem" between digital (bistable magnets) and analog (spin currents) without the need of a converter.

Some of the issues to solve are the variability due to the temperature and the sensitivity in using a spin current and a channel material.

1.2.3. *ITRS*'s benchmarking

ITRS have done a benchmarking in order to compare some characteristics of these devices. As will be introduced in section 2.2, ReRAM devices can be considered as Memristor.



ITRS's benchmarking Memory devices [4]

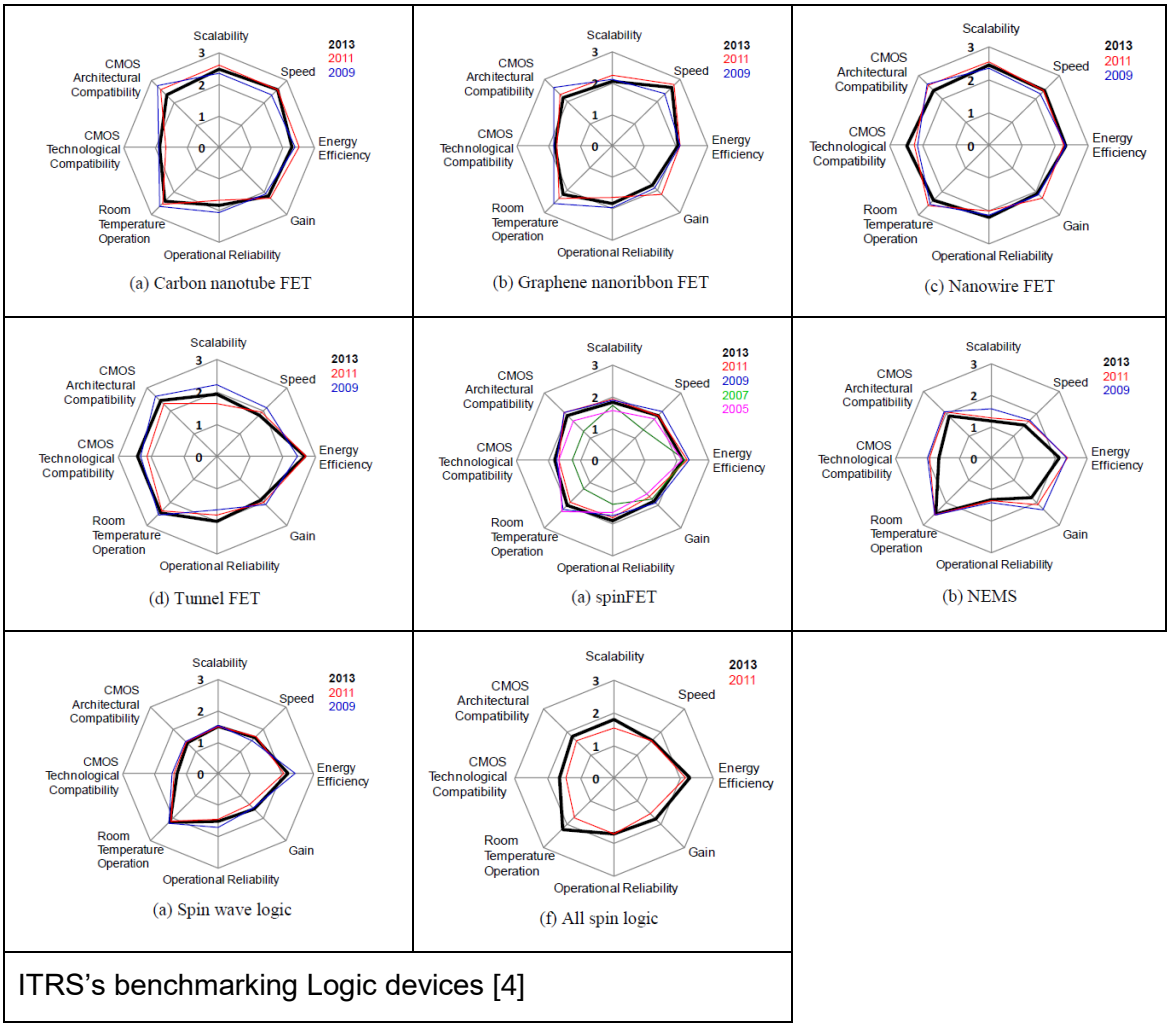


Table 1. ITRS's Memory and Logic Devices benchmarking.

2. State of the art of the technology used or applied in this thesis:

Once the state of art of the current technology have been introduced, this section will introduce an emerging device called memristor and computer architectures different to Von Neumann's.

First, the memristor concept will be introduced and also, the most important memristor models and physical technologies will be detailed. A known memristor application is the development of memory arrays, so it will be interesting to understand this project. After that, an introduction of computer architectures different to Von Neumann's will be explained focusing in Quantum and Neuromorphic Computing, and Computing in Memory Architectures. Furthermore, the concept of Material Implication will be defined which, in addition with memristors, are the focus of this project. And finally, it will be explained why current physical memristors are not useful to develop material implication without CMOS technology.

2.1. Memristor concept: Technologies and Models

The memristor is the 4th fundamental passive circuit element. In the past we knew that there were 3 passive devices:

$$R=v/I \qquad L=d\varphi/di \qquad C=dq/dv$$

In the 1971s Professor Leon Chua realized that there was a missing passive component dominated by the electric charge and the magnetic flux [8].

$$M=d\varphi/dq.$$

He started the study of this component and he realized that the main characteristic was that this component varies its resistance and behaves as a non-volatile memory.

From the relation between flux and charge we obtain two equations. One equation states that there is a quasi-static relation between voltage and intensity, but the resistance depends of the state that the memristor have. The other one, is a time depending equation that defines how the state variable evolve in time. [9]

$$v=R(w)I \qquad dw/dt=i$$

[*Figure 2 and 3*] shows the hypothetical memristor's behavior developed by Leon Chua. It is possible to differentiate two resistences R_{ON} and R_{OFF} controlled by the voltage applied to the memristor.

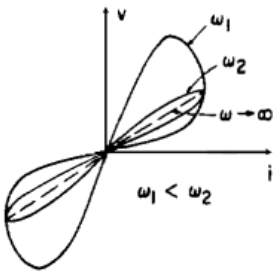


Figure 2. Memristors states using diferents frequencies on input signals

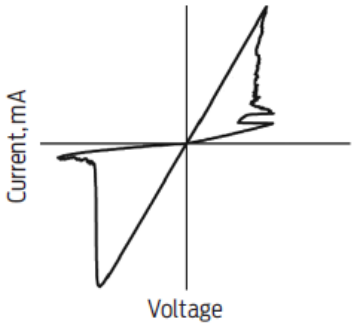


Figure 3. I-V curve square signal

If initially the resistance of the memristor is R_{OFF}, while the voltage is increasing, the current increase until the maximum is reached. At this point the resistence of the memristor turns to R_{ON}. Now while the voltage decreases, the current slowly start to decrease until reach the threshold value, now the resistance turns again to R_{OFF}.

2.1.1. Memristor models

Many models of memristor have been developed in order to perform simulations using theorical real memristors. The following sections will expose the key memristor models that are used in this field of research [10] [11] [12].

2.1.1.1. Linear Ion Drift Model

In this model there are two variable resistors connected in series: one defining the high doped area (R_{ON}) and the oxide area (R_{OFF}), the total resistance is the sum of both resistors. The resistance of both variable resistors are controlled by the length of the doped and undoped area divided by the total length of the device.

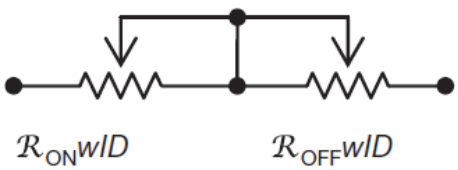


Figure 4. Linear Ion Drift Model

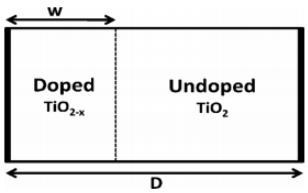


Figure 5. Real memristor performance

This model can be defined as a variable two-state resistor controlled by symmetric threshold voltages.

2.1.1.2. Nonlinear Ion Drift Model

Is the same idea as Linear Ion Drift Model but with intermediate states of resistance. The state can be selected by a parameter within [0, 1] called w.

2.1.1.3. Simmons Tunnel Barrier Model

This model offers a better implementation of memristor where it is possible to obtain non-linearities and asymmetric switching behaviour due to an exponential dependence of the state variable

2.1.1.4. ThrEshold Adaptive Memristor (TEAM) Model

It is a general memristor model. The advantages of it are a current threshold and the tuneable-polynomial non-linearity dependence between the current and the derivative of the internal state variable.

2.1.1.5. Voltage ThrEshold Adaptive Memristor (VTEAM) Model

Similar as his processor model (TEAM) it has a tuneable non-linearity. The unique difference is that the threshold is now controlled by a voltage.

In this model the current–voltage relationship is undefined and can be freely chosen from any current–voltage characteristics

2.1.2. Memristor Technologies

Many different technologic processes has been presented in literarure showing the behaviour of emerging memristive devices.

2.1.2.1. TiO_2 HP Labs Memristor

Thirty five years after L. Chua theorized the memristor, HP Labs were the first in discover a practical memristor component [9] [13]. What they did was a passive component with the properties of non-volatile current controlled resistor.

The component is made by a doped and undoped current-controlled TIO_2. The doped and undoped area can be considered as RON and ROFF. With the current it is possible to control the doped area.

In this technology is possible to stay in intermediate states.

2.1.2.2. Knwom Memristor

Knowm Memristors are devices which its principal performance is done because an electric field. This electric field induced ions through a metal multilayer material stack [*Figure 6*].

These devices are fabricated with an easily oxidable metal layer near an electrode. When a positive voltage is applied to this electrode, the metal starts to oxidize producing ions. The ions flows to the lower potencial electrode passing to an amorphous chalcogenide active layer who starts to metalize producing pathways between both electrodes and reducing the device resistance. Producing a reverse potential, the device increase its resistance [14].

The resistance is related with the amount of metal in the amorphous chalcogenide active layer.

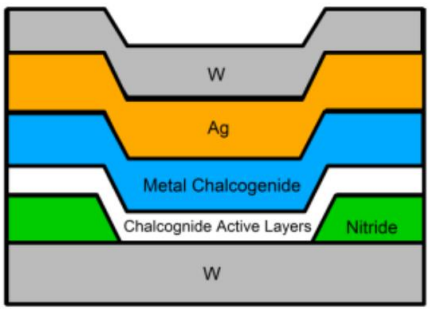


Figure 6. Knwom Memristor Material Stack [14]

2.2. Memristors and memory

After the first conception of a memristor by HP Labs, in 2011 Leon Chua defines a memristor as any device which as a hysteresis loop in I-V curve applying a bipolar periodic signal [15]. For this reason, we can say that ReRAM memories (introduced in section 1.2) are memristors. In addition, other devices as ferroelectric and spintronic memristors are included in the memristor concept. [16]

Some of the most important advantages in ReRAM Memristors, are that they are two-terminal devices compatible with CMOS production and highly scalable. Having said that, might it be insert a higher volume of ReRAM memristors obtaining a more memory capacitance inside the same dimensions.

The design of crossbar arrays using ReRAM memristors is an exciting method to develop memory units. In it, ReRAM memristors are placed in a NxN matrix interconnected by two nanowire layers, controlling rows and columns accesses each of these two [*Figure 7*]. Using this architecture, is easier the write and read processes.

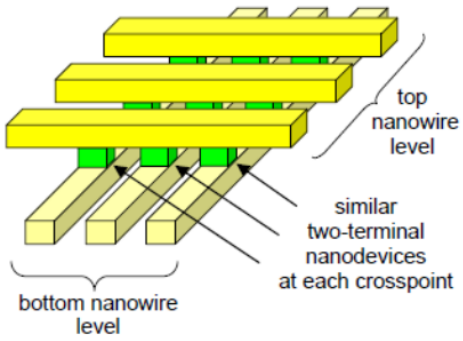


Figure 7. Exemple of a crossbar [16]

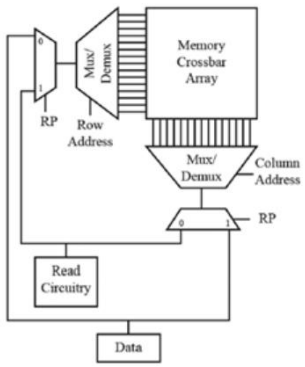


Figure 8. Exemple of a memrestive crossbar memory [16]

Unfortunatly, using current ReRAM memristes is not possible to achieve this Memory Unit without using CMOS technology [16]. Mixing both technologies is it possible to perform a memrestive crossbar array with the needed peripherals inside the Memory Unit [*Figure 8*].

A non-adaptive memory NxN array is that one which is divided by mxm subarrays with the objective to prevent errors storing data. Only a mxm subarray is active and, when a memristor inside it, is near the end of its life time, the operational mxm skips to the next mxm. With this, is possible to extend the crossbar's lifetime. But, what about useful memristors? This reconfiguration is not the most efficient. For that, adaptive methods as a shift mxm subarray can extend still more the crossbar lifetime [16].

2.3. Von Neumann Computing Alternatives Principles

2.3.1. Quantum Computers

Quantum computers are those where its operations are governed by quantum mechanicals laws. Is it true that in a quantic point of view, every computer is governed by quantum physics but, the concept of quantum computers concerns all systems that experiments special transformations in its internal state. These transformations can be controlled just by quantum mechanics [17].

Diverging with current computing, which uses bits (called Cbits in this section), quantum computers uses Qbits (usually known with this name in quantum computers literature). As

is it known, Cbits have switching performance with two defined states, |0} and |1}. In a Qbit, is it possible to have both states in the same Qbits, meaning state |0} and |1} in the same Qbit at the same time. This is possible beacause quantum objects as photons or electrons can have positive or negative spin, adding to this the concept of superposition, the state of the Qbit is defined by |ψ}.

$$|\psi\} = \alpha_0|0\} + \alpha_1|1\} = \begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix} \quad (1)$$

Where α_0 and α_1 are complex numbers which have the condition of make |ψ} a unitary vector.

$$|\alpha_0|^2 + |\alpha_1|^2 = 1 \quad (2)$$

The most interesting idea of this computers is that while Qbits are grouped by N Qbits, is it possible to transmit the same information than 2^N Cbits [17]. This means that is possible to reduce exponentially the number of operations to do.

$$|\psi\} = \alpha_{00}|00\} + \alpha_{01}|01\} + \alpha_{10}|10\} + \alpha_{11}|11\} = \begin{pmatrix}\alpha_{00}\\ \alpha_{01}\\ \alpha_{10}\\ \alpha_{11}\end{pmatrix} \quad (3)$$

Nevertheless, when a Qbit is measured, it must be showed in one of its basic states. This produces the loss of the rest of states before be measured. For that reason, quantum computers tries to compute with the minimum of Qbits considering that superposition cannot be read.

According to this information, quantum computers are not considered as substitution of current computers. They are just useful in special calculations where all superposition parameters are able at the same time.

2.3.2. Neuromorphic Computers

The term neuromorphic describes those systems that tries to imitate neuro-biological architectures in the nervous system. The main difference between Von Neumann and neuromorphic computers is that they are inspired in the human brain. In brain, a device called neuron can process data and store itself. Neurons of brain are interconnected by synapsis. Also, synapsis can connect neurons and nerves. [18] [19]

These computers has a great potential to achieve intelligence based on unreliable devices similar to those found in neuronal tissue. Talking about human-like tasks, neuromorphic computers have a great advantage against Von Neumann computers. This is because the needed hardware to develop this kind of tasks Von Neumann computer augment exponentially with the input complexity. [4]

Neuromorphic computers usually use the noise received to increase the effectivity in its actions. They are low-power machines, nevertheless, they offer a high-density of operations. In addition, unless obtain a high signal-noise rate, neuromorphic computers still working while in Von Neuman computers it stops to correct that error. With low noise in both computers, neuromorphic computers rarely reach the performance levels for wich Von Neumann computers were designed (100% accurancy and $<<10^{-9}$ error rate). [4]

According to ITRS [4], neuromorphic computers just have specific applications. They offer advantages mixing it with Von Neumman computers that offer universal computation ability.

2.3.3. Computation in Memory (CIM) Architecture

The main objectve of this architecture is unify the computation and storage processes in a crossbar array. It has been proposed to solve big data computation problems. This proposition is a memristor-based design but, CMOS devices can appear as periferics to get control from/to the crossbar.

Some of the advantages are the high parallelism that the crossbar array offer, high density of memristors because it's high scalability and low power consumption.

Two operations in conventional computing and in CIM has been compared in [20] obtaining betters results in CIM. The most important challenge in CIM is the evolving of memristors explained before.

CIM, in comparison with Quantum and Neuromorphic architecture, is a very polyvalent architecture that could replace conventional computing in future [20].

The computation and the storage process are made by Material Implication (IMPLY) and NOT logic functions explained in the next chapter.

2.4. Computing concept by Material Implication

Material Implication (IMPLY) [21] [22] [23] [24] is a two-state (B=[0,1]) memristor-based logic operation. In this operation, the state 0 means that the memristor is in high impedance and 0 low impedance.

An advantage of using IMPLY by a memristive-crossbar is the possibility to store and process data simultaneously. This produce a higher process parallelism as well as an access time reduction.

IMPLY operation is often performed by two memristor normally called *p* and *q*. The operation can be expressed like *p->q* or *p IMP q*.The most exciting characteristic is that the operation result is stored in memristor *q*. That means that is not necessary to add a third memristor to store the result.

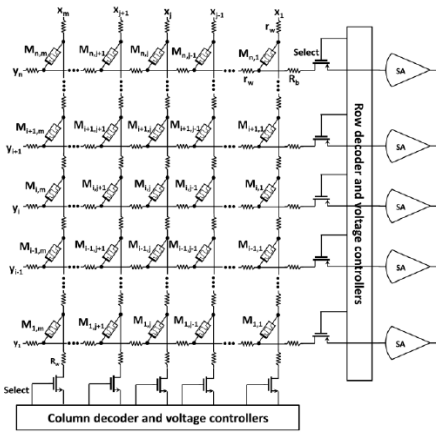


Figure 9. Memristive IMPLY Crossbar

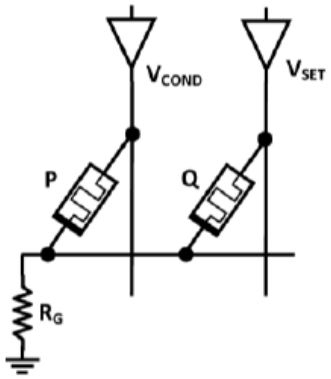


Figure 10. Basic IMPLY Crossbar

The memristors interconnection in IMPLY can be observed in [*Figure 10*]. A voltage V_{COND} is applied in memristor *p* while VSET is applied in *q*. VCOND is always lower than - VTH with VTH the memristor switching voltage (state 0 to 1) - and VSET have to be greater than both other voltages. Memristors p and q are connected to a resistance R_G whose value must be inside (R_{ON}, R_{OFF}).

$$V_{COND} < V_{TH} \leq V_{SET} \quad (4)$$

p	q	p IMPLY q
0	0	1
0	1	1
1	0	0
1	1	1

Table 2. IMPLY's True Table

[*Table 2*] shows the IMPLY true table. From it is possible to observe four different cases. Since now till the end of this section R_{OFF} and R_{ON} will be assumed as a resistances near to open and short circuit respectively to facilitate the understanding of IMPLY operation.

- In the first case, memristor p and q are in high impedance. By applying V_{COND} and V_{SET} the voltage terminals of q is V_{SET}, enough voltage to produce a change of state to low impedance.
- In the second case, memristor p is in high impedance while q is in low impedance. Again the voltage in the terminals of memristor q is V_{SET}. But now, memristor is in low impedance, so there no change of state.
- In the third case, memristor p is in low impedance and memristor q is in high impedance. Being p in low impedance, potential differential produced in it is to low so the voltage applied in terminals of q is V_{SET}-V_{COND}. This voltage is not sufficient to produce a change of state so q remains in high impedance.
- In fourth case, memristor p and q are in low impedance so the applied voltage in terminals of q is not enough to produce a change of state. In addition to that, memristor q would not change its state because is necessary a change of polarity.

These table is easy demonstrable with ideal memristors, while developing it with real memristors, for example model VTEAM, non-linear effects can be observed on the first and third cases. These effects are called drift state and write time.

Write time can be observed on first case. It consist in a reduction of resistance in time till arrive to the desired low impedance meaning state 1. Third case is more sensitive to suffer drift state. If voltages of third case are applied during certain time, the resistance of memristor q tend to decrease [21]. Due to this, is necessary decide the necessary time to apply V_{SET} and V_{COND} to satisfy IMPLY requirements and, a refresh operation after a number of cycles of IMPLY operations in order to refresh the 0 state of a memristor produced by the third case.

It has been demonstrated that using IMPLY and NOT logic operations – being NOT the application of a voltage V_{CLEAR} in a memristor enough to produce a change from 1 to 0 – is possible to develop any other logic function [21]. This is a high strength for IMPLY because with the same hardware configuration is possible store and process data simultaneously. Finally, using both operation it would not be necessary a refresh process of memristors because NOT can be used as well as a refresh operation.

2.5. Real Memristor Device Evaluation

This section demonstrates why is not possible perform IMPLY easily with a physical memristor device. These test have been done by the research group of the University's Electronics Engineering department using real physical memristors from Knowm Company.

Ideally, when a memristor is excited with a periodic signal with an enough amplitude to produce a switch of state, its I-V curve should describe a loop with two well defined lines. The slope of these lines should have as values R_{OFF} and R_{ON}.

[*Figure 11*] shows the I-V curve of Knowm memristors when a periodic signal is applied with enough amplitude to produce switching. Form it is possible to observe different curves by applying different signal periods. This produce a high variety on parameters R_{ON}, V_{SET} and V_{RESET} – being here V_{SET} and V_{RESET} voltages to switch from sate 0 to 1 and from 1 to 0 respectively.

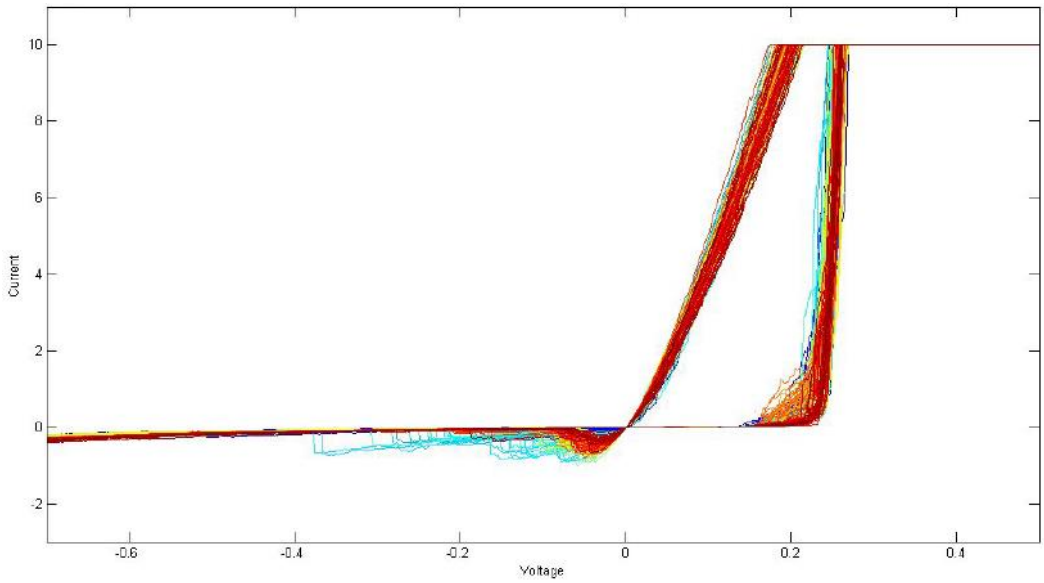


Figure 11. Knowm Memristor Experience [25]

It can be observed that parameter ROFF remains stable, but there is a high random variability on RON (15-10 kΩ), VSET (1.7 – 2.2 V) and VRESET (-0.1 - -0.01V approximately). With these parameters it is no possible do IMPLY as is demonstrated in [24].

3. Methodology used in the project development:

This section deals about the methodology used to the project development. As will be explained in each subsection, simulations with memristor model VTEAM have been done in order to prove Material Implication. Finally, the design and development of an electromechanical memristive emulator have been done with the goal of create an ideal memristive computing architecture based in Material Implication Logic.

3.1 Simulation with Memristor Model. Computing Application

In order to demonstrate that is possible develop IMPLY with a real reliable memristor model it was decided to develop simulations with memristors model VTEAM. These simulation consist in obtain the I-V characteristic curve of memristor and each of the states from IMPLY observing the resistance switching on memristor *q*. To do that, the library [26], wrote in Verilog-A, was used in Cadence IC6 Virtuoso as PSpice tool.

The memristor was characterized with values 100Ω and 100kΩ for R_{ON} and R_{OFF} as well as the width (D) of 3nm. [*Table 3*] show the rest of parameters on the model used [27].

model	*4*	**w_multiplied**	10^{8}	**i_on**	$-8.9\cdot10^{-6}$	**v_on**	*0.3*
window_type	*0*	**p_coeff**	*2*	**x_c**	$1.07\cdot10^{-10}$	**v_off**	*-0.35*
Dt	10^{-10}	**J**	*1*	**B**	$5\cdot10^{-3}$	**iv_relation**	*0*
init_state	*1*	**p_window_ noise**	10^{-18}	**a_on**	$2\cdot10^{-9}$	**x_on**	*1.2n*
Roff	*100k*	**threshold_voltage**	*0*	**a_off**	$1.2\cdot10^{-9}$	**x_off**	*1.8n*
Ron	*100*	**c_off**	$3.5\cdot10^{-6}$	**k_on**	$1.32\cdot10^{-6}$	**alpha**	*2*
D	$3\cdot10^{-9}$	**c_on**	$4\cdot10^{-5}$	**alpha_on**	*10*	**beta**	*9*
uv	10^{-15}	**i_off**	$115\cdot10^{-6}$	**alpha_off**	*10*	**c**	*0.01*
G	*4*	**N**	*14*	**q**	*13*	**a**	*4*

Table 3. Verilog-A Library Parameters

3.1.1. I-V Characteristic Memristor Curve Simulation

To obtain the I-V curve, the schematic that illustrates [*Figure 13*] was used applying a periodic signal with an amplitude of 2 V and a period of 100ns. [*Figure 12*] shows the I-V curve, which is similar to the obtained with Knowm memristors.

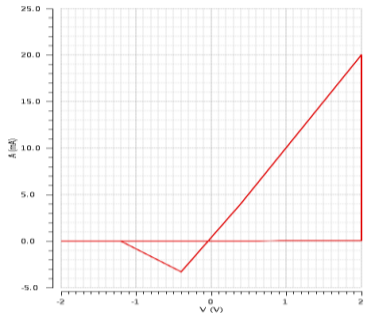


Figure 12. Simulation I-V Curve

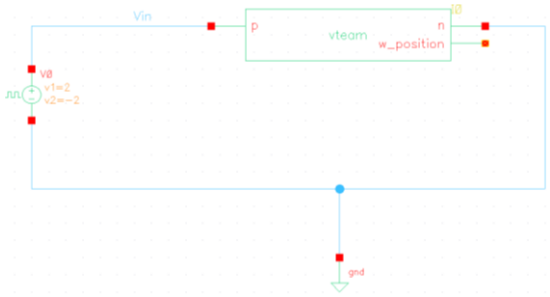


Figure 13. Simulation Schematic

3.1.2. Material Implication Simulation

From the curve I-V, it was decided to apply values of 1 and 2 V to IMPLY voltages V_{COND} and V_{SET} respectively. In addition, a resistance of 1kΩ was selected as R_G.

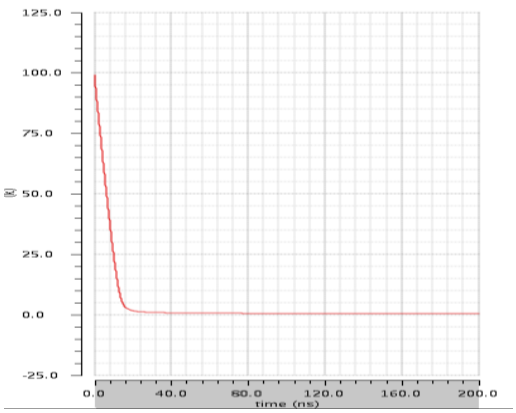


Figure 14. Case 1 IMPLY. Time write 40 ns

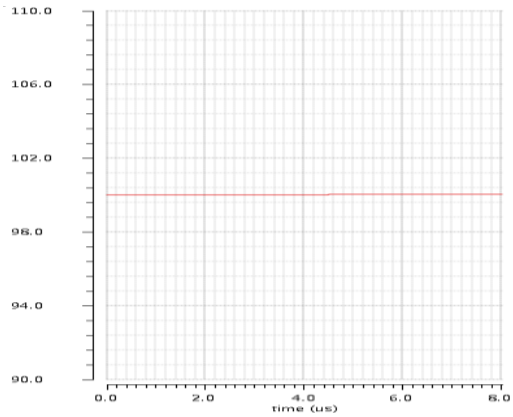


Figure 15. Case 2 IMPLY

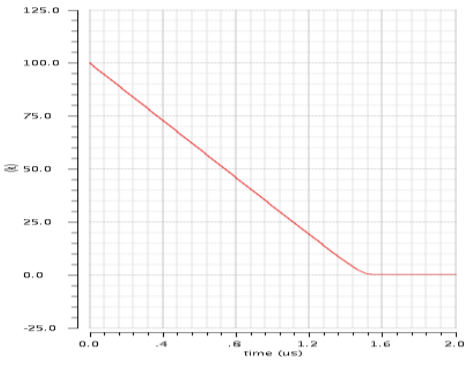


Figure 16. Case 3 IMPLY. State drift reduced due to use a time write of 40 ns

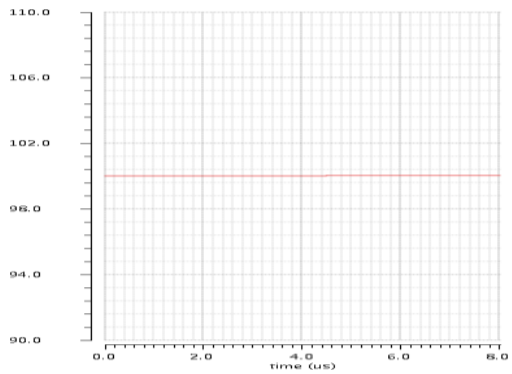


Figure 17. Case 4 IMPLY

As was commented on section 2.2, the non-linear effects are presented on the first and third cases. It can be observed in [*Figure 14*] that the minimum write time must be 40ns. Then on the third case [*Figure 16*], after 40 ns the resistance on memristor *q* as a value on the range of 99.5kΩ. Selecting a write time of 40 ns, is possible to obtain a well operation of IMPLY during many cycles.

3.2. Synthesis of a Memristive Emulator Circuit: Experimental Structure

Once the simulations coincide with the theory, it was decided to develop a memristive emulator circuit of ideal memristor, in order to verify experimentally the correct behavior of IMPLY operation. After verify it, a 7-memristors IMPLY circuit was developed to achieve some of the arithmetic-logic instructions of Intel i4004 microprocessor [2] using two two-bit registers.

3.2.1. Basic Design

The basic design consisted in achieve the following requirements:

- The conception of a passive circuit
- A non-volatile device
- Two fixed states. Without intermediate states
- Fixed switching tensions with the minimum variety that guaranties IMPLY behavior

To achieve this design, a 3-terminal bistable relay was selected. The relay was controlled by the current that flows through a SET and RESET inductors. With this inductors connected in series and with the correct polarity, is possible control the relay with the same signal. In addition, two resistances modeling R_{ON} and R_{OFF} are connected to a pair of terminals of the relay

A Zener diode barrier was used to fix the SET and RESET emulator voltages. These are defined as:

- $V_{set\_emu} = V_{th\_set} + V_{Z1} + V_{\gamma 2}$ (5)
- $V_{reset\_emu} = -V_{th\_reset} - V_{Z2} - V_{\gamma 1}$ (6)

Where V_{set_emu} and V_{reset_emu} are the required voltages to SET and RESET the emulator, V_{th_set} is the necessary voltage in the inductor terminal, $V\gamma$ the forward current of diodes and V_Z the zener voltage.

It is easy to observe that this circuit can be adapted to the required user necessities just changing the Zener diodes and resistances. [*Figure 18*] shows the basic memristive-emulator circuit schematic.

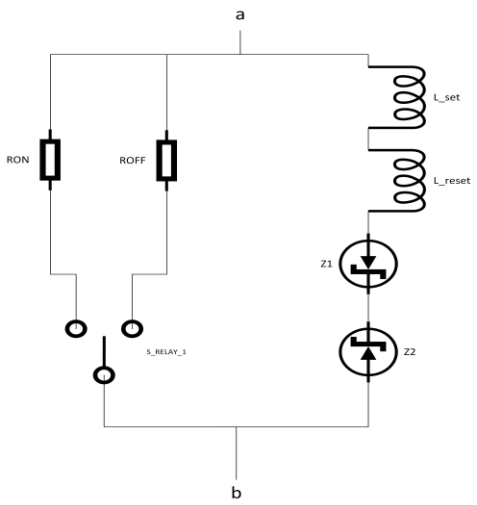


Figure 18. Emulator Basic Design

3.2.2. Complete Design

In the final design, just a Zener was used, with that the switching voltages are now asymetrics being SET voltage higher, in module, than RESET. With that, is possible to obtain a similar I-V curve than Knowm memristors. Finally, the selected resistances of R_{ON} and R_{OFF} were 10 and 100 Ω respectively.

Besides the emulator design, the manual switching skill was introduced to select the state of the emulator as well as a lit up pilot of state. Both circuits are separately sourced. [*Figure 19*] display the final circuit schematic.

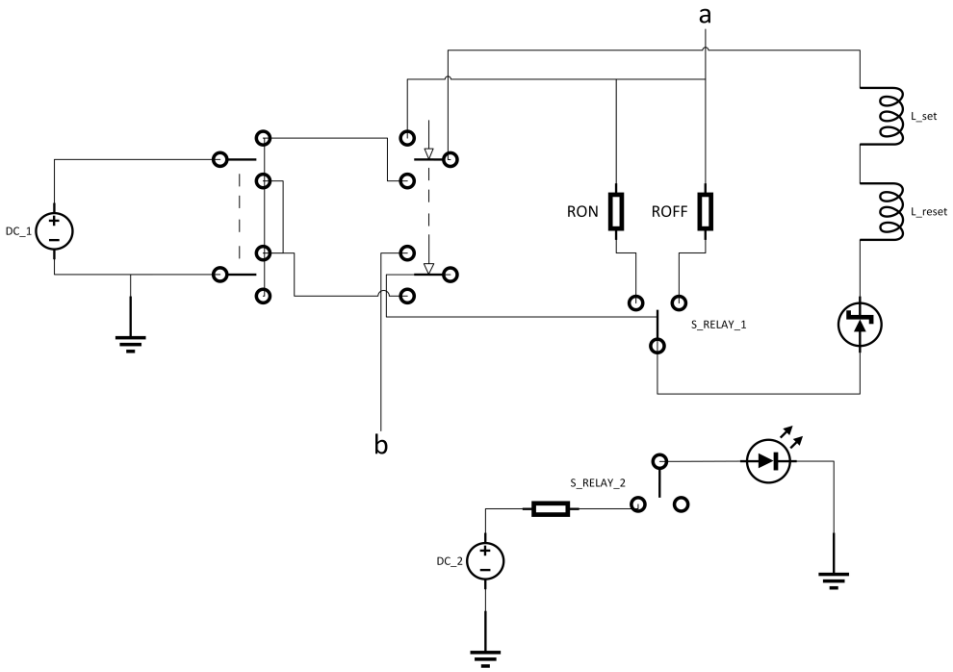


Figure 19. Emulator Complete Design

To design the manual switching, a double 3-way-switch was used to change the polarity of the power supply DC_1. Then with a double 3-way-push button execute the operation or remains in the basic circuit.

The pilot is always in active mode even if you push the button or not. In its design was used the second relay that is controlled with the same inductors. Both relays are inside the same PDIP package.

3.2.3. Connectors

This design was conceived as a modular system using a protoboard as motherboard. So it was necessary to add a pin-array as connectors with a step of 2.64 mm.

The circuit has six pins in total:

- Two as terminals of the emulator
- Two as power supply of the manual switch
- Two as power supply of the lit up pilot

3.2.4. PCB Design

The design of the PCB board have been developed using Kicad software. The main objective was decreas the dimensions of the emulator circuit, with that, was posible to

guarantee the device modularity. The board have been fabricated in the university at the docent laboratory. It dimensions are: 0.5 mm width tracks and 0.9 mm diameter via. Due to the not metalized vias, it has been necessary to insert thin cables through each one, creating now contact between both layers. Having contact between both layers, it has been possible to add ground planes. The final dimensions of the board have been 35.6 x 27.94mm.

Most of the component footprints have been designed manually with Kikad using the measures provided by the manufacturer.

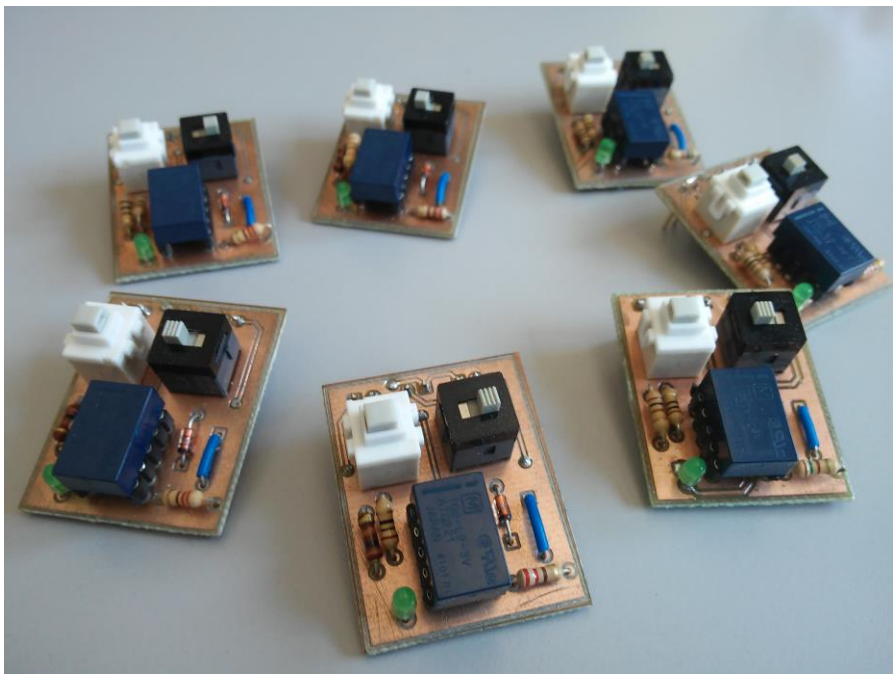


Figure 20. Seven-set memristive emulator circuits

3.3. Development of a Memristive Circuit able to Compute Arithmetic-logic operations based on IMPLY and Memristors

The purpose of the 7-memristor IMPLY circuit set-up was the development of some of the i4004 arithmetic-logic instructions following a nonconventional track meaning Memristors and IMPLY. Different from i4004, which has two 4-bit registers, this design will use 2-bit registers, and each bit is represented by an emulator.

As opposed to Intel i4004, in this design is possible to select where store the result and it is stored while is processed.

3.3.1. Sequences

Once the hardware was checked, the first thing to do was determine the instructions to perform. These instructions are: AND, OR, NAND, NOR, XOR, INC, DEC, R, ADD and SUB defined in [2]. Next step was obtain the operation sequences for each instruction [*Table 4*]. Finding them, was realized that seven memristors was necessary to perform them. Adding three extra memristors to the first memristors called workers.

Instruction	Sequence
R2 **AND** R1	False(W0), R20→W0, R10→W0, False(R10), W0→R10, False(W1), R21→W0, R11→W0, False(R11), W1→R11
R2 **OR** R1	False(W0), R20→W0, W0→R10, False(W0), R21→W0, W0→R11
R2 **NAND** R1	False(W0), R20→W0, R10→W0, False(R10,W1), W0→W1, W1→R10, False(W1), R21→W1, R11→W1, False(R11, W0), W1→W0, W0→R11,
R2 **NOR** R1	False(W0), R20→W0, W0→R10, False(W1,W0), R10→W0, W0→W1, False(R10), W1→R10, False(W0), R21→W0, W0→R11, False(W1,W0), R11→W0, W0→W1, False(R11), W1→R11
R2 **XOR** R1	False(W0,W1), R10→W0, R20→W1, R20→R10, W1→W0, False(W1), W0→W1, R10→W1, False(W0,R10), W1→W0, W0→R10, False(W0,W1), R11→W0, R21→W1, R21→R11, W1→W0, False(W1), W0→W1, R11→W1, False(W0,R11), W1→W0, W0→R11
R2 **INC** R1	False(W2,W0), R10→W0, R11→W0, W0→W2, False(W0,W1), R10→W0, R11→W1, R10→R11, W0→W1, False(W0), W1→W0, R11→W0, False(W1,R11), W0→W1, W1→R11, False(W1,W0), R10→W0, W0→W1, False(R10), W1→R10
R2 **DEC** R1	False(W1,W0), R10→W0, R11→W1, R10→R11, W0→W1, False(W0), W1→W0, R11→W0, False(R11), W0→R11, False(W1,W0), R10→W0, W0→W1, False(R10), W1→R10, False(W2,W0), R10→W0, R11→W0, W0→W2
R2 **R** R1	False(W0,W1), R10→W0, R11→W1, False(R10,R11), W1→R10, W0→R11
R2 **ADD** R1	False(W0,W1,W2), R20→W0, R10→W0, W0→W2, False(W0,W1), R20→W0, R10→W1, R20→R10, W0→W1, False(W0), W1→W0, R10→W0, False(W1,R10), W0→W1, W1→R10, False(W0,W1,R20), R11→R20, R21→W1, R20→R21, R11→W1, W1→W0, R21→W0, False(R11), W0→R11, False(R21,R20), W2→R20, R20→R11, W2→W0, W0→R21, R11→R21, False(R11), R21→R11, False(W2), W1→W2, W0→W2,
R2 **SUB** R1	False(W0,W1,W2), R10→W0, R20→R10, R10→W2, False(R10), W0→R10, False(W0,W1), R20→W0, R10→W1, R20→R10, W0→W1, W1→W0, R10→W0, W1→R10, W0→W1, W1→R10, False(W0,W1,R20), R21→W0, R11→W1, R11→R20, R21→R11, W0→W1, False(W0), W1→W0, R11→W0, False(R11,W1), W2→R11, W0→W1, W2→W0, R11→W1, False(R11), W1→R11, W0→R11, False(W0), R20→W0, False(R20,W1), R21→W1, W2→W1, W1→R20, False(W1), W2→W1, W1→R21, False(W1), R21→W0, W0→W1, False(W0,W2), R20→W2, W2→W1, W1→W0, False(W2), W0→W2

Table 4. Table of Sequences, result stored in R1

The sequences are controlled by a FPGA and a voltage driver. With the driver is possible get control from the FPGA's GPIOs to the voltages V_{SET}, V_{COND} and V_{CLEAR}, needed by IMPLY and NOT operations.

3.3.2. Required Devices

In addition to the seven emulators, has been necessary a device to control the operations of an instruction and a power supply.

3.3.2.1. Voltage Driver

First, was necessary a voltage driver. With it is possible to select which voltage apply to the emulators. Its design consist on a 7x3 matrix of photoMOS relay components.

These devices are just two p-MOS phototransistors interconnected by the drain and source of each one. Is possible to control them with the light of LED inside the package. This device is normally closed (low impedance). These devices has an internal impedance in the range of 30Ω. This parameter influence in the values of V_{SET}, V_{COND} and V_{CLEAR}.

Each photoMOS relay is controlled by a GPIO of a FPGA. The high value of the GPIO signal is 3.3 V. To reduce the consumptions of current, resistors in series have been connected in each GPIO.

Each row of the 7x3 matrix controls the voltage applied to each emulator. For that reason is necessary a software control that just put in low impedance one of these three components. In addition, is necessary to set in high impedance all the photoMOS relays before plug voltages V_{SET}, V_{COND} and V_{CLEAR}.

Voltages V_{SET}, V_{COND} and V_{CLEAR} are plugged by six banana connectors, being three of them it's respectively grounds. In the circuit, all grounds are interconnected. Each signal is connected to a column of the matrix 7x3.

This design was implemented in a PCB board [*Figure 21*]. The dimensions are again 0.5 mm width tracks and 0.9 mm diameter via, the holes are not metalized so it was necessary to insert thin wires and solder it to both layers.

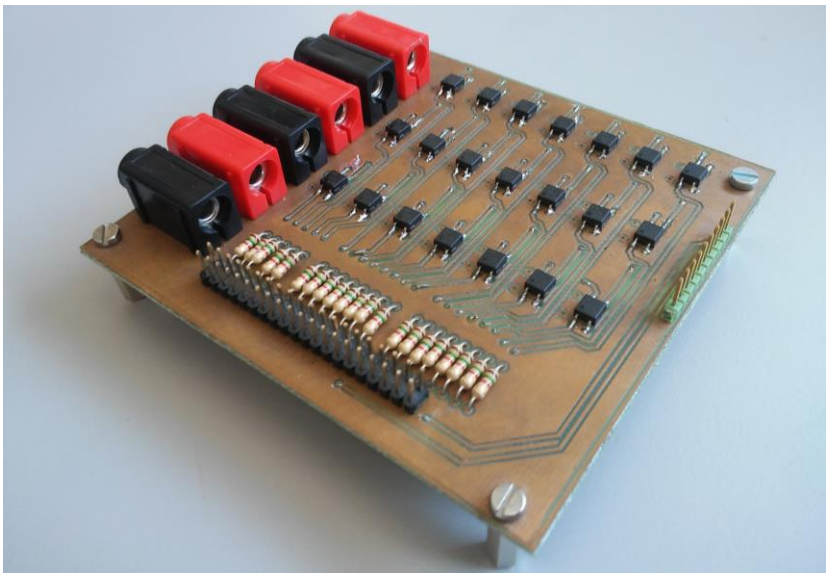


Figure 21. Voltage Driver, PCB Design

3.3.2.2. Altera DE2 development kit

The FPGA selected was the Altera DE2 development kit, which offer a high variety of peripherals. This device is used to control the instruction to execute using its peripherals.

The used peripherals are: five shift switches and two push button switches. With the shift switches, is possible to select the instruction to do. Finally, the instruction is triggered by using a push button. The last push button is the system reset, it consist in set-up to high impedance all photoMOS relays from the driver.

The FPGA is programmed with an algorithm posteriorly commented.

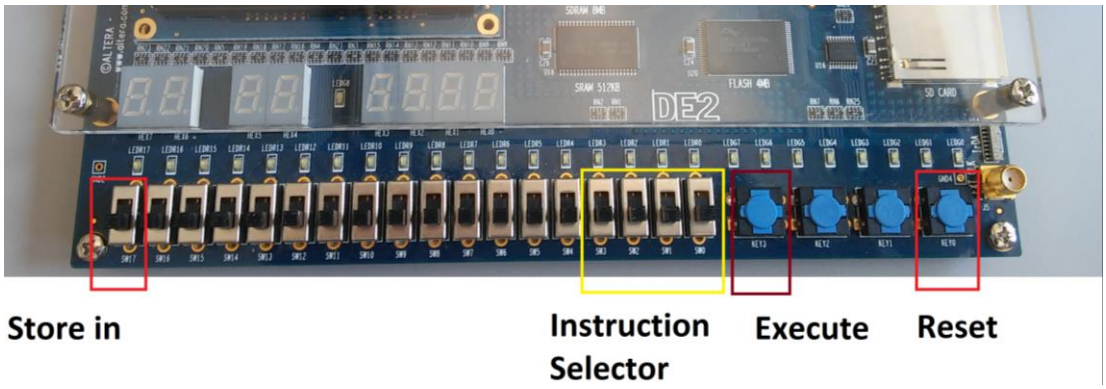


Figure 22. FPGA Peripherals

3.3.2.3. Power Supply

To set-up the voltages V_{SET}, V_{COND}, V_{CLEAR} and the required power supplies of the FPGA and emulator's manual switch and lit up pilots. It is necessary to use two power supplies model HM7042-5 fabricated by Hameg. Moreover, they are useful because it's current limitator. When current is higher than defined, automatically turns off.

3.3.3. Interconnection

As has been definen in section 2.4, the physical interconnection of memristors in IMPLY is the same for any of the desired instructions. As the resistance R_{ON} of the emulator is too small the major part of current will go through it to R_G, producing not enough potencial differential to produce a switching with a nominal value of V_{CLEAR}.

There were two options to solve the problem: increase the applied voltage V_{CLEAR} or reduce the resistance R_G. It was decided to use the second option. So since now, during a NOT operation the resistor R_G will be replaced by a short circuit. To do that, a solid state relay is used due to its lower internal impedance. The package is the same as biestable relays but, here there are just one inductor. Its operation is the same as a current relay, while the inductor is polarized resistor R_G is short circuited, if not the current flows through R_G. The inductor is controlled by a GPIO.

The interconnection of registers (R1 and R2) and worker (W0, W1, W2) memristors can be observed on [*Figure 22*]. The system is modular and uses the protoboard as motherboard. On the motherboard are also connected the necessary power supplies of emulator's manual switching and lit up pilot.

The voltage driver is communicated with the FPGA by a 20x2 pin array and with a 10x1 pin array with the motherboard. The 20x2 connector is connected to the GPIO_0 port of the kit and has the control of every photoMOS. On the other hand, the first seven pins of the 10x1

pin array controls the emulators, the 8th controls resistor R_G, the 9th is the 3.3V pin of GPIO_0 and the 10th is ground terminal.

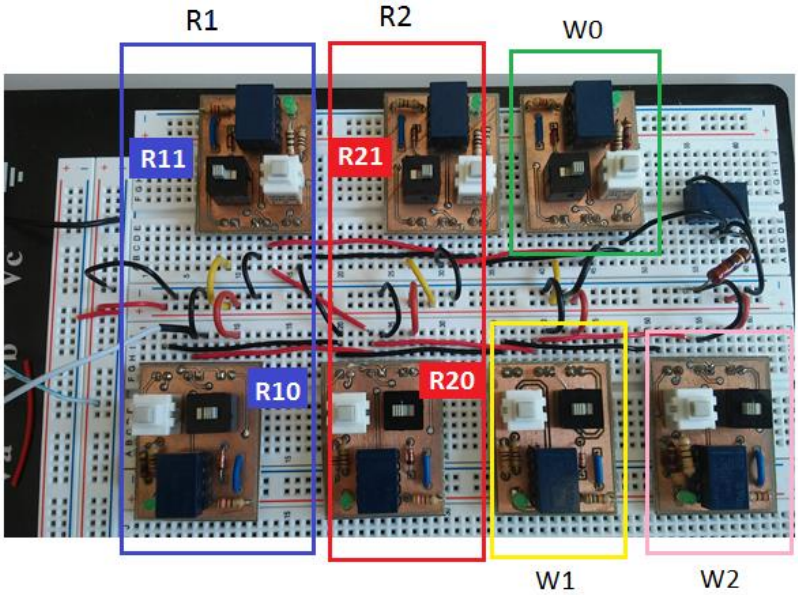


Figure 23. Emulator interconnection on Motherboard

3.3.4. Final Hardware Set

With the required devices achieved, was time to prove the system behaviour. With resistances R_{ON} and R_{OFF} with values 10 and 100 Ω, a resistor R_G with a value of 22Ω was selected conforming [Equation 4] and having in account the decision assumed on section 3.3.3.

After that, was necessary to select voltages V_{SET}, V_{COND} and V_{CLEAR}. The nominal switching voltages on the emulator's design were 7.5 and 2.5V for SET and RESET. Due to the internal impedances in photoMOS components as well as the insertion of R_G, it was necessary to increase these voltages in order to produce switching. The selected voltages were 10, 7.5 and -7 V for V_{SET}, V_{COND} and V_{CLEAR} respectively.

Finally, the final system design is displayed on [*Figure 23*].

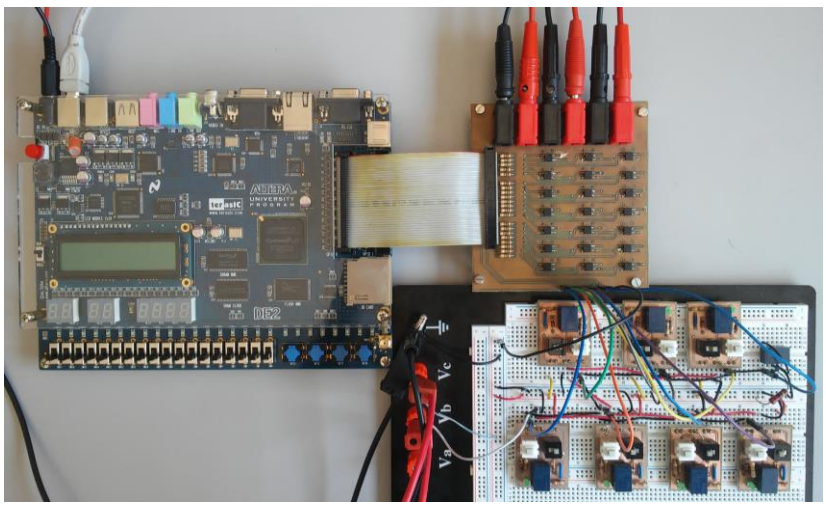


Figure 24. Final System

3.3.5. Sequence Controller Algorithm

The algorithm has been wrote in VHDL. This source code is used by the development kit of FPGA, Altera DE2. As input peripherals, it has the switches commented before.

Inside the algorithm exists three modules:

- **Frequency divider**. Its mission is divide the native FPGA clock frequency of 50MHz to 1 kHz. It is necessary because the relay's switching slopes.
- **Instruction detector**. This module gets the combinations of the shift switches [*Table 5*]. It will be reading until a raise edge on the set-button is detected. After that, it will transmit the instruction order. This order consist in five bits, the MSB define in which register store the result (low store in R1 and high store in R2), and the rest selects the instruction. While an instruction is running, it would not be possible execute another until it ends.
- **Sequence Trigger**. When this module receive the instruction order the instruction runs. Once is finished, the module send an instruction finished to the Instruction detector.

Instruction	**AND**	**OR**	**NAND**	**NOR**	**XOR**	**R**	**INC**	**DEC**	**ADD**	**SUB**
Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Table 5. Sequence code using shift switches

In addition, a reset bit can be used by pressing the reset push button. After a raise edge on this button is detected, all photoMOS relays remains in high impedance and sequence trigger would not execute any instruction until Instruction detector sends another instruction order.

3.3.6. Instruction Execution

Each instruction as an operation sequence. The FPGA execute them activating and deactivating GPIOs. As has been said, a sequence is created by IMPLY and NOT operations.

A IMPLY operation is performed by: 1st deactivating the GPIO which controls V_{COND} of the selected *p* emulator to set the voltage on it. 2nd deactivating the GPIO which controls V_{SET} of the selected *q* emulator to set the voltage on it. 3rd activing the GPIO which controls V_{SET}, and finally activing the GPIO which controls V_{SET}. It is possible activate and deactivate both signals together, but it is necessary high accuracy because V_{SET} can change the state of an emulator if V_{COND} is turned off.

By the other hand, NOT operation is performed by: 1st activating the GPIO which unplug resistor R_G, 2nd deactivating GPIO which controls V_{CLEAR} of the selected emulator, 3rd activating GPIO which control V_{CLEAR} and finally plugging again R_G.

4 Results

As main result a new design of electromechanical memristive emulator has been developed. Adapt it to other requirements is simple, just is necessary to change the device resistors and Zener diodes in order to parametrize R_{ON}, R_{OFF} and the switching voltages to SET or RESET the memristor.

Once the system is operative, is time to get the clock speed as well as the required time to execute an instruction and obtain its results.

Instruction	Time of execution (ms)	Number of pulses applied	Number of internal operations
AND	160	40	10
OR	96	24	6
NAND	192	48	12
NOR	256	64	16
XOR	352	88	22
R	96	24	6
INC	320	80	20
DEC	304	76	19
ADD	560	140	35
SUB	816	204	51

Table 6. Instructions Results

Has was said in section 3.3.5, the clock frequency of the FPGA is divided to a value of 1kHz. So the required time to perform an operation is this frequency divided by the number of steps in each operation, in this case 4. The clock speed is defined by 250 Hz.

Finally, [*Table 4*] shows the time performances of execution for each instruction at a clock speed of 250 Hz.

An article of this project has been submitted on the *2016 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2016)* organized in Athens, Greece [28].

5. Improvement Opportunities

5.1. Memristive Emulator

Actually this system is working as an ideal memristor. For some application might be interesting consider non-linear effects. An emulator of model VTEAM could be interesting to be implemented.

By other hand, the dimensions of the device can be reduced using SMD components. As in this component is just necessary use short-time pulses, it has been possible work with 250 mW resistors. If the emulator is exposed to longer periods, is necessary to work with power resistors.

5.2. Final System

If the FPGA has a high accuracy it could be possible to reduce the number of steps inside an operation to two. Doing that, the clock speed of the system could be doubled.

Another way to increase the clock speed is using a different photoMOS or a different device. This device has a delay produced by the switching. When the gate is activated there is a necessary time to change of state. This delay is in the order of 1 ms.

6 Budget

In this section will be introduced the project budget according to hardware, software and hours dedicated in the development of the project.

In the budget will be considered the availability of an electronic laboratory, including here the basic instrumentation, the two need power supplies and the IC tool.

5.3. Staff Costs

Description	Costs per hour (€/h)	Duration (h)	Taxes	Amount (€)
Project Development	8	400	IVA 21%	3200
Project director advices	20	30	IVA 21%	600
PCBs Creation	10	3	IVA 21%	30
			Subtotal	***3830 €***
			Taxes (IVA 21%)	804.30 €
			Total	**4634.30 €**

Table 7. Staff Costs

5.4. Hardware Costs

5.4.1. Emulator Circuit Costs

Component	Unit Price (€/u)	Quantity	Taxes	Amount (€)
Push Button	0,738	7	IVA 21%	4,27
Shift Switch	0,854	7	IVA 21%	4,94
LED	0,102	7	IVA 21%	0,59
Resistor 100Ω ¼ W	0,04	7	IVA 21%	0,23
Resistor 10Ω ¼ W	0,04	7	IVA 21%	0,23
Resistor 220Ω ¼ W	0,04	7	IVA 21%	0,23
Zener Diode 3.3V	0,011	7	IVA 21%	0,06
TQ Relay 3V	2,711	7	IVA 21%	15,68
Pin array 1x10 M-F	1,239	7	IVA 21%	7,17
Pin array 1x6 M-F	1,239	7	IVA 21%	7,17
			Subtotal	***40.57 €***
			Taxes (IVA 21%)	8.52 €
			Total	**49.09 €**

Table 8. Hardware Costs: Emulator Circuit

5.4.2. Voltage Driver Costs

Component	Unit Price (€/u)	Quantity	Taxes	Amount (€)
Banana Conector	0.64	6	IVA 21%	3.83
Pin array 2x20 M-M	1.92	1	IVA 21%	1.92
Pin array 1x10 M-M	1.239	1	IVA 21%	1.24
Resistor 1.2 kΩ ¼ W	0.04	21	IVA 21%	0,84
Photomos Relay	3.57	21	IVA 21%	74.97
			Subtotal	***84.72 €***
			Taxes (IVA 21%)	*17.79 €*
			Total	**102.51 €**

Table 9. Hardware Cost. Voltage Driver

5.4.3. FPGA Costs

Device	Unit Price (€/u)	Quantity	Taxes	Amount (€)
Altera DE2 Development Kit	199.85	1	IVA 21%	199.85
			Subtotal	***199.85 €***
			Taxes (IVA 21%)	*53.12 €*
			Total	**252.97 €**

Table 10. Hardware Costs: FPGA

5.5. Summary

Staff Cost	4634.60 €
Hardware Cost	404.57 €
Total Cost	**5039.17 €**

Table 11. Budget Summary

As is observed the major part of the total cost is payed to the staff, so the project is totally rentable and viable.

7 Conclusions

The Moore's Law ending is a fact we are facing, and is necessary to develop a new computing paradigm for the future. May be, nanowire FET's offers the possibility of extend the transistor's life during a while. For that reason is necessary to find a different device who acts as a logic switch.

The memristor, a new paradigmatic component, is the 4th passive two-terminal component that offers interesting characteristics. Theoretically, it has two defined states of resistance, so it could be used as a logic device. In addition, it is a non-volatile component, so it could be used as a memory too. Unfortunately on the practice, is observed a high variability to be overcome.

The emulator's development makes easier the physically verification of IMPLY. The emulator shows an ideal memristor behavior and it is easier to be customized changing some components.

With the development of the final system, the power of IMPLY was demonstrated. IMPLY and memristors are technologies which can be useful in the design of future computers based on computing in memory.

A new computing structure have been implemented by using the concept of material implication function and the developed quasi-ideal memristor emulator. We have implemented the practically complete set of arithmetic-logic instructions of the i4004. It can be seen that the structure of the architecture based on material implication is independent

of the set of selected instructions. It is the sequence of pulses what implements one instruction or other. This a new and interesting new computing paradigm.

The idea could be extended easily to registers of 4 or 8 bits. This first work could be a first step for a research field of computing with newemerging devices.

Bibliography

[1] G. E. Moore, "Cramming more components into integrated circuits," *Electronics,* vol. 38, no. 8, 1965.

[2] Intel Corporation, *4004, Single chip 4-bit, P-channel microprocessor,* 1967.

[3] M. M. Waldrop, "More than Moore," *Nature,* pp. 144-147, 2016.

[4] ITRS, *EMERGING RESEARCH DEVICES,* 2013.

[5] J. R. Powell, "The Quantum Limit to Moore's Law," *Proceedings of the IEEE,* vol. 96, no. 8, pp. 1247 - 1248, 2008.

[6] A. Todri, S. Kundu and P. Girard, "A Study of Tapered 3-D TSVs for Power and Thermal Integrity," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 21, no. 2, pp. 306 - 319, 2013.

[7] N. Muralimanohar, H. B. Yoon and N. P. Jouppi, "Multi-Level Cell Memory". United States of America Patent 20150162078, 06 11 2015.

[8] L. Chua, "Memristor: The missing Circuit Element," *IEEE Trans. Circuit Theory,* vol. CT, no. 18, pp. 507-519, 1971.

[9] D. B. Strukov, G. . S. Snider, D. R. Stewart and . R. S. Williams, "The missing memristor found," *Nature,* pp. 80-83, 2008.

[10] S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. C. Weiser and E. G. Friedman, *Models of Memristors for SPICE Simulations,* 2012.

[11] S. Kvatinsky, E. G. Friedman, A. Kolodny and U. C. Weiser, "TEAM: ThrEshold Adaptive Memristor Model," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS,* pp. 211-221, 2013.

[12] S. Kvatinsky, M. Ramadan, E. G. Friedman and A. Kolodny, "VTEAM - A general model for Voltage Controlled Memristors," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS,* pp. 1-5, 2014.

[13] R. S. Williams, "How We Found the Missing Memristor," *IEEE Spectrum ,* pp. 29-35, 2008.

[14] Knowm Inc, *Knowm Memristors,* Santa Fe, New Mexico, 2015.

[15] L. Chua, "Resistance switching memories are memristors," *Applied Physics,* vol. 102, no. 4, pp. 765-783, 2011.

[16] P. Pouyan, "Reliability-Aware Memory Design Using Advanced Reconfigurations Mechanisms," Barcelona, 2015.

[17] N. D. Mermin, Quantum Computer Science: An Introduction, Cambridge, 2007.

[18] I. G. and S. Liu, "Memory and information processing in neuromorphic systems," *Proceedings of the IEEE,* 2015.

[19] S. Hyun Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder and W. Lu, "Nanoscale Memristor Device as Synapse in Neuromorphic Systems," *Nano Letter,* vol. 10, no. 4, pp. 1297-1301, 2010.

[20] S. Hamdioui, L. Xie, H. A. Du Nguyen, M. Taouil, K. Bertels, H. Corporaal and H. Jiao, "Memristor Based Computation-in-Memory Architecture for Data-Intensive Applications," *Design, Automation & Test in Europe Conference & Exhibition (DATE),* pp. 1718-1725, 2015.

[21] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny and U. C. Weiser, "Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS,* pp. 2054-2066, 2014.

[22] K. Lehtonen and M. Laiho, "Stateful Implication Logic with Memristors," *IEEE/ACM International Symposium on Nanoscale Architectures,* pp. 33-36, 2009.

[23] S. Kvatinsky, A. Kolodny, U. C. Weiser and E. G. Friedman, "Memristor-based IMPLY Logic Design Procedure," pp. 142-147.

[24] Y. T. Xudong Fang, "Circuit analysis of the memristive stateful implication gate," *Electronics Letters,* pp. 1282-1283, 2013.

[25] M. Escudero, "Memristor 6," Barcelona, 2016.

[26] S. Kvatinsky and D. Fliter, "Technion - Israel Institute of Technology," December 2015. [Online]. Available: http://webee.technion.ac.il/people/skva/Memristor%20Models/Verilog-A/memristor%20model.txt. [Accessed 04 2016].

[27] D. Fliter and K. Talisveyberg, "Manual for Using Memristor Models," December 2011. [Online]. Available: http://webee.technion.ac.il/people/skva/Memristor%20Models/Manual.pdf.

[28] S. Zuin, M. Escudero-López, F. Moll, A. Rubio, I. Vourkas and G. C. Sirakoulis, "Experience on Material Implication Computing With an Electromechanical Memristor Emulator," *IEEE Symposium Series on Computational Intelligence,* 2016.

Glossary

ITRS International Technology Roadmap for Semiconductors

FPGA Field Programmable Gate Array

CNT Carbon NanoTube

MIM Metal Insulator Metal

NWFET Nanowire Field Effect

TFET Tunnel FETs

NEMS Micro/Nano-Electro-Mechanical Switches

SWD Spin Wave Device

ASL All Spin Logic

TEAM ThrEshold Adaptive Memristor

VTEAM Voltage ThrEshold Adaptive Memristor

ReRAM Redox RAM

Cbits Current bits

Qbits Quantum bits

CIM Computing in Memory

IMPLY Material Implication logic operation

NOT Not logic operation

SET Switching memristor from 0 to 1

RESET Switching memristor from 1 to 0

AND AND logic gate

OR OR logic gate

NAND NAND logic gate

NOR NOR logic gate

XOR XOR logic gate

INC Increment

DEC Decrement

R Rotate

ADD Addition

SUB Subtraction