UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
Zepler Institute for Photonics and Nanoelectronics

Experimental Demonstration of RRAM-based Computational Cells for Reconfigurable Mixed-Signal Neuro-Inspired Circuits and Systems

DOI: TBA

by

Georgios Papandroulidakis
MEng, MSc
ORCiD: 0000-0002-9203-2557

A thesis for the degree of
Doctor of Philosophy

November 2021

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
Zepler Institute for Photonics and Nanoelectronics

Doctor of Philosophy

Experimental Demonstration of RRAM-based Computational Cells for Reconfigurable Mixed-Signal Neuro-Inspired Circuits and Systems

by Georgios Papandroulidakis

Modern electronics drive a shift from distributed, cloud and/or mainframe computing towards the 'edge'. To realise this vision, we need access to hardware technologies that are both energy and scale efficient. During the last decade, the introduction of Resistive Random Access Memory (RRAM), also known as memristors, has fuelled interest in extending conventional circuits' capabilities. Specifically, their capacity to act as scalable, non-volatile, finely tuneable, electrically programmable resistive elements render them promising candidates for future computer architectures. RRAM technologies have been considered by many as a promising candidate for implementing reconfigurable neuro-inspired circuits and systems capable of processing data in both digital and analogue formats. Presently, there is no extensive study of the behaviour of such circuits when realised physically with real RRAM devices. Hence, there are ample opportunities for developing novel electronic circuits for reconfigurable mixed-signal data processors in silico.

This thesis explores the design, implementation and testing of in-silico data processors capable of mapping data from one information domain to another and enabling a mixed-signal data processing. Through this research, I am introducing RRAM-based circuit designs operationally validated through simulations with state-of-art RRAM device model and then practically implemented proof-of-concept designs of these hybrid RRAM-CMOS circuits on hardware. The hardware implementation and testing of low-complexity primitive RRAM-based circuits that can process information in the analogue domain due to the introduction of programmable RRAM devices is the main contributions through this project. In this work, findings are presented regarding the implementation and testing in hardware of a RRAM-based primitive Multiply-Accumulate circuit, RRAM-enhanced Threshold Logic Gate design and as well as larger circuits on these primitive circuits that are easily integrated into RRAM-based In-Memory Computing (IMC) and Near-Memory Computing (NMC) architectures.

Contents

List of Figures

List of Tables

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as: Papandroulidakis et al. (2018)
   Papandroulidakis et al. (2019b)
   Papandroulidakis et al. (2019a)

Signed:... Date:..................

Acknowledgements

Thanks to my family and friends for their support throughout my studies. Special thanks to my father Vasileios Papandroulidakis for his tireless support and trust in me which was very valuable and encouraging to me every time I needed it the most during my journey. I would also like to thank my wife Aleksandra Nazeraj for her crucial support without which I wouldn't be able to complete my PhD program. Furthermore, many thanks to my supervisors and colleagues for their tremendous help during my PhD. I would like to thank Prof. Themis Prodromakis, Prof. Geoff V. Merrett and Dr. Alexander Serb for their continuous support and helpful supervision throughout my doctoral studies. Additionally, many thanks to my PhD viva examiners, Dr. Basel Halak and Dr. Hadi Heidari, whose feedback was important and helpful.

To my wife Aleksandra

Definitions and Abbreviations

MIM	Metal-Insulator-Metal device topology
MOSFET	Metal-Oxide-Semiconductor Field Effect Transistor
pMOS	p-type Metal-Oxide-Semiconductor (Field Effect Transistor)
nMOS	n-type Metal-Oxide-Semiconductor (Field Effect Transistor)
TLG	Threshold Logic Gate
IMC	In-Memory Computing
MAC	Multiply and Accumulate
WTA	Winner Take All
WUC	Wake Up Circuit
RRAM	Resistive Random Access Memory
DRAM	Dynamic Random Access Memory
SRAM	Static Random Access Memory
NVM	Non-Volatile Memory
CMOS	Complementary Metal Oxide Semiconductor
CiM	Compute in Memory
LiM	Logic in Memory
NMC	Near-Memory Computing
PCM	Phase Change Memory
CwM	Compute with Memory
DCCML	Dual Clocked Current Mode Threshold Logic Gate
MCMTLG	Memristor-based Current Mode Threshold Logic Gate
CIAL	Coupled Inverters with Asymmetrical Loads
GND	Ground node connection in circuits and systems

Chapter 1

Introduction

1.1 Motivation

In recent years, hardware technologies are moving away from older methods of computing, such as mainframe computing, and instead are focusing on the new era of highly distributed computing, also known as edge computing. Towards materialising the added requirements for edge computing new design methodologies need to be found, methodologies that are both energy and scale-efficient. At the same time, in order to accommodate system designs capable of performing a variety of data processing applications efficiently, modern electronic systems are often required to be reconfigurable. Metal Oxide Semiconductor Field Effect Transistors (MOSFETs) and Complementary Metal-Oxide Semiconductor (CMOS) technologies cannot alone deal with the increasing computing needs of modern systems and thus potential solutions are sought in hybrid systems incorporating emerging technologies in computer architecture design, as discussed by Hamdioui et al. (2016). Towards accelerating computing, a few important operations have been selected as the main computationally primitive computing operations that complex algorithms and applications can be ”disintegrated” into, as discussed by Serb et al. (2018a). The observation that most computing-demanding algorithms can be decomposed into a large number of fundamental operations, such as multiplication, accumulation, comparison etc., drove the need to design specialised hardware towards accelerating these specific operations, with examples shown by Huang et al. (2016); Carlson et al. (2014); Shafiee et al. (2016), which resulted in the acceleration of larger systems and computer architectures that base their operation in these fundamental computing tasks. Through accumulated research effort, findings suggest that computational concepts from the biological neural networks can be adapted into artificial neuro-inspired circuits with positive results in the acceleration of fundamental computer arithmetic and logic operations, as showcased by Ielmini and Wong (2018). Many such neuro-inspired computing circuits and systems showcased that massively parallel computing structures and networks of simple computing components can be an important design paradigm for approaching accelerator designs as suggested by many researchers including the work by Furber (2016); Indiveri and Liu (2015).

Most of the neuro-inspired circuits and systems suggested over the years are based on conventional MOSFET and CMOS technologies that defined computers for more than 40 years. One of the earliest examples that showcases a simplified Artificial Neural Networks (ANNs) fitted for computer logic can be found in the work of McCulloch and Pitts (1943) where the concepts of the primitive perceptron circuit and perceptron-based networks were introduced. More recent implementations suggest artificial synapse-neuron circuits (the basic computing node of ANNs) with analogue output with some relevant examples including the results showcased by Anderson et al. (1992); Mead and Allen (1991); Douglas et al. (1995); Indiveri (2001); Indiveri et al. (2006). One family of primitive circuits for MAC, the neuro-inspired artificial neuron category of circuits, has been evolved over the last few decades to become one of the most important hardware solutions to perform massively parallel digital or analogue Multiply-Accumulate (MAC) operations, a cornerstone arithmetic operation highlighted by Indiveri et al. (2011); Indiveri and Liu (2015); Payvand et al. (2018); Dara et al. (2013); Bobba and Hajj (2000); Papandroulidakis et al. (2018).

Towards building better hardware accelerators for edge computing, that can process data in a massively parallel manner, new computing methodologies have been developed, i.e. the design concepts showcased by Kang and Shanbhag (2016); Zhu et al. (2013); Zhang et al. (2020b); Gallo et al. (2017). In recent years, the development of novel big data hardware accelerators has been extensively studied, e.g. by Santoro et al. (2019); Jeloka et al. (2016); Gonugondla et al. (2018), and led to the rise of the modern post-von Neumann design paradigm known as In-Memory Computing (IMC). IMC is based on the use of memory systems that can simultaneously be employed as processing units (for processing data in place). In such systems, specific logic functions can be performed with the data stored inside the same memory enabling a logic and memory co-location computing concept to be realised. The fundamental concept of IMC is based on the memory and logic co-location circuits, another neuro-inspired technique discussed by many researchers including Seshadri et al. (2017); Li et al. (2017); Kim et al. (2018); Yin and Jiang (2020).

Recent breakthroughs indicate that some emerging technologies can be used alongside conventional MOSFET/CMOS-based circuits towards providing new capabilities in computing systems, e.g. as suggested by Serb et al. (2017); Bayat et al. (2018); Danial et al. (2018, 2019). More specifically, advances in emerging memory technologies introduce a new component for future computer circuit design. Memristor (abbreviation from memory-resistor) devices, also known as Resistive Random Access Memory (RRAM), are two-terminal, tuneable, non-volatile, nanoscale resistive memory devices.

RRAM has many traits over conventional memory, such as the capability to store multi-bit information per single device, e.g. as shown by Stathopoulos et al. (2017); Sebastian et al. (2018, 2020). Additionally, as highlighted by Sebastian et al. (2020); Zhang et al. (2020b) and other researchers, RRAM can be integrated to implement computationally-capable memory-based systems with low power consumption, an important advantage for large MAC-based ANN systems implemented as edge computing solutions.

Given the numerous advantages of RRAM device technologies, much effort has been dedicated to the design of novel post von Neumann computer systems, with some examples shown by Dastanova et al. (2018); Kvatinsky et al. (2012); Papandroulidakis et al. (2017, 2018). The usage of memory-centric topologies (such as crossbar arrays), as suggested by Zha and Li (2017); Chakrabarti et al. (2017), can provide a very efficient topological organisation of memristor devices and thus memristor circuits that can be employed to accelerate MAC operations. Towards covering all the requirements for a new generation of accelerators for big data processing and solve the limitations conventional technologies impose, highlighted by Mutlu et al. (2015), hardware solutions that employ memristors/RRAM devices have been used to develop novel IMC systems. As shown by Zha and Li (2017), hybrid RRAM-enhanced IMC systems can be employed to better capture the basic principles of biological neural-systems for in silico implementations that are capable of performing low-power logic operations inside a memory architecture. This can result in novel systems that showcase competitive traits against conventional accelerator designs. The advent of emerging memory technologies, such as memristor/RRAM, provided new opportunities to develop nano-electronic programmable logic fabric that is power and area efficient, with similar hybrid IMC-based concepts being tested, for example, by Hu et al. (2016b); Kumar Maan et al. (2016); Zha and Li (2017); Chakrabarti et al. (2017).

Naturally, as presented by Ielmini and Wong (2018); Serb et al. (2018b), with the advent of RRAM devices there has been a great interest in novel circuits designs that attempt to introduce solutions inspired by biological neural networks, i.e. using RRAM devices to emulate biological synapses in hardware ANNs. To that end, emerging memory technologies, including RRAM, is considered a catalyst for implementing analogue programmable inference engines by efficiently emulating synaptic weights as well as initiating a next generation of analogue and mixed-signal computer architectures at the nano-scale, for example, novel computing systems such as the one showcased by Gallo et al. (2017). Hence, it is envisioned that RRAM-based (also mentioned as memristive) inference engines can be used for performing analogue or mixed-signal reconfigurable classification operations in silico. This can in principle be achieved via arbitrarily shaping the classification decision boundary of perceptron units through tuning individual memristive states, as shown by Serb et al. (2018a). Logic-wise, this means that it is possible to design circuit concepts that are applicable for dealing with both analogue and digital signal processing modes. Presently, there is no extensive study of the behaviour of classification circuits realised physically with state-of-art memristor devices, providing ample opportunities for a PhD programme in this area that can contribute to the aforementioned needs of modern electronics.

1.2 Research Aims

The aim of this PhD is to build a RRAM-enhanced reconfigurable computing system capable of performing neuro-inspired primitive classification operations. For this purpose, I am following the main principles of IMC design paradigms towards being capable of integrating the RRAM and CMOS technologies into novel IMC circuits and system that are still compatible with the general existing accelerator design methodology. The main concept studied, explored and then materialised through hardware experimental proof-of-concept implementations and additional simulations in this thesis is the design of primitive circuits capable of fundamental computer operations such as multiplication, comparison, classification etc. aimed for integration into IMC architectures. The systems employ the naturally analogue RRAM technology, as discussed and presented by Stathopoulos et al. (2017); Sebastian et al. (2020); Ielmini and Wong (2018), to increase the benefits of IMC-related circuit in performing reconfigurable mixed-signal logic operations, with the main part of the computing being performed in the analogue domain and the inputs and/or outputs being digital. The bigger picture is the implementation of a reconfigurable sea-of-gates system, centred around a RRAM-based memory system, where each memory-based gate is based on the aforementioned functionality of programmable converting logic, an important step towards future field programmable mixed-signal arrays.

The main concept investigated in this work is the neuro-inspired concept of memory and logic co-location computing which is the baseline concept for the area of in-memory computing (IMC). The main objective is to physically implement and test reconfigurable RRAM-based hybrid primitive circuits that base their operation in low-complexity but powerful computing operation. The main goal of this thesis is the introduction of novel IMC circuits and systems for implementing low-complexity fundamental logic operations, such as multiply and accumulate (MAC) and analogue comparison operations. I am testing the functionality of the proposed designs through the practical implementations of the circuits using real RRAM devices alongside discrete conventional electronic devices and circuits. Through these implementations I am showcasing the capabilities of the specific RRAM technology under test in primitive neuro-inspired circuits, thus extrapolating the computational capabilities of building larger systems that employ such primitive hybrid circuits.

Furthermore, an important trait of RRAM-based circuits can be considered the use of RRAM devices as the building blocks of associative computing memories, hence enabling the configuration of generalised analogue Look-Up-Tables (LUTs) for IMC accelerators. By studying and testing low-complexity hybrid RRAM-MOSFET circuits and circuit networks, I am showcasing the potential capabilities in introducing such novel type of computationally-flexible data conversion inside memory (with regards to IMC implementations). Hence, the exploitation of this converting logic (CL) concept, based on neuro-inspired designs, with some of the earliest ones showcased by McCulloch and Pitts (1943); Bobba and Hajj (2000), alongside the RRAM technology, e.g. as discussed in the work of Chua (2014), are particularly interesting for introducing novel and computationally flexible systems. Additionally, the exploitation of the naturally analogue computing performed by the RRAM devices is employed to implement circuits that can naturally convert information between different domain (analogue or digital) and thus mitigating the need for additional conversion circuitry in some applications. Through this scope, I investigate multiple different configurations of low-complexity RRAM-based circuits to implement and test a variety of operational modes. Additionally, the circuit designs and implementations of this work are used to complement the existing RRAM-based circuit designs by adding more designs of reconfigurable primitive circuits. Hence, I aim at providing novel solutions that supplement the existing and well-documented digital-to-digital and analogue-to-analogue memristor-based logic schemes, with some of the different implementations being presented by Serb et al. (2018a); Kvatinsky et al. (2011); Vourkas and Sirakoulis (2016), thus providing novel methods of manipulating data in-place using RRAM-based logic.

1.3 Objectives

Towards providing evidence on the potential IMC performance improvements by introducing analogue RRAM technologies for the design and implementation of novel neuro-inspired memory and logic co-location computing, which are the basis of IMC systems, I explore the computational capabilities of hybrid RRAM-MOSFET circuits and systems. An important objective is to physically implement and test reconfigurable RRAM-based hybrid primitive circuits that base their operation in low-complexity but powerful computing operation. The operational validation through experimental demonstration of the RRAM-based primitive neuro-inspired data processors is important for future hybrid IC design since such circuits and systems are relatively under-documented with regards to the research area of hardware realisation and measurements. By providing novel experimental measurements of RRAM-enhanced neuro-inspired circuit that can be programmed on-the-fly to alter the information domains used as input/output I am showcasing novel findings that support the future implementations of hybrid ICs for IMC-based acceleration. This could potentially be employed towards creating a hybrid reconfigurable computer architecture for performing mixed-signal data processing. Through the findings of this thesis, I am adding practical knowledge to the area of RRAM-based neuro-inspired circuitry. Through the findings of these experiments, an important trait of RRAM-based circuits can be considered the use of RRAM devices as the building blocks of analogue reconfigurable associative computing memories. Based on the findings of this thesis, the concept of interpreting such circuit designs as a form of reconfigurable associative memory for performing information mapping between different data domains is reinforced. By experimenting with simple RRAM-MOSFET circuit networks, I am showcasing the potential capabilities in introducing such novel type of data conversion inside IMC architectures.

More specifically, I experimentally demonstrated several versions of the RRAM-based neuro-inspired circuits for data processing. The main effort can be found in the implementation and testing of in silico associative analogue computing memories and primitive classification blocks in the form of a 1T1R memory array and a current-mode TLG design, respectively, using state-of-art RRAM technology ($Pt/TiO_x/AlO_x/Pt$ RRAM devices proposed by Stathopoulos et al. (2017)) as programmable artificial synaptic weights. I designed and implemented in simulation and in hardware a RRAM-based mixed-signal MAC circuit (shown in Chapter 3) and a RRAM-based TLG which is performing the main computing in the analogue domain (i.e. analogue RRAM-based multiplication and accumulation in current mode) while the classification result is a binary/digital decision (shown in Chapter 4). An extension of the MAC and TLG circuits is showcased in Chapter 5 where the different modes of operation of a combined RRAM-based MAC-TL circuit are explored through the implementation and testing of mixed-signal circuit modalities that affect the way the circuit receives inputs from its environment and the way it decides how to classify this data. Additionally, further findings that highlight the computational flexibility of the RRAM-based circuits are investigated through the design and test in Cadence Virtuoso's Spectre simulation of proof-of-concept Winner Take All (WTA) and Wake Up Circuit (WUC) systems that base their operation on the proposed primitive RRAM-based MAC circuit and simple comparator circuits (such as those implemented for the proposed RRAM-based TLG).

The reconfigurable primitive RRAM-based MAC circuit that is showcased in Chapter 3 was presented in two international conferences (ISCAS 2019, Papandroulidakis et al. (2019a), and MEMRISYS 2019, please see Section 1.4). An investigation of a RRAM-based MAC circuit that performs MAC operations in the analogue domain and its hardware experimental realisation of that circuit is an important step for this study (shown in Chapter 3). The same RRAM-based MAC circuit is tested in Cadence Virtuoso's Spectre simulation as a programmable RC (RRAM-Capacitor) delay component for a proof-of-concept Winner Take All (WTA) network. The proposed RRAM-based

Winner-Take-All (WTA) system is competitive with existing state-of-art designs showcasing 300fJ energy consumption and 1ns delay per operation. Furthermore, the detailed implementation of the RRAM-enhanced TLGs, presented in Chapter 4, were disseminated in two international conferences (ISCAS 2018, Papandroulidakis et al. (2018), and MEMRISYS 2018, please see Section 1.4) and published in a journal (IEEE TCASI, Papandroulidakis et al. (2019b), please see Section 1.4). The findings on the RRAM-based TLG design proposed in this thesis showcase a competitive power dissipation of 1.79uW and a very fast operation of 0.12ns decision time (faster than existing state-of-art RRAM-based TLGs) while simulated in Cadence Virtuoso using 65nm MOSFET technology node and the RRAM model proposed by Messaris et al. (2018). The theory and design methodology derived by the analysis of the showcased RRAM-MOSFET circuits organised as hybrid memory arrays were presented in an international conference (ICEIC 2020, please see Section 1.4). Through the findings of this thesis, I contribute to the existing body of literature regarding memristor logic techniques and provide new methodologies of performing memory and logic co-location computing using RRAM.

1.4 Contributions of this work

This sections is referencing all the relevant contributions of the thesis until the date of submission. The research conducted throughout this PhD programme resulted in a journal publication and many presentations in international conferences. The list of publications (journal articles and proceedings in international conferences) is as follows:

Journal:
1. **G. Papandroulidakis**, A. Serb, A. Khiat, G. V. Merrett and T. Prodromakis, “Practical Implementation of Memristor-Based Threshold Logic Gates”, in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 8, August 2019

Conference paper:
1. J. Szypicyn, C. Papavasiliou, **G. Papandroulidakis**, A. Serb, S. Stathopoulos, Geoff V. Merrett and T. Prodromakis, ”Reconfigurable Memristor Integrated Circuits”, International Conference on Electronics, Information, and Communication (ICEIC2020), 2020

2. **G. Papandroulidakis**, A. Serb, A. Khiat, Geoff V. Merrett and T. Prodromakis, “Practical Implementation of digital in-analogue out memristor-based logic circuit” in International Conference on Memristive Materials, Devices and Systems (MEMRISYS) 2019, 2019.

3. **G. Papandroulidakis**, L. Michalas, A. Serb, A. Khiat, Geoff V. Merrett and T. Prodromakis "A Digital-In-Analogue-Out Logic Gate Based on Metal-Oxide Memristor Devices", in IEEE International Symposium on Circuits and Systems (ISCAS) 2019, 2019

4. A. Serb, **G. Papandroulidakis**, A. Khiat, and T. Prodromakis, "Plane-Splitting Logic Techniques using Hyrbid CMOS-Memristor Circuits and Systems," in International Conference on Memristive Materials, Devices and Systems (MEMRISYS) 2018, 2018.

5. **G. Papandroulidakis**, A. Khiat, A. Serb, S. Stathopoulos, L. Michalas, and T. Prodromakis, "Desing and Practical Implementation of Memristor-based Threshold Logic Gate," in International Conference on Memristive Materials, Devices and Systems (MEMRISYS) 2018, 2018.

6. A. Serb, **G. Papandroulidakis**, A. Khiat, and T. Prodromakis, "Processing big-data with Memristive Technologies: Splitting the Hyperplane Efficiently," in IEEE International Symposium on Circuits and Systems (ISCAS) 2018, 2018.

7. **G. Papandroulidakis**, A. Khiat, A. Serb, S. Stathopoulos, L. Michalas, and T. Prodromakis, "Metal Oxide-enabled Reconfigurable Memristive Threshold Logic Gates," in IEEE International Symposium on Circuits and Systems (ISCAS) 2018, 2018.

Chapter 2

Memory and logic co-location circuits for in silico classifiers

In this section, I am going to present the fundamental theory and circuit design methodologies required for the design and implementation of RRAM-based reconfigurable neuro-inspired logic circuits. More specifically, special important cases of neuro-inspired computing, usually taking the form of the emerging In-Memory Computing (IMC) computer architecture, will be investigated and the concepts of memory and logic co-location primitive gates, mainly in the form of Threshold Logic Gates (TLGs), will be highlighted. The important connections of emerging RRAM technology as a strong candidate for synaptic emulation circuits will be analysed and their consequent connection with ANNs and neuro-inspired computing will be showcased and discussed with specific circuit design examples. Through this background analysis, I am showcasing the fundamental ideas that will lead on designing and understanding the behaviour of a hybrid CMOS-RRAM memory and logic co-location circuit for developing novel IMC paradigms.

2.1 Circuits and Systems for Data Processing

One of the cornerstones of computing is the data processing, thus how data are manipulated through the use of logic circuits towards providing some meaningful results as the output response of these circuits. For the study of electronics, a data processing engine can be considered any form of circuit capable of manipulating an input to produce an output useful for the cascading circuits. Generalising this concept, we can assume that that the data processing engine maps a signal from one form to another, thus it can be used as a sort of computational bridge that connects different interpretations of data formats. Although most modern electronics in computers are manipulating binary data for reinforcing their operation against noise, the analogue signals processing remains a necessary and important area of computing for many applications, with an example shown in the work by Serb et al. (2018a,b). The introduction of novel solutions for computing in analogue or mixed-signal with low power dissipation and small chip area, as shown by Serb et al. (2018b), is rejuvenated by the need to perform massively parallel digital operations in accelerator systems, operations that can be reduced in number if we assume an analogue data processing domain. This is particularly interesting with the rise of emerging technologies, such as RRAM (and other memristor-like technologies discussed by Chua (2014); Strukov et al. (2009)), that have the potential of becoming the catalyst in the design of novel energy- and area-efficient circuit designs for processing in fully digital, fully analogue or mixed signal at the nano-scale.

One of the most widely documented data processing operation is the data classification which can be considered a fundamental processing operation found in most Machine Learning (ML) applications implemented by ANNs. Data classification may be implemented in several specific forms, but all of these possible implementations involve the identification of an input and its appropriate mapping into the output domain space. For example, even a simple and conventional Boolean AND gate can be seen as splitting a 2-dimensional binary space into two categories, the input stimulus that result in logic ′0′ and the input stimulus that result in logic ′1′. Similarly to the simple Boolean AND gate, a typical 2-input artificial neuron uses its weighted input sum and threshold to linearly separate its input space (2-dimensional continuous space) into two output domains, as discussed by Serb et al. (2018a). It is worth noting that, in general, simple Boolean digital logic gate, such as AND, OR, NAND and NOR gates, can be interpreted as a special cases of a quite simple artificial neural network with unity weights where a threshold, defines by the configuration of the CMOS implementation per logic gate case, linearly separates the input space. This fundamental concept can of course be generalised to an input space of higher dimensions, where the widely used 2-input digital logic gates become digital CMOS-based primitive classification gates. The operation of these gates can be directly related to simplified and restricted artificial neurons and thus ANNs, where all the input weights are equal to a reference unity weight and classification operation depends on the threshold level. The property of linear separability is maintained by all potential gate designed for low-complexity and computing of fundamental operations, thus similarly to simple neuron model (i.e. perceptron) a single classification gate cannot compute functions such as XOR but instead an interconnected network of multiple layer of such gates is required to achieve that.

Introducing programmable non-binary weights at the inputs of simple classification gate turns them into a form of more configurable ANNs by allowing the programming of appropriate synaptic emulation components, with the important category of these primitive gates being well-documented for conventional technologies implementations, as highlighted by Beiu et al. (2003b). In most cases, the input space remains binary (i.e. a digital input vector is used) since most implementations are aimed for compatibility with existing conventional digital computer architectures, but the contribution of each input is moderated by a continuous weight, with a recent example of such a hybrid design showcased by Papandroulidakis et al. (2018). For the case that the input space is continuous (or in general multi-bit) we obtain a typical perceptron with a step activation function, one of the first ever such designs was proposed by McCulloch and Pitts (1943).

A wide variety of approaches exist for implementing data mapping and classification operations in hardware. These range from gates and digital system (e.g. Field Programmable Gate Array (FPGA)) analogue computing blocks, such as MOSFET-based fuzzy gates by Merrikh-Bayat et al. (2011); Merrikh-Bayat and Shouraki (2013); Tarkhan and Maymandi-Nejad (2018), and many other designs and computing solution in between. Recently, with the rise of 'edge' computing, Internet-of-Things computing networks and distributing sensing system monitoring patients, building, local climate, etc. there is a higher than ever demand for power and area efficient hardware solutions capable of performing a variety of data processing away from large computing systems. This rising demand in novel solutions is usually divided among fixed-structured Application Specific Integrated Circuits (ASICs) and computationally flexible and programmable FPGA-based designs based on the specific demands of each application. In cases where area and power are quite limited (such as in the case of 'edge' computing discussed by James et al. (2014); Krestinskaya et al. (2018a)), ASIC designs are usually preferred. On the other hand, in cases where the hardware need to be programmable and reconfigurable at the hardware-level due to irregular computing demands (thus there are requirements for different computing data-paths to be accelerated), FPGAs are the key to build stable and long-lasting data processing infrastructure that reduces the costs of re-deployment and replacement of systems, as shown by Gaillardon et al. (2015); Kuon and Rose (2006); Vestias and Neto (2014). Although the exact computer architecture that would be preferable in most cases is still under investigation, with the increase in maturity of emerging technologies, such as RRAM devices, unique opportunities arise where the design and implementation of both power-efficient and reconfigurable systems can co-exist in the same IC design. In the next section (Section 2.2), I am going to discuss some of the most important system that are consider ideal candidates for enabling the design of such an attempt for novel data classifiers at the ”edge”.

2.2 Computer logic inspired by biological neural processing

As discussed earlier, much of computing can be deconstructed into a set of fundamental operation many of which can be considered neuro-inspired since the same fundamental logic seems to be at the core of the inner workings in biological neural networks as well. Thus, an important aspect of designing novel hardware can be found through the integration of ANN-related concepts into state-of-the-art computer architectures, with a recent example shown by Wan et al. (2019). Through the examination of the neuro-inspired design concepts, we are able of developing important computer circuits and systems that base their design and functionality on what we can observe as data processing in biological neural networks. Furthermore, in many cases, neuro-inspired circuits and systems can provide better solutions in terms of area and/or power compared to their conventional counterparts.

The importance of neuro-inspired computing concepts is manifold as they present our best idea of the basic operational principles of logic circuits co-located with memory modules. Neuro-inspired designs position themselves as a true alternative of the conventional von Neumann architectural since they can mitigate or even eliminate the von Neumann computing-memory bottleneck. Through these novel computing concept not only we can gain significant performance compared to more conventional architectures but also we can provide novel solutions in organising a reconfigurable computing substrate based on primitive circuits, another design idea derived from biological neural networks. More specifically, many neuro-inspired designs are based on relatively simple primitive computing cells that can change their operation/behaviour depending on their environment and operational state of the system, thus defining a guiding design methodology for conceptualise and implementing reconfigurable computers. These aforementioned two focus points are the main reason that a lot of attention has been given on designing computers with technology adapted to emulate the functionality of biological neuron activity, as proposed by McCulloch and Pitts (1943).

With regards to the design of primitive computing cells for neuro-inspired systems a detailed study of the synaptic-neural behaviour is required. Neural activity of integrate-and-fire is a primitive processing operation that is defined as the fundamental function many consider as the enabling process behind the observable brain-related processes. In the last few decades, an active scientific research effort is aimed towards designing and implementing circuits for performing brain-inspired computations, thus building artificial brains or (the more computation-oriented term) artificial neural networks (ANNs). Integrate-and-fire (IF) operation can be considered as the process of accumulating information from an input signal (e.g. voltage in a capacitor component) and when a specific level of accumulation is reached then an event is triggered (e.g. the ”firing” of a neuron, thus the transmission of a voltage signal). Usually the concept of IF operation is connected with synaptic behaviour where the input signals vector is passed through a set weights that map the input information into an appropriate processed signal that is then used to as input to the actual neuron system. The main computation performed with the synaptic behaviour is that of the fundamental computer arithmetic operation of multiplication, since we consider that the input signal is multiplied by a corresponding contribution parameter (in this case the corresponding input weight). Thus, in general, implementations of synaptic emulator circuits are centred around the multiplication of signals. For the case of multiple input signals connected to the input of a neuron through multiple synapses (a design inspired from the observations in biological neural networks), each input has different contribution to the final accumulated sum which controls the behaviour of the neuron based on the mostly simple event of the total weighted input sum surpassing or not a specific threshold value (also referred to as bias). Early studies that considers the usage of these simplified observations on how the synapses and neuron work as a form of primitive circuits can be traced to the important work of McCulloch and Pitts (1943). An example of a thresholding function can be considered the case where a continuous input signal is fluctuating, thus resulting is passing through the line defining the threshold value. When the signal is larger than the defined threshold value then, by convention of this example, the output is set to logic '1' while in different case, thus while the signal is below the line defining the threshold, the output is set to logic '0'.

The well-documented effort in designing modern electronic circuits and systems is usually focused to the implementation of efficient hardware solutions through optimising networks of basic Boolean operations, as exhibited in the results of Leshner et al. (2010); Neutzling et al. (2018). Although some circuit design methods, such as CMOS, have many advantages that enabled them to be the catalyst for much of the current electronics, limitations imposed by fundamental design techniques create need for novel solutions in terms of computer architecture design as well as in terms of technologies, as discussed by Hamdioui et al. (2016). An important requirement for today's electronic systems is the search towards scalable hardware architectures. Unfortunately, as discussed by Hamdioui et al. (2016); Wilkes (1995), this search for novel design techniques that could lead to better scalable systems is hindered by the limitation of the CMOS-centric circuit design tools as well as due to the physical limitations that the MOSFET devices are imposing upon the conventional computer architectures. Many consider that towards solving the scalable circuit design problems further solutions among the lines of neuro-inspired design must be found, with examples shown by Hamdioui et al. (2016); Serb et al. (2018a). Similarly to what can be observed in biological brains, novel computers could be based upon a primitive functional computing structure corresponding to an equivalent synapses-neuron emulation computing model. A system based on this design would be implemented mainly through a large-scale connection of multiple primitive neuron components through emulated synaptic weights, a concept central to this category of circuits and systems from the initial concept till recently, as shown by McCulloch and Pitts (1943); Maan et al. (2016). The primitive neuron circuits communicate through electronic signals that are mapped from one information domain to another through the synaptic weights network that exist between the communicating circuits. In the simplest cases, this mapping can be performed through the use of resistive networks to emulate synapses with each synaptic weight imposing a different resistance to the current that flows between the communicating neurons, as exhibiting in the recent work by Papandroulidakis et al. (2018). The above-mentioned concepts can enable the design of a highly scalable and massively parallel architecture inspired by the computational template found in biological neural networks. Such designs have been preliminary tested in many forms mainly as different implementations of ANN-related architectures, as discussed by Maan et al. (2016); McCulloch and Pitts (1943); Santoro et al. (2019).

Deeply embedded within the concept of neuro-inspired computing and ANN implementations is data classification. The idea of classification is a cornerstone of the ANN computing and can easily be quantised to describe the equivalent threshold logic (TL) operations found in digital electronics logic (thus logic that is based in the comparison of an input with a threshold level). As discussed in the previous section (see Section 2.1), even basic logic functions, such as 2-input AND logic gate, works by creating two clusters of possible outputs (classified input data). This is similar to the operation of a typical 2-input neuron which uses its weights and activation function threshold to separate its input space into two output data domains of valid clusters, as shown by Bayat et al. (2018). Notably, the AND gate can be interpreted as a special case of a neural network where an appropriate threshold line linearly separates the input space. The same applies for multi-input OR, NAND, NOR gates as well as for all the majority logic gates. These gates are also known as threshold logic gates (TLGs) (an important circuit category whose circuits implementations are discussed mainly in Section 2.4) mainly when aimed at analogue or mixed-signal circuit and system designs, thus when there is a more direct connection to their neuro-inspired design and operation.

Although many simple designs exist that support the concept of neuro-inspired computing the main area of study is concerned with modeling the group of logic circuits that can compute linear separable logic functions. This circuit group is based on the initial research results of the simplified models of neural activity that were centred around the concept of comparison of a weighted input with a threshold value, thus called the threshold logic (TL), with this terminology appearing initially in the work by McCulloch and Pitts (1943). The TL concept can be considered a main component in understanding and emulating how networks of simple and primitive logic operations can be used to design complex systems that computes high-level operations, such as image processing, data classification, pattern recognition etc., with few example including the work shown by Maan et al. (2016); James (2016); Sebastian et al. (2018). In many cases, the novel computer architectures that arise from the introduction of the neuro-inspired TL design methodologies were in the form of sea-of-computing-nodes implementations where primitive operations, such as multiplication and comparison, as well as major architectural design shifts, such as memory co-location with the computing modules, are key characteristics of the design, as showcased by James (2016); Rojas (1996); Zhanbossinov et al. (2017).

An important case of memory and logic co-location circuits can be found in the first simplified model of synapse-neuron functionality was introduced as a result of the evaluation of the neuron functionality as a threshold logic gate (TLG). The so-called

TLG, following the fundamental principles of neuro-inspired computing as well as being compatible with conventional design methodologies, was a model for performing the comparison of a threshold value to the weighted sum of an input vector. Each weight is assigned to each input and defines the impact of that input to the logic operation performed by the integrate-and-fire (IF) function. As discussed earlier, the TL computing concept was introduced initially by McCulloch and Pitts (1943) as a method of emulating the neural activity observed in the brain using relatively low complexity circuit designs. The modelling of neural functionality as a complex system of TL primitive logic blocks has been the basis for the analysis and design of ANN-inspired computer architecture implementations. Hence, although these basic logic gates perform only fundamental signal processing tasks, the interconnection of those circuits adds to the emergence of complex classification function, as shown by Guo et al. (2015); Merrikh-Bayat et al. (2015); Wen et al. (2018). This has the consequence that larger systems that use these primitive gates could be capable of advanced computing when computing processes can be disintegrate into basic fundamental operation. Another important detail is that TLGs are considered a group of circuits that initiated a different approach of conceiving the design of mixed-signals logic circuits in computer architectures. This was based on the emergence of memory-centric reconfigurable logic systems, enabled by the memory cells incorporated into a generalised gate array towards reaching the parallelism and power efficiency of biological neural activity, as shown by Li et al. (2016); Alibart et al. (2016). More details on some important TLG design methodologies and examples will be presented in the following sections of this thesis.

Although TLGs and similar circuits stand as important examples of neuro-inspired computing (architectural organisation around primitive logic blocks and use of analogue or mixed-signal designs) there are many other concepts of neuro-inspired computing that have been adapted from biological neural networks for modern computers, such as the memory and logic co-location solution of systems. These methodologies, in many cases, are based on incorporating novel device and circuit solutions as design modifications of existing circuit design.

2.3 In-memory computing

Neuro-inspired computing concepts are increasingly being considered as a more efficient method of implementing computers, as showcased by Sebastian et al. (2020). As discussed in previous sections, the conventional separation between computing and memory systems impose important limitations to computers' performance. Although many approaches to introduce novel computing architectures are based on neuromorphic designs that attempt to mimic the biological brain at different levels of abstraction, one of the most prominent concepts is the implementation of post-von Neumann architectures by retaining some of the existing conventional design but with important changes in the functionality of each system. In-Memory Computing (IMC) is a well-studied concept introduced towards mitigating architectural performance problems in conventional computer architecture designs (such as the von-Neumann bottleneck which occurs due to the limited bandwidth communication channel between processing units and memory modules), with some examples being the work exhibited by Agrawal et al. (2018); Si et al. (2019); Yu et al. (2020); Zhang et al. (2017); Su et al. (2020); Si et al. (2020); Seshadri et al. (2017). In this new type of computer architecture, memory systems are considered as a natural place to build upon novel logic circuits and systems that employ mainly the memory circuitry to perform computations. The process usually involves the introduction of additional circuitry and/or technologies towards enabling the memory system to operate as massively parallel data processing accelerators, as discussed by Santoro et al. (2019). This concept matured into the IMC design paradigm which are one of the most important class of future computer architectures, such as the cases shown by Sebastian et al. (2020); Ielmini and Wong (2018).

There are many different approaches in implementing IMC systems, as exhibited by Chakrabarti et al. (2017); Santoro et al. (2019); Ielmini and Wong (2018); Sebastian et al. (2020). Four of the most important are Computation-near-Memory (CnM), Computation-in-Memory (CiM), Computation-with-Memory (CwM) as well as the Logic-in-Memory (LiM). CnM technique is based on the introduction of computing systems attached to memory modules. This means that CnM can perform computations essentially in the enhanced memory module before sending data to a central processing unit. Depending on the level of processing, the CnM can be considered a pre-processor or a full parallel bit-wise accelerator capable of performing multiple MAC operation per clock cycle. CiM is similar to the CnM but with the difference that no additional computing circuits are necessary. Instead, in CiM, in-memory computing is performed by reading data in parallel from multiple rows and using the sense amplifiers (SAs) to compute some primitive logic operation, such as bit-wise AND, OR, etc. The computation results can be written back in memory with an additional write-back data-path making possible the sequential execution of parallel bit-wise operations in a set of data. In CwM architecture designs the memory itself is as the means to perform computing. The memory modules in this technique are essentially Content Addressable Memories (CAMs) where the read operation is performed in multiple rows and can retrieve stored data as a form of Look Up Table (LUT), as discussed by Santoro et al. (2019). The CAM and LUT systems have an important connections with the neuro-inspired architectures since they can be considered as different forms of associative memories, a very interesting and valuable computing concept inspired by the brain. The final LiM technique is different than all the other techniques in the sense that the data is computed locally without the need to move them outside of the array. The memory components themselves are capable of performing simple computations and data are manipulated within the memory system which can work as a full computer with memory and logic co-located, as shown by the work of Santoro et al. (2019); Agrawal et al. (2018); Si et al. (2020, 2019).

Most IMC have system design characteristics that have been adapted straight out of the main design methodologies found in memory system architectures. One of the most important feature of memory units is the uniformly distribution of primitive storage cells in a 2-dimensional or 3-dimensional topological structures, as the work of Chakrabarti et al. (2017); Lastras-Montaño et al. (2017) showcase. One of the most well-documented and widely employed 2-dimensional structure for the topological organisation of memory cells in a system, that later was adapted for use in IMC systems, is the crossbar array that provides the denser method of organising cells, as shown by Afifi et al. (2009); Zha and Li (2016); Adam et al. (2016). A crossbar array constitutes a dense configuration of nanoscale devices or circuit, such as a DRAM cell or a RRAM device. In particular, a crossbar array consists of two sets of wiring connections crossing perpendicularly as sets of row wires and column wires. Each crosspoint area of the array, hence each area crossing between a row wire and a column wire, is connected to a two-terminal component (one terminal connected to the row wire and the other to the column wire). For the case of natural two-terminal devices, such as most of the RRAM device technologies, no further array-level connections are needed. On the other hand, for crosspoint components with multiple terminals, such as the RRAM-based 1-transistor-1-resistor (1T1R) or DRAM's 1-transistor-1-capacitor (1T1C) cells (i.e. at least three terminals per cell in the form of MOSFET gate, BE and TE), the additional terminal (usually the gate of the access transistor) need to be accessed per row (usually referred to as wordline) or per column (usually referred to as bitline) with an additional set of row or column wires, respectively. Various switching elements can be used in crossbar's junctions and the conductivity of this element adjusts the intensity of the connection between each pair of horizontal (row direction) and vertical (column direction) wire. Naturally of particular interest is the case of integrating emerging memory technologies, such as the RRAM, with a crossbar array architecture, as straightforward realisation of synaptic connections of a neural network if we consider neuron circuits connected at the edges of the rows and columns, a main design methodology that is shown by many examples including the work by Liu et al. (2016); Cai et al. (2019); Liu et al. (2015); Zha and Li (2016).

At the centre of most ANN-related solutions is the cornerstone operation of Multiply and ACcumulate (MAC) which is extensively employed as a primitive function for most machine learning (ML) algorithms, as discussed by Ielmini and Wong (2018); Sebastian et al. (2020); Xia and Yang (2019). The multiplications and summation are two fundamental computer operation. In their simplest form we can consider that the primitive AND and OR Boolean logic operations are basic equivalents of binary multiplication and addition, respectively. Many pattern recognition algorithms include massive amounts of MAC operations usually in the form of matrix to vector (MVM) or matrix to matrix multiplication (MMM). The MAC operation of two matrices is also known as dot-product computation. Although, in digital logic the operators of AND and OR can be employed to create large dot-product networks, in many cases this is too costly in terms of area and power dissipation. Instead the main effort of implementing massive networks of MAC circuits have been placed in memory-centric architecture designs where low-complexity reconfigurable cells, such as DRAM and SRAM that can store binary information, are used for the MVM MAC operations. Thus, the strong connections between IMC design and MAC-based massively parallel primitive operations has been identified and exploited in most memory and logic co-location systems.

Most of the early IMC designs were based on conventional MOSFET technology and CMOS design and, more specifically, around circuits already used as part of existing functional memory solutions, such as CMOS-based memory cells (i.e. SRAM cache memory), sense amplifiers, data buses, multiplexing and decoding networks etc., as shown by Agrawal et al. (2018); Si et al. (2020, 2019); Yu et al. (2020). Many interesting architecture have been proposed throughout the last decades showcasing the continuous effort towards providing additional findings for IMC design methodologies. In the work by Zhang et al. (2017), a 6-transistor (6T) SRAM memory architecture is proposed for implementing a digital MAC-based system for pattern recognition and classification. The input stimulus for classification is in analogue format (driven by custom CMOS network of binary-weighted current sources) while the classification engine is build around the SRAM cells and the sense amplifiers, with additional circuitry for data conversion and memory control in the periphery of the main IMC topology. Similarly to most IMC designs the peripheral circuitry has been adjusted, compared to conventional memory control circuitry, to access multiple rows and/or columns per operation. This is a feature that defines the IMC schemes since massive parallelism for primitive computations is also a neuro-inspired concept (as is the aforementioned memory and logic co-location paradigm). In the research conducted by Kang et al. (2018), an IMC system for inference operation is proposed. The system is named Deep In-Memory Architecture, and similarly to other CMOS-based implementations the main memory/computing circuit used is the SRAM cell. The system is build around a conventional SRAM memory bank but with custom peripheral circuitry to enable multi-row and multi-column dot-product operations towards enabling massively parallel access to computing memory arrays. In the work by Seshadri et al. (2017), a novel computational concept build around the use of conventional DRAM cells alongside custom sense amplifiers was proposed. In this IMC design, multiple DRAM-based capacitor elements were connected simultaneously to the sense amplifier to perform massively parallel Boolean bit-wise logic functions, as exhibited in the design proposed by Seshadri et al. (2017). Similarly to most IMC techniques the logic functions performed are of low-complexity (i.e. simple Boolean AND, OR, etc. gates) but the main feature is the large scale of parallelism of data processing in the memory-centric accelerator using conventional DRAM memory modules.

More recent implementations proposed the use of custom MOSFET circuitry alongside traditional memory cells to enhance the computational capabilities of IMC systems. In the work by Su et al. (2020), another IMC design (also referred to as Computing In-Memory (CIM) accelerator category) is proposed for applications in pattern recognition and classification Integrated Circuits (ICs) aimed for edge computing. A custom two-way transpose (TWT) SRAM cell is implemented to support the parallel MAC operations of the system. Additional custom circuits to accommodate for process variation in deep sub-micron MOSFET technologies and a custom sense amplifier adjusted for enhanced sensitivity to small read margins are also part of this IMC system. In the work by Si et al. (2020), an IMC design with a custom Twin-8T SRAM cell is proposed for CNN implementations in processors at the "edge". A parallel system with dual memory word-lines is implemented to store and read positive and negative weight values. The custom 8T SRAM cells includes a read port which enables the decoupling of the memory cell from the long bit-line connecting all the cells together (a parallel access if required for a dot-product operation) which enhances the speed of operation.

All in all, the main effort in CMOS-based IMC systems can be reduced in the use of all the main components that exist traditionally in conventional memory modules, such as in DRAM and SRAM cache memory banks, while employing usually small circuit alterations and/or additions to the memory cell and sense amplifier. Additionally, usually the IMC solutions require a more complex control system to generate appropriate control signal to enable parallel computing by accessing and reading multiple word-lines of memory cells. As is evident from the examples of the IMC implementations, the concept of IMC is inspired by the neuro-inspired memory and logic co-location idea with the memory banks being loosely adapted to the role of artificial synapses and the sense amplifiers to the role of artificial neurons. The IMC designs is among the most interesting attempts to employ neuro-inspired computing inside conventional computers and exist as a strong paradigm of realistically feasible future accelerator systems. But limitations imposed by technologies (limited analogue computing capabilities from MOSFET devices) impedes the realisation of many of the most promising IMC solutions that employ analogue and mixed-signal circuitry to accelerate fundamental arithmetic and logic operations, as discussed by Sebastian et al. (2020); Ielmini and Wong (2018). The rise of novel memory technologies that are better suited for the role of artificial synaptic weights as candidates for future hybrid IMC systems (i.e. employing both emerging and conventional electronics) can result in an important enhancement of the IMC capabilities (an important evolution of the IMC paradigm that will be discussed in Section 2.7).

The concepts of IMC system design and MAC-based logic function are considered tightly related since the crossbar arrays and the memory word-line access is well-fitted for dot-product MVM operations as well as the customised sense amplifiers or capacitor-based circuitry can be employed for the accumulation operations. The MAC-based logic mapping can easily be scaled into massively parallel computing platforms making the inherently parallel structure of IMC an ideal computing solution for MAC acceleration. Although many MOSFET-CMOS-based IMC implementations exist, with some of them having good performance as shown for the design proposed by Seshadri et al. (2017), limitations of MOSFET devices increase the need to introduce novel devices and technologies into a similar set of IMC architecture designs, as discussed by Hamdioui et al. (2016). The fact that the MOSFET device is usually operated as a binary switch as well as the difficulty to scale reliably in very small technology nodes showcase that devices such as RRAM could alleviate many design constraints of previous IMC systems by introducing new hybrid MOSFET-RRAM solutions, as discussed by Hamdioui et al. (2016). Additionally, the IMC concept itself requires novel technologies towards achieving analogue memory and logic operations in-situ, a trait that in most cases is missing from IMC solutions build around conventional technologies, as shown by Ielmini and Wong (2018). The analogue data processing can be a significant solution for the next generation of accelerators especially for the case of edge computing where we would like to achieve very low power designs, as showcased by Ielmini and Wong (2018); Krestinskaya and Pappachen James (2018). Of course, the introduction of hybrid emerging-conventional implementations is just one part of the solution for future accelerators. Additionally to the device technology updates, architectural changes will be important in developing truly next generation IMC accelerator with enhanced capabilities, as shown by Mehonic et al. (2020).

At the same time that IMC was gaining ground as an emerging alternative to conventional von Neumann computer architecture, other supplementary aspects of neuro-inspired computing were also increasingly considered as viable alternatives to the CMOS-based Boolean logic that dominates modern computers. The rise of novel emerging memory technologies and their maturity into future widespread memory solutions only enforce the idea of using neuro-inspired mixed-signal logic inside the memory units as a main method of performing massively parallel operation in future computers.

2.4 Threshold Logic Computing Circuit Design

Towards creating hardware that can be efficiently employed for the next generation of IMC architectures a lot of effort has been placed in the design of circuit that can compute massively parallel multiplication operation fast. The neural activity and processing is capable of solving complex problems through primitive and fundamental logic techniques. Many contributions have been made towards the implementation of a digital circuit that can exploit the same powerful operations. Over the last few decades the

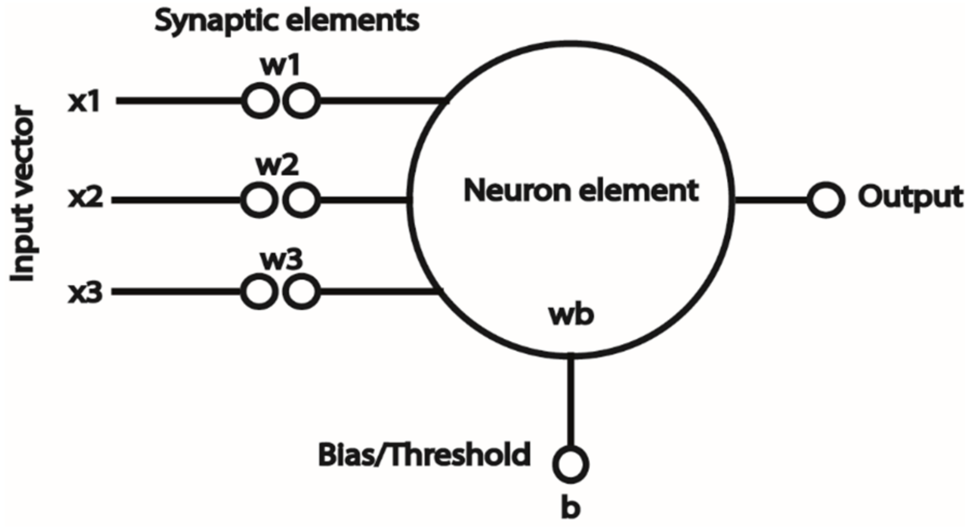

FIGURE 2.1: Schematic of the fundamentals of an artificial neurons design. Synaptic connections are mapping the input space into an appropriate format through the use of synaptic weights. The synapse-dependent transformed signals are then used as inputs for the neuron which decides whether to fire or not depending on the total effect of the input signals compared against a biasing signal.

concept of TLGs is becoming increasingly stronger with a wide range of electronic implementations developed since its inception, as extensively highlighted and discussed by Beiu (2003); Beiu et al. (2003d). TLGs, in their most wide-spread form, are based on the operational characteristics of single perceptron nodes with binary weighting configuration and digital input signals, thus fitted for conventional computing systems. As discussed briefly in Section 2.1, a single TLG can perform a single threshold logic (TL) operation, from simple function such as AND, OR and their complements to other more complex majority logic functions. One important limitation is that a single TLG can compute only linearly separable function, similar to what the perceptron theory and application dictates, as presented by McCulloch and Pitts (1943). In order to implement a non-linearly separable function more complex multi-layer networks of TLGs need to be employed. Hence the operational traits are comparable with that of multilayer neural network, as discussed by Tran et al. (2012).

Furthermore, similarly with perceptron's functionality, the functionality of TLG is defined through the input weights and threshold value (also referred to as bias) of the unit, thus the same network structure can be used to compute different majority functions by using different sets of (input) weights and/or threshold, as shown by Bayat et al. (2017). Although the possible sets of classification configurations permitted by assuming the TLGs to binary weights is limited (which is true for the case of implementing weights with MOSFETs), through the employment of analogue weights the configurability of the circuit can be greatly enhanced, as shown by Maan et al. (2016). This enables the TLG circuits to truly be an ideal candidate for a reconfigurable logic gate, while retaining a low-complexity primitive logic block design. Furthermore, based on the IMC design methodology discussed in Section 2.3, TLGs seem to be a good memory-dependent reconfigurable primitive gate choice for building IMC systems.

Although, the general belief is that a neuron in reality is a more complicated system and the simplification towards TLGs is questionable, these circuits can provide a powerful tool in the hands of electrical engineers through the form of a range of efficient and programmable logic gate families aimed towards the implementation of high performance and power efficient computer architectures, with some examples showcased by Krestinskaya and Pappachen James (2018); Beiu et al. (2003b); Bobba and Hajj (2000); Xia et al. (2009). Many different implementations of threshold logic gates have been proposed with different trade-offs regarding their power-noise ratio performance. Recently, the use of Current-Mode Differential TLGs seems to winning ground as one of the most fast and low-power TLG implementation, as exhibited by Dara et al. (2017); Bobba and Hajj (2000). In this section, we are going to study and discuss the history of TLG implementations and the advances of the TLG design paradigm throughout the recent years, with a focus on the Differential TLGs.

In the literature we can find many different TLG implementations where the integration of TL circuits with a wide variety of components both conventional and unconventional is proposed, e.g. transistors, as shown by Bobba and Hajj (2000), capacitors, as shown by Medina-Santiago et al. (2018), resistors, as shown by Vrudhula et al. (2015), resonant tunnelling diode, as shown by Pacha et al. (1998); Pettenghi et al. (2008a) etc. The main goal of this search of a TLG design that can be efficiently used as a primitive logic blocks for ANNs implementations. From the early days of TL introduction and the first emergence of TLGs till now many researchers have tried to design novel TLG circuits and systems towards increasing the advantages and limiting its disadvantages that this logic technique exhibits (i.e. no arbitrary function mapping etc.). As the number of proposed TL circuits and systems is on the order of hundred we are not going to present each one of these implementations in this report but instead we are going to focus on some of the most important modern TLG, including the conductance-based TLGs and the differential TLGs.

TLGs can be considered as one of the main example of implementing circuit emulators of synapses-neuron circuits, thus a lot of effort has been focus in developing many different designs, with some of the different implementations showcased by Pettenghi et al. (2008a,b); Strandberg and Yuan (2000); Bobba and Hajj (2000); Leshner et al. (2010). One of the most important early types of TLGs are the conductance or current-based implementations. These type of TLG designs initiated during the mid-1940s, as shown by Beiu et al. (2003b); Beiu (2003), through the introduction of MOSFETs-resistor circuits and exhibit important traits such as easy integration with conventional CMOS Boolean logic gates (e.g. for the case of multi-type logic cascade) and robust and reliable operation. Later other conductance-based designs included different main computing components such as Bipolar Junction Transistors (BJTs) -based and fully MOSFET-based TLGs, as shown by Beiu et al. (2003b). A closer examination in a number of TLG designs can provide interesting insights regarding the main design methodologies adapted to implement digital artificial neurons. Additionally, the analysis and dissemination of MOSFET-based TLG designs will clearly highlight the parts of the design that can be enhanced by using emerging post-MOSFET devices and post-CMOS circuitry, as shown byBeiu (2000); Quintana and Rueda (1995); Beiu et al. (2003a,c).

The main computing concepts behind conductance-based TLG designs is the use of a parallel reconfigurable network that can map a programmable conductance value (i.e. parallel configuration of simple synaptic weight emulators) and, usually, a CMOS-based gate is used as a thresholding element capable of firing when the total conductance is above a specific programmable threshold value. The thresholding element essentially performs the comparison between the threshold and the combined weighted input of the input network and is usually implemented with low-complexity circuits such as a simple CMOS inverter, as shown by Beiu et al. (2003b,d). For the CMOS inverter to be programmable additional parallel pMOS and nMOS device are connected to the circuit in order to configure on-the-fly the composite width of the pMOS (W_p) and nMOS (W_n), for the pull-up and pull-down networks, respectively. This enables the simple CMOS inverters to have a programmable voltage threshold on which the inverter switches between logic levels. There is a wide variety of different conductance-based TLGs each one following design methodologies that enhance some characteristics of the gate's behaviour with each circuit configuration modifying some aspects of the baseline TLG design, as shown by Beiu et al. (2003d); Beiu (2003). Usually the modifications are aimed at improving the speed and reducing the complexity and power dissipation of each newer version of the TLG. At the same time, many of these modifications are related with specific MOSFET-CMOS design optimisations at a circuit-level. Some of the early conductance-based TLGs have static power dissipation, thus they are inefficient from a power consumption perspective especially when compared to available CMOS-based circuits. While many interesting ideas have been employed towards the minimisation of static power consumption, the problem still exists in some designs and the main focus at designing novel TLGs is at mitigating the power dissipation issues through combinations of different design methods as well as the introduction of small additional modifications on the operation cycle of the gate, as shown by Dara et al. (2017); Leshner et al. (2010); Leshner (2010).

An important circuit evolution of the traditional TLG design methodology is the differential design. Differential TLG designs are considered some of the most computationally versatile types of TLG mainly due to their capability in representing efficiently both positive and negative weights (as well as positive and negative threshold/bias) while at the same time retaining low-complexity design and generally occupying small chip area. At the same time, they usually providing easier methods for representing a threshold value, for example as an additional network of MOSFETs, resistors, etc. instead of the fixed CMOS (with pMOS and/or nMOS custom widths) inverter type of thresholding element, as shown by Bobba and Hajj (2000). The early CMOS-based differential TLGs are based in the employment of two MOSFET-based networks that map the weighting per input. Due to the use of networks that connect MOSFETs in parallel, a single MOSFET is considered a unity weight with different weights mapped with the input being connected simultaneously to a larger number of parallel MOSFETs. Thus, the use of MOSFET devices restricts the capability to efficiently represent a wide variety of weights, especially weight values that are not derivatives of the unity weight. This observation applies to all TLGs that base their weight mapping capabilities on MOSFET devices. Similarly to the basic design principles of conductance-based TLGs, a differential design usually employs a CMOS-based thresholding circuit that compares the weighted sum of one network against the weighted sum of the other network (thus compares the competing differential networks). The main difference of the thresholding element, as compared with non-differential designs, is that the comparator is made to compare two voltage (for voltage-mode designs) or current (for current-mode designs) signals that are competing with each other. Both of the competing signals are generated by input vectors that control either a positive/negative input weight contribution or a positive/negative threshold/bias contribution, as shown by Bobba and Hajj (2000); Dara et al. (2017). In general, differential TLG solutions are capable of faster, reliable and, maybe more importantly, low-power operation compared to non-differential implementations. The main advantage of the differential designs is mainly considered the low power consumption since since the TLG can be optimised to perform the same computation but with additional biasing signals to speed the comparison operation and thus minimise the dynamic power dissipation. With total power consumption of the TLG reduced sufficiently, some designs are considered as interesting alternative to replace some Boolean CMOS networks, as shown by Beiu et al. (2003b); Leshner et al. (2010); Maan et al. (2016). More specifically, TLGs can potentially can potentially be considered as an alternative in circuits and systems where a network of reconfigurable majority gates can be beneficial for the delay and power consumption of the system-level logic operation. This is especially important for implementing reconfigurable MAC networks, as discussed by Sebastian et al. (2020), thus TLGs are promising candidates for implementing large ANNs, as shown by Krestinskaya and Pappachen James (2018).

The first differential TLG design introduced in 1964, as shown by Beiu et al. (2003b,c). This implementation was based on a resistor-diode (R-D) network as well as bipolar transistor technology, with the R-D network mapping the weighting and the threshold emulated by the characteristic of the BJTs, as shown by Beiu et al. (2003b). Since this implementation many advancements in device technology and circuit design have been performed and in many cases each new technology find its way as part of TL-based

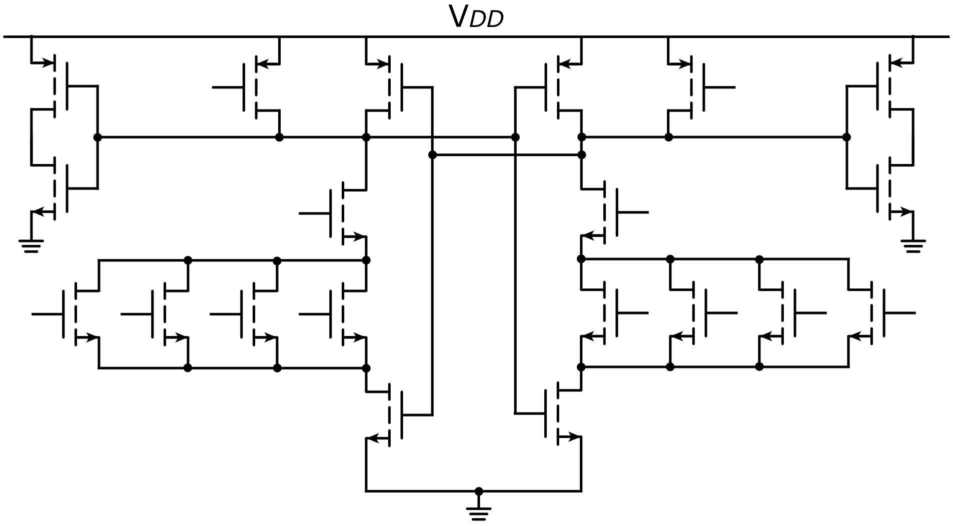

FIGURE 2.2: Schematic of LCTL design proposed by Strandberg and Yuan (2000). Schematic adapted from Strandberg and Yuan (2000).

circuit implementations, as shown by Beiu et al. (2003d). Some of the more recent CMOS-based TLG solutions are based on the use of CMOS sense amplifier (i.e. what is essentially a large SRAM memory cell design) as the CMOS-based comparator circuit which, being essentially a simple digital neuron, is emulating the thresholding element that defines when to fire or not. The sense amplifier (a design usually found in conventional DRAM and SRAM memory banks) can perform fast comparison operations between the two competing differential signals through the use of a positive feedback inverter circuit (i.e. CMOS-based latching element). Thus the same TLG output can occur by using either direct or inverted inputs and through this process each output is enforcing the final stable memory state of the sense amplifier. The sense amplifier detects which signal is higher and amplifying this difference to a full voltage swing, thus performing stable memory latching (i.e. bi-stable memory latching). The amplifier can be tuned to detect very small differences between the competing signals and be sensitive to low signal values, thus it can perform a fast latching while the input-weight differential network will be charged with small voltages (i.e. lower than the supply voltage V_{DD} of the sense amplifier), as shown by Vrudhula et al. (2015); Strandberg and Yuan (2000). Due to its design as a latching element, the sense amplifier can provide simultaneously the output of the differential comparison (canonical output by convention) as well as its complement without any additional circuitry. It worth noting that in most designs additional CMOS-based inverters are used for the sense amplifier's output for isolation, thus the TLG will cascade the output of the result of the computation and not the competing signals pre-computation's intermediate results (isolation from the actual signals generated by the differential network).

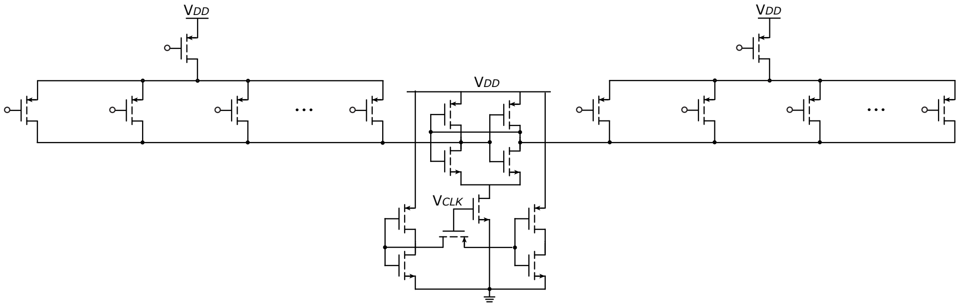

FIGURE 2.3: Schematic of CIAL design proposed by Hidalgo-Lopez et al. (1995). Schematic adapted from Hidalgo-Lopez et al. (1995).

Some differential TLGs that followed the above-mentioned design methodology can be found in the implementation of the Cross-Coupled Inverters with Asymmetrical Loads (CIAL) proposed by Hidalgo-Lopez et al. (1995) and shown in Fig. 2.3. The CIAL TLG was used as a digital comparators circuit to accelerate systems that otherwise will use conventional Boolean logic gates. Additionally, another similar design approach can be found in the implementations of the generic Latch-Type TL (LCTL) gates proposed by Strandberg and Yuan (2000) and shown in Fig. 2.2. These differential designs showcase a similarity in the method they implement their thresholding element, thus their neuron emulator circuit, as well as the method the competing signals are generated. Specifically, based on the input logic vectors controlling the differential parallel MOSFET-based arrays, one of the two sensor nodes will be enforced to change state through higher current flow (i.e. current mode operation). This is defined by the number of parallel MOS devices being conductive at each sense-and-compare (i.e. essentially a customised memory readout) operation. Through the implementation of differential arrays, the TLGs can compare either two sets of input vectors or a set of input vector and a set of threshold vector, thus providing flexibility of operation and much easier programming of the threshold value compared to the earlier conductance CMOS-based TLGs. Each input of threshold signal can be assigned to multiple parallel transistors, thus increasing the contribution of this input/threshold to the comparison operation. Furthermore, another different implementation proposed to improve the speed of the LCTL circuit is the Cross-Coupled inverters with asymmetrical loads Threshold Logic (CIAL-TL) design which sets its differential arrays outside of the sensing element (Fig. 2.4). A main difference with the LCTL and similar designs, similar to the designs shown by Leshner (2010); Tripathi et al. (2017); Bobba and Hajj (2000), compared to the CIAL and CIAL-TL designs, thus designs similar to the work by Hidalgo-Lopez et al. (1995); Leshner et al. (2010); Beiu et al. (2003b), is the placement of the differential arrays. For the case of the LCTL circuit category, the differential arrays are within the sense amplifier circuit path (similarly to the work by Bobba and Hajj (2000)) while in the CIAL type TLGs the arrays are placed outside of the sense amplifier circuit. The CIAL-TL is able of performing faster comparisons due to the avoidance of long feedback caching of

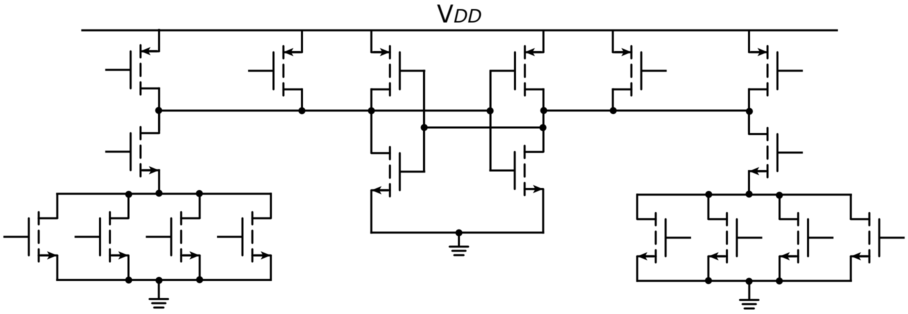

FIGURE 2.4: Schematic of CIAL-TL design proposed by Leshner et al. (2010). Schematic adapted from Leshner et al. (2010).

LCTL, as shown by Leshner et al. (2010); Beiu et al. (2003b). Another important part of the TLG design methodology is the inclusion of small circuit modifications to the sense amplifier comparator part towards of this whole family of circuits is that they need an equalisation circuit to "initialise" the state of the sensor part before every comparison (evaluation) the state of the sensor. The initialisation of the sensor part is in fact a technique of placing the latching element in an unstable state, i.e. both positive feedback inverters are "pushing" logic '0'. When the differential values are introduced to the sensor the latch is forced to take one of the binary possible states, thus "evaluating" the differential input current flows.

It is important to note that the differential TLGs, similarly to most other modern VLSI TLG designs that use deep sub-micron MOSFET devices, suffer from sensitivity to noise as well as from process variations and device mismatch since essentially the comparison is between two analogue current/voltage values. This results in necessary limitations to their maximum size of input/threshold vectors that they can use. Of course, different techniques widely used in modern electronics to battle these issues can be used in these types of TLGs as well. Techniques such as parallel arrays of same size transistors (in order to lower the statistical parameter variations) as well as other analogue layout and circuit design methods, can be used to increase the fan-in of the gates. Such solutions increase significantly the chip area for implementing the same logic and at the same time do not eliminate the computing reliability issues, as discussed by Beiu et al. (2003b); Maan et al. (2016).

Regardless of these known issues that requires circuit design adaptation to mitigate, differential TLGs are used extensively as the baseline design of TL circuit implementations. In recent decades, the introduction and utilisation of Current Mode (CM) differential TLGs (CMTLG) attempts to further reduce power consumption as well as the delay of computation, issues that usually affect TLGs, as discussed by Bobba and Hajj (2000). During the early CMTLG implementations and due to attempts to minimise the necessary MOS transistors needed to perform the operation, the sensor part, which

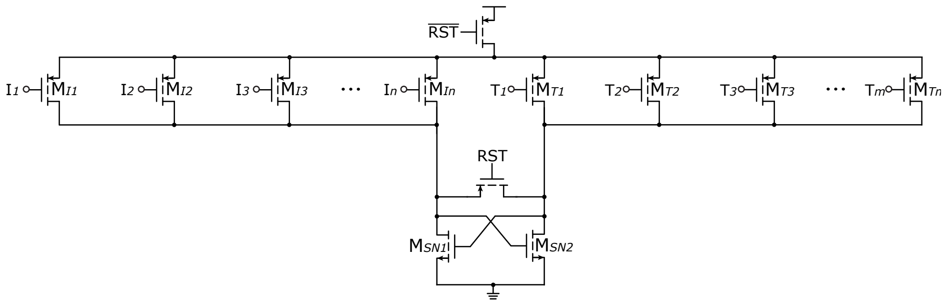

FIGURE 2.5: Schematic of ECMTLG proposed by Bobba and Hajj (2000). Schematic adapted from Bobba and Hajj (2000).

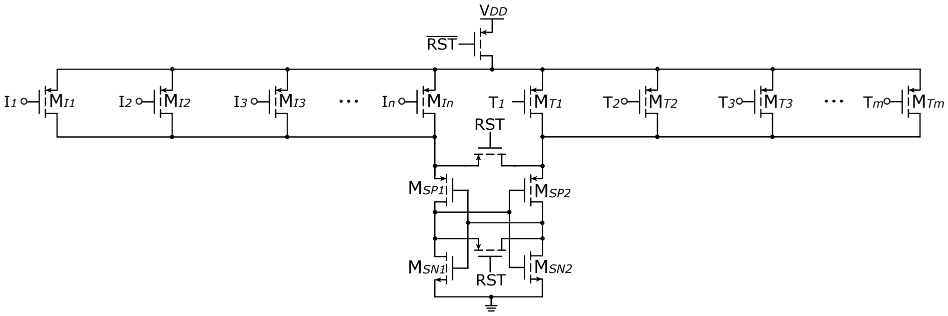

FIGURE 2.6: Schematic of the DCMTLG proposed by Bobba and Hajj (2000). Schematic adapted from Bobba and Hajj (2000).

decides the output of the gate, consisted of a half latch circuit (Fig. 2.5), as exhibited by Bobba and Hajj (2000).

Some interesting differential TLGs that inspired many modern designs, such as the one showcased by Dara et al. (2017), were initially introduced by Bobba and Hajj (2000). These were the discharged CMTLG (DCMTL) (see Fig. 2.5) and the equalised CMTLG (ECMTL) (see Fig. 2.6). The only difference between the two designs is the equalisation circuit implementation, as shown by Bobba and Hajj (2000). For the case of DCMTL, two parallel nMOS was connected in the sensor part, one per branch of the half latch element to short-circuit the half latch nMOS (discharge them), while, for the case of ECMTL, a nMOS transistor that connect the two half latch outputs, and equalise them during the equalisation phase, was added. In a way, the basic concept was to incorporate a parallel array of pMOS components as the pull-up networks of the inverters comprising the latch. Hence the TL enabling transistor array was “embedded” into the actual sensor part. Additionally, for both DCMTL and ECMTL designs the addition of a power gate that disconnects the circuit from the power supply during the equalisation phase was added to eliminate the static power consumption during equalisation. At the same time, by using these half-latch structures static power consumption problems arise for the evaluation cycles, thus using full latching element for sensing and comparison in differential TLGs is an important aspect of the design. By using the CMOS sense amplifier as sensor part the static power consumption during evaluation phase (phase when the sensor outputs the result) is eliminated, as shown by Bobba and Hajj (2000). Although the dynamic power consumption is an existing issue with the sense amplifier (power dissipation during the state transition) implementations with sub-threshold CMOS latches and differential weight banks could mitigate this issue, as discussed earlier in this section.

Based on the aforementioned brief analysis of main CMOS-based TLG design advancements throughout the recent years it is clear that extensive research has establish some fundamental design methodologies for developing artificial neuron circuitry. The main computation performed in TL circuits is centred around the comparison between the two "competing" signals. In every TLG, the main design effort is found in the configuration of a low-complexity and low-power primitive classification circuit that applies a decision boundary upon the input information. As expected this is exactly in line with the primitive functionality of simple neural activity, thus these kind of circuits are emulating simple logic components of what we observe being the method of the nature to organise the biological data processing system around us. But the efficiency of these transistor-based emulator of the biological intelligence is limited by our capability to manipulate signals in the nanoscale. Although the MOSFET is the component that is driving the computer technology throughout the last 5 decades, it cannot easily be incorporated into a system that will compete with the biological neural networks in both performance and power consumption, as shown by McKee (2004); Ielmini and Wong (2018). As emerging technologies mature, new concepts arise of how we could build a better emulator of biological neural networks. In the following sections of this chapter we are going to discuss how the introduction of RRAM in IMC, TLG and other neuro-inspired circuits and systems provided novel solutions of ANN computer architectures.

2.5 Associative Memories and Data Conversion

Memory association is a very interesting mechanism found in biological neural networks. Similarly to the rest of the aforementioned design methodologies, memory association is being investigated and transferred from the biological NNs to ANNs and computer architectures. Essentially, we can consider memory association systems, based on their computer architecture implementations, as a category of systems that map a specific set of data to another set of data. The data can have the same or different format, i.e. one set of data can be digital while the other analogue etc., with some such example showcased by Alkabani et al. (2019); Karam et al. (2015); Yavits et al. (2015); Zhou et al. (2019); Kaur (1998); Li et al. (2019).

Although the memory association mechanism found in nature is relatively complex and more research is required to enable a full understanding of the biological operation, this has not impeded its adaption in modern circuits and systems design. Circuits and systems that are inspired by the memory association mechanism are very popular and have enabled the development of powerful computational circuits. Some examples of this category of circuits can be found in the form of associative memories (AMs), content addressable memories (CAMs), look-up tables (LUTs) etc., as shown by many researchers including Alkabani et al. (2019); Karam et al. (2015); Junsangsri et al. (2017); Ullah et al. (2012); Guo et al. (2017); Xie et al. (2016); Hongal et al. (2014).

The introduction of novel memory technologies enables new possibilities of enhancing the computational capabilities of associative memory circuits. Through associative memories we can perform data conversions between different data sets. In the case we introduce devices capable of performing computing in the analogue domain, such as in the case of RRAM devices, then we can integrate physical data conversion operation simultaneously with other data processing. This has the potential of mitigating the disadvantages of using CMOS-based data converters, thus providing gains in area and power dissipation of hybrid RRAM-CMOS systems.

2.6 Metal Oxide Resistive Memory Technology

Although the MOSFET-based circuits and systems design enabled the current advances of digital computing and became the catalyst for significant gains of computational power in computer architecture, new technologies are in need towards performing a similar increase in computational jump for the next generation of computing systems. Many emerging technologies are being actively investigated towards enhancing our understanding regarding their capabilities and limitations. An increasingly larger focus is being given to the family of emerging memory technologies since developing novel solutions for mitigating the von Neumann ”memory wall” issue is being considered of high priority in future computer architecture designs.

Over the last few decades, memory has become the forefront of novel computer solutions. Many approaches aim at increasing the performance of computers while keeping the power consumption low by introducing new memory technologies to existing system designs. At the same time, other approaches aim towards eliminating architectural inefficiencies of conventional computer, such as the von Neumann bottleneck (”memory wall”) occurred due to the separation of logic and memory, as discussed in Section 2.3. Many different technologies are suggested as promising candidates in novel systems that rethink the design of the fundamental computing components. One of the most interesting memory device category is RRAM (Resistive Random Access Memory) also known as part of the memristor family of technologies. RRAM family of devices have a rich history firstly introduced mathematically as concept, as shown by Chua (2014), and then as early attempts to develop non-volatile switches by Strukov et al. (2009). In the last decade since the reintroduction of RRAM as a prominent nanoscale non-volatile memory element, as exhibited by the work of Strukov et al. (2009, 2010), many circuits concepts and designs around the use of analogue non-volatile memory matured due to the rapid advancements in RRAM technology. Nowadays it is considered by many as one of the most significant technologies that can relatively soon be integrated into computer design to introduce novel post-von Neumann architectures, such as the designs proposed by Santoro et al. (2019); Kvatinsky et al. (2012); Chua (2014, 2015); Prodromakis et al. (2011); Mehonic et al. (2020).

For more than a century, experimental clues have been accumulated that indicate that for some electronic devices a “strange” electrical characteristic appears within the otherwise “normal” behaviour, but these observations were usually overlooked as some sort of defectiveness of some device’s mechanism or in random events of device fabrication, as discussed by Prodromakis et al. (2011). It took many decades, as highlighted by Sebastian et al. (2020), from the early observations of uncommon device behaviour observations to the final establishment of the circuit theory around this group of ”stange” circuit responses by Chua (2014) who, through this work, described the behaviour of a new fundamental electronic device. In the work of Chua (2014), it is highlighted that there should be a fourth ”missing” fundamental electronic element that cannot be emulated by any combination of the other three circuits. The term for this fundamental components was memristor, as introduced by Chua (2014), an abbreviation for words memory resistor, and it was considered the electronic equivalent of a connecting link between magnetic flux and electric charge, similarly to how a resistor element is connecting the voltage with the current. The basic reason that memristor is defined as fundamental, while still maintaining unique traits that other fundamental circuits do not have, is due to its inherent behaviour of “remembering” the amount of voltage applied on it and for how long, thus preserving memory of its past (see Fig. 2.7 for an example of a memristor’s behaviour), as showcased by the initial findings of Strukov et al. (2009).

The fundamental mathematical observation and concept was materialised in the form of resistive memory, a technology investigated for many decades but was re-introduced in the last decade with a relatively recent research of a specific RRAM type by Hewlett Packard Laboratories (HP Labs) in 2008. In this more recent analysis of the RRAM devices, the connection with the memristor theory was established which enabled researchers to better understand the role of these devices not just as a replacement of conventional memories. With the findings of HP Labs’ RRAM device as well as the many different configurations of resistive memory device structure and technology developed in the following years, the start of a new era for emerging memory circuits and systems was enabled, as highlighted by Ielmini and Wong (2018). As discussed by Strukov et al. (2009), the fundamental structure of the RRAM was two-terminal titanium dioxide-based (TiO_2) capacitor-like Metal-Insulator-Metal (MIM) device, which behave with a similar characteristic pinched hysteresis loop thus having memory behaviour. Since this initial RRAM configuration many more MIM-based RRAM devices have been proposed with most of these device having a similar structure as showcased by Strukov et al. (2009).

The link between the theory and the practical implementation of RRAM technology is continuously reinforced through an increasing number of research findings proposing a wide variety of devices, circuits and architectures especially over the last decade based around the RRAM device, as shown by Prodromakis and Toumazou (2010). One of the most important applications of RRAM is the efficient emulation of the synaptic weights of a biological brain, with an example shown by Zhang et al. (2020b). This is achieved through its ability to assume continuously programmable conductance state, based on the voltage that is applied to its terminals, and retain that conductance state if not further voltage is applied. Furthermore, the capability to store multi-bit information on a single device renewed the interest in developing multi-valued memory modules capable of storing large amounts of data, as shown by Xia and Yang (2019); Zhang et al. (2020b). Hence, by replacing conventional memory elements with RRAM an important re-examination of ANNs-based computers set in motion, as shown by Sebastian et al. (2018).

The development of materials and devices that exhibit memory-resistive (memristive) behaviour, and thus can be utilised as RRAM, is an active area of research. Many of RRAM behavioural features are still fine-tuned with continuous advances in the respective technology and usually different types of RRAM have a unique set of traits and advantages (over competing technologies) as well as characteristics that require further improvements. RRAM devices can be implemented with many different methods and techniques with many recent research findings suggesting a wide variety of different materials and implementations, as shown for example by Stathopoulos et al. (2017); Pi et al. (2013); Li et al. (2015). We consider that a main feature for RRAM specific implementations is the MIM design. Additionally, most of the proposed RRAM device designs have common traits such as the low-power operation, the compatibility with existing MOSFET-CMOS technologies and the compact form factor that enables dense arrays of such components, as discussed by Kvatinsky et al. (2012); Chua (2015); Strukov et al. (2010). The presentation and analysis of the technology details behind the development of RRAM devices is out of scope with regards to this thesis. We can gather from literature how important RRAM technology is considered among researchers that work towards integrating MOSFET-CMOS with emerging memory devices and circuits, as discussed by Sebastian et al. (2020); Li et al. (2015); Ielmini and Wong (2018). The foundations of most of these proposed concepts and implementations is centred around the effort of enhancing computers through a technology-level

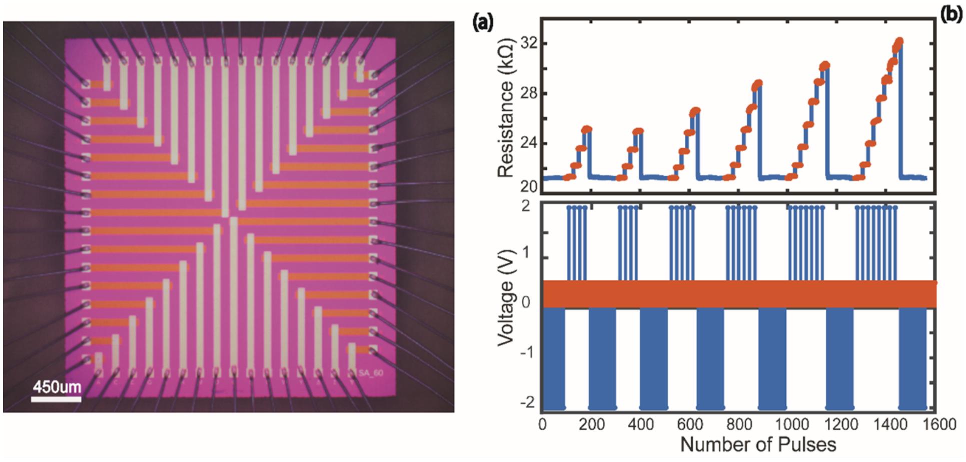

FIGURE 2.7: Die photo of RRAM devices developed by Stathopoulos et al. (2017) and example behaviour of multi-state programming of the devices (details of the experiment and detailed device response are exhibited by Stathopoulos et al. (2017)).

co-operation between the RRAM and MOSFET.

One of the most important traits of the the RRAM devices is scalability. RRAM can be easily organised into dense memory arrays due to its small feature size. In literature we can find preliminary findings suggesting very small (nanoscale) RRAM feature sizes with capabilities in scaling below the available MOSFET feature size, as discussed by Stathopoulos et al. (2017); Pi et al. (2013); Li et al. (2015). The implementation of crossbar circuit topology is one of the most well-documented method of organising RRAM memory arrays in dense configuration. A significant issue raised in the case of crossbar arrays is the sneak path problem. The behaviour of crossbar RRAM system can be essentially simplified to that of a resistive network. Hence, if the crossbar is not appropriately controlled, signals will cross-talk through different RRAM devices and charge different nodes from the ones we want to access. Thus appropriate design of the whole circuit with careful examination of the crossbar programming/reading schemes and accompanying selector devices should be performed before a RRAM-based system implementation. Towards enhancing the control of the memory components, the introduction of the 1-transistor-1-RRAM (1T1R, also referred to as 1-transistor-1-resistor or 1-transistor-1-memristor) components to organise dense RRAM memory banks is used. The 1T1R component will be examined more extensively in later sections and its behaviour will be tested in the circuits presented later in this study.

An important aspect of understanding the physics behind the intrinsic behaviour of the RRAM devices is the need to develop realistic and useful models of the RRAM to incorporate into a digital laboratory for testing circuit and systems performance before the practical circuit implementation, as shown by Hajri et al. (2017, 2020); Xia and Yang (2019). In the last decade, many different models for RRAM devices have been proposed. A variety of RRAM models, similarly to other non-volatile memory device families, has been extensively studied in the literature showcasing a wide variety of different behaviour (i.e. binary, analogue, non-linear, etc.), as shown by Reuben et al. (2019); Biolek et al. (2016); Reuben et al. (2017).

The fundamental RRAM behaviour can be described by the following mathematical differential equations for a current-controlled RRAM device:

$$v = R(w) \times i \tag{2.1}$$

$$dw/dt = i \tag{2.2}$$

, where w is the state variable of the RRAM and R is a generalised function of the RRAM resistance. The RRAM resistance is dependant upon the state of the device. Starting from this basic general theory form much effort has been dedicated in developing better RRAM models based on the observable behaviour studied under different RRAM device implementations with only few of the available model showcased by Biolek et al. (2009); Hajri et al. (2020); Prodromakis et al. (2011); Biolek et al. (2013); Kolka et al. (2015); Siemon et al. (2019); Reuben et al. (2019).

One of the most recent RRAM models that showcase the increasing maturity of the device modeling area of research can be found by Messaris et al. (2017, 2018, 2019). In the work by Messaris et al. (2018), a flexible data-driven voltage-dependent RRAM model is showcased. The model is build for simulating devices exhibiting non-volatile memory behaviour and bipolar switching operation, as shown by Messaris et al. (2018); Hajri et al. (2020). The model is also centred around empirical measurements of the RRAM devices aimed to simulate with the measurement being performed through the employment of ArC One instrumentation board (developed by ArC Instruments, UK) which is showcased by Serb et al. (2014). The mathematical description of the model is based on the following differential equations, a system that is based on the simple fundamental mathematical model of the memristor element:

$$i(R,v) = \begin{cases} \alpha_p(1/R)\sinh(\beta_p v) & v > 0 \\ \alpha_n(1/R)\sinh(\beta_n v) & v \leq 0 \end{cases} \tag{2.3}$$

$$dR/dt = g(R,v) = s(v) \times f(R,v) \tag{2.4}$$

From equation 2.4, we can further explain the $s(v)$ sensitivity function and the $f(R,v)$ window function. These equations are reproduced below for convenience. More information with regards to the exact methodology of developing the specific model can be found in the work by Messaris et al. (2018, 2017).

$$s(v) = \begin{cases} A_p(-1 + \epsilon^{t_p|v|}) & v > 0 \\ A_n(-1 + \epsilon^{t_n|v|}) & v < 0 \end{cases} \tag{2.5}$$

$$f(R,v) = \begin{cases} -1 + \epsilon^{\eta \kappa_p (r_p(v) - R)} & v > 0 \\ -1 + \epsilon^{\eta \kappa_n (r_n(v) - R)} & v < 0 \end{cases} \tag{2.6}$$

An example of the model behaviour can be seen in Fig. 2.7 which is adapted from an example presented by Stathopoulos et al. (2017). The model presented in Messaris et al. (2018) is capable of emulating accurately the behaviour of RRAM devices. Although it is build with specific RRAM technology in mind, as exhibited by the combined findings of Messaris et al. (2018); Stathopoulos et al. (2017), the model itself is very flexible and can be adapted to accurately simulate the intrinsic behaviour of many different implementations of RRAM devices, as shown by Messaris et al. (2018). Furthermore, the modeling is focused in many aspects of the intrinsic behaviour of the RRAM devices such as non-linear behaviour for the below-threshold region of the RRAM current-voltage characteristic behaviour.

An important note is that, similarly to some other modern RRAM models, the model by Messaris et al. (2018) is developed in Verilog-A and can be easily integrated into circuits and systems simulations in Cadence Virtuoso environment (Spectre simulation). The use of Verilog-A to describe the RRAM behaviour is important since it is a new emerging standard for device modeling in the semiconductor industry and enables the easy and flexible simulation of electronics.

Furthermore, one of the most important features of the model of Messaris et al. (2018) is the capability to be fitted to specific RRAM device response. The model can be adapted to specific observed RRAM behaviour by changing appropriately the fitting parameters of the model through a specified fitting process. The fitting process includes the measurement of the RRAM's response under specific stimulus, as shown by Messaris et al. (2018). Additionally, an algorithm has been developed to translate the data from the fitting measurements to the fitting parameters of the Verilog-A model discussed and showcased byMessaris et al. (2018). This fitting process alongside the high-precision of the model enables the simulated instances of the RRAM devices to capture many interesting parts of the RRAM device's behaviour such as the non-linearity, etc.

2.7 RRAM-based Logic Circuits

Although we have discussed the importance of emerging technologies, such as RRAM devices, we need to introduce some of the most importance design methodologies for logic and memory circuits using RRAM. Through the exploration of RRAM-based circuit designs a better understanding of how conventional designs of memory systems can be adapted towards implementing neuro-inspired modes of computing, as shown by Zidan et al. (2018). The RRAM-based logic circuits have recently been adapted to so many implementation that the organisation into circuit families is necessary, as shown by Mehonic et al. (2020). Additionally, the introduction of specific design methodologies of employing RRAM devices in circuits need to be explored before delving into the different circuit groups.

Although RRAM devices exhibit important traits compared to conventional memory components, the specific characteristics of RRAM need to be examined towards being capable of integration into circuits and systems. Due to the inherent analogue nature of the RRAM devices, a networks of RRAM-based circuits is essentially a resistive network where nodes are connected with a variable programmable resistance, as exhibited by Stathopoulos et al. (2017). This can create problems in dense memory arrays (e.g. crossbar arrays) when attempting accessing/reading specific parts of the RRAM-based resistive networks (i.e. accessing in parallel a memory word-line or multiple word-lines for IMC schemes). This is due to the existence of parasitic paths in the network and the lack of sufficient control to guide the accessing signals over the required path by using only the RRAM devices themselves. The above-mentioned requirements can be fulfilled by the usage of a versatile and programmable basic building component inspired by conventional DRAM memory design (i.e. 1-transistor-1-capacitor), the 1-transistor-1-RRAM (1T1R) composite circuit which is widely used by many different implementations, as shown for example by Ielmini and Wong (2018); Mehonic et al. (2020). An example array of such structure can be seen in Fig. 2.8. The 1T1R can be considered as a fundamental memory and computing circuit which is based on the serial connection of its digital mask, implemented by the MOSFET part, that is enabling or disabling the access to the RRAM device and the analogue weight, implemented by the programmable RRAM element acting as a trimming element that increases/decreases the contribution of the signal passing through. The 1T1R unit can naturally perform the multiplication of a voltage signal passing through a binary transistor-based mask and the RRAM-based weight (i.e. physical occurrence of Ohm's law). In other words, we have two current flow regulation elements, one acting as a controllable access point for the second element, the non-volatile memory. This primitive computational unit can be used to build resistive networks that can map complex logic functions, as shown by Mehonic et al. (2020); Sebastian et al. (2020); Maan et al. (2016).

Many different RRAM-based logic gates have been proposed using different configurations of RRAM-based networks, such as simple and dense RRAM passive arrays (i.e. use of only RRAM devices for the logic and/or memory network), universal hybrid MOSFET-RRAM gates, custom CMOS circuits enhanced with RRAM for specific applications, etc., studied by many researchers including for example the work by Kvatinsky et al. (2014); Serb et al. (2017); Kvatinsky et al. (2012); Xie et al. (2017); Kim et al. (2019); Vourkas et al. (2016); Vourkas and Sirakoulis (2016). The common characteristic for most of the proposed techniques is the use of RRAM devices simultaneously as memory components as well as logic switches. Thus, the main effort towards exhibiting

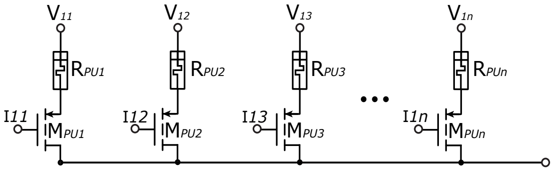

FIGURE 2.8: Schematic of a 1-transistor-1-RRAM (1T1R) array that is employed for building active RRAM-based memory arrays. The use of accompanying MOSFET devices as selector devices enable the better control of the access of the RRAM devices and eliminates issues such as the parasitic sneak-path problem that occurs in passive arrays Vourkas et al. (2016).

RRAM-based logic is focused around the implementation of IMC systems. The RRAM as logic switch can be operated both as digital or as analogue (multi-valued) computing element depending on the device technology as well as on the logic/control scheme employed per computing paradigm. With the RRAM devices being essentially a candidate for future non-volatile memory components the need for RRAM-based circuits to be adapted in designs easily integrated within a dense memory array is in many cases an important parameter of the design methodology. As discussed previously, the crossbar array organisation is one of the most widely used topologies for RRAM logic networks since it provided the maximum component density for a given area of integration, as shown by Papandroulidakis et al. (2017); Xie et al. (2015); Truong et al. (2016a); Hu et al. (2014). At the same time, other designs are less focused towards adapting logic inside memory but instead follow a different method in implementing IMC-like computing schemes into logic data-paths. Hence, many novel RRAM-based computing schemes are introducing RRAM devices into logic circuits, thus bringing neuro-inspired elements into standalone gates. Of course, the different design configurations depend mainly in the need and/or capability to implement massively parallel simpler gates or fast serial complex logic functions. In both methods, the RRAM devices greatly enhance the capabilities of the circuits (for the cases of adapting existing designs with additional RRAM devices) or provide novel ways of computing.

The different implementations can be categorised based on how they operate (voltage or resistance -based logic) as well as if they are used for storing/computing binary or multi-bit information. Many of these features are dependent not only on the application under test but on the actual RRAM technologies employed for each implementation. It is worth mentioning that in many cases other technologies similar in functionality with RRAM are employed in similar circuit configurations (i.e. Phase Change Memory (PCM) is another important emerging Non-Volatile Memories (NVM) that is considered part of the memristor family of technologies). With regards to the main design methodologies for NVM-based logic circuits and systems, most of the related work can be considered essentially as technology-agnostic with the selection of a specific NVM

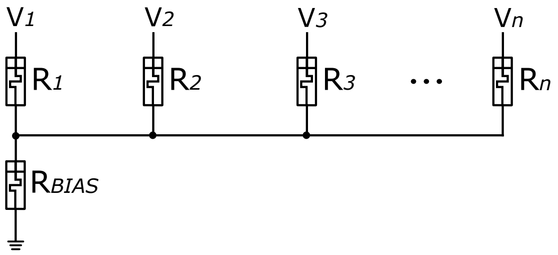

FIGURE 2.9: Schematic of a RRAM-based passive array that can implement IMPLY logic.

technology defining the exact requirements and limitations of the design under test. In the work by Kvatinsky et al. (2011, 2014), some interesting RRAM-based logic circuits are proposed showing the capabilities of resistive networks to map logic functions. An important RRAM-based logic has been proposed by Kvatinsky et al. (2012) in the form of the MAGIC gates. A MAGIC gate is operated in a digital fashion making use of a high ON/OFF state ratio of the RRAM devices to form resistive networks that perform logic operation. The operation of MAGIC gates requires the programming of RRAM during the logic operation (also known as 'stateful' logic). Another important technique is the IMPLY logic that can be easily mapped in a RRAM-based array, as showcased by Siemon et al. (2019); Linn et al. (2013), to perform massively parallel IMPLY operations cascading more complex logic functions through multiple computation steps. These techniques requires high ON/OFF RRAM state ratio and the usage of programming voltages (usually much higher that reading voltage pulses) to perform logic operation in-situ. Logic design techniques similar to the IMPLY-based logic mapping on RRAM-based memory arrays have been proposed by Rahman et al. (2016); Lalchhandama et al. (2016). Other work that base their primitive gates in the MRL design Kvatinsky et al. (2012) have been proposed showcasing how additional circuits and systems can be implemented using this low-complexity RRAM-based networks, as shown by Qu et al. (2019); Emara et al. (2016); Teimoori et al. (2016).

Another prominent method of implementing logic circuits using RRAM can be found in the form of dot-product based operations with program-once-read-many operation scheme applies. This method is usually found in RRAM-based ANNs solutions as well as in other neuro-inspired systems whose computing greatly depends on massive parallelism of operations such as multiplications, as shown by Hu et al. (2018, 2016b); Lastras-Montaño et al. (2017). One of the most important and low-complexity logic circuits of this category is the TLG which can employed to perform massively parallel neural emulation operations inside the memory, as shown by Mozaffari et al. (2016); Mozaffari and Tragoudas (2018). Although there are many different techniques included in this family, the main computation concept is based in the neuro-inspired RRAM-based weighted multiplication and the use of a neuron emulator including thresholding elements (i.e. CMOS inverters, latches, operational amplifier-based comparators, etc.), as shown by Papandroulidakis et al. (2018); Dara et al. (2013); Cheng and Strukov (2012).

Modeling a neuron behaviour into circuits, that fires when the input reaches a threshold, has resulted in the TLG design paradigms that have been the basis of many CMOS-based ANNs implementations, as shown by Beiu (2000); Beiu et al. (2003b); Bobba and Hajj (2000); McCulloch and Pitts (1943). Scaling issues of the MOSFET/CMOS technologies and reconfigurability limitations, both important traits for ANN design towards implementing large networks that are easily re-programmable, resulted in a continuous search for emerging technologies that could eliminate or at least mitigate these issues, as discussed in Section 2.4. As showcased in Section 2.6,it has been found that some of the best synaptic weight emulations can be implemented using emerging non-volatile memory (NVM) technologies, such as PCM, RRAM, etc. and other technologies that are part of the memristor family. By using these components many neuro-inspired computing concepts gain important new traits by taking advantage the benefits of NVM memories, such as the small area of integration, low-power non-volatile operation and multi-state (i.e. capable of handling analogue information) storage per single cell. The NVM traits of nanoscale size and analogue behaviour helped with the area and power constraints of large ANN networks since these devices were able to emulate, in a much more efficient manner, what was previously achieved through large MOSFET-based circuits. Many such implementations are centred around the RRAM technology which can employed efficiently as synaptic emulator into the in silico neuro-inspired classifiers, as shown by Krestinskaya and Pappachen James (2018); Mehonic et al. (2020), thus TLGs as well Maan et al. (2016).

As discussed previously, the 1T1R can be used as a primitive computing element of RRAM-based arrays. The employment of 1T1R networks in ANN implementation, a similar primitive block to other RRAM-based logic solutions, is due to the use of RRAM-based crossbar arrays in the physical computing of MAC operation. More specifically, the 1T1R array can easily perform a physical parallel MAC operation through the application of multiple voltage signals at the bitlines of the array and with the final accumulated output at the word-line. The main operations of multiplication and accumulation are performed physically through the Ohm's law and the Kirchhoff's law, respectively, as discussed by Jeong and Lu (2018). The accompanying MOSFET devices are used usually for logic masking purposes, thus selecting which RRAM is part of the physical multiplication. Due to these inherent characteristics of both the RRAM technology and its preferred organisation into larger hybrid RRAM-MOSFET systems, researchers have been able to exploit more efficient methods of building reconfigurable and brain-inspired circuits and systems. The 1T1M is a better building block for developing memory-centric computer systems and architectures (such as ANNs etc.) where the co-locality of logic and memory and on-the-fly reconfigurability are essential. The non-volatility and scalability of RRAM devices are important for ANN applications, as discussed by Ielmini and Wong (2018); Sun et al. (2018a). Simultaneously, the ability of the RRAM-based circuits to replace relatively complex CMOS circuits like CMOS-based memory cells, CMOS adders etc. can potentially lead to miniaturisation of previously slow and power-hungry systems, as shown by Tran et al. (2012); Adhikari et al. (2012).

An important part of the RRAM-based ANN design is the exploration and analysis of the simplest circuit implementations that operate under the notion of data classification and can easily be integrated inside the memory, thus showcasing potential for being scaled-up to an IMC-based ANN architectures, as highlighted by Maan et al. (2016). These RRAM-based TLG designs where the basic concept of replacing complex and large CMOS-based synaptic and neuron circuits with RRAM-based equivalent ones. The implementation of some of these concepts is aimed mainly towards designing primitive synapse-neuron emulator for use in large ANNs or simply the design of basic computing blocks for developing IMC accelerator working alongside conventional computers. Regardless of the exact aim of each implementation, all of the suggested solutions are build around some common methodologies and all of gates can be assigned in a unified family of logic circuits.

A low-complexity (thus capable of being integrated into small chip area) RRAM-based TL circuit implementation was introduced by Rajendran et al. (2011). In that design each input of the input vector is connected to a composite circuit with a RRAM device (implementing fully analogue synaptic emulators) and a current mirror circuit in series. The RRAM device is used as the weight based on which the input is multiplied, while the current mirror ensures that there will be no short between two inputs in one input is connected to logic '1' and the other to logic '0', thus ground. The threshold value is represented here by a current source connected to a current mirror, as shown by Rajendran et al. (2010, 2011). All the outputs of the current mirror circuits (both for the inputs and the threshold) are connected to a common node which is used to drive a CMOS inverter network. The first CMOS inverter is used as a sense output circuit thus performing the thresholding operation based on the composite current of the common input node (i.e. circuit node that all RRAM-MOSFET branches are connected). The rest of the CMOS inverters in the output network are connected serially to the sense output inverter and are employed for isolation as well as for performing the complementary output of the gates canonical output (i.e. inverter-driven comparison result). As proposed by Rajendran et al. (2010, 2012), such TL circuit have the potential of replacing Look Up Tables (LUTs) due to lower power consumption and much smaller area of integration.

Another important TLG design is the programmable TLG proposed by Gao et al. (2013b). The circuit uses a hybrid RRAM-CMOS circuit that is showcased in performing simple TL operations such as NAND and NOR, thus implementing TL-based universal gates for replacing larger CMOS networks. Compared to traditional implementation of CMOS-based universal gates, this implementation has a higher fan-in, thus being

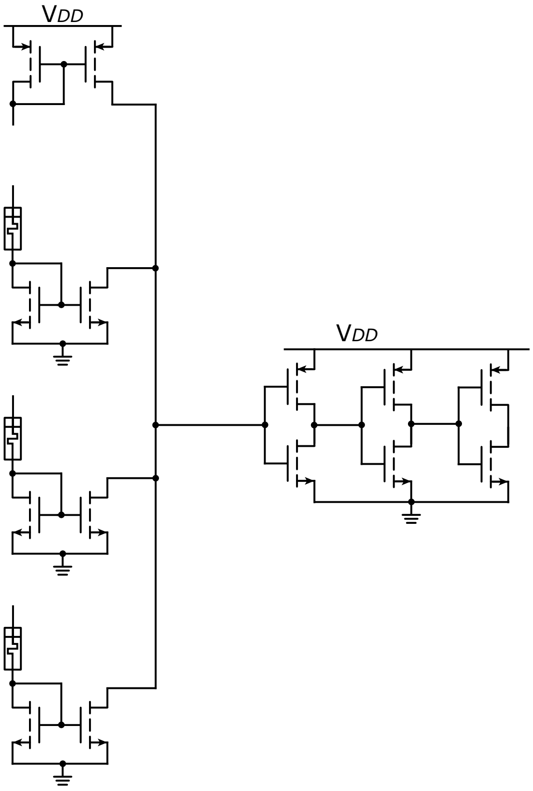

FIGURE 2.10: Schematic of Memristor-based Threshold Logic (MTL) design proposed by Rajendran et al. (2010).

able to perform a TL function for large input vectors. The RRAM-resistor input network is performing a ratioed logic operation where the RRAM devices are connected to the input vector, while the resistor defines the pull-down contribution. The CMOS D Flip-Flop (DFF) is used as the comparator part of the TLG that senses the output of the intermediate node. The in-place reconfigurability of the gate is crucial for its high adaptability to workflows, as shown by Maan et al. (2016).

Although the main focus of this work is around circuit-level implementations of hybrid RRAM-CMOS neuro-inspired designs, it is important to provide the basic computing architecture where these circuits will be incorporated. The main computing concepts followed for the organisation of these gates is usually found inside memory system (i.e. IMC paradigm). Thus, the hybrid RRAM-CMOS circuits are usually part of a larger RRAM-based IMC architecture, as discussed by Ielmini and Wong (2018); Sebastian et al. (2020); Mehonic et al. (2020); Gallo et al. (2017). The main IMC neuro-inspired features were presented in Section 2.3 but it will be useful for establishing the design methodologies specifically of RRAM-based IMC towards providing a brief analysis of higher-level RRAM-based system implementations.

An interesting RRAM-based IMC design is proposed by Halawani et al. (2019). In this work, Halawani et al. (2019) showcase a RRAM-based crossbar memory is used for designing ANN accelerators in real-time search engine applications. A low-complexity RRAM networks (forming a simple voltage divider topology) is used to perform XNOR Boolean operations using appropriate voltage reading signals as well as specific resistive memory configurations (thus implementing a 'stateful' logic scheme). The concepts are combined to showcase a novel CNN for feature extraction.

Another important RRAM-based IMC system is showcased by Gallo et al. (2017). In this case, Gallo et al. (2017) showcase an mixed-precision RRAM-based system that bypasses the limited precision of the RRAM devices. The RRAM-based IMC is capable of massively parallel and high-throughput operations. This speed and parallelism is exploited by the RRAM-based IMC accelerator to perform multiple operations on the same data until the error with the pre-calculated ideal output is very small. An integrated CMOS-based high-precision systems is also used near the IMC for a final processing.

A similar category of RRAM-based IMC accelerators can be found in the work of Shafiee et al. (2016). In this work, Shafiee et al. (2016) shows a CNN accelerator implemented using RRAM devices for IMC analogue computing. The main computing performed by the RRAM-based memory is the dot-product operation. The operation is performed in the analogue domain with CMOS data converters employed to connect the analogue IMC with the rest of the conventional CMOS-based CNN implementation.

From a brief examination of the above-mentioned IMC designs it can be identified how the hybrid RRAM-based neuro-inspired circuits are highly tuned for memory-centric implementations and can provide improvements over older CMOS-based IMC computers, as discussed by Ielmini and Wong (2018); Sebastian et al. (2020). RRAM's analogue nature and easy integration in massively parallel arrays, that can perform physical computing, as shown by Xia and Yang (2019); Burr et al. (2017), are key features of many RRAM-based IMC implementations. Thus, the introduction of DRAM-like crossbar memory array to organise dense RRAM-based IMC systems is considered by many the main topological design of novel neuro-inspired computer architectures aimed for the basis of novel IMC paradigm.

2.8 Conclusions

In this technology background examination, I identified the main computational concepts that define the neuro-inspired circuit and systems design methodology. A selection of different neuro-inspired circuits was presented and discussed towards analysing and disseminating the practical implementation methodology of these family of circuits. An initial analysis of the MOSFET/CMOS designs was presented and after the introduction of the RRAM device theory and technology a group of hybrid RRAM-CMOS neuro-inspired circuits were presented towards showcasing the enhanced capabilities of the hybrid designs. The review presents in detail some of the most important memory and logic co-location systems and explains in detail how RRAM can have an important role in expanding their capabilities for data processing. Another area of focus, highlighted in this review, is the enhanced reconfigurability provided by such hybrid system design with examples of what neuro-inspired reconfiguration in electronics looks like and how the traits of RRAM can help in designing novel solutions. A main example of logic gates that gather all the aforementioned traits and benefit from RRAM are the TL-based circuits, with an extensive review of their CMOS-based design and their memristively-enhanced counterparts being provided in this chapter.

Although there are many important computing concepts proposed based on the use of 1T1R computing-memory array, the majority of such circuits are implemented and studied mainly in computer simulations, as highlighted by Maan et al. (2016). Hence the practicality of these suggested designs needs experimental demonstration using proof-of-concept circuits of these concepts. This is especially important when novel and hence more mature RRAM technologies are introduced, thus the experimental validation of many RRAM-based circuits, that require specific traits not available at the time of the design proposal, is possible. It is really important after establishing the fabrication of a stable continuously tuneable RRAM technology to study the practical real-world constraints found in the physical hardware implementation of such circuits and research in depth what are the requirements and what are the real-world responses of the RRAM based TL.

Neuro-inspired circuits are based on simple and fundamental data processing operations such as vectors multiplications, accumulation and comparison, operations that can be found in relatively simplified versions of the showcased model for synaptic and neural activity, i.e. in TLGs. Although, MOSFET technology is used extensively for many different in-memory and neuro-inspired implementations, the addition of emerging technologies is considered highly beneficial. This is especially true if we closely examine the circuits based on transistors-RRAM presented throughout this chapter. An important concept that needs to be thoroughly investigated is deconstruction of the more complex neuro-inspired gates towards identifying primitive circuit structures that can be exploited in providing new levels of reconfigurability in systems design. Towards satisfying that need and based on the existing work I am proposing my contributions regarding primitive RRAM-based data processing circuits and systems in the following chapters.

Chapter 3

RRAM-Based Reconfigurable MAC Circuit for IMC systems

In this chapter, I am showcasing the design and behaviour of a RRAM-based Multiply and Accumulate (MAC) circuit. Initially, I am briefly discussing some of the design methodologies of the state-of-art Multiply-Accumulate (MAC) circuits alongside their most common architectural organisation into IMC accelerator systems. Then, I am presenting and discussing the mapping of the different operational modes of a low-complexity hybrid RRAM-CMOS circuits based on the placement of the components in the topology. In order to develop a system that accommodates the ever-increasing requirements for accelerator systems, such as IMC accelerators, I am proposing the design of a primitive digital-in-analogue-out MAC circuit mixed signal gate that is inspired by a simple memristor-based linear neuron model circuit. This gate showcases all the necessary traits towards being employed as a primitive and programmable logic gate for the next generation of reconfigurable computing systems.

I am providing hardware implementation to validate the RRAM-based MAC circuit that can map different memristor-based logic techniques on that primitive logic gate topology. The memristor logic is implemented as a reconfigurable memory-dependent voltage divider circuit. An adversarial memristor-based voltage divider is used to provide a biasing/thresholding signal against which the intermediate node voltage of the memristor logic topology is compared against. The design and operation of the proposed gate and both experimental measurements of a practical implementation and simulation of an example configuration of the gate are presented. Real RRAM devices are characterised and appropriate fitted parameters are extracted to introduce a realistic memristor behaviour into the simulation environment. Finally, a circuit-level design of a voltage racing Winner-Take-All circuit is simulated based on the fitted memristor models and the proposed RRAM-based MAC circuit with the MAC circuit being employed as a programmable RC generating circuit.

3.1 Multiply and Accumulate Circuits Design Methodology

As discussed in Chapter 2, many of the most demanding computer algorithms are based around the MAC operation. MAC is as a cornerstone logic operation for many computer applications that are based on data classification, filtering, neuro-inspired computing, etc., as discussed by Furber (2016); Indiveri and Liu (2015). Different implementations of MAC circuits have been proposed based on different combinations of weighted or unweighted multiplications with digital or analogue inputs. Most of the MAC circuits and systems suggested over the years are based on conventional MOSFET and CMOS technologies that have defined computers for more than 40 years. Their circuit design is based around the conventional DRAM and SRAM technologies and their fundamental computation concept is based on neuro-inspired circuit design methodologies. One of the earliest examples that showcases the relation between MAC operations and ANNs can be found in the work by McCulloch and Pitts (1943) where the concept of the perceptron was introduced. More recent implementations suggest artificial synapse-neuron circuits with analogue output, as proposed by Douglas et al. (1995); Indiveri (2001); Indiveri et al. (2006). The concept of MAC operation can be found even in the traditional Boolean logic gates where the multi-input AND/NAND and OR/NOR gates can be interpreted as simple unweighted implementations of MAC circuits with an added activation function that compares the relative position of the MAC result to a specific threshold.

A very important family of primitive circuits designed specifically for performing MAC operations is the neuro-inspired artificial neuron category of circuits. Over the last decades, the artificial neuron circuits has been implemented in many different forms for a wide variety of computer applications. Currently with the advent of novel circuit design methodologies and computing devices, neuron emulators have been evolved to become one of the most important hardware solutions for performing massively parallel MAC operations usually inside IMC system accelerators. Many different circuit solutions have been suggested for both analogue and digital computing and for low-complexity or high-complexity circuits (usually depending on the level of neuronal activity emulation). From the many different available implementations, some of the proposed designs are complex and are used to mimic more closely the functionality of biological neural networks, thus emulating a detailed set of neuronal mechanisms, as shown by Indiveri et al. (2011); Indiveri and Liu (2015); Payvand et al. (2018). At the same time, other designs are aimed towards replacing large networks of digital Boolean logic with a more power-efficient digital or hybrid RRAM-CMOS neuron-based circuit with some such examples including Dara et al. (2013); Bobba and Hajj (2000); Papandroulidakis et al. (2018); Leshner et al. (2010).

One of the most important sub-systems in computer architectures are the computing accelerators used to perform faster and efficiently many algorithms that are usually optimised for massively parallel computation of simple fundamental operations, as discussed by Kang and Shanbhag (2016); Zhu et al. (2013); Zhang et al. (2020b); Gallo et al. (2017). In recent years a lot of effort has been placed in developing better hardware accelerators for processing data, as discussed by Kang and Shanbhag (2016); Zhu et al. (2013); Zhang et al. (2020b); Gallo et al. (2017). Additionally, the development of novel big data hardware accelerators has been extensively studied and led to the rise of IMC, as shown by Santoro et al. (2019); Jeloka et al. (2016); Gonugondla et al. (2018). By attempting to match the hardware capabilities of computers with the need to process faster increasingly larger amounts of data, a new effort to develop novel big data hardware accelerators has been a priority for the numerous proposed applications of emerging technologies and novel circuits and systems that derive by incorporating them into computer architectures. Such hardware accelerators need to be low-power towards being compatible with computing systems at the edge where power and area of hardware solutions is limited, as highlighted by Sebastian et al. (2020); Zhang et al. (2020b). At the same time, the computational requirements for these accelerators are ever-increasing, thus novel solutions that are capable of exploiting parallelism in data and perform data processing using mixed signal techniques, when an analogue method is more efficient compared to a digital one, are needed, as discussed by Seshadri et al. (2017); Li et al. (2017).

RRAM can be integrated to implement memory systems with low power consumption, an important advantage for large MAC-based ANN systems, as shown by the recent work of Sebastian et al. (2020); Zhang et al. (2020b). Given the numerous advantages of RRAM device technologies, highlighted by Serb et al. (2018b); Bayat et al. (2018); Danial et al. (2019), such as the capability to store multi-bit information per single device, showcased by Stathopoulos et al. (2017); Sebastian et al. (2018, 2020), much effort has been dedicated to the design of novel post-von IMC systems based on hybrid RRAM-CMOS circuits, with such design examples including Dastanova et al. (2018); Kvatinsky et al. (2012); Papandroulidakis et al. (2017, 2018). Similarly to the main design methodologies of conventional CMOS-based IMC, hybrid IMC systems need to employ memory-centric topologies, such as crossbar arrays, a topological structure that can accommodate a large number of memory-computing RRAM devices as well as support massively parallel operations, thus enabling the acceleration of MAC operations, as shown by Zha and Li (2017); Chakrabarti et al. (2017). The family of IMC designs have been recently reinforced with the introduction of multiple hybrid RRAM-CMOS implementations making use of emerging memory technologies, as shown for example by Santoro et al. (2019); Sebastian et al. (2020); Merrikh-Bayat et al. (2017). Furthermore, another important aspect is the design of electronics systems as reconfigurable and thus capable of being altered into a variety of hardware logic datapaths by exploiting arrays of primitive computing structures, for example the work of El-Chazawi et al. (2008); Rahimi et al. (2015); Koenig et al. (2017), a computing requirement that can be satisfied by employing IMC techniques, as discussed by Meyer et al. (2019); Shafiee et al. (2016);

Liu et al. (2015); Yu et al. (2019). The advent of emerging memory technologies, such as RRAM, have provided new opportunities to develop nano-electronic programmable logic fabric that is power and area efficient, as shown by Hu et al. (2016b); Kumar Maan et al. (2016); Zha and Li (2017); Chakrabarti et al. (2017).

The increasing maturity of RRAM technology has offered an ultra-compact and efficient way of performing element-wise multiplications using physical computing (Ohm's law). Various architectures have been proposed thus far for RRAM-based MAC under various assumptions (see Chapter 2). In this chapter, I study, practically implement and test a minimalistic low-complexity MAC system with the following properties: i) single-ended design for reducing the number of components, ii) exclusively digital inputs, to allow signals to be received from long distances without fear of analogue distortion and iii) analogue output, which preserve the information richness of the MAC computation and can be processed locally before digitisation and broadcasting. I then proceed to show how it is possible to obtain usable results from a system designed using nothing else other than modular 1-transistor-1-RRAM (1T1R) blocks, shunting transistors and a capacitor, before illustrating how easily the system can be used to create a simple Winner-Take-All (WTA) network. Based on the findings of this chapter, it is shown that in the limit of many MAC inputs the incremental cost/input reduces to an extremely frugal 1T1R + parasitic capacitance overheads (including programming infrastructure), that the system can operate at frequencies of 100 MHz with a practicably low load capacitance and that power dissipation of 300 fJ/operation are realistically achievable for a full on-chip implementation.

3.2 1T1R-based Circuits Configuration Map for In-Memory Computing

The composite 1T1R computing array have the potential of being employed as a computationally flexible primitive building block that combines a switch with a resistive tuning element. Depending on the logic function implemented, further digital and/or analogue logic processing can be added at the output of the computing array. In the case of simple digital processing, such as inverters, latches, etc., a binarisation layer can be added at the output of a 1T1R array to project signals onto a discrete binary domain. This is important for enabling the cascading of conventional CMOS circuitry with 1T1R-based circuits and networks. As discussed in Chapter 2, many different configuration of 1T1R circuits can be found in the literature, such as the work on analogue inverters, as shown by Serb et al. (2018a,b, 2017).

By analysis many existing solutions few common design pattern for RRAM-based circuit emerge and are centred around the concept of implementing complex logic functions using low-complexity networks of primitive computing elements, as shown by

Serb et al. (2018a). Based on the map shown in Fig. 3.1, the similarity in the different circuit configurations, used to implement all possible combinations of input and output domains, can be observed. In fact, it could be summarised that what distinguishes a 1T1R-based system designed for digital vs an analogue output is a binary resolution element, such as a thresholding element, here illustrated as an example by an inverter, but it can be configured with any other electronic circuit which response is affected by different voltage levels (for the case of the voltage mode voltage divider structure). Without such element for output classification, the output of the shown structures can in general be analogue, as defined by the multiplication and subtraction (since they pulling to different voltage supplies) of the two 1T1R components. With regard to the differentiation between design for analogue and digital inputs, the contribution of the accompanying MOSFET devices can be amplified in order to enable either a abrupt on/off characteristic (which you can see I the upper part of the table represented by transistors with always well-defined V_{GS} voltages) or instead to operate more as a smooth and ideally linear signal amplifier from the control gate to the main current branch (here shown as source-degenerated transistors yielding a linear-like signal transfer, albeit with a threshold penalty).

Similarly to other proposed IMC designs, the RRAM-based circuit design showcased in this chapter is underpinned by considering how simple networks of the 1T1R computing elements can be added in a design methodology family of similar circuit designs that perform conceptually similar computations and communicate with either analogue or digital data. All the aforementioned circuits share a common trait which is that the main computation is performed in the analogue domain due to the physically analogue multiplication and accumulation taking place in 1T1R computing arrays. As showcased in Fig. 3.1, the computational separation of hybrid 1T1R-based circuits highlights the flexibility of the RRAM technology to be incorporated into many different logic operations based on its connectivity.

Neuro-inspired circuits can easily be integrated into computing systems with the aim of performing power-efficient conventional digital or mixed signal logic and have great potential of being used as the main programmable logic element of primitive logic gates, as shown by Seshadri et al. (2017); Hu et al. (2016a); Chakrabarti et al. (2017). Circuits that perform a logic function mapping in RRAM-based IMC systems can be used as a form of miniaturised associative memories. The associative memories, as discussed in Chapter 2, are essentially performing a mapping from one information domain to another. In this chapter, the RRAM-based MAC circuit under test is receiving a digital word, that can be interpreted as a vector of neural spikes, and responds with an analogue MAC result (the result of the parallel input vector to weight vector MAC operation), which is a physical method to create information in a compact format. Further processing of the analogue output can be performed by cascading circuits (such as thresholding elements, etc.) depending of the specific application that these circuits are employed for. An important such example can be found in the work of TLGs, as shown by Sun et al. (2018b); Alibart et al. (2016); Papandroulidakis et al. (2018), where a basic and simplified model of synaptic-neural operation is used to perform primitive inference operations. Hence, primitive MAC circuits can be used to develop neuro-inspired circuit designs for IMC applications, as shown by Valavi et al. (2019); Verma et al. (2019); Halawani et al. (2019); LIu et al. (2020). The operations are capable of being naturally transferred between information domains (binary, continuous) and thus converting the information into formats that can be easily harnessed by other cascading circuits. This lack of restriction of the operated data domain can lead to significant advances in the performance of future hardware accelerators, as shown by Zha and Li (2017).

Another important trait of incorporating memristors in a reconfigurable primitive circuit is that given its analogue nature, having multi-bit resistive switching behaviour, the employed hybrid logic circuits can have the potential of being operated under mixed-signal domain. Mixed-signal operation is a highly valuable capability for introducing a new era of nano-scale analogue electronics since some operations, such as MAC operations, can be accelerated if performed in the analogue domain while other operations, such as comparison, is best suited for fast CMOS circuits. RRAM-based circuits enable mixed-signal IMC systems to be easily integrated with conventional digital electronics while retaining the aforementioned power and area efficiency, as shown by Li et al. (2014). Additionally, the organisation of such gates need to be accommodated into such a topological structure that will enable the operation in a massively parallel manner, thus exploiting the intrinsic parallelism found in many data processing applications. The usage of crossbar arrays can lead the neuro-inspired circuits under test to further accelerate cornerstone operations, such as MAC, in IMC systems by introducing low-complexity networks capable of massively parallel operation.

In Section 2 the existing landscape of memristor logic techniques and a set of novel methods for enhancing logic circuit with RRAM devices for data processing has been showcased and discussed (see Fig. 3.1 in Chapter 2). The focus of the proposed solutions was on data processing that connects the analogue signal world to the digital one, thus developing computing paradigms for data mapping between binary and multi-level inputs/outputs. To that effect, implementations already developed for analogue-in-analogue-out 1T1R-based circuits and especially the inverse digital-in-digital-out 1T1R-based logic gates, as shown by Serb et al. (2018a), are used as guiding examples of 1T1R-based circuit design methodology for designing a new digital-in-analogue-out 1T1M-based primitive circuit for MAC operations. The design and analysis of these circuits will enable us to provide complementary computing schemes to the existing groups of memristive logic gates thus supporting the completeness of the available memristive circuit solutions for data processing.

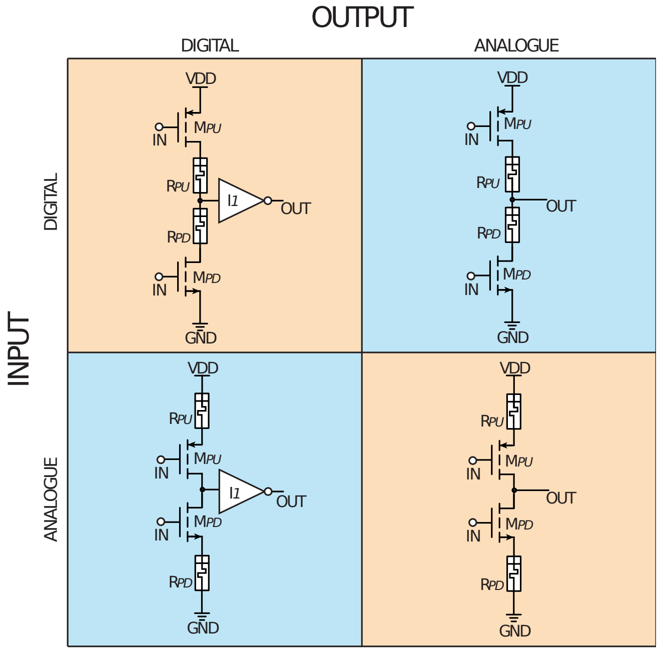

FIGURE 3.1: Map of different primitive RRAM-based circuit configurations of elements introducing a hard switch/selection (i.e. MOSFET devices) and elements introducing a continuous value of contributions (i.e. RRAM devices). Additional hard-threshold circuits (such as CMOS-based inverters and latching elements) can be used to binarise an output signal generated by the hybrid RRAM-MOSFET circuits. By enabling small differences in the circuit configuration and circuit control scheme a variety of reconfigurable circuits can be designed that will be fitted towards receiving inputs and emitting outputs in our choice of analogue or digital format. Given the analogue nature of RRAM devices operation and behaviour, the main computation (i.e. MAC operations) is always performed in the analogue domain. The implementation that connects the same signal domain is considered as non-Converting Logic (non-CL), while the circuits that bridge different signal domains are considered as Converting Logic (CL) gates. In the map, the non-CL circuits are denoted with orange while the CL circuits with blue. This map of primitive hybrid RRAM-CMOS circuits showcase the computationally flexibility of such circuits and exhibit their capability in being employed into novel implementations of analogue or mixed-signal associative memory systems.

3.3 MAC Circuit Design

In this section, I present the version of the RRAM-based MAC circuit that implements a simplified version of a synapse-neuron model. I can employ the MAC circuit as a primitive cross-data domain logic gate between the digital vector and analogue output. The setup and results of hardware implementation of the RRAM-based circuit are showcased and afterwards an example proof-of-concept of a circuit-level Winner-Take-All (WTA) network is tested in simulation (Cadence's Virtuoso Spectre environment) using the computationally flexible RRAM-based MAC circuit as reconfigurable RC delay component with measurement-based extracted RRAM models.

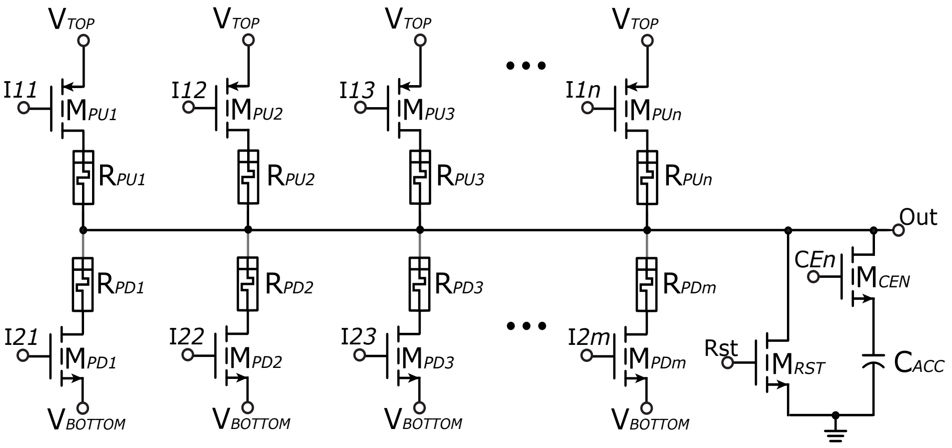

FIGURE 3.2: Schematic of the RRAM-based Multiply-Accumulate (MAC) circuit implementation of this work. The multiplication operation performed per each 1T1R composite element depends on the RRAM memory contents while the accumulation of all the active 1T1R element contributions is performed naturally in current form through the common output node. The circuit can be employed as a form of simplified heteroassociative memory system used to map a digital memory word to an analogue output value. The input digital word can be considered as a train of digital spikes send as stimulus to the hybrid RRAM-CMOS associative memory thus further enhancing the concept of neuro-inspired design in such circuits. The output being could be stored in a integrating capacitor to retain the information-rich analogue signal. The heteroassociative memory mapping capabilities can be easily expanded by increasing the size of the 1T1R memory arrays, thus the number of 1T1R component for the pull-up and/or the pull-down network. Furthermore, this circuit can be easily integrated as part of a crossbar memory array (i.e. the pull-up and pull-down 1T1R-based networks can be easily integrated within a RRAM-based crossbar memory topology), thus it can easily be incorporated into massively parallel IMC system architectures.

The RRAM-based MAC circuit, shown in Fig. 3.2, is based on the primitive but computationally flexible circuit of a the parallel connection of multiple 1T1R composite components to form a 1T1R-based computing array. The MAC circuit (being implemented conceptually as part of a IMC system) is performing essentially an in-memory mapping between a digital word that controls the accompanying MOSFETs of the 1T1R-based array and an analogue value that is generated through the corresponding voltage-divider action between the 1T1R components pulling to V_{DD} and the 1T1R components pulling to GND. This can be interpreted as a miniaturised associative memory that receives a binary word and generates a corresponding analogue value. An additional integrating capacitor alongside a nMOS used to discharge the capacitor can be used for operating the same circuit in charge accumulation mode (strobe the pMOS part of the input vector, then strobe the nMOS part and finally read the output as the resulting voltage accumulated at the capacitor). Similarly to fundamental neuro-inspired design methodology, the circuit is based on the design of a primitive neuron models with the pull-up 1T1R components modeling the input synapse connections and the pull-down 1T1R components creating a composite "inhibitory" biasing for the output of the neuron.

Regarding the operation of the MAC circuit under test, the similarity to a synapses-neuron emulation is explicit: the input stimulus is in the form of binary event input vectors (similarly to spiking pulse trains) programming/reading the weights of the artificial synapses and the output itself is treated fundamentally as analogue (due to the physical analogue processing imposed by using RRAM devices) and can be thought of as the generalised membrane potential (driven by the result of the MAC operation) of the neuron emulator circuit. Naturally, multiple possible implementations exist, including presence or absence of an output capacitor, additional circuitry for modifying the effective activation function etc. The very same circuit can be also set-up and interpreted as a standard Digital-to-Analogue converter (DAC) (where, for example, binary-coded RRAM devices to map with standard fixed steps the conversion could be used). This is where the memristor trimming helps with obtaining an accurate conversion function. The differentiating factor in the showcased MAC circuit design approach compared to other dot-product-based RRAM-MOSFET designs is mainly found in the use of a capacitive output that maintains all the information-rich, analogue signal instead of digitising the output immediately and thus flattening out the information before it can be used.

3.4 MAC Circuit Operation and Experimental Measurements

I demonstrate the operation of the MAC circuit through two experimental measurements: i) one hardware implementation with off-the-shelf discrete components and real $Pt/TiO_x/AlO_x/Pt$ RRAM devices, and ii) a hybrid experiment of a hardware measurement and a Cadence's Spectre simulation on 65nm CMOS, with the RRAM models being extracted from the same RRAM devices that were used in the hardware measurement. Type NDP5020P (1H10AA) pMOS and SUP85N02-03 (T32BAA) nMOS discrete transistors were used for the practical circuit implementation. The control signals for the pull-up and pull-down networks were generated through microcontroller programming (MBED NXP LPC1768 MCU module). The experimental setup was based on the use of probe-station where a wafer with in-house fabricated RRAM arrays was employed. Devices from topologies of standalone RRAM were connected to a prototyping breadboard that includes the discrete transistor components and the microcontroller board. The microcontroller was connected to the gates of all transistors, including the pull-up/pull-down nodes to control the digital input vector and read the analogue voltage value from the intermediate node of the circuit. An overview of the experimental setup is showcased in the form of block diagram in Fig. 3.3.

The MAC circuit can have multiple modes of operation by virtue of being implemented as part of the 1T1R array. In "combinatorial mode" (direct voltage divider) the pull-down 1T1R network is always co-activated alongside the pull-up 1T1R network and the integrating capacitor is disconnected from the circuit. In "integrative mode" the

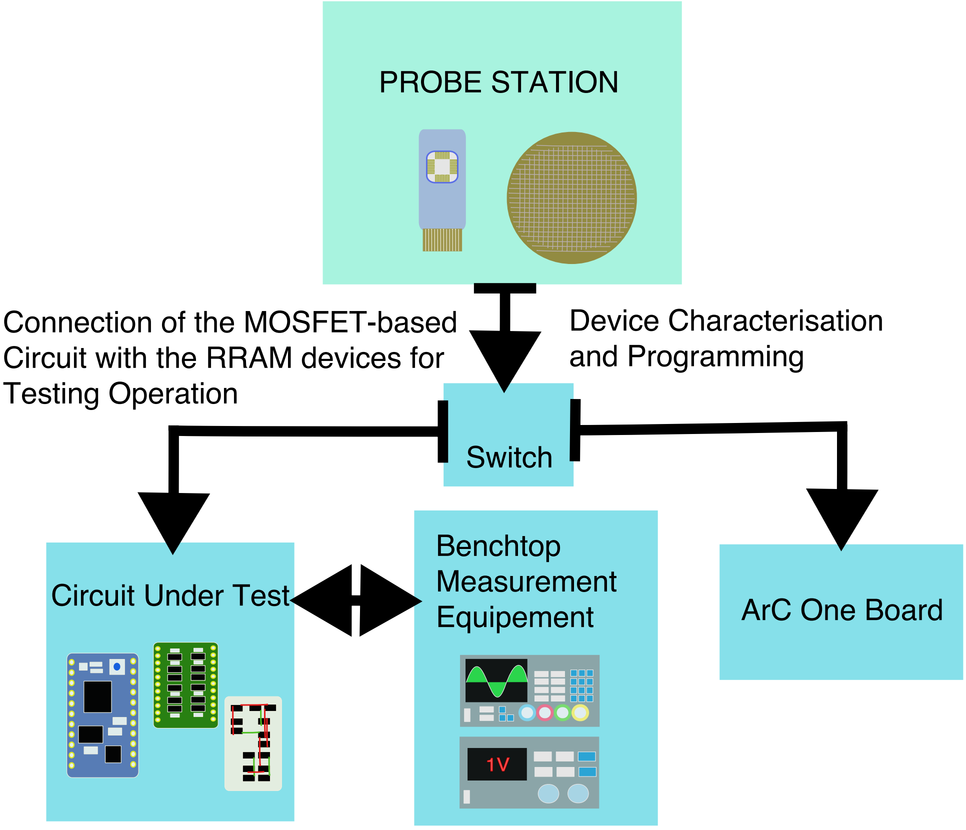
PROBE STATION
Connection of the MOSFET-based Circuit with the RRAM devices for Testing Operation
Device Characterisation and Programming
Switch
Circuit Under Test
Benchtop Measurement Equipement
1V
ArC One Board

FIGURE 3.3: Block diagram of the setup for the hardware experimental measurements. The hardware experiments consist of programming appropriately a specific number of RRAM devices. The RRAM devices are accessed through a probe station and programmed through the use of the ArC One instrumentation board (ArC Instruments, UK). When the device preparation phase is finished, the RRAM devices (still accessed through the probe station are disconnected from the ArC One board and connected to the appropriate hardware prototype that includes the other part of the hybrid RRAM-CMOS circuit under test, thus the MOS components. Alongside the circuit under test other smaller accompanying boards (e.g. voltage dividers for lowering the voltage levels), micro-controller or single board PCs (e.g. RPi for sending appropriate control signals) are employed. Additional bench-top measurement equipment (e.g. voltage sources, oscilloscopes, etc.) is also used to test the circuit and measure its response.

pull-down 1T1R network may or may not be co-activated with the pull-up network but the integrating capacitor is connected to the circuit. Depending on the whether the pull-down 1T1R network is connected or not, the capacitor acts either as a dynamic damper (”damper mode” - damper voltage divider) or as an integrator (”integrator mode” - summing capacitor), respectively. In the case of the integrator mode there is never a DC path to GND. These configurations showcase the operational flexibility of the circuit.

An initial example is shown Fig. 3.6 where experimental measurements of the RRAM-based MAC circuit are presented. For the configuration of the gate in this example

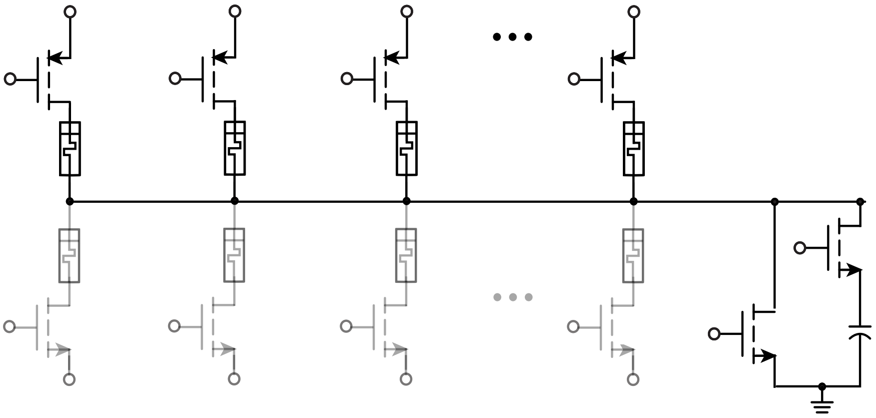

FIGURE 3.4: Schematic of the RRAM-based MAC circuit focusing on the integrator mode of operation. In this mode the pull-up 1T1R network is used to charge the accumulating capacitor and store the weighted sum in an analogue format. This mode is essentially performing a current mode dot-product operation.

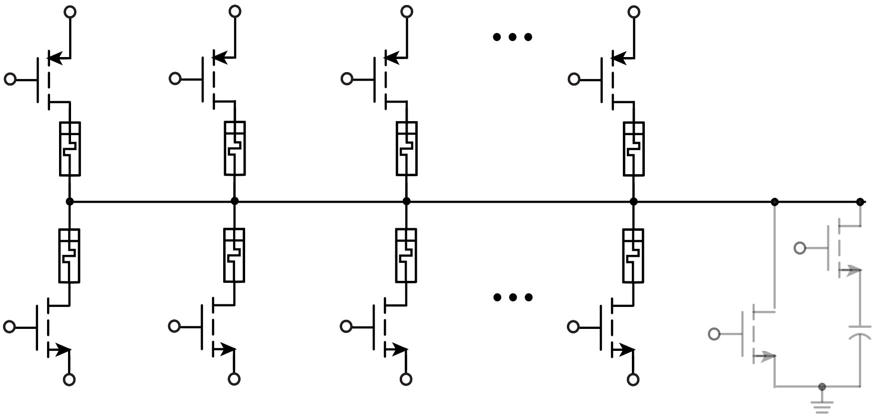

FIGURE 3.5: Schematic of the RRAM-based MAC circuit focusing on the combinatorial mode of operation. In this mode, the output accumulating capacitor is disconnected since for the specific operation mode a fast voltage divider -based operation of the RRAM-based network is required. The pull-down 1T1R network is connected and act as a biasing network that subtracts from the pull-up signal.

setup, 3 pull-up 1T1R components (PU: $5.5k\Omega, 11k\Omega, 11.5k\Omega$) and 2 1T1R pull-down components (PD: $7.3k\Omega, 6k\Omega$) components were employed. The logic mapping operation between the digital input vector and the analogue output of the primitive MAC circuit is shown in Fig. 3.6b.

Regarding the programming of the RRAM elements inside the circuit: i) To program a pull-up RRAM device, I use an additional transistor to set the intermediate node to an appropriate voltage V_{SET} and connect the pull-up supply terminal to GND for a SET operation. For a RESET operation, I connect a suitable voltage V_{RESET} to the pull-up supply terminal and connect the intermediate node to GND. ii) To program a pull-down, I use a similar configuration. For a SET operation, I connect V_{SET} to the intermediate node and connect the pull-down terminal to GND while for the RESET operation, I connect V_{RESET} to the pull-down terminal while I connect the intermediate node to GND. The circuit was operated in combinatorial (direct voltage divider) mode. The pull-up MOSFET gate voltages should be set within a range that will not inflict resistive state change in the RRAM devices. The pull-down network can also be engineered with such protections, but in this case (Fig. 3.2) I illustrate an example where the pull-down is optimised for current drive. Also note that the mismatch in the drive MOSFETs of the pull-up 1T1R network can to some degree be compensated by adjusting the resistive states of the RRAM elements.

For the Cadence's Virtuoso Spectre simulations, I used an existing instrumentation platform, as shown by Serb et al. (2014), (ArC Instruments, UK) which has a bespoke software module, showcased by Messaris et al. (2017), for extracting RRAM device models, exhibited in the work of Messaris et al. (2018), (and briefly shown in Chapter 2 through 2.3-2.6 equations). For the case study, four RRAM devices were tested and their model parameters were extracted, which I then used in the Verilog-A version of the model for subsequent Spectre circuit simulations with the results shown in Fig. 3.7 (more information about the model fitting process can be found in the work of Messaris et al. (2018)). Because in this work I consider static RRAM devices (i.e. no resistive state switching is induced during operation), I am only interested in the extraction of the parameters a_p, a_n, b_p and b_n which fully define the static I-V characteristic. As discussed in Chapter 2, the I-V equations used in the model are reproduced from Messaris et al. (2018) for convenience:

$$i(R,v) = \begin{cases} a_p(1/R)sinh(b_p v) & v > 0 \\ a_n(1/R)sinh(b_n v) & v \leq 0 \end{cases} \tag{3.1}$$

The device parameters (a_p, a_n, b_p, b_n) I extracted and used in this work are shown in Table 3.1. Note that R in equation 3.1 is the standardised resistance of the device and it is measured at V_{READ}=700mV read-out voltage for this work. The typical V_{READ}=700mV

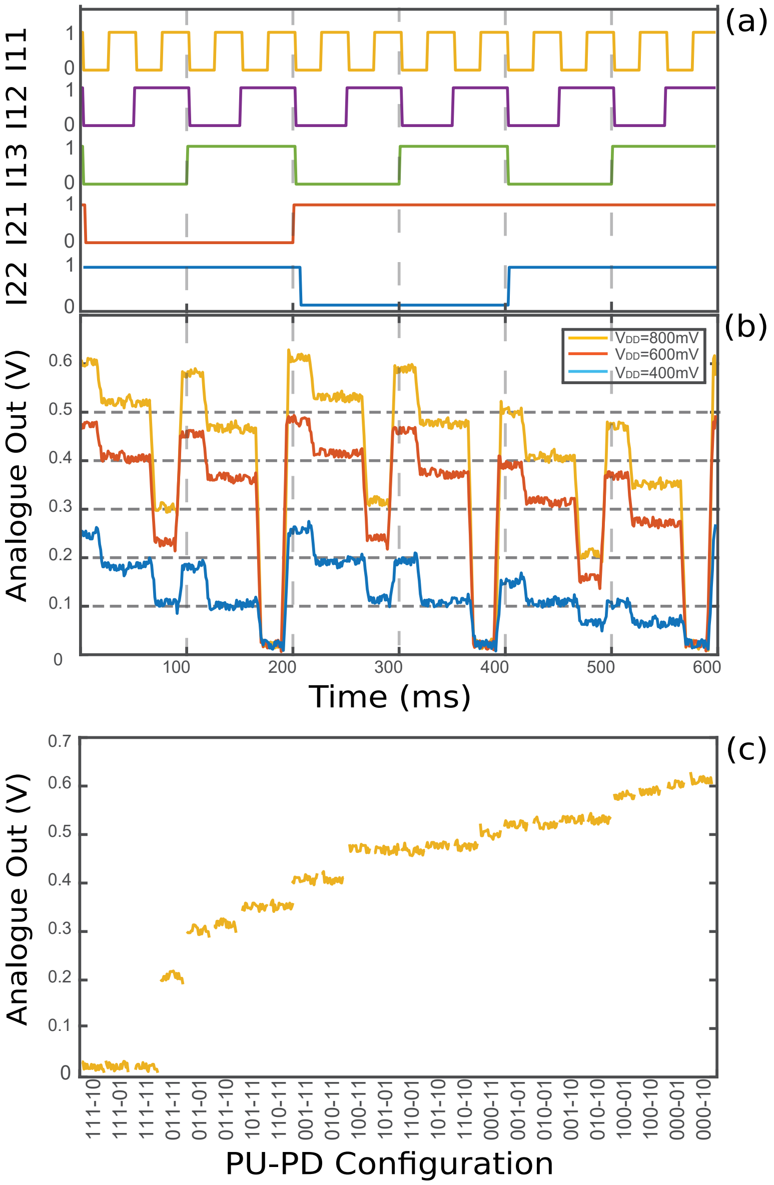

FIGURE 3.6: Measured results of the first hardware experiment. In this first case study, a 3 1T1R pull-up array (PU: 5.5$k\Omega$, 11$k\Omega$, 11.5$k\Omega$) with a 2 1T1R pull-down array (PD: 7.3$k\Omega$, 6$k\Omega$) was implemented and experimentally measured. The MAC circuit was realised by using $Pt/TiO_x/AlO_x/Pt$ RRAM devices, as shown by Stathopoulos et al. (2017). In Fig. 3.6a the digital input vectors controlling the transistors of the 1T1R array are shown with the I_{11}, I_{12}, I_{13} the control signals for the pull-up components and I_{21}, I_{22} for the pull-down components. The pull-up 1T1R network is using pMOS devices, thus the high logic '1' and low logic '0' denotes a non-conductive and conductive transistor states, respectively, 1T1R. In Fig 3.6b, the circuit response ($V_{AnalogueOut}$) for all the different configuration vectors of the control signals (I_{11}, I_{12}, I_{13}, I_{21} and I_{22}) are shown under 400mV, 600mV and 800mV power supply voltages. In Fig 3.6c, the response for 800mV power supply is rearranged into an analogue voltage output ($V_{AnalogueOut}$) versus digital vector input map (signals I_{11} I_{12} I_{13} - I_{21} I_{22} are encoded into a digital control word displayed along the x-axis). Multiple output levels can be discerned.

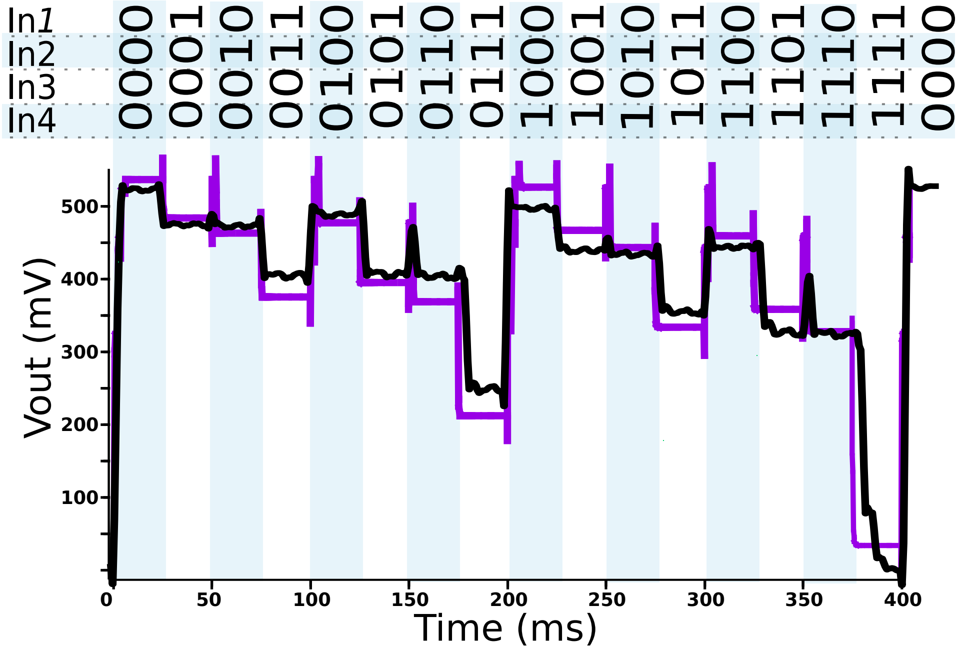

FIGURE 3.7: Graph exhibiting the results from the second experiment. In this case study, a hardware MAC circuit of 4 pull-up 1T1R components (PU: 461$k\Omega$, 508$k\Omega$, 534$k\Omega$, 539$k\Omega$) and 1 pull-down biasing component (PD: 330$k\Omega$) using $Pt/TiO_x/AlO_x/Pt$ devices, as exhibited by Stathopoulos et al. (2017), was implemented and measured. The same configuration is used for the simulation of the circuit in Cadence Virtuoso Spectre environment towards comparing the response of the circuit using real devices and simulated device models. For the MOSFET models, I used a commercially available 65nm technology node library. The RRAM device models are based on the Messaris et al. (2018) and the RRAM model instances are fitted based on the same devices used for the practical circuit implementation. The black trace is the experimentally measured response while the purple line is the simulated circuit response. The comparison highlights that the simulated circuit follows the response of the real hardware implementation. These preliminary results suggest the circuit robustness against transistors variations (in one case large discrete components and in the other case 65nm CMOS).

read-out voltage is the maximum operating voltage the RRAM was expected to be subject to during these experiments. Resistance values quoted henceforth correspond to this definition of R. The standardised resistance for each extracted model is shown next to each device name in Table 3.1. The read-out voltage was connected to the pull-up V_{PULLUP} terminals of each 1T1R component (shown in Fig. 3.2). The results of the second experiment (combined hardware experiment and simulation test) are shown in Fig. 3.7.

It is useful to note that different technologies for the MOSFET are employed between the hardware experiments and simulations. More specifically, in hardware experiments I am using 0.25um node of-the-shelf discrete components while for the simulations I am using the most modern MOSFET technology node available to me which is a commercially available 65nm node. This enabled to test in hardware the response of the

TABLE 3.1: Fitting Parameters from Extracted Models

RRAM Device ($k\Omega$)	**Fitting Parameters**			
	a_p	a_n	b_p	b_n
D1 (461)	0.379	0.327	2.18	3.38
D2 (508)	0.183	0.418	3.32	2.45
D3 (534)	0.021	0.04	5.32	3.96
D4 (539)	0.298	0.27	2.59	3.04

proposed circuit using a larger and power consuming technology (0.25um MOSFET technology) while at the same time performing extensive testing in simulation with a relatively modern (65nm MOSFET technology) and much less power-demanding components to showcase some preliminary results of what the behaviour of a future IC chip implementation of the circuit could potentially be like. Although the transistor technology nodes are different by a large margin, the circuit proposed in this chapter showcase a similar behaviour that seemingly does not deviate from the expected RRAM-MOSFET circuit response at any significant part to the relative technology of the transistor components. This can be observed through the results exhibited in Fig. 3.7. Due to the use of 0.25um MOSFET devices alongside $Pt/TiO_x/AlO_x/Pt$ RRAM devices for the hardware experiments and of 65nm MOSFET technology with the RRAM model which is fitted to experimentally measured devices for the simulations it is expected that circuit behaviour mismatches might occur. For that reason, the hardware and simulation proof-of-concept tests can be studied as separate cases of the same circuit under test since, due to the employment of different MOSFET technologies, the performance and detailed behaviour might showcases differences. From the results exhibited on Fig. 3.7, the experimentally measured results from the hardware implementation of the MAC circuit match closely the simulated results of the circuit using the same RRAM devices fitted into distinct model instances. By extrapolating these results, this could be interpreted as a preliminary indication that we can expect a specific circuit behaviour from vastly different MOSFET technologies.

3.5 MAC Circuit employment in Winner-Take-All System

An important and well-documented family of ANNs is the Winner Take All (WTA) configuration, as showcased by Rahiminejad et al. (2019); Fernando et al. (2018); Hung et al. (2003); Wang et al. (2019). Such networks are implementing the simple but computationally powerful max functions where from a group of firing neurons they can classify the winning neuron (neuron with the highest/fastest response) into one category and all the other (losing) neurons into another category.

WTA networks are essentially performing a multi-input binary comparison and classification not unlike what TLGs are performing but with a single composite input (the dot-product of an input vector multiplied with the weight vector) and a threshold (defined by a biasing signal). Many different implementations of WTA systems exist and some of the most recent ones are making use of emerging memory technologies, such as RRAM, as shown by Fernando et al. (2018); Doevenspeck et al. (2018).

In the presented design of the WTA networks (see Fig. 3.8), I am using the classification circuit presented by Wu et al. (2017) which is a racing signal-based classifier. Compared with the more conventional signal level circuits, such as the conventional CMOS-based Boolean gates, the racing signal circuits base their operation in the speed of the signals propagation from one node of a circuit to another, as shown by Madhavan et al. (2014). In many implementations racing signal designs can showcase higher speed of operation and are thus preferred. Since racing signal circuits are operating similarly to a form of simplified event-triggered circuits not all logic networks can be easily implemented with the appropriate timing requirements for a racing signal implementation, as shown by Pan et al. (2019); Moisiadis et al. (2001); Madhavan et al. (2014).

In this (voltage-driven) signal racing WTA design, shown in Fig. 3.9, a low-complexity CMOS latch-based classifier (essentially a half-latch design) with additional CMOS circuits to enable sensitivity to a global trigger signal is used to implement the max function circuit, as shown by Wu et al. (2017). In this work, the WTA network employs the MAC circuit instead of using the more IC-area-demanding and more complex custom pass-transistor MOSFET-based networks. As shown in Fig. 3.2, the RRAM-based MAC circuit as a programmable RC delay network.

The use of this 1T1R-based artificial neuron (i.e. the MAC circuit) as a reconfigurable composite resistance is based on the parallel connection of the programmable RRAM devices. Through the configurable composite serially connected resistance (of the 1T1R array) and accumulating capacitor C_{ACC} the implementation of a programmable RC-delay generator is possible. Due to the 1T1R array pulling to V_{READ} node (enable 1T1R elements contributing to the common node accumulation) the capacitor charges with a rate defined by the composite trimming resistance.

Each RRAM-based MAC circuit-neuron is connected to a homogeneous structure that compares and classifies the delay of every neuron output through a voltage racing latching technique. The classification is based on an event-triggered latching of the winning (fastest) half-latch which generated a global trigger V_{GT} that locks all the other half-latches into a losing state. The additional power gate controlled by V_{STRB} enabling/disabling the pull-up voltage cutoff of the neuron model ensures a power-efficient operation of each circuit mode. The latching elements of WTA store the results of the classification operation(V_{LO1}, V_{LO2} and V_{LO2}) with the winning neuron acquiring '0' logic value, while the losing neurons are denoted with '1' logic value. A reset

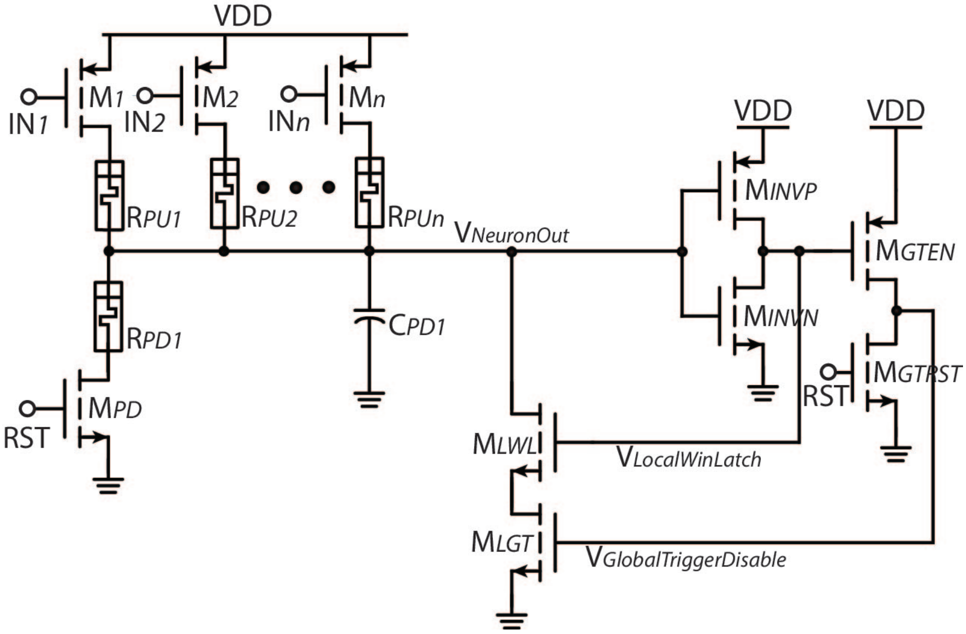

FIGURE 3.8: Schematic of a single Winner-Take-All (WTA) computing node. The circuit consist of the RRAM-based MAC circuit showcased in previous sections of this chapter. The WTA operation is based on the mapping of a specific digital input word to a specific RC-delay value which is then classified by an appropriate CMOS-based voltage racing classification circuit Wu et al. (2017). In this specific figure only one neuron model and classification circuit is presented and more such circuits should be wired together in the V_{GT} global trigger connection for a comparison between multiple neuron to be performed.

operation controlled by the V_{RST} signal is used to clear the state of the WTA output (discharge capacitors and initialise the global trigger) and prepare the system for the next classification operation.

For the WTA network to operate, the winning neuron's M_{LL} nMOS opens the pull-down discharge path due to winning latch output. Thus, the global trigger V_{GT}, controlling the M_{GT} nMOS, does not affect the winning neuron's discharge path since the path is opened. At the other hand, the discharge paths for the losing neurons are enabled by V_{GT} forcing the input to the half latch to logic '0'. Due to the design of the system, if multiple neuron nodes have the same RC-delay all such neurons will latch into the winning state. Hence, the output binary word of the WTA can include multiple '0' denoting the winning neurons and '1' denoting the losing neurons.

An example of the WTA operation is shown in Fig. 3.10 where few cycles of classification operations are showcased. The circuit configuration tested is employing the extracted RRAM models ($D1, D2, D3andD4$ extracted devices used in the second MAC circuit experiment) presented in Table 3.1. The device configuration as well as the input vector connectivity V_{IN1-12} is shown in Fig. 3.9. The outputs of the MAC circuits are shown by the V_{NO1}, V_{NO2} and V_{NO3} signals while the outputs of the WTA latches are shown by the V_{LO1}, V_{LO2} and V_{LO2} signals. The V_{IN} is shown as a hexadecimal value of

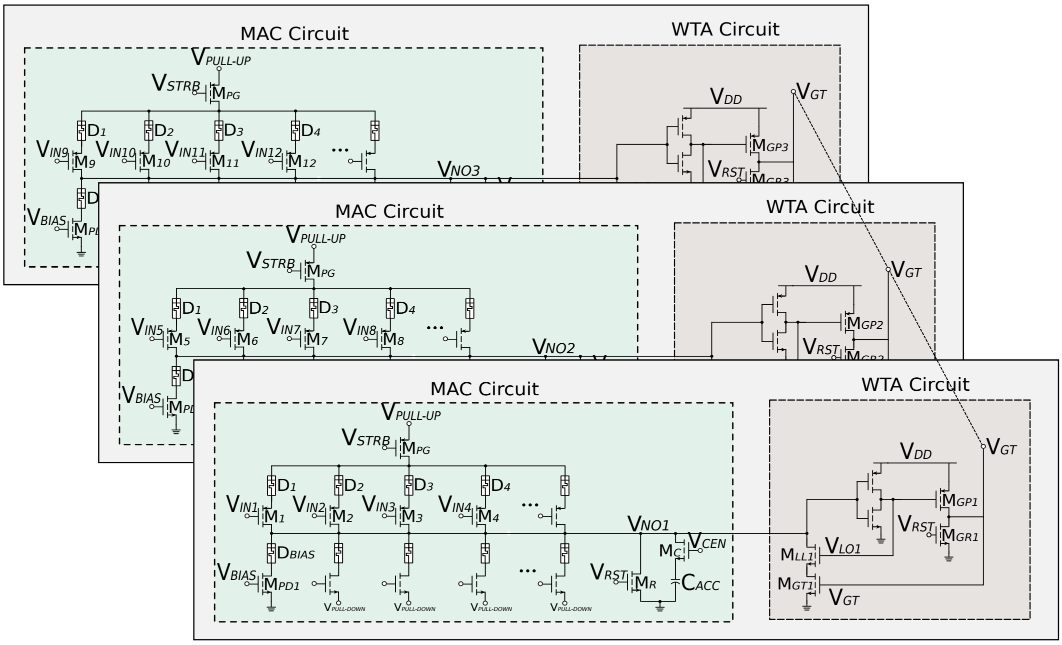

FIGURE 3.9: Schematic of the proof-of-concept Winner-Take-All (WTA) system that was simulated (using Cadence's Virtuoso Spectre) in this work. The WTA tested includes three neuronal (RRAM-based MAC) circuits, operating as programmable RC delay generators, alongside a CMOS-based max function classifier networks based on the work of Wu et al. (2017), that is used to identify the winner of the RC delay-based voltage racing operation. The WTA system is used to find the fastest neuronal response being a voltage racing implementation, as shown by Wu et al. (2017). The MAC circuit output nodes (also referred to as neuron output (NO) nodes) are labelled as V_{NO1}, V_{NO2} and V_{NO3}. When the fastest RC path is identified (shown by the output of the half-latches sensing circuits, thus latch output (LO) nodes V_{LO1} V_{LO2} and V_{LO2}) a generated global trigger signal V_{GT} locks this neuron into winning state while all the other neurons are locked into losing state, until the next evaluation phase. The operation cycle is controlled by the V_{RST} signal. The WTA system in this example operates at V_{DD}=1.2V for the latch-based sensor (WTA circuit) part while a suitably low reading voltage V_{PULLUP}=V_{READ}=0.7V is used for the RRAM-based MAC circuit. The accumulating capacitor value is set to $C_{ACC} = 20fF$.

the binary input vector V_{IN1-12} and the conversion used a $\{MSB-(V_{IN12}\ V_{IN11}\ ...\ V_{IN2}\ V_{IN1})-LSB\}$ convention. The generated V_{GT} ends each cycle of classification when a winning node has been evaluated (see Fig. 3.11). The WTA stores the complement of a one-hot encoded digital word per evaluation operation. The V_{RST} signal controls the cycle of the system with the V_{STRB} enabling a short pulse of V_{READ} voltage to pass through the artificial neuron RRAM-based network and generate a delayed output. As shown in Fig. 3.10 and Fig. 3.11, all the artificial neurons outputs are racing to lock the WTA but only the fastest neuron succeeds in completing this process as expected from the operation of the voltage racing WTA system.

Since the CMOS-based detector WTA circuit is based on a half-latch, it is possible to integrate it as part of a parallel sense amplifier network (which is based on cross-coupled

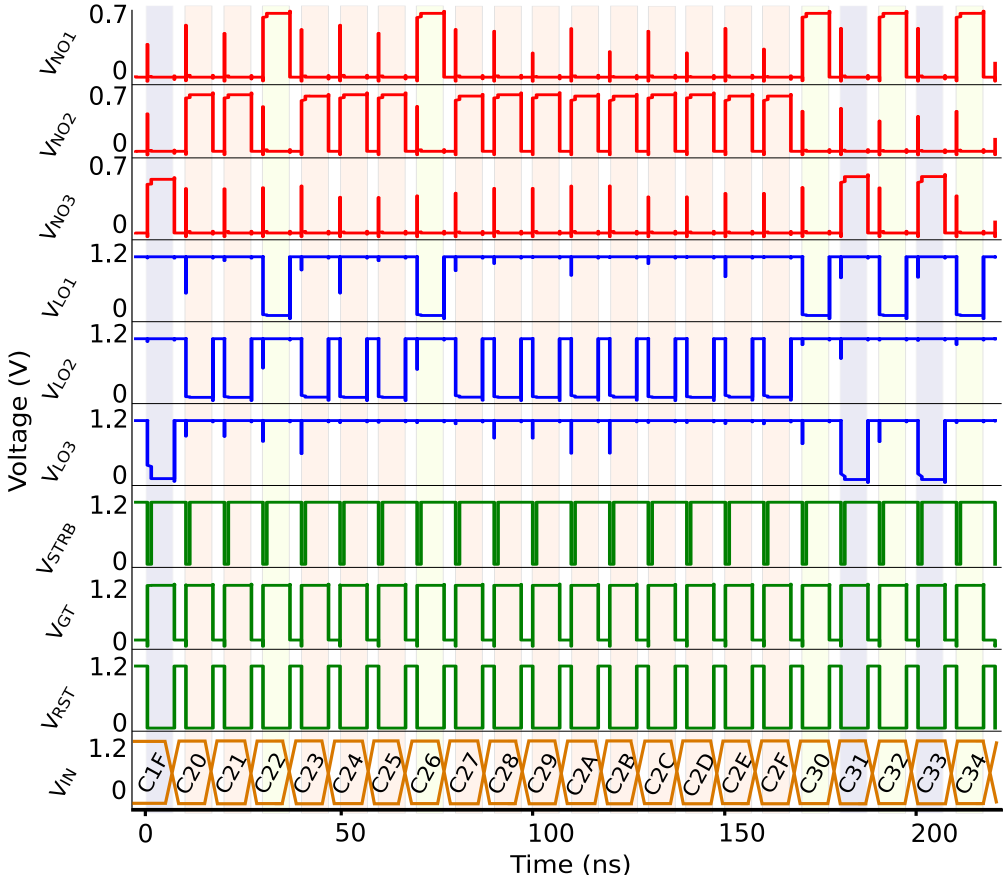

FIGURE 3.10: The results of the Winner-Take-All (WTA) network are showcased in this figure. The results are based on the example 3-neuron circuits presented in Fig. 3.9. The waveforms for the latch outputs (V_{LO1}, V_{LO2} and V_{LO2}), that comprise the classification word encoding of the WTA lateral network state, the neuron output (V_{NO1}, V_{NO2} and V_{NO3}) that feeds into the WTA classification system during the evaluation phase, as well as the control signals that enable the cycles of operation for the presented network (V_{STRB}, V_{GT}, V_{IN}). The V_{IN} is encoded into a hexadecimal value representing the input vector V_{in1-12} controlling the pull-up 1T1R network used for this example. The connectivity of the V_{IN} vector is shown in Fig. 3.9. The artificial neurons (see Fig. 3.9) were based on multiple instances of the extracted RRAM models as described in Section III.A (see Table 3.1). The simulation was performed in Cadence's Virtuoso Spectre using a commercially available 65nm technology node for the transistor devices. It can be clearly shown that when a winning node is evaluated by the WTA by observing the V_{GT} signal. Distinct coloured shading has been added to highlight when one neuron is winning over the rest.

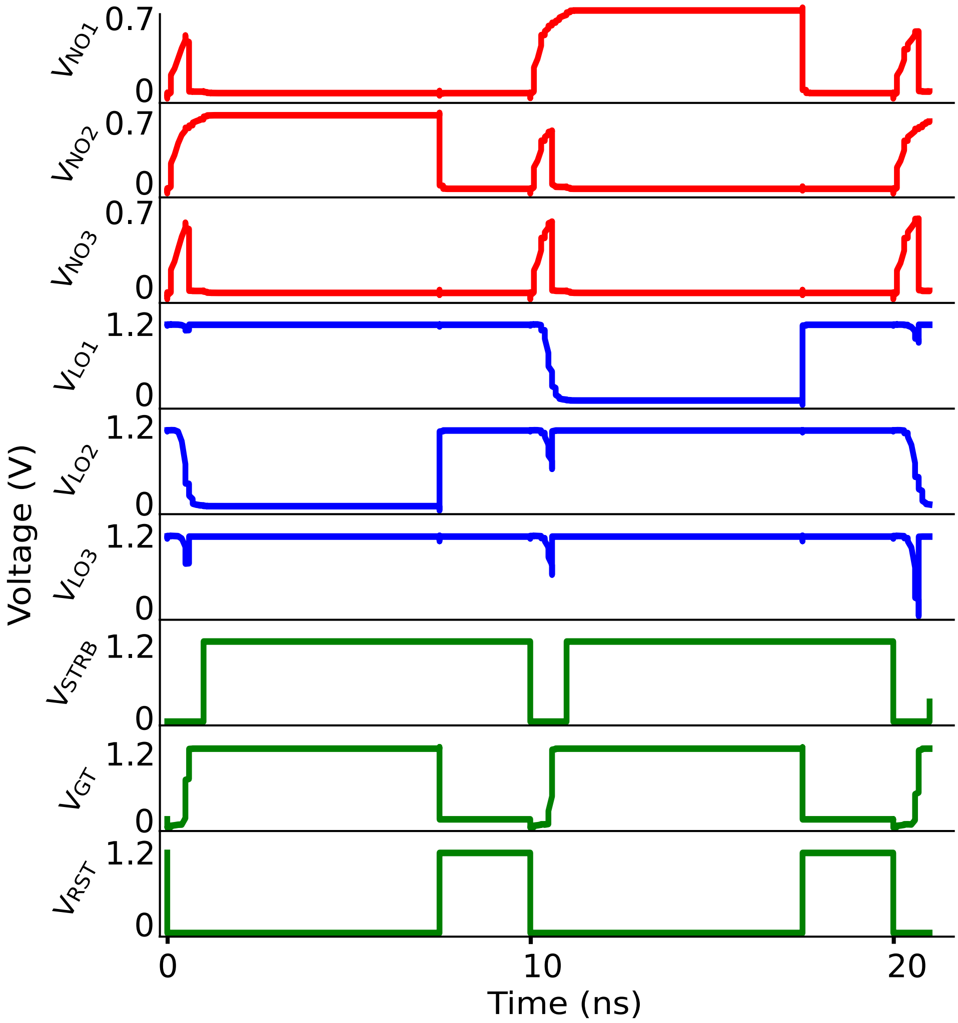

FIGURE 3.11: In this figure, a focused analysis of the WTA operation cycle is showcased. The red traces (V_{NO1}, V_{NO2} and V_{NO3}) showcase the outputs of the 3 artificial neurons while the blue traces (V_{LO1}, V_{LO2} and V_{LO3}) showcase the latches output (winning node is the node with latch output at logic '0'). The green traces are the signals controlling the operation cycle of the WTA. The V_{RST} initialise the system when set to high (logic '1') and the V_{STRB} is the signal controlling the power gates of the artificial neuron. The V_{GT} is low when the WTA is initialised/evaluating and high when the winning neurons has been evaluated. A single period of operation (defined by V_{RST}) is 10ns from this proof-of-concept example and the classification operation requires t_{op}= 1ns. The voltage racing operation of the circuit can be highlighted from the artificial neurons outputs that before the winning neuron is found all the traces are racing to the top but the fastest one is locking the latches to an one-hot encoded digital word.

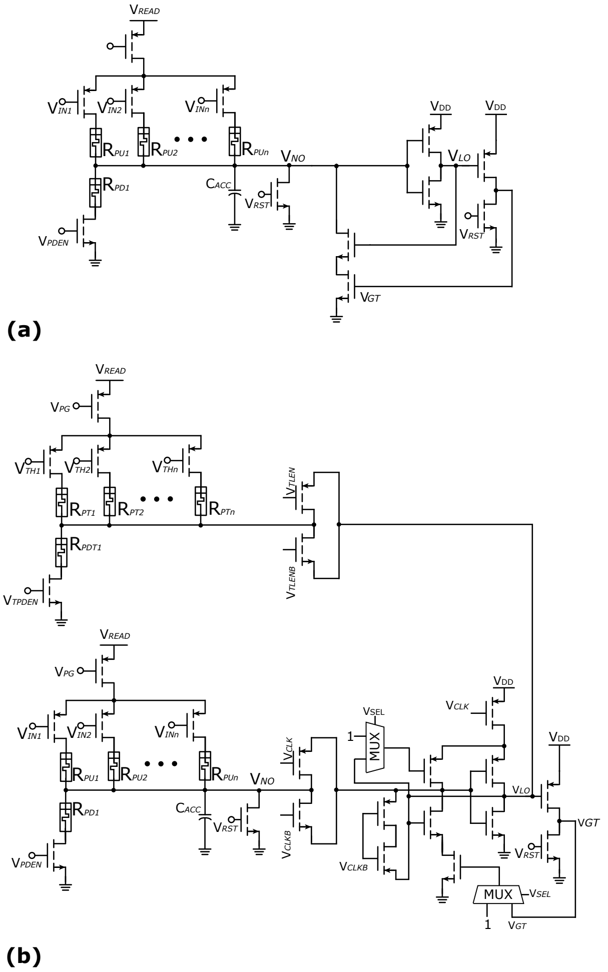

FIGURE 3.12: Schematic of the proposed Winner-Take-All (WTA) circuit in (a) and its adaptation (b) for enabling the implementation of a programmable classification circuit. In (a), the MAC circuit is connected through the V_{NOut} node to a half-latch-based circuit that is based on the implementation presented by Wu et al. (2017) as shown in Fig. 3.8 and Fig. 3.9. In (b), the WTA part of the neuron model can be implemented by connecting/disconnecting appropriate pMOS and nMOS MOSFET devices of a CMOS full latch, similar to the main sensor component used in differential TLG designs, as discussed in Chapter 2, as shown by Papandroulidakis et al. (2018); Dara et al. (2017); Mozaffari and Tragoudas (2018). Thus, the design employed here shows compatibility with some RRAM-based TLG implementations that can be used to efficiently implement a novel sea-of-gates reconfigurable computer, as shown by Beiu et al. (2003b). The configurable CMOS-based classifier can be used as periphery circuitry to a RRAM crossbar array where multiple 1T1M arrays that can be used to create a configurable neuro-inspired system capable of easily reconfiguring the IMC-based classification operation.

latch design). Having a switch network that can connect multiple columns of memory through the sense amplifier network, TLG structures for comparing two differential RRAM arrays can be easily configured. In the Fig. 3.12, I am showcasing that with additional control CMOS circuits, a differential mode Memristor-based Current Mode TLG (MCMTLG) can be configured, as shown by Papandroulidakis et al. (2018), thus the RRAM-based TLG design showcased in Chapter 4, to implement the partial WTA classification circuit. Thus, the near-memory (Near-Memory Computing (NMC) -based) classification network can be transformed into different operation modes depending on the requirements of each application. By switching between the different version of the classification structures, performing a different type of neuro-inspired computing is possible. The switch between different configurations of the classification engine is implemented using appropriate multiplexing circuits to enable/disable the pull-up pMOS and the additional globally triggered (in WTA mode) pull-down nMOS found of the first inverter that is used to comprise the latching element. There is a multitude of methods to process digital signalling and combine a relatively complex function into a compact analogue result. This category of computation is particularly interesting with the rise of technologies that can be employed to physically perform analogue computing, such as RRAM. From the different designs following a similar concept the key factor is the decision boundary which can be very finely tuned by the RRAM devices (physical analogue computing) even if the results is ultimately transformed into a binary output to ensure compatibility in mixed logic architectures (i.e. cascading between different logic technologies).

3.6 Comparison of WTA case study with state-of-art

A wide variety of different WTA implementations have been suggested with some example being proposed by Serrano-Gotarredona and Linares-Barranco (1995); Hung et al. (2003). Naturally, many emerging technologies found their way as part of novel WTA circuits and systems, as shown by Truong et al. (2016a,b); Hasan and Taha (2017). The main categories of WTA are level-based voltage-mode/current-mode parallel comparison or a racing-based signal transmission delay parallel comparison. The design of the max function implementation is usually based on simple lateral networks of CMOS-based circuits for comparison (i.e. latching elements etc.). Due to the close integration of WTA classification networks with artificial neuron circuits, novel hardware solutions of hybrid CMOS-RRAM WTA have been an proposed.

Some of the most important and recent implementations include designs of both fully conventional and hybrid technologies. A CMOS-based WTA with high resolution capabilities has been proposed by Baishnab et al. (2009, 2017). As shown by Baishnab et al. (2009), a fully custom CMOS WTA network is showcased. The implementation is tested on Cadence's Spectre and exhibits a fast classification operation with t_{op}=40ns-80ns for the case of 100fF load capacitance. The circuit is converting an input voltage to current and then amplify the difference between the input current and the bias current through a custom current mirror network. A final CMOS inverter gate is employed for isolation of the output and ensuing cascading capability of the WTA computing node. The fully-CMOS circuit was operated at V_{DD}=1.8V power supply and requires 12 MOSFET devices per WTA computing node. In Moro-Frias et al. (2011), a CMOS-based current-mode WTA system with fast operation and low-complexity design (4 MOSFET devices and 2 current sources per WTA computing node) was proposed. The current mirrors are implemented using current mirrors. The specific circuit operates under the principle of continuous analogue feedback between all the available input current nodes as well as the connection of these nodes to a common current sink node. When one input current is larger than the other inputs then the WTA node affected by the larger input is fully activated (in proportion to the input current) while the other nodes are cut-off due to the increased contribution of the winning node to the common node. The WTA in Moro-Frias et al. (2011) showcases low power consumption of P=281.7uW alongside the employment of a low number of devices per WTA comparison node. This circuit implementation operates within t_{op}=4.74ns under a supply voltage of V_{DD}=2.5V.

The increasing maturity of RRAM devices introduces new solutions for hybrid MOS-RRAM ANNs. Among the proposed hybrid ANN implementations some of the most interesting designs involve the use of WTA systems, as shown by Sheridan et al. (2016); Ebong and Mazumder (2012). In the work of Sheridan et al. (2016), it is proposed a WTA system as part of a Locally Competitive Algorithm hardware implementation, using a RRAM-based crossbar and CMOS control circuitry, for feature extraction operations. The WTA design is based on the use of a single-RRAM-per-node memory array with additional peripheral circuitry for driving the RRAM devices and enabling the appropriate signals for the algorithms under investigation. The implementation by Sheridan et al. (2016) uses a V_{READ}=0.8V for the RRAM devices. Furthermore, in the work of Ebong and Mazumder (2012), a clocked hybrid CMOS-RRAM design of a WTA as part of a RRAM-based Spike-Timing-Dependent-Plasticity (STDP) ANN was provided showcasing fast neural operation. The general structure of the WTA design is centred around a RRAM memory array that employs recurrent connections through a parallel array of CMOS-based neuron-emulating circuits. The memory array is a fully passive one incorporating single RRAM devices per memory node. Hence, the STDP mechanism is implemented mainly through the CMOS-based neuron circuits and other peripheral circuitry around the array. The CMOS neuron is based on an integrate-and-fire design, as shown by Ebong and Mazumder (2012). The system operates at 1kHz and has a dynamic power consumption of P=15.6uW.

Towards better illustrating the expected performance of most modern WTA designs, I organised performance metrics of important state-of-art WTA designs. The available performance metrics are showcased in Table 3.2. It is useful to note that in many cases an exact circuit description of the implementation was not provided thus making a direct comparison challenging. Hence, these metrics are employed mainly to provide a better landscape of the power dissipation and delay per operation that WTA systems are usually showcase.

TABLE 3.2: Performance of Previous State-of-Art WTA systems

	WTA State-of-Art Designs based on Literature			
Metrics	***Energy*** E_{op}	***Delay*** t_{op}	***#Device***	***Design***
Baishnab et al. (2009)	N/A	40-80ns	12	CMOS
Moro-Frias et al. (2011)	1335fJ	4.74ns	4+2(I_s)	CMOS
Ebong and Mazumder (2012)	1560pJ	100us	N/A	Hybrid
Sheridan et al. (2016)	86pJ	100ns	N/A	Hybrid

Another important CMOS-based WTA design is presented in Wu et al. (2017) where a signal racing-based WTA networks is proposed. This implementation uses 40nm MOSFET technology node and is operated at V_{DD}=0.9V with a high throughput of 1.0GHz for 16-bit resolution (simulation performed in HSPICE). The design is based on the use of custom pass-MOS networks that employ multiple different lengths of pMOS devices. Since I base the design of the CMOS part on the work of Wu et al. (2017), a direct comparison in that case can provide some useful insights. The comparison between the two designs, the hybrid RRAM-MOS I proposed and the fully CMOS design proposed by Wu et al. (2017), are showcased in Table 3.3. In Table 3.3, under the number of devices per computing node column, the m the RRAM devices used per artificial neuron (thus per instance of the RRAM-based MAC circuit used for the programmable RC generator). Our approach of hybrid WTA, as compared with the fully CMOS WTA from Wu et al. (2017), exhibits some interesting traits in terms of energy per operation and speed of the classification operation. More specifically, our hybrid approach (with E_{op} = 300fJ) showcase lower energy per operation which is reinforcing the concept of implementing such systems using RRAM-based artificial neurons. The measurements regarding our approach were based on the proof-of-concept WTA system presented in Section III.B. Our WTA system operates at V_{DD}=1.2V for the CMOS part and V_{READ}=0.7V for the hybrid CMOS-RRAM part. Additionally, our RRAM-CMOS approach showcases faster classification operation with a system response within t_{op}=1ns compared to the fully CMOS racing voltage WTA equivalent. It is also worth noting the reduced number of devices required for a similar resolution of generated delayed signals compared to Wu et al. (2017) due to the replacement of the complex pass-MOS network with the RRAM-based MAC circuit.

Towards better understanding the performance comparison of Table 3.3, we need to consider that the hybrid implementations proposed in this thesis is build using 65nm MOS technology while the fully CMOS design of Wu et al. (2017) is employing 40nm

TABLE 3.3: Comparison between Proposed Hybrid RRAM-CMOS and State-of-Art Fully CMOS-based Racing Voltage WTA

Designs	WTA implementations			
	Energy E_{op}	***Delay*** t_{op}	***#Device***	***MOSFET tech.***
Wu et al. (2017), Fully-CMOS	4602fJ	4ns	24	40nm
My approach, Hybrid RRAM-MOS	300fJ	1ns	12+4(m)	65nm

technology node that enables the design to operate at 0.9V voltage supply. We can see that although the fully-CMOS implementation is employing a smaller MOSFET technology node and smaller supply voltage for the CMOS part the performance of the hybrid RRAM-MOS design is competitive. This can be partially explained due to the smaller number of MOS devices and the fast RC delay generation and comparison operation driven by the RRAM-based 1T1R arrays. The global comparison circuit (CMOS-based max function operation of the WTA) is similar in design and operation for both cases shown in Table 3.3 with the only difference being the different MOSFET technology node and supply voltage. These preliminary results are showcasing the potential of hybrid ANN implementations, an observations that is documented by the literature. Compared to the state-of-art implementations discussed earlier, it can be observed that the proposed design shows good performance since we can achieve competitive energy consumption and system delay from the state-of-art designs of Moro-Frias et al. (2011); Ebong and Mazumder (2012); Sheridan et al. (2016); Baishnab et al. (2009), which include both hybrid and fully CMOS designs. Although it is not possible to compare the different implementation with each other due to limited information regarding the actual hardware design and disparity of technologies employed and/or applications, the results exhibited in this section showcase that my hybrid WTA design can be competitive with other state-of-art designs.

3.7 Conclusions

In this chapter, I presented an implementation of a Multiply-Accumulate circuit based on RRAM devices. The circuit was used to convert information from the digital domain to analogue data, thus employing the RRAM array as a natural DAC with reconfiguration capabilities (unlike their fixed resistor-based counterparts). The operation of such a nano-scale reconfigurable RRAM-based DAC, as discussed and showcased, have potential for incorporating important data conversion operations in IMC paradigms. Thus, this type of area and power efficient data converters can be used to enable better neuro-inspired hardware accelerators implemented inside or near the memory units. Such design solutions are important towards performing more efficiently massively parallel data processing. I showcased a proof-of-concept RRAM-based primitive logic gate that can be used to convert information from the digital domain to analogue data thus operating as a mixed-signal hetero-associative memory. This type of area and power efficient data converters can be implemented to enable novel neuro-inspired hardware acceleration for massively parallel data processing at the edge. The circuit's functionality was experimentally demonstrated using real $Pt/TiO_x/AlO_x/Pt$ RRAM devices. Furthermore, RRAM models were fitted based on their hardware counterparts and then used to test in simulation a proof-of-concept Winner-Take-All network as an example of the potential use cases of the MAC circuit.

More specifically, the basic concept explored through this work is the design and validation of a generalised circuit capable of receiving binary data vectors and classifying it into a continuous output space, thus introducing a generalised RRAM-based data mapping function. This idea here is materialised through the design of a RRAM-enhanced associative memory computing block. The configuration tested here is based on two 1T1R arrays for pull-up (pMOS devices) and pull-down (nMOS devices) networks. The inputs are the digital gate signals for the pMOS and nMOS of the 1T1M arrays, while the output is the analogue voltage read from the intermediate node. More specifically, in this work, I showcase an experimental demonstration of a RRAM-based MAC circuit assuming: i) the use of digital input, which can be transmitted along great distances with little distortion, ii) single-ended design to reduce component count and thus power dissipation and chip area, iii) analogue output due to the fast physical computing of the resistive arrays as well as to preserve the maximum available information content from the MAC operation. The MAC circuit design is reminiscent of some of the earliest artificial neuron designs, e.g. shown by Douglas et al. (1995), but in the current version it has been adapted to be optimised for the advantages introduced by the RRAM technology. To support the RRAM-based MAC circuit showcased in this chapter, I presented the design and operation of the MAC circuit and provide experimental evidence to support the proof-of-concept. Additionally, I employed the RRAM-based MAC circuit to test a Winner-Take-All (WTA) network as an example of the computing flexibility of the primitive circuit, although its applicability is much more general. Throughout the chapter I discussed how the 1T1R-based MAC circuit can be exploited towards implementing sea-of-gate reconfigurable computer architectures.

Through the scope of the aforementioned presented findings regarding the RRAM-based MAC circuit, I can see that the flexibility of primitive RRAM-CMOS circuits to be adapted into a wide variety of different computational modes. At the same time, by enabling the MAC operations in the analogue domain through the use of RRAM devices, RRAM-based circuits are capable of accelerating MAC operations. Through the presented results, the practical implementation of 1T1R-based neuro-inspired systems seems feasible. The relatively simple modeling approach presented in this work is sufficient for an initial testing of performance for such systems. Furthermore, I can gauge the importance of RRAM-based MAC circuits that connect digitally to the environment, thus without the need of specialised conversion circuitry, but perform the computations physically in the analogue domain as viable solutions for the next generation of neuro-inspired computers.

Chapter 4

Memristively-Enhanced Threshold Logic Gate

On the computation/architecture front for novel computer designs, there has been a sustained effort for many decades to develop neuro-inspired computation concepts, mostly in the form of ANN-based systems. Research on ANNs has thus far spanned the entire interval between the first simplified models of all-or-none hardware neurons, the main computational concept proposed by McCulloch and Pitts (1943), and the current state-of-the-art massively parallel GPU-based ANN implementations, showcased for example by Abadi et al. (2016); Vestias and Neto (2014); Krizhevsky et al. (2012). However, one often overlooked example that somewhat defines the basic operational principles of ANN-like computation can be found in the form of its quantised, digital counterpart, the so-called threshold logic (TL), the term proposed by McCulloch and Pitts (1943) to describe the main neural functionality as mapped into computer logic. TL is a computational model for performing a comparison between a threshold value and the weighted sum of an input vector aimed at implementations for computer logic.

The employment of TLGs as fundamental computational units in neuro-inspired post-von Neumann computing schemes with the recently demonstrated multi-bit capabilities and fine tuning of metal-oxide-based RRAM devices raises the prospect of a RRAM-based reconfigurable fabric. In the following sections, I present my design of a RRAM-based current mode TLGs (also referred to as RRAM-based current mode threshold logic gate, MCMTLG). Alongside the design and operation of the hybrid RRAM-CMOS TLG I am showcasing experimental results using a discrete component-on-breadboard circuit implementation of the proposed design with real RRAM devices. More specifically, regarding the physical implementation of a Memristor (RRAM) -enhanced Current Mode Threshold Logic Gate (MCMTLG), the implementation was build using discrete components in a breadboard. The pMOS and nMOS discrete devices used for this experiment are NDP6020P and SUP65N02-03, respectively. The RRAM devices used for this experiment are MIM stacks constituting of Pt/Al2O3/TiO2/Pt/Ti (10/4/25/10/5) nm, as showcased by Stathopoulos et al. (2017). The measurements performed through this breadboard experiment enable us to define/test this versatile memory and logic co-location circuit design.

4.1 RRAM-Enabled TLGs Design Methodology

As discussed in Chapter 2, a basic computational unit in TL is called a Threshold Logic Gate (TLG) and it corresponds to the equivalent computational unit of artificial neuron found in ANN designs. TLGs were introduced as a method of describing and modeling neural activity in the brain through conventional electronic circuits and systems, as shown by McCulloch and Pitts (1943); Bayat et al. (2017). Although TL is effectively a simplification of the main ANNs functionality adapted for digital computers, TLG-based logic families have been shown to be capable of fast and low-power operation as evaluated by the power-delay trade-off metric, as discussed by Beiu et al. (2003b); Leshner et al. (2010); Bobba and Hajj (2000). Many of these neuro-inspired TLG implementations showcase that computer designs incorporating such technologies to accelerate ANN-based operations could benefit the area and power dissipation of these systems. Similarly to what some recent work explores, TLGs are a promising candidate for IMC and ANN hardware implementations and further research is required to showcase how TLG can be integrated with emerging technologies to enhance the existing advantages even more, as highlighted by some research findings such as the ones by Tran et al. (2012); James et al. (2014); Krestinskaya and Pappachen James (2018).

Many competing emerging memory technologies are part of the RRAM technology family, such as PCM, ECM, VCM etc., as showcased by Gao et al. (2013a); Edwards et al. (2015). A number of these technologies have been studied as an important part of novel reconfigurable circuit and system aimed at ANNs and IMC designs, as proposed by Guo et al. (2015); Gallo et al. (2017); Ambrogio et al. (2018); Burr et al. (2017) (including TLG implementations proposed by Gao et al. (2013a); Nukala et al. (2014); He and Fan (2017); Alibart et al. (2016) – showcasing exclusively simulated results). In principle, any RRAM technology featuring non-volatile resistive switching, sufficiently high ON/OFF ratio and not excessively high or low resistance levels can be introduced into an appropriately designed TLG. For cases other than binary weights TLG designs, the employment of analogue RRAM devices as weight storage elements is preferred since they enable us to store multi-bit information per single device and thus further accelerate potential MAC operations by compressing multiple binary multiplication layers into a single one. Among many competing technologies one of the most promising ones can be found in the form of RRAM devices and more specifically continuously tuned RRAM capable of multi-bit storage such as the work shown by Stathopoulos et al. (2017).

Towards designing a TLG that can easily integrate RRAM devices as the weight representation elements a lot of work has focused into testing low area and low power design. As showcased in Chapter 2, some promising TLG implementations are based on the differential synaptic weight circuits. There are some differential TLG designs that implemented the weight memory array through conventional electronics, such as capacitors and resistor, such as the work by Seshadri et al. (2017); Mozaffari and Tragoudas (2017), while other recent implementations take advantage of emerging nanoelectronics devices such as single-electron technology (SET), as shown by Inokawa et al. (2003) and negative resistance devices (NRD), as shown by Pettenghi et al. (2008a); Mirhoseini et al. (2010). More importantly, recently, some TLG designs, such as those proposed by Rajendran et al. (2010); Dara et al. (2013), have shown that RRAM technology can efficiently be incorporated into TLG designs, hence becoming the catalyst of significant power consumption and noise sensitivity reduction, as well as logic and area scaling in TLGs, compared to conventional Boolean logic gate. Introducing the RRAM devices as analogue weights in a digital logic gate family, has the advantage of enabling highly localised, continuously tuneable, minimal front-end footprint and low-voltage operated non-volatile memory into the TLG, thus providing a potentially decisive advantage in the implementation of memory-heavy ANN accelerators, especially the ones centred around IMC-based designs, for example those proposed by Gao et al. (2013a,b); Kumar et al. (2016); Kulkarni et al. (2016); Yang et al. (2014); Vrudhula et al. (2015)

From the available RRAM-based TLG implementations, the computing schemes of differential memristively-enhanced load comparison TLGs are shown to have advantages over simpler RRAM-based TLG designs, such as in the work of Maan et al. (2016); Tatapudi and Beiu (2003); Lee and Hwang (2008). More specifically, the differential implementations, in general, as shown by Beiu et al. (2003c), showcase delay and energy improvements over non-differential Memristor (RRAM) -based TLG (MTL) designs, as showcased by Rajendran et al. (2012); James et al. (2015a); Maan et al. (2016); James et al. (2013). At the same time, there is a significant trade-off between energy, delay and computationally flexibility (reconfiguration capabilities) as well as between area and complexity for these two main groups of TLGs. While the differential TLG group is optimised for performance and logic-centric features (e.g. positive and negative weight configurations similarly to the design proposed by Dara et al. (2017)) the non-differential MTL gates provide a lower-complexity gate structure where a resistive network (weighted inputs) is fed into a simple digital gate, such as an inverter, operating as a thresholding element, thus a circuit that perform a comparison between the input and a bias. It is worth noting that different variants and configurations of the non-differential MTL designs can provide novel solutions to RRAM-enhanced TLG-based computer architectures and are capable of competing against conventional systems in applications such as object recognition, FPGAs, etc., with some examples highlighted by James et al. (2014); Krestinskaya et al. (2018b); James et al. (2015b); Maan et al. (2015); Zhu et al. (2013); Zhang and Kaneko (2016).

RRAM-enhanced TLGs can compete with other RRAM-based logic circuits, depending on the application and logic function implemented. More specifically, RRAM-based TLG can outperform many RRAM-based non-TL logic gates if the function can be compressed using TL, as showcased by Leshner et al. (2010); Maan et al. (2016). Technologies such as RRAM-based Look-Up-Tables (LUTs), proposed by Chen et al. (2012); Kumar et al. (2014), and RRAM-based universal logic gates, with a variety of implementations existing as shown by Teimoori et al. (2016); Emara et al. (2016); Kvatinsky et al. (2012), may be preferable in some architectures and/or applications over RRAM-enhanced TLGs. But the RRAM-based TLG's requirements for a state-of-the-art multi-bit RRAM technology, such as the one showcased by Stathopoulos et al. (2017), favours the implementation of non-uniformly behaved programmable analogue resistive elements. At the same time, TL computing schemes do not require frequent switching of the memristive weights, as they are mainly used as programmed-once-read-many reconfigurable logic, thus do not impose high requirements of switching endurance in RRAM devices. In contrast, LUTs techniques require stable and hard-defined RRAM resistive states to operate correctly, being mainly used in digital circuits, while other logic techniques that make use of memristive networks to perform state-based logic, for example as shown by Kvatinsky et al. (2012), require total homogeneity of device behaviour and high endurance in large crossbar arrays to be viable as true alternative post-von Neumann solution. While unorthodox by mainstream conventional systems, the implementation of future computers that make use of non-uniformly behaved components might be the key to a new era of computing. Neuro-inspired logic schemes, such as TLGs, are ideal to 'assimilate' such 'imperfect' technologies, i.e. technologies that do not offer better reliability and resistive state control compared with existing conventional digital electronics, and use them to build new generations of computers, similar to what biological brains seem to achieve in nature's biological neural networks.

4.2 Overview of RRAM Devices Characterisation

Towards identifying and preparing the RRAM devices appropriately a few important steps in testing the behaviour of the devices and setting it in an appropriate resistive state are taken. Although the characterisation and programming of the device is beyond the scope of this thesis, it is useful to provide relevant information regarding the general process in preparing the RRAM devices before they are deployed in the experimental setups and/or extracted for use in a simulated environment. Towards better explaining the characterisation followed for preparing the device before the hardware measurements and/or the parameter extraction for simulation testing a few example of RRAM devices under distinct phases of characterisation are shown in this section.

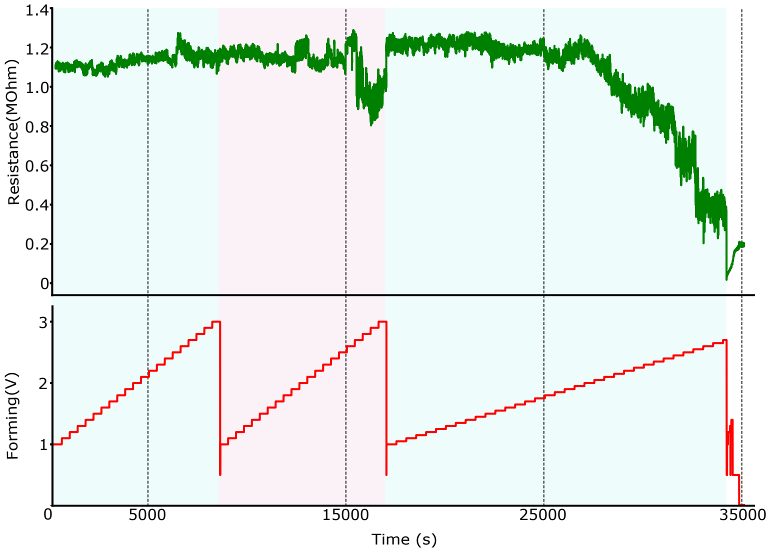

FIGURE 4.1: Example of the forming process applied to pristine RRAM devices. The *Forming* (V) signal is the stimulus to the RRAM device and the *Resistance* $(MOhm)$ signal is the resistance progression read of the device under test. A staircase-like train of pulse of increasing voltage amplitude is applied to the RRAM device. In this example, three phases of programming staircase-like pulses from 1V to 3V are applied. At the end of the third phase the resistance of the RRAM device under test has been set to approximate 200kΩ. This process is theorised that changes the geometry of the RRAM devices towards enabling it to form conduction paths that are necessary for the correct operation of the RRAM which includes the programming to different states. The details of the RRAM mechanics and chemistry is out of scope of this thesis and more information can be found in the work of Stathopoulos et al. (2017); Michalas et al. (2017).

All the RRAM devices that were employed for the purposes of the experiments and simulation in this thesis are based on the $30 \times 30mm^2$ $Pt/TiO_x/AlO_x/Pt$ RRAM technology, showcased and tested in detail by Stathopoulos et al. (2017). It is worth noting that for the purposes of the hardware measurements and simulations of the circuits and systems proposed in this thesis, the more standard program-once-read-many operation scheme is assumed for the RRAM devices. Thus, the main focus of the device characterisation, with regards to the experiments of this thesis, is the preparation and programming of one or more RRAM devices to a wanted resistance state and then the testing of the stability of the device under test at this resistive state under specific stimulus.

The RRAM devices employed for the purposed of the experiments in this thesis were characterised, programmed and tested on wafer through the use of a probe station. For the experimental setup, I am using an ArC One measurement board (ArC Instruments, UK), alongside the aforementioned probe station, which includes all the necessary hardware measurement equipment and software programs to enable the manual and automated testing of RRAM devices. The experimental process is shown in the form of block diagram in Fig. 3.3, in Section 3.4, which showcases the main components

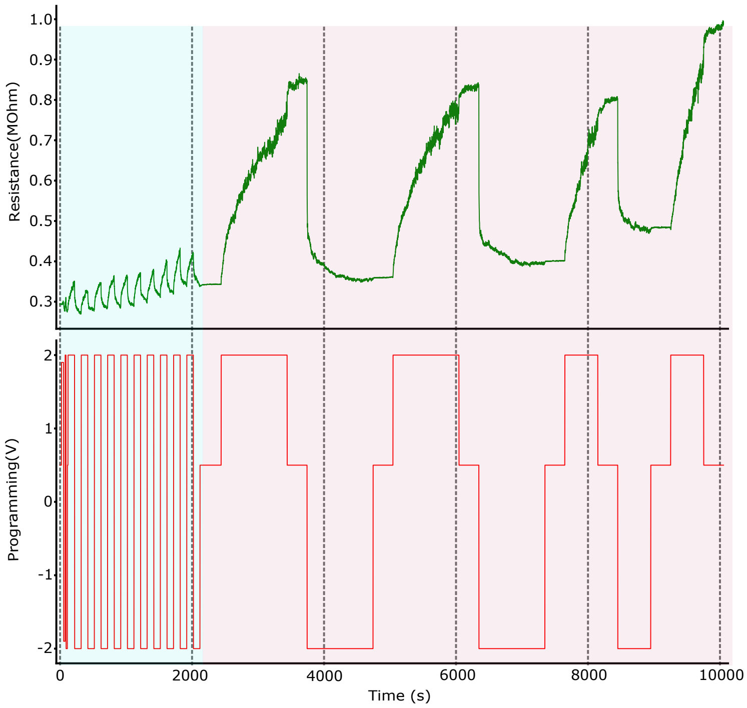

FIGURE 4.2: Automated programming testing can also be employed to map the RRAM programming behaviour between resistance states. The *Programming*(V) signal is the stimulus to the RRAM device and the *Resistance*$(MOhm)$ signal is the resistance progression read of the device under test. This process can test the endurance and stability of the switching operation between two or more resistive states and how reproducible is the switching before a large enough state drift occurs that requires correction. Programming voltage pulses of different amplitude and duration are applied to the device towards setting the device to an appropriate resistance level as dictated by requirements of the experiment. Reading pulses are also applied usually in low amplitude range of 200-500mV (*Reading*(V), Reading signal) towards avoiding unwanted programming during the reading operation. Through the reading pulsed I am performing the appropriate testing the programmed resistance value. More information regarding the process of setting up an automated process for programming and testing a RRAM device are showcased by Serb et al. (2014, 2015); Messaris et al. (2018).

of the experimental setup implemented for performing the hardware experiments as well as the parameter extraction employed to create specific RRAM model instances for the simulations.

The first important operation performed on pristine RRAM devices is the forming operation. This characterisation phase is important for creating appropriate conductive paths in the RRAM device thus enabling it to attain stable resistive states across a range of resistance, as discussed by Stathopoulos et al. (2017). The details of the RRAM mechanisms and thus a detailed explanation of the forming process is out of scope of this thesis. Information can be found in the relevant material science literature for RRAM device, as discussed by Mehonic et al. (2020); Ielmini and Wong (2018); Zhang et al.

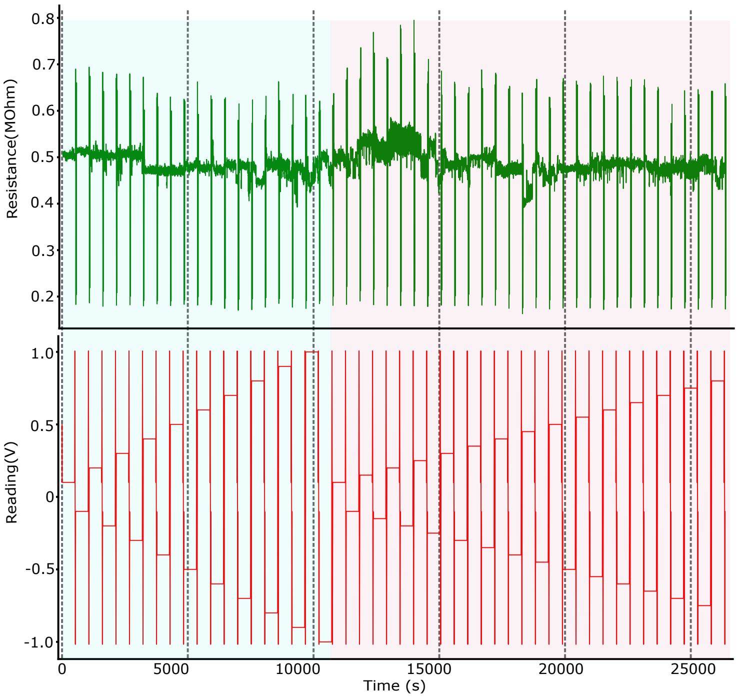

FIGURE 4.3: Example of detailed RRAM device response mapping for a specific range of input stimulus. The *Reading*(*V*) signal is the stimulus to the RRAM device and the *Resistance*(*MOhm*) signal is the resistance progression read of the device under test. This process is followed during the parameter extraction of the devices that is used in order to create unique instances of the Verilog-A RRAM model, as discussed by Messaris et al. (2018). The process includes the continued pulsing of the RRAM device under test with a specific voltage amplitude thus testing the endurance of the device to retain the specific resistive state under the stress of the stimulus (state drift test). Afterwards, a full current-voltage characteristic graph is created for a specific range of voltages to test the non-linearity of the device under test. The process is sequentially repeated for a predefined set of voltage amplitudes similarly to what is shown in this example. The detailed process is discussed by Messaris et al. (2018). Since the experiments and simulations performed in this thesis are focused around the program-once-read-many operation, the devices at this stage of the characterisation process are tested for the non-switching part of their response. This is usually found in the range of 0V to approximately 1V. Hence, for this range of voltage stimulus the devices under test do not exhibit any unwanted resistive state alteration and instead keep relatively stable the state in which they were programmed to in previous steps of the characterisation and preparation for employment process.

(2020a). More specifically, details closely related to the specific RRAM technology employed for the experiments in this thesis (thus $Pt/Al2O_3/TiO_2/Pt/Ti$ RRAM devices) can be found in the work of Stathopoulos et al. (2017); Michalas et al. (2017). In Fig. 4.1, an example of the forming process is showcased. Usually for the initial forming process an approximate resistive state of below 700kΩ-1MΩ is set as the goal of the initial state of the formed devices (thus the device that completed its forming process). A train of voltage pulses are applied to the device under test. after each programming pulse a reading pulse is applied to check if the RRAM device under test is formed (thus achieve a predefined resistive state). The reading pulse is of predefined amplitude and duration characteristics that are not altered throughout the duration of each forming process. The programming pulse are progressively increased in magnitude during a single automated process. In most cases, programming pulses of short duration are preferred to form successfully RRAM devices.

After the forming of the RRAM device is complete, a process of testing and programming the device to an appropriate resistive state, with regards to the specific experiment or simulation aimed to be deployed on, can be initiated. Through this process, I am deciding on a specific range of resistive states that this device can be programmed on and be compatible with the specific requirements of the CMOS circuit part of the experiment. The usual goal is the initial programming of the RRAM at approximately 400-500kΩ as the main upper limit of stable reconfigurable operation. Depending on the requirements of specific experiments under test, an appropriate number of RRAM devices are programmed to specific resistive states from this common initial state. For higher current limiting effect higher resistance states are set while for faster operation (thus smaller RC delay in circuits) smaller resistance states are preferred. The best performance of the specific RRAM devices under test (thus $Pt/Al2O_3/TiO_2/Pt/Ti$ (10/4/24/10/5) nm configuration) can be observed for the resistive range of below approximately 150kΩ where better controllable resistance programming operations and more stable non-drifting reading operation can be achieved. Resistive state above 150kΩ showcase also good programming capabilities that exhibit relatively stable programming and reading operation but usually require more extensive characterisation and testing processes to verify that the endurance is within the specification of each experiment and/or simulation. The behaviour of the RRAM technology under test is also showcased and discussed in more detail by Stathopoulos et al. (2017). The programming can be achieved through automated or manual programs as enabled by the ArC One measurement board. Although, in many cases, the automated programming process can provide satisfactory accuracy in setting a device to a specific resistive range. In other cases, the automated process that is applied programs the device under test either on or close to the required resistive range and afterwards a brief manual programming procedure is required to set the device to an appropriate resistive state. Examples of a RRAM device programming is shown in Fig. 4.2.

Finally, additional testing is required to showcase the endurance and stability of the programming state of the device, its behaviour within a specific range of voltage stimulus. A detailed characterisation for a specific range of voltages that the device is expected to be operated under is also performed. Through this process we can better understand the non-linearity of the current-voltage characteristic behaviour. The significance of this process is two-fold. Firstly, data gathered from this process can showcase if the device is expected to show resistive state drift under the reading operations (usually below 1V) of the hardware experiments. Secondly, this process is employed for extracting the appropriate parameters towards mapping a real-world device response to a unique RRAM device model instance. An example of this process can be seen in Fig. 4.3. Through this process we can extract the parameters of large sets of devices and have a set of different model instances fitted closely to the actual behaviour observed for the real devices' measurements.

The figures presented in this section showcase examples of the processed described above for different RRAM devices. For each device employed in the following sections of this thesis, similar procedures are used to characterise and prepare the devices that are needed for the experiments. For the experiments and simulations we are performing static RRAM-based operations which means that the main process followed is the forming and programming of a RRAM device to an appropriate resistive state and then the testing of its stability under a specific pattern of stimulus. All the aforementioned programming and testing procedures are included as software programs for the ArC One measurements board and details about these procedures can be found in the relevant description of the specific equipment by ArC Instruments (UK).

4.3 Design and Operation

As discussed in Chapter 2 (shown in Section 2.4), differential CM-TLG designs consists of two parts, as shown in Fig. 4.4. These parts are the differential (showcased with red shading in Fig.4.4) and the sensor part (showcased with the green shading in Fig.4.4) . The differential part consists of the input and threshold branches, controlled by the input and threshold input vectors, respectively. Within each branch, the RRAM-based resistive weight arrays are implemented by 1T1R computing arrays. Each composite 1T1R element essentially forms a RRAM-based resistively source-degenerated pMOS transistors. Each 1T1R ensemble receives a digital input signal controlling the gate of the pMOS transistor; a single element of the branch's input vector, with the accompanying RRAM device defining the contribution of each such vector element. If the input is low (active), then a RRAM-dependent current flows from that 1T1R sub-branch towards the sensor part. Additionally, each of the differential branches is power-gated by a serially connected back-to-back (BtB) pMOS circuit. Through the use of the power gate on the differential arrays reading voltage connection, I am controlling the power-off of the input and threshold arrays during the evaluation phase of the TL operation.

The sensor part is the thresholding element of the circuit is essentially performing a simple analogue comparison operation between the differential in-flowing currents. Due to the design of the sensor partially as a sense amplifier for memory reading operations (i.e. CMOS-based latching element) the output of the sensor is binary. By convention, the canonical output of the sensor (thus the TLG) is indicating if input 1T1R-dependent analogue current signal is greater than the threshold current or vice versa. For all purposes the sense amplifier is structurally similar to a SRAM memory cell but with usually larger MOSFET devices since the sensing and latching based on relatively small differences in the charging/discharging of bit-lines, e.g. in DRAM banks. More specifically, the latching element consisting of two back-to-back connected CMOS-based inverters, forming a positive feedback loop (shown in the sensor part of Fig. 4.4). Due to the use of similar circuits extensively in conventional memory systems when reading from memory, naturally, the RRAM-based TLG design is IMC compatible if the fact that the 1T1R arrays are memory words (similarly to 1T1C in DRAM memory banks) and the sensor is a sense amplifier with additional equalisation circuits is considered. Furthermore, towards enhancing the cascading capabilities of such TLG designs, two additional CMOS inverters (one per sensor's output) can be added at the outputs of the TLG. The additional output inverters can be used essentially for buffering of the output and towards avoiding any voltage level degradation and isolating the sensor part from the circuitry connected further down the logic cascade. The power supply to the sensor part with the isolation inverters is controlled by a pMOS power gate to enable a full turn-off the sensor when the circuit is initialising. A transistor-level schematic of the implemented circuit, that contains more details of the actual transistor and 1T1R arrays placement, is provided in Fig. 4.6a (depicting the case study of 2-input RRAM-based TLG).

The main design features of my proposed RRAM-based TLG are based on a set of different TL circuits from the literature, such as the designs presented by Dara et al. (2013); Mozaffari et al. (2018). Similar to Dual Clocked Current Mode Threshold Logic Gate (DCCML) design, showcased by Dara et al. (2017), I used a common voltage supply for both sensor and differential parts. Furthermore, the differential 1T1R banks were connected to the outputs of the sensor, thus speeding up the sensor's decision-making operation (differential current comparison) by removing the RC paths introduced by the 1T1R array path, a design feature similar to transistor-based coupled inverters with asymmetrical loads (CIAL), as shown by Hidalgo-Lopez et al. (1995), threshold logic CIAL (CIAL-TL), as shown by Ramos et al. (1998), and their RRAM-based counterpart shown by Dara et al. (2013).

In more details regarding the operation of the showcased gate, similarly to other current mode design, the RRAM-based TLG performs a current comparison operation in

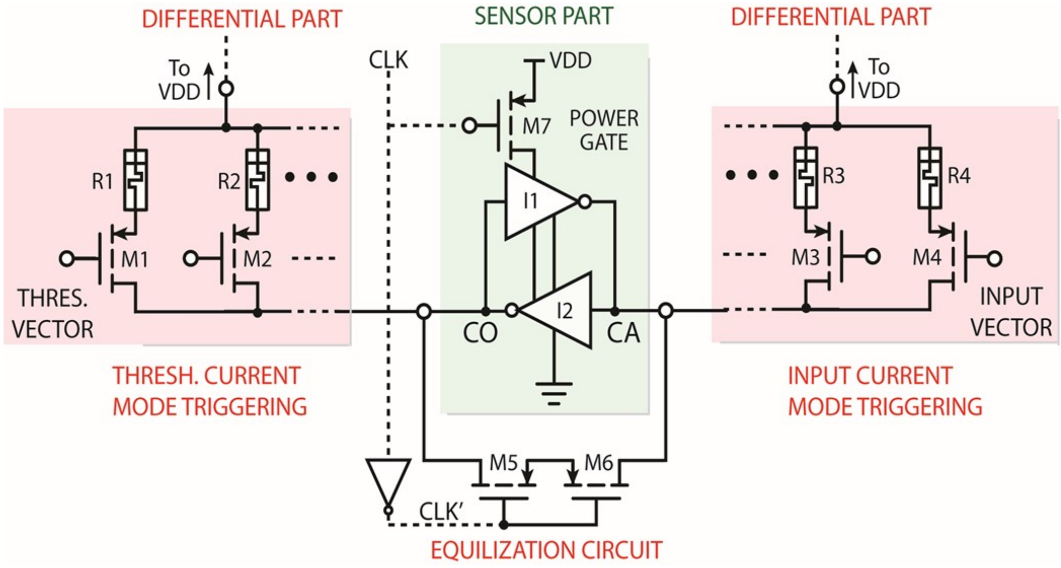

FIGURE 4.4: Schematic of a RRAM-based differential current mode TLG design. The specific design implemented and tested in this chapter is also referred to as RRAM-based Current Mode Threshold Logic Gate (MCMTLG). The two basic parts of design are the RRAM-based 1T1R array performing a dot-product multiplication between the RRAM memory contents (resistive memory state also referred to as memristance, i.e. memory resistance) and the binary input vector controlling the accompanying pMOS MOSFET device, and the sensor determining which 1T1R array outputs greater current. Further details for the differential TLG design can be found in Chapter 2. The canonical (CA) and complementary (CO) circuit nodes are the output nodes during the evaluation phase while they are connected to the differential 1T1R arrays (thus the output of the MAC operations are used as input to the sense amplifier). The outputs are available during the evaluation phase when the differential current flows have been compared and a final stable state of the sensor part has been obtained. The clock signal (CLK) and the complementary clock signal (CLK′) are controlling the sensor's power gate and the equalisation circuit, respectively. Hence, the CLK signal defines the transition between the two TLG operation phases, equalisation (reset) and evaluation (set).

two phases. During the equalisation phase, the differential part is powered-on and the sensor part is powered-off. At the equalisation phase the voltages at CO and CA are forced to be almost equal by the shunting BtB pMOS devices between the branches. The sensor part is not yet powered-on thus no comparison is performed. Next, in the evaluation phase, the inter-branch shunting is released, the differential part is powered-off and the sensor part is powered-on. This has the effect of forcing the sensor part into an unstable equilibrium where the two inverters in the positive feedback loop are pushing towards changing state. This allows differences on nodes CO and CA to be amplified by the positive feedback action of the BtB-connected inverters of the sensor part with the node having seen the larger analogue current signal to be quicker in winning the ”competition” between the inverters of sensor. Notably the differential part is cut-off from the voltage supply during the evaluation phase, thus disabling the current flow towards the sensor, leaving only a brief window for the sensor (during the short period of CLK falling and its complementary signal CLK' rising) of achieving a stable and correct transition to a binary memory state, based on the small voltage differences settled during the equalisation.

4.4 Experimental Setup and Measurement Methods

For this practical RRAM-enhanced circuit implementation I used RRAM devices designed and fabricated in-house by Stathopoulos et al. (2017). All the RRAM devices used in the experimental setups are in 3x3mm2 chips that are wire-bonded to PLCC68 packages. Each RRAM device is a 60x60 um2 cross-point of top and bottom electrodes. All circuits implemented throughout this work rely on the rich dynamic behaviour of an in-house Metal-Oxide RRAM technology employing MIM structured devices based on the specific device configuration presented by Stathopoulos et al. (2017). Originally, the devices were fabricated on 6-inch SiO_2/Si wafer with bottom electrodes (BEs) and top electrodes (TEs) patterned using optical lithography, e-beam evaporation and liftoff processes. Similar processes were adopted for the active layer patterning, except that sputtering was used for the deposition with a magnetron-sputtering tool. The active layer is constituted of TiO_2 and Al_2O_3 thin-film metal-oxides. After dicing, 3x3 mm^2 wire-bonded chips containing RRAM device devices were obtained, with MIM stacks constituting of $Pt/Al2O_3/TiO_2/Pt/Ti$ (10/4/24/10/5) nm. The RRAM devices employed for the purposes of the TLG-related experiments are usually programmed to below 150kΩ resistive range towards showcasing faster operation of the TLG circuit due to lower RC delay as well as due to higher controllability of programmable resistive states and endurance under reading stimulus (as discussed in Section 4.2).

All hardware experiments exhibited in this chapter were carried out on a breadboard proof-of-concept circuit based on the design showcased in Fig. 4.4 and Fig. 4.6a. An bench-top power supply was used to provide the power rails V_{DD} and GND of the implemented circuits. The results were gathered through a Rigol MSO4000 oscilloscope. For the experiments on the different weight configuration both packaged devices and wafer devices, accessed through a probe station, were used. The devices were connected to a breadboard hardware circuit using a breakout board alongside the MOSFET discrete components. The power supply used for the experiments was 0.65V, to avoid any unwanted state programming through the trains of reading pulses applied to the differential part of the circuit. For the pMOS devices I used the NDP5020P (1H10AA) model while for the nMOS devices I used the SUP85N02-03 (T32BAA) model.

I measured the circuit response through the Rigol MSO4000 Oscilloscope. The input vector and the clock signal were produced through Python programming language scripting, in software, (use of mini computer RPi3 Model B and Python programming language to employ the input-output pins for generating the appropriate signals) and custom resistive voltage divider -based converter, in hardware, to appropriate configuration-dependent voltage levels. It worth noting that for the case of 3-input and 4-input experimental configurations of the RRAM-based TLG I measured the input vectors and the clock signal through the Logic Analyser (LA) digital probes, due to the limited number of analogue probes available from the oscilloscope. In each experiment, the memristive devices used were programmed in the required state using an ArC ONE instrument board (ArC Instruments, UK). All devices used for all the experiments were located on the same die, i.e. only one memristive device package containing a total of 32 RRAM devices. Having decide upon the details of the components of my practical implementation, I demonstrate experimental setups to validate the functionality of the circuit and gaining insights regarding its real-world constrained operation.

4.5 Measurements and Logic Operation Validation

The RRAM-based TLG showcased in this chapter was built using discrete MOSFET components and real RRAM devices connected on a breadboard. Similar to Dual Clocked Current Mode Threshold Logic Gate (DCCML) design, as shown by Beiu et al. (2003b), I used a common voltage supply for both sensor and differential parts. Furthermore, the differential 1T1R memory arrays are connected outside of the sense amplifier structure and more specifically to the inputs-outputs ports of the sensor. This design enables the speeding up the sensor decision-making operation (thus comparison and classification) regarding the performed current comparison by removing the RC paths of the parallel 1T1R-based network. If the RC delay is introduced by the differential arrays being inside the dual V_{DD}-GND path of the sense amplifier this impede the compare-and-classification operation of the sensor, as shown by Beiu et al. (2003b); Hidalgo-Lopez et al. (1995); Bobba and Hajj (2000). This practice is similar to the Coupled Inverters with Asymmetrical Loads (CIAL) technique, showcased by Kulkarni et al. (2016). For my physical implementation with discrete components, the differential circuit uses two pMOS transistors back-to-back for power-gating. This is done to control the connection of the digital input and threshold vectors to the reading voltage supply, thus avoiding logic state degradation of the latching element during evaluation phase, as well as improved operational stability even with noisy input vectors. The BtB pMOS circuits also enable lower power consumption, due to the fact that the differential part is cutoff and does not consume power during the evaluation phase. The voltage supply V_{DD} used in the proposed design is V_{DD}=0.65V, thus ensuring that the memristive devices being use cannot be accidentally programmed during TLG operation. Furthermore, a BtB pMOS circuit was used also for the equalisation circuit that reset the sensor part before the evaluation being performed. V_{CLK}, which control the operation cycle of equalisation/evaluation, and the input vector's high voltage levels, are set to 0.9V. The low logic level for both the V_{CLK} and the input vector is set to 0V (GND node). A small computer, and more specifically a Raspberry Pi 3 Model B, is used to generate the clock signal as well as the digital input and threshold vectors used for the experiments of the RRAM-based TLG described below. The appropriate outputs are driven by the pin connections of the computer.

In the 2-input circuit experiments, the threshold branch consists of a single 1T1R-based memory/logic element (TH) while the input vector consists of two 1T1R-based elements. The example can be seen in Fig.4.6 where the 2-input RRAM-based TLG schematic is shown in Fig.4.6a and the measured results are shown in Fig.4.6d-h. Simple Boolean logic gates, such as AND and OR gates, can be interpreted as different flavours of majority gates. For example, a 2-input MAJ-1 gate is equivalent to a 2-input OR gate and a 2-input MAJ-2 is essentially the same function as a 2-input AND gate. Towards mapping these logic function in 1T1R-based memory arrays, RRAM resistive weights that are programmed in such way that some simple inequalities are realised are required. More specifically, for an OR gate logic mapping, the RRAM-based weights need to be set that either input has a larger weight than the threshold branch (i.e. $M1, M2 < TH$). Similarly, for the case of AND gate logic mapping the requirements of: $M1, M2 > THandM1||M2 < TH$ need to be satisfied in hardware by programming appropriately the RRAM weights. This can be extended to much more complex threshold functions, as shown by Leshner et al. (2010); Leshner (2010), with the important requirement that the logic function need to be linearly separable, as discussed in Chapter 2 regarding the simple artificial neurons operation.

For the case of the 2-input AND experiment (mapped in the proposed RRAM-based TLG), I used the RRAM-based weight configuration of $(M1, M2; TH)$ = $(60.5k\Omega, 60k\Omega; 33k\Omega)$, thus satisfying the requirements for the AND TL inequality equation (majority-2 function). From the CA output it is possible to measure the AND function circuit response, while from the CO output the complementary function, NAND, can be measured. The measured response of the AND/NAND TLG configuration is shown in Fig. 4.6c and the input vector is showcased in Fig. 4.6e and in Fig. 4.6f for the first input (IN1) and the second input (IN2), respectively. The clock signal V_{CLK} that controls the TLG operation (by enabling/disabling the equalisation circuit) is shown in Fig.4.6g. Due to the use of binary input vectors the quantised corner points of the 2D area (Fig.4.5b,c) area of interest.

The canonical (CA) output, which is defined by convention where AND and OR functions can be measured, and complementary (CO) output, where the complementary NAND and NOR functions can be measured, can be seen in Fig. 4.6a. Fig. 4.6c showcase the CO output of the circuit configured to perform the 2-input AND/NAND gate

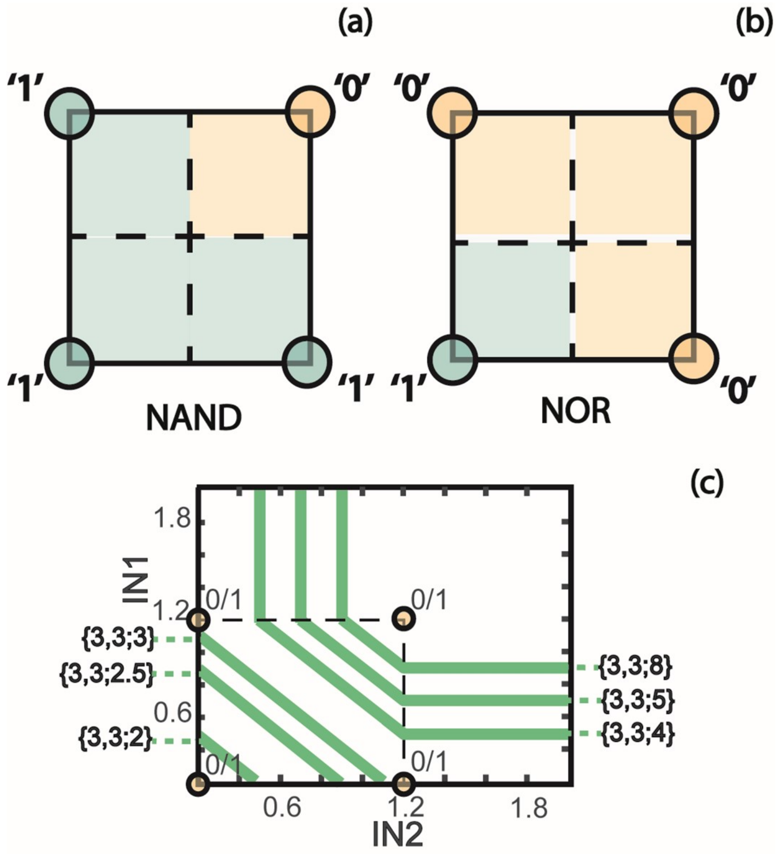

FIGURE 4.5: Concept-level figures highlighting the space separation performed by TLGs and the programming capability of the threshold. (a), (b) Indicative 2D input space splitting performed by the 2-input RRAM-based TLG design, for the NAND case (a) and the NOR case (b) respectively. (c) LTSpice simulation of this TLG design shown in Fig. 4.4 using resistors instead of RRAM devices. Changing the threshold weight alters the decision boundary of the classification operation performed in the TLG. Each weight configurations denoted on the side of each case, in the form of a, b: c, in units of MΩ, where a, b are the weights mapped in the input array and c is the threshold/bias resistive weight which defines the classification boundary.

(NAND(CO)), while in Fig. 4.6d the measured CO output for the OR/NOR configuration (NOR(CO)) is shown. The clock signal is determining the equalisation/reset phase (clock HIGH) and evaluation/set phase (clock LOW) cycle of operation. The outputs NAND(CO) and NOR(CO) are valid during the evaluation phase, while during the equalisation it it is shown that the output signals stay at an intermediate unstable logic level. The V_{DD}=0.65V and the V_{CLK}=V_{IN}=0.9V (for the logic '1'). It is worth noting that due to the use of pMOS devices in the 1T1R sub-circuits of the differential part, the logic for HIGH input voltage the corresponding input is non-conductive (logic '0') while for LOW input voltage the corresponding input is conductive (logic '1').

For the second case presented in Fig. 4.6, the differential part configuration was set as $(M1, M2; TH) = (33.8k\Omega, 18.3k\Omega\;\; 41.6k\Omega)$ for mapping a 2-input Boolean OR function. These values are chosen from the available dynamic range of the RRAM resistive state

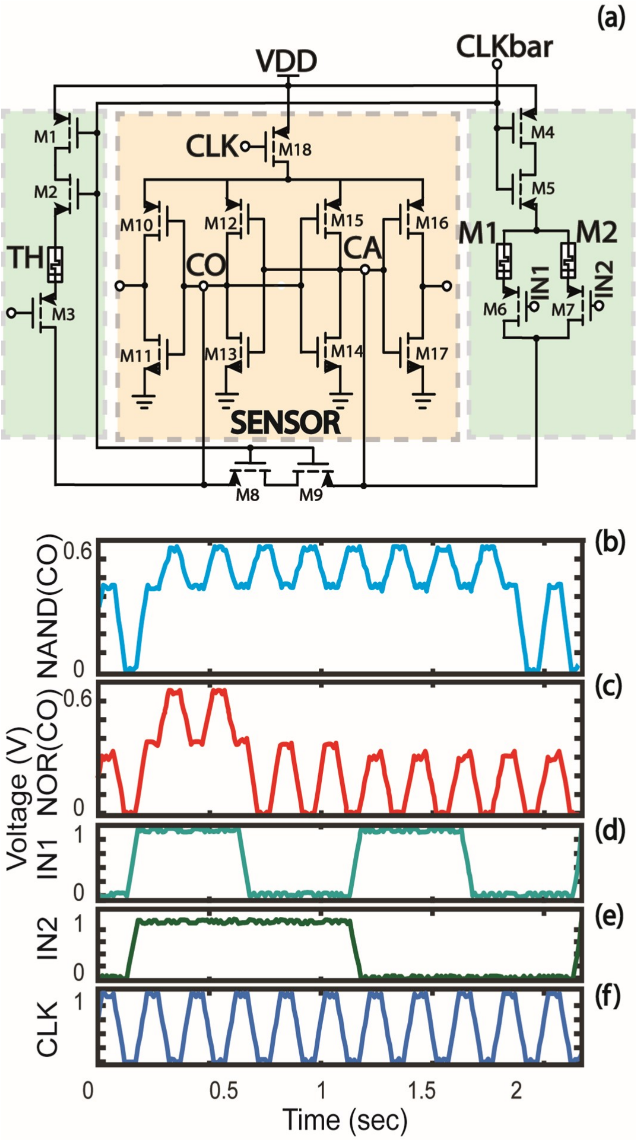

FIGURE 4.6: The circuit schematic and the measured response of the practically implemented 2-input RRAM-based TLG configuration. (a) Circuit schematic of the practically implemented RRAM-based TLG design showcased here (also referred to as RRAM-based Current Mode Threshold Logic Gate (MCMTLG)). (b) Measurements taken from an experimental implementation of a RRAM-based TLG configuration showcasing NAND functionality, thus an appropriate RRAM-based weight configuration used for mapping AND/NAND logic function. (c) Similarly to the previous case a different RRAM-based weight configuration showcase a NOR functionality as measured from the CO output of the RRAM-based TLG configured to map OR/NOR logic function. (d, e) The 2-input digital input vector signals are showcased. (f) The clock signal controlling the equalisation/evaluation phases. Regarding the RRAM weight configurations, I employed the following sets in the experimental hardware implementation:M1, M2; TH = 60.5kΩ, 60kΩ 33kΩ 33kΩ, 18.3kΩ 41.7kΩ for the AND/-NAND function and OR/NOR function, respectively. Due to the use of pMOS devices as the accompanying MOSFET device for the 1T1R differential arrays, the HIGH and LOW voltage level, of the input vector, corresponds to non-conductive and conductive MOSFET device, respectively.

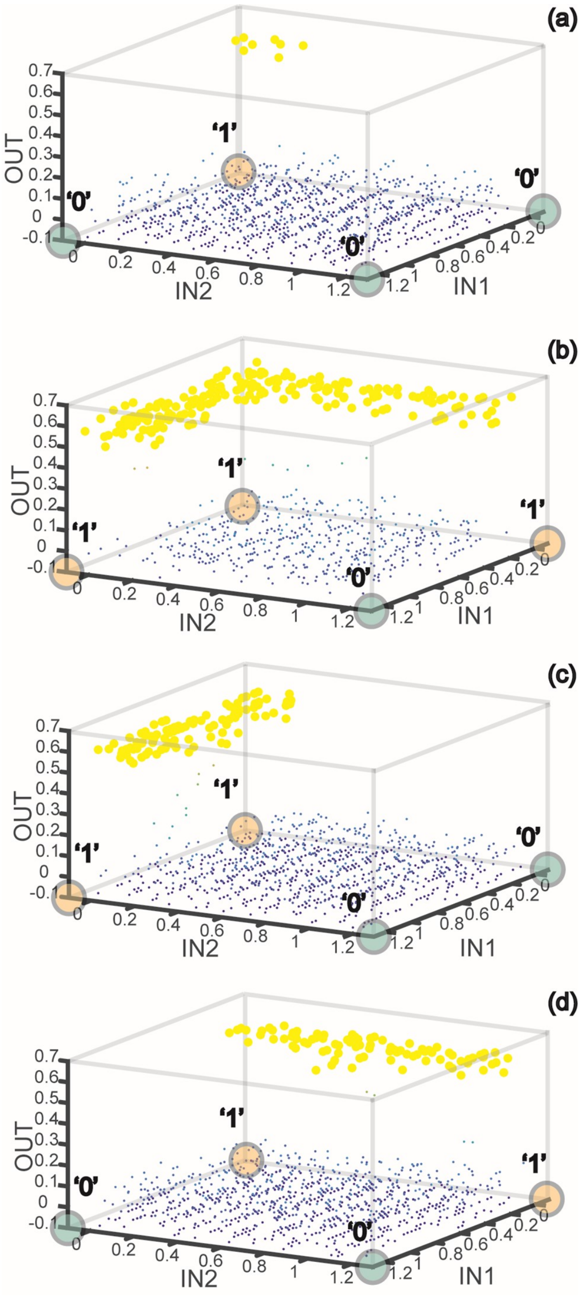

FIGURE 4.7: Experimental measurement of the RRAM-based TLG under test. In this example, the TLG configuration classifies a 2-input vector, thus 2 1T1R composite components are employed from the input array. By using analogue inputs (ramps from 0V to 1.2V per input) a clearer map of the circuit response is created. Four different cases are presented in this experiment, mapping every possible TL Boolean function. (a) AND TLG (M1, M2 ¿ TH and M1——M2 ¡ TH), (b) OR (M1, M2 ¡ TH), (c) OUT = IN1 (M1 ¡ TH, M2 ¿ TH) and (d) OUT = IN2 (M1 ¿ TH, M2 ¡ TH). The memory configurations for each case of the experiment is set as follows: AND 109.1kΩ, 105.7kΩ; 86.7kΩ, OR 83.6kΩ, 85.9kΩ 262.5kΩ, IN1 78.4kΩ, 233.2kΩ; 109.1kΩ and IN2 253.5kΩ, 82.8kΩ 39.2kΩ. The input/output mapping on the quantised, binary space is also included. An input activates (deactivates) a specific RRAM device branch of the 1T1R array by getting LOW (HIGH), due to the use of pMOS as the transistors in 1T1R computing arrays.

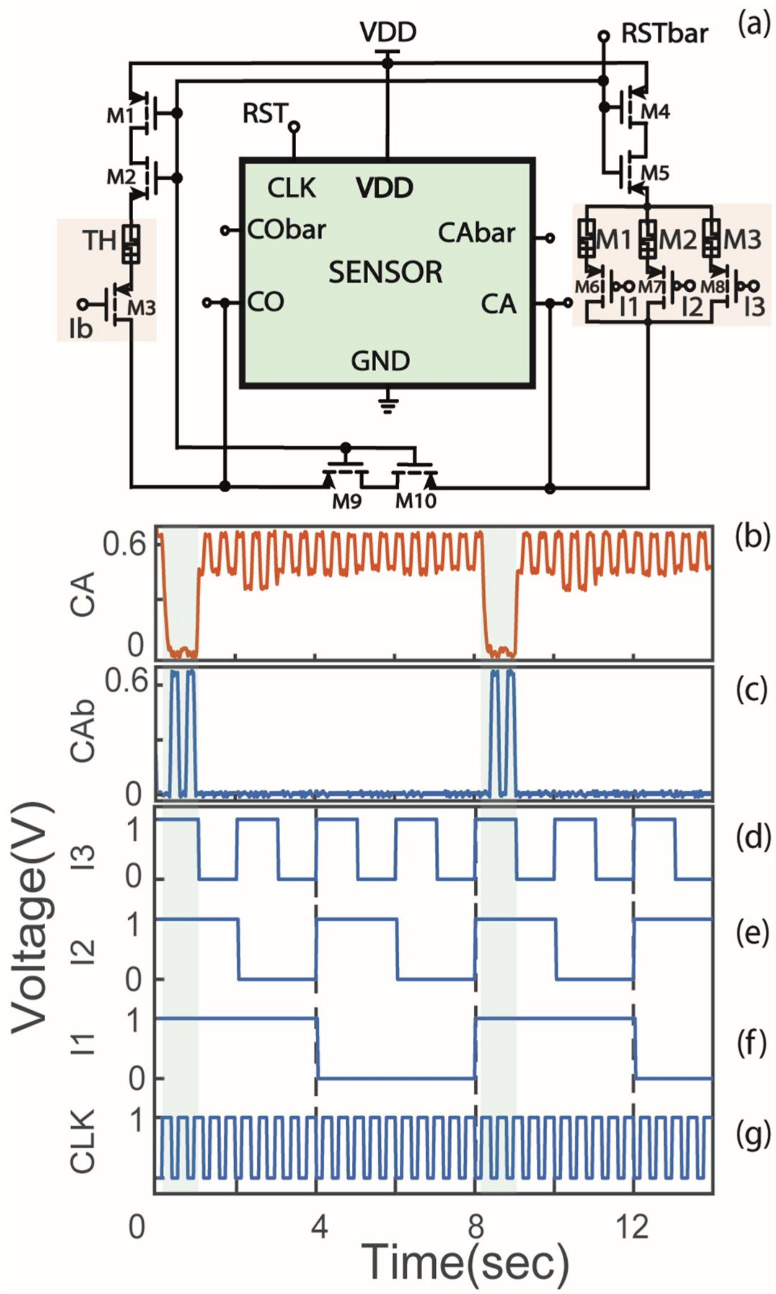

FIGURE 4.8: Practical implementation of 3-input RRAM-based TLG experiment. (a) Schematic of 3-input MCMTLG, where the sensor component now includes the CMOS latching element plus two additional restoration inverters per output. An example of how a 3D input space is split by the TLG. (b) Canonical (CA) output and (c) and the complementary (CAb) output as buffered by a CMOS inverter gate to isolate the sense amplifier's output nodes from the following circuit cascade. As can be seen in Fig. 4.8(c) the CAb output does not include the full voltage information of the CA but instead "selects to transmit only the final output state available during the evaluation phase. (d-g) control signals (3-input vector I1, I2, I3 plus clock). The clock signal is the same as the RST signal in this schematic since the resetting/initialisation operation (performed during the equalisation phase) is controlled CLK. The weight configuration for the OR3/NOR3 is M1, M2, M3; TH=31.5kΩ, 30kΩ, 28.2kΩ 68.2kΩ.

programming capability. Similarly to the configuration of the AND/NAND case study, the input vector (Fig. 4.6e-f) and the clock signal (Fig. 4.6g) are the same for this operational configuration. Both OR and NOR, as well as the AND/NAND, functions are performed simultaneously due to the complementary bi-stable operation of the sensor part.

To highlight the reconfigurability of the MCMTLG and its effect on the defined classification boundary, additional SPICE simulations (for this case the LTSpice simulation environment was employed for these preliminary simulations) were performed using an resistor-based emulator version of the proposed circuit. For the transistor components of the simulated emulator circuit I used commercially available 0.35um technology node. In Fig. 4.5c different plane classification thresholds are showcased by changing the threshold RRAM-based resistive weight values of the 2-input MCMTLG emulator configuration and recording the decision boundary for each case. The first three RRAM-based weight configurations $3M\Omega$, $3M\Omega$; $2M\Omega$, $3M\Omega$, $3M\Omega$; $2.5M\Omega$, $3M\Omega$, $3M\Omega$ $3M\Omega$ will result in an OR/NOR logic gate (considering the generation of both canonical/complementary outputs from the TLG design) while the remaining three configurations $3M\Omega$, $3M\Omega$; $4M\Omega$, $3M\Omega$, $3M\Omega$; $5M\Omega$, $3M\Omega$, $3M\Omega$ 8MΩ in an AND/NAND logic gate. The different resistance scale used in the simulations (in $M\Omega$), compared to the hardware experimental setups (in $k\Omega$), is based on the goal of testing the MCMTLG for the $M\Omega$ memristance range. The high current limiting effect of these resistances in parallel 1T1R computing arrays is moving closer to an implementation of a process invariant TL circuits.

In Fig. 4.7, I am showcasing a practical experiment to validate a 2-input RRAM-based TLG with sweeping inputs (essentially analogue inputs from logic LOW to logic HIGH), thus getting a detailed view of the inner working of the physically implemented circuit. The circuit has been developed for binary input space (the input vector control the accompanying transistors of the 1T1R) and responds in binary values, but it is possible to capture the full behavioural aspect of the RRAM-based TLG by introducing analogue ramp signals as inputs to its input 1T1R array. The analogue ramp signals are connected to the gates of the accompanying MOSFET devices (or the 1T1R array). As shown in the results of this experiment, the full evaluation response of a 2-input RRAM-based TLG can be mapped. For the four cases of the response in analogue input signal I used 3 RRAM devices, one for the threshold 1T1R array and 2 for the input 1T1R array. I program the RRAM devices for performing all the possible 2-input TL function. The logic functions mapped in RRAM devices for the TLG are the $AND(M1, M2 > TH$ and $M1||M2 < TH)$ (see Fig.4.7a), OR $(M1, M2 < TH)$ (see Fig.4.7b), $IN1(M1 < TH, M2 > TH)$ (see Fig.4.7c) and $IN2(M1 > TH, M2 < TH)$ (see Fig.4.7d). The RRAM memory configurations for each case of the experiment is set as follows: AND $109.1k\Omega, 105.7k\Omega; 86.7k\Omega$, OR $83.6k\Omega$, $85.9k\Omega$ $262.5k\Omega$, IN1 $78.4k\Omega, 233.2k\Omega$ $109.1k\Omega$ and IN2 $253.5k\Omega$, $82.8k\Omega$ $39.2k\Omega$.

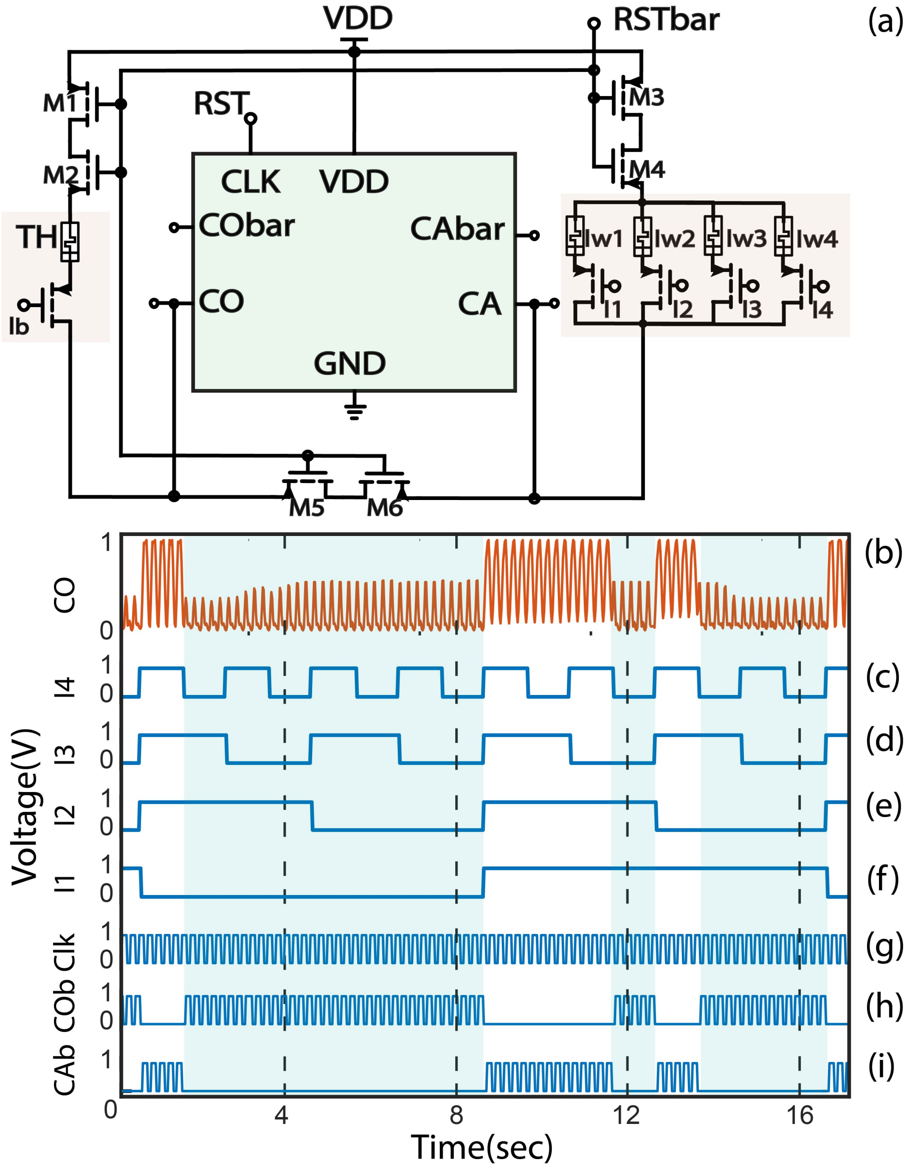

FIGURE 4.9: Practical implementation of 4-input MCMTLG. (a) Schematic of the 4-input MCMTLG. (b) Circuit response measured at the complementary output (CO). (c-f) The signals for the digital 4-bit input vector (I1, I2, I3, I4). (g) The clock signal controlling the circuit. The clock signal is the same as the RST signal in this schematic since the resetting/initialisation operation (performed during the equalisation phase) is controlled CLK. Additionally, to the analogous measured CO output of the sensor, the threshold-based processed responses from the isolation inverters are also shown (h,i). The memristive weight configuration is M1, M2, M3, M4; TH = 30kΩ, 21.6kΩ, 31.2kΩ, 25.2kΩ 19.1kΩ. This configuration of the MCMTLG performs the following Boolean function: F= MAJ2(I1,I2,I3,I4), thus needing at least two conductive inputs for the total input current to be larger than the threshold current.

In Fig. 4.9, a 4-input RRAM-based TLG is tested, performing a MAJ-2 (majority-2) TL function, implemented by configuring the 1T1R TL arrays as follows: M1, M2, M3, M4 TH = $30k\Omega$, $21.6k\Omega$, $31.2k\Omega$, $25.2k\Omega$ $19.1k\Omega$. In Fig. 4.9a the schematic of the circuit practically realised is shown, which include an additional 1T1R for the input network of the differential part. In Fig. 4.9b-i the circuit response and the corresponding control signals (input vector and clock) are shown. It is worth noting that the CAb and COb sensor outputs are the outputs of the isolation inverters, included into the sensor part by the showcased TLG design.

Differential arrays with a larger number of 1T1R composite elements enable increased fan-in capabilities of the TLG design Thus, the complexity of linearly separable functions that can represented on the 1T1R-based memory arrays is higher and further logic scaling can be achieved by eliminating larger Boolean gate-based networks with TLGs, as shown by Leshner et al. (2010). The replacement of large multi-stage Boolean CMOS gate logic networks, with an equivalent TLG circuit inside an IMC system, can result in significant reduction in hardware cost, in power consumption and IC area as well as reduction of critical signal path length, thus even further improvements in performance and reduction in power and area, as discussed by Leshner (2010).

4.6 Comparison of proposed TLG design with state-of-art

Through the aforementioned measurements an investigation of the feasibility of such circuits using state-of-the-art RRAM technology is performed. Although, the main contribution presented in this work is the practical implementation of a RRAM-based TLG using real RRAM components, the specific design investigated here is not the only embodiment of RRAM-enhanced TL circuits and systems. A wealth of other designs can be found in literature, albeit only examined through simulations, thus making it difficult to carry out an immediate comparison. In order to provide comparison data, I have modelled the RRAM-based TLG design presented in this chapter in the industry-standard Cadence tool using commercially available 65nm MOSFET technology node, for the CMOS parts of the design, and a Verilog-A RRAM device model, showcased in Messaris et al. (2018) and discussed briefly in Chapter 2. The RRAM model is fitted based on measurements taken from real RRAM devices, through a method showcased by Messaris et al. (2018). For the simulations performed in this section, the RRAM model parameters are being set to the parameters showcased in the example case study exhibited by Messaris et al. (2018). Thus, the specific RRAM model has an initial state of approximately 16kΩ and can be programmed in controllable small steps from 5kΩ up to 32kΩ. For the proof-of-concept performance test of this section, no additional RRAM device was modelled through parameter extraction process. Instead multiple instances of the example case shown by Messaris et al. (2018) were used. Through this testing, it is possible, by extrapolating my findings, to predict the performance of my

RRAM-based TLG design if implemented in deep-submicron technologies and at the same time providing a better framework for comparison with existing RRAM-based TLG designs.

Towards comparing the simulated version of the TLG design presented in this chapter, I am providing here information regarding state-of-art RRAM-based hybrid TLG designs. The state-of-art design were selected towards representing the main design directions in TLG implementations. The Memristor-based TL (MTL) (shown by Rajendran et al. (2012)) design was one of the first TLG implementations that employed RRAM devices as its reconfigurable input weights. In MTL the memristive weights are isolated from the actual thresholding units (implemented in the form of a cascade of inverters) through the use of nMOS-based current mirrors. The current mirrors are preventing the reverse current flow from the accumulation node (the node that is connected with the CMOS output circuit) towards the passive RRAM devices. Each RRAM device is accompanied by a nMOS-based current mirror, thus making this implementation a 2-transistor-1-resistor (2T1R) type of circuit design. An additional current source accompanied by a pMOS-based current mirror is used to provide a reference current. Depending on the current drain each RRAM device is enabling through each corresponding nMOS-based current mirror, a final analogue voltage is used as the input to the output CMOS comparison and isolation circuit. The first CMOS inverter gate of the output gate cascade is performing the comparison operation which essentially translates to the capability of the remaining voltage at the accumulation node to drive the inverter gate or not. The Resistive TL (RTL) (shown by James et al. (2013)) is similar to the MTL, but with the simplification that the memristive network is directly connected to the output CMOS inverter gate, thus without the use of current mirrors per RRAM device. Instead a fully passive RRAM network is considered for the RTL implementation. Although this may introduce issues such a sneak path currents in a large array, the advantage is the capability for introducing much higher number of RRAM devices compared with other hybrid RRAM-MOS array nodes. The output inverter is employed as the comparator of the specific design similarly to how the comparison is effectively performed in other designs as well including the aforementioned MTL circuit. Operational-wise the RTL gate employs a type of low-complexity ratioed memristive network, thus a reconfigurable voltage divider network with pull-up components (mapping the input) and a single pull-down component (mapping the bias), thus having potentially better support for higher fan-in capabilities of the gate. As showcased by James et al. (2013), additional fully CMOS networks can be introduced as alternative output readout circuits depending on the requirements of the logic cascade for which the RTL design is employed for. For the purposed of comparing different TLG designs the initial simpler design with the single CMOS inverter gate is sufficient and also is creating a very competitive design in terms of number of devices required.

Another interesting implementations is the CMMTL design (shown by Dara et al. (2013))

which was one of the first differential TLG concept that used RRAM devices. In terms of design, CMMTL is similar to my proposed MCMTLG implementation, having its differential part separated from the sensor part, thus out of the current paths of the latching component a design more popular in older fully CMOS designs, as shown by Bobba and Hajj (2000). The design proposed by Dara et al. (2013) is employing 1T1R-based arrays for mapping the input and threshold vectors. The differential part in this design is logically separated into two sections of positive and negative weights applied for both the input array and the threshold array. The negative weights are introduced as weights contributing the current to the opposite current injection node of each related weight array. For example, a negative input weight is a weight placed in the threshold array controlled by the specific input, thus its contribution is competing with the canonical input array (which in this case is considered positive input array). This simple but effective methods is used to provide a solution to the requirement of both positive and negative weights for DNN-related applications. The CMMTL design provided promising results that reinforce the findings suggesting that the differential implementations have important performance advantages over non-differential TL designs, as discussed in Section 2.4 and Section 4.1. For example, as shown in Table 4.1, the CMMTL design showcased better power dissipation and delay metrics over the MTL implementation, as shown by Dara et al. (2013). Furthermore, another important implementation is the first order Threshold Function memristive Threshold Logic Gate (1-TF mTG), a more recent state-of-art differential current-mode RRAM-based TLG showcased by Mozaffari et al. (2018). In the work of Mozaffari et al. (2018), the hybrid RRAM-MOS TLG design is optimised for minimal transistor count and can achieve very low power consumption and delay. The differential part is using 1T1R-based arrays for the input and threshold vectors. In terms of design, the 1-TF mTG incorporates in a way the differential part into the sensor part, with the differential arrays being part of the pull-up network of the gate. The design is very interesting and is similar to the work of Bobba and Hajj (2000), which was one of the first to proposed CMOS-based CM-TLGs. The RRAM-based design is also compared with a fully CMOS equivalent design proposed by Mozaffari et al. (2017). The results shown by the more recent work of Mozaffari et al. (2018) showcase that the RRAM-enabled design showcasing a reduced number of transistor devices due to the use of RRAM devices for the weight reconfiguration in contrast with the larger MOS-based networks per input that were employed in Mozaffari et al. (2017). Better performance with the RRAM-devices is also observed.

For the performance metrics of the state-of-art hybrid TLG designs (shown in Table 4.1), I use published data that support the metrics of power consumption and delay. For the case of CMMTL, I used data for the delay metric is measured by Dara et al. (2013) using 45nm Berkley's Predictive Technology Model (PTM) technology node and for 3-input gate. For the power estimation of the CMMTL I could not gather relevant data from Dara et al. (2013), but in the comparative study by Maan et al. (2016) an estimation is provided of power supply with a commercially available 0.25um MOSFET technology node and for 2-input case study. The MTL data were gathered from Rajendran et al. (2012), using 45nm PTM technology node for a 3-input case study. The RTL metrics were gathered by James et al. (2013), using a commercially available 0.25um MOSFET technology node and the data refers to 2-input gates. The 1-TF mTG power and delay was measured by Mozaffari et al. (2018) using 45nm PTM technology node and the used values are for an indicative 3-input gate. With regards to the RRAM model that was employed for the simulation of each case, CMMTL, MTL and RTL used HP RRAM device model, as showcased by simulations of Rajendran et al. (2012); James et al. (2013); Dara et al. (2013), while 1-TF mTG simulations were based on VTEAM RRAM device model, as shown by Mozaffari et al. (2018). An overview of performance metrics for the aforementioned state-of-art TLG designs is shown in Table 4.1.

TABLE 4.1: Overview of State-of-Art RRAM-based TLG Designs Performance

	Metrics			
Hybrid TLG Designs	***Power (P_{op})***	***Delay (t_{delay})***	***#Device***	***MOS & RRAM Tech.***
Dara et al. (2013)	118.22uW	0.44ns	24	0.25um & HP
Rajendran et al. (2012)	35uW	6.1ns	24	45nm & HP
James et al. (2013)	9.2uW	0.45us	5	0.25um & HP
Mozaffari et al. (2018)	0.27uW	0.195ns	15	45nm & VTEAM

Although the experimental measurements of my RRAM-MOS TLG design is build upon of-the-shelf components (0.25um MOS devices), in simulation I was able to test my design using a modern 65nm MOS technology. Towards optimising my proposed RRAM-based TLG design (MCMTLG design) for the commercially available 65nm technology we can reduce the power of both operation phases (equalisation and evaluation). For the case of the MOSFET technology, I am employing W_p= W_n= 200nm L_p= L_n= 60nm devices. For the differential part we for the sensor part we use V_{READ}= V_{DD}= 700mV. The sensor part, build around the 65nm technology, is operating at subthreshold the evaluation operations are performed correctly for all the performance tests. The lower the differential parts reading voltage, the lower the sensitivity of the sensor part due to the more limited current values is. At the same time, smaller voltage swings are performed by the sensor part due to a sub-threshold voltage supply. Additionally, the clock period is set to t_{CLK}=30ns. Based on the performed simulation results, the t_{CLK} is optimised for performing the fastest cycle of TL operation while, simultaneously, ensuing that the timing constraints required for the sensor part to complete the comparison operation without errors are satisfied. The RRAM memory configuration used for the 2-input MCMTLG case is 30$k\Omega$, 30$k\Omega$; 18$k\Omega$, for the 3-input MCMTLG case is 30$k\Omega$, 30$k\Omega$, 30$k\Omega$; 18$k\Omega$ and for the 4-input MCMTLG case is 30$k\Omega$, 30$k\Omega$, 30$k\Omega$, 30$k\Omega$; 18$k\Omega$. As discussed earlier in this section, all the RRAM model instances are based on the example case study showcased by Messaris et al. (2018). The comparison-and-classification operation delay was measured for the start of the evaluation phase until the voltage difference between the two complementary outputs, CO and CA, of the sensor, reached approximately 90%×V_{DD}=630mV. The power of this simulated TLG configuration was measured by calculating the product of IxV (with I being the current flowing from the voltage supply and V the value of the voltage supply), over a single clock period (thus including both equalisation and evaluation phases) and for the case where all 1T1R branches of the input array are conductive. Thus, when the digital input vector case is set to all logic ’0’ per input since the accompanying MOSFETs in the 1T1R arrays are pMOS devices.

A performance overview of the proposed RRAM-based MCMTLG is shown in Table 4.2. By employing a competitive MOSFET technology (i.e. 65nm technology node for MOS devices) the simulated hybrid RRAM-CMOS MCMTLG design we can showcase competitive power (P_{op}) and delay (t_{delay}). This can be observed by comparing the performance metrics shown in Table 4.1 against the 3-input MCMTLG case showcased in Table 4.2. In terms of device count (transistors and RRAM devices), the proposed MCMTLG and CMMTL have higher count of devices when compared to the much simpler design of RMTL has the least components. In terms of delay, my proposed MCMTLG design is comparable with the state-of-the-art 1-TF mTG Mozaffari et al. (2018) implementation. It can be observed that, in general, differential implementations (i.e. MCMTLG, 1-TF mTG, etc) have lower t_{delay} delay per operation that non-differential ones.

TABLE 4.2: Comparison of Proposed RRAM-based MCMTLG for Different Fan-In Cases

	Metrics			
Fan-In	***Power*** **(*P_{op}*)**	***Delay*** **(*t_{delay}*)**	***#Device***	***MOS Tech.***
2-input MCMTLG	1.64uW	0.203ns	14×MOS+3×RRAM	65nm
3-input MCMTLG	1.79uW	0.124ns	15×MOS+4×RRAM	65nm
4-input MCMTLG	1.89uW	0.122ns	16×MOS+5×RRAM	65nm

It is worth noting that the power and delay of the operation are dependent of the exact RRAM-based weights comparison performed. For the operation cases tested in this section and showcased in Table 4.2, I am measuring the power and delay for the case when all input 1T1R branches are active. Thus, the 2-input, 3-input and 4-input MCMTLG are performing R_{IN}=15kΩ, R_{IN}=10kΩ and R_{IN}=7.5kΩ against R_{TH}=18kΩ (common threshold for all cases), respectively. Thus, is can be observed from Table 4.2 that the highest delay is shown for the 2-input case due to the relatively to the exhibited cases close comparison between 15kΩ and 18kΩ.

Furthermore, it is useful to note that the resistive weights are an important part of the experimental and simulated behaviour of the proposed TLG under test. Thus, for a set of RRAM-based weights that is programmed in a different range of resistance values (e.g. in hundreds of kΩ or tens of MΩ) a calibration will probably be required towards finding the optimal operational parameters, such as the minimum t_{CLK} and appropriate voltage supply. From this example, it is easy to showcase that the importance of exhibiting experimental setups to provide working proof-of-concepts does not mitigate how significant is the testing in a simulated environment with a much modern MOS technology.

4.7 Conclusions

In this chapter, I have presented a physical implementation of a proof-of-concept RRAM device-enhanced TLG. The reconfigurable memristive loads were employed to enable mixed signal data processing as they exist as analogue inference system to "external" stimuli (input vector applied to the differential network). The final output of this primitive analogue RRAM-based inference engine is processed by a low-complexity sensor circuit implemented by a sense amplifier (thus my design is easily integrated into IMC systems). Through the presented experimental results, I have shown that the comparison operation between the memristance of the threshold device and the composite impedance of the input network defines the circuit functionality, thus enabling a memory-dependent data classification. One important drawback that impedes the adoption of TLGs into VLSI systems and computer architectures is the lack of sufficient tools to synthesise TL-based circuits and systems inside computer architecture design, thus the implementation of specific TL-based logic structures remains mostly as part of custom IC design, as discussed by Beiu et al. (2003b). The design, testing and validation of these TL circuits in a practical setting and especially enhanced with emerging electronic devices, such as RRAM, can enable the better understanding of constraints that need to be considered before moving towards developing more realistic synthesis tools for TLG-based IMC architectures.

Through the presented experimental results, I have practically demonstrated the functionality of 2, 3-, and 4-inputs RRAM-based TLGs, showcasing the RRAM-enabled reconfigurability of the base design for all validated cases. Furthermore, I have investigated how RRAM-based TLGs behave when used outside their intended, digital-input operating regime. Experiments show the decision boundary shapes, which approximate bevelled L-shapes with lines parallel to the input voltage axes. Further investigation through simulations confirms these shapes and shows graphically how altering the resistive states of the memristive devices affects the specific decision boundary locations (most notably which points in the binary-quantised input space lie on the 'output = 1' side of the boundary or the 'output=0'). Finally, I implemented and simulated a

3-input TLG in a commercially available 65nm technology for the purposes of comparison with some state-of-art RRAM-CMOS TLGs. Towards that purpose, I have utilised the RRAM model by Messaris et al. (2018) which takes into account the non-linearity of the memristive device IV (a typical feature of many technologies, as discussed by Joshua Yang et al. (2013)). Results testify to the robustness of the TLG concept by confirming no perturbation of functionality and power/delay figures comparable, and indeed competitive, vs. state-of-art. My work thus provides experimental backing to a considerable body of literature where simulation work indicates the potential for highly energy- and area-efficient TL implementations exploiting experimental RRAM devices.

The results of the proposed proof of concept circuits and systems indicate that the specific non-volatile memory technology being tested could have great impact on the physical implementation of ANNs through truly mixed-signal implementation, with analogue processing being efficiently integrated into a otherwise digital system. As showcased, the RRAM-based analogue processing can have the potential in enhancing the speed of fundamental arithmetic and logic operations such as MAC computations etc. Thus, one of the most prominent applications of this logic technology is the implementation of unconventional more-than-Moore computer circuits, systems and architectures such as novel neuromorphic computing systems. Based on the findings on RRAM-based TLGs, it can be highlighted how low-complexity logic gates can have potentially a great impact on the building of powerful data processing systems. TLGs being a basic memory and logic co-location circuit is an ideal circuit structure for RRAM devices to be integrated with. The introduction of mixed-signal data processing at the nanoscale will negate the necessity for expensive data converters since the conversion from digital to analogue and vice versa can be done in place using the 1T1R-based computing arrays. The results exhibited in this chapter provide further insights towards developing primitive mixed signal computing blocks for hardware acceleration of cornerstone arithmetic and logic computer operation. At the same time, the RRAM-enhanced designs showcase the greater energy and area -efficiency of the hybrid CMOS-RRAM reconfigurable systems compared to conventional solutions.

Finally, the present work suggests some routes for further investigation. Notably, it is logical to expect that the capability of RRAM devices to support fine tuning of their resistances to become particularly valuable in TLGs of even higher dimensionality, thus with a higher number of 1T1R components for the input array. In a 5-input TLG, the number of possible points increases to 32 whilst the number of memristive devices only rises to 6. It would be useful to investigate how the number of possible majority gates implementable with n inputs increases vs the number of available RRAM devices $(n+1)$, and computing a 'logic density' metric. I conjecture that a higher number of inputs n will lead to a higher 'logic density' burden, which eventually reaches the maximum number of resolvable states attainable by the RRAM device. How this limits practical performance remains to be revealed in future work. Another avenue of investigation pertains to linking multiple TLGs together in larger combinatorial blocks and understanding, e.g. how the RRAM resistance-dependent delays may affect overall timing constraints (particularly in domino logic-type systems), in future prototypes for hybrid IMC systems.

Chapter 5

Composite application for mixed-signal data classifiers

As discussed previously in Chapter 3 and Chapter 4, key computational concepts for implementing neuro-inspired reconfigurable computing circuits are memory and system co-located. Thus, it will be useful to investigate how RRAM-based IMC systems can be employed to enhance the reconfiguring capabilities of computer architectures further than is possible through only CMOS-based conventional electronics.It is worth noting that TL operations are widely used in modern electronics, e.g. in the form of sense amplifiers inside DRAM/SRAM memory modules, simple non-linearly separable Boolean gates (i.e. AND, OR, and others), etc. Some techniques have already showcased methods for employing the circuitry of conventional memory arrays to perform simple bit-wise parallel Boolean operations, such as multi-input AND or other $(n-1)$-majority gate where n is the number of the inputs, as shown by Seshadri et al. (2017). The majority gates are part of the TLG family as discussed in Chapters 2 and 4.

Reconfiguration capability of a homogeneous sea-of-gates type of computing topology that uses memory-based primitive circuits is one of the most important traits of an IMC architecture design and the RRAM technology can enhance the processing capabilities by introducing analogue computing at the nano-scale. In terms of massively parallel data processing, an IMC system needs to be configured in such a way that the width of the parallel operations (due to the data vector size aimed to be processed) are accommodated in order to maximise the acceleration of the whole algorithm. If the IMC system has more resources that are typically used in a specific algorithm, part of the remaining resources should be altered to include the computing of other algorithms at the same time, thus enhancing the parallel computing capabilities of accelerators, thus enhancing parallelism not only data-wise but operation-wise. Towards reinforcing the IMC-based parallel processing, architectural reconfigurability, thus the capability of the system to adapt its accelerator structure depending on the operations it need to process in parallel, is considered a significant attribute for future accelerators and can have an important impact on neuro-inspired IMC accelerator architectures. Hence, the introduction of a low-complexity primitive logic circuit that can be configured to operate under a variety of mixed-signal computing modes can be highly beneficial to future IMC accelerators.

An important aspect of reconfigurable circuits and systems is the usage of circuits that can be easily operated between different computation modes without any extensive rewiring networks which can add significant area and power overhead to the system. The use of complex reconfiguration networks can defeat the purpose in introducing hardware-level programmability for accelerator designs. It is important for primitive reconfigurable circuits to be capable of re-arranging their computational structure with minimal hardware overhead in order to be capable of being integrated into a sea-of-gates system, i.e. one of the common configuration for IMC accelerator designs, with enhanced re-programming capabilities. Towards exploring further the computational capabilities of the RRAM-based IMC design, extensions of the previous work, shown in Chapters 3 and 4, are showcased in this chapter. A combined gate for enhanced reconfiguration capabilities for IMC as well as a proof-of-concept example for a RRAM-based Wake Up Circuit (WUC) system are shown and tested through Cadence Virtuoso Spectre simulations. The simulations are performed for circuit-level descriptions for both the RRAM-based combined logic circuit and the proof-of-concept WUC, thus providing additional information regarding our understanding of the expected performance of the proposed system.

5.1 Combined circuit for MAC and TL operations

The RRAM-enhanced data processing circuits, such as the ones showcased in Chapter 3 and 4, can be used to design low-power and area efficient IMC systems for performing massively parallel fundamental operations (e.g. mixed-signal vector multiplication etc.). As it can be observed from the analysis thus far, a valuable and important computational theme is that most RRAM-based logic techniques can be implemented as a form of programmable resistive networks that can operate either in voltage mode, i.e. by forming voltage divider structure, or in current mode, i.e. a set of voltages applied per each 1T1R component and the output is read through a common node. In many cases, additional circuitry for classification are employed at the output of the RRAM-based networks towards processing the output signal and ensuing cascading capability of the logic operation with other hybrid RRAM-CMOS or fully conventional CMOS circuitry, as discussed by Serb et al. (2018a).

The RRAM-based network is usually employed to implement a specific logic function where input control signals (binary or analogue) trigger the mapping of that input into

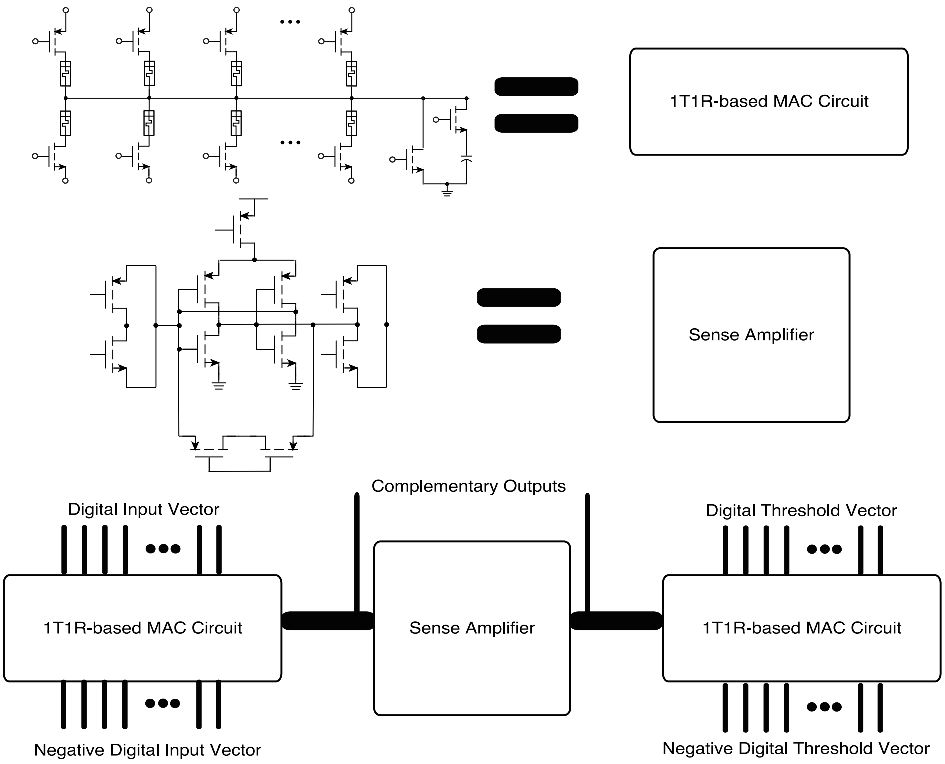

FIGURE 5.1: Schematic of a combined near-memory primitive circuit for MAC operations and TL computing. The circuit is based on the employment of two 1T1R computing arrays and a single sense amplifier. The circuits can be reconfigured to either the MAC circuit (shown in Chapters 3) or the RRAM-based TLG (shown in Chapters 4). The additional capacitor at the output of the MAC Circuit can provide analogue integration and storage capabilities and is also available for the combined computational circuit. The positive digital input/threshold vector is controlling the pull-up 1T1R array while the negative digital input/threshold vector is controlling the pull-down 1T1R array.

an output (in a true associative memory type of computing), as described with the computational engine concept in Chapter 2 (see Sections 2.1 and 2.7). The main differentiating factor among the different implementations is the input/output biasing scheme applied/expected from the logic network. Thus, in this chapter, I combine the design methodologies from the previous chapters and I propose a higher-complexity reconfigurable logic circuit that can be programmed to operate in either mode of data processing (i.e. current/voltage mode MAC operation and current/voltage mode TLG configuration) towards creating a combined RRAM-based programmable computing cell for IMC (also referred as combined RRAM-based logic circuit or simply combined logic circuit in the following sections).

As has been established in previous chapters, the 1T1R-based computing arrays are the main primitive logic structure for performing mixed-signal data processing and can be encountered in many different forms, as shown by Ielmini and Wong (2018); Xi et al.

(2020). For the case of the static RRAM-based logic circuits category, the logic function is mapped as memory-dependent weights and with the computation occurring either naturally (i.e. physical laws for signal multiplication and accumulation) or through the design of simple circuit structures (i.e. CMOS latching elements), as shown and discussed by Papandroulidakis et al. (2019b). Through the proposed configurations of the previously tested MAC circuit and TLG paradigms, I design a circuit that combines the different circuit structures, thus performing efficiently mapping between different signal domains. The aim of combining the different modes of operation into a unified programmable gate is worth investigating if the need of organising such systems into a sea-of-gates computer architecture is considered. Hence, these gates have the potential of being considered the basic logic unit of future field programmable gate arrays (FPGAs) for mixed-signal computing where the main computing elements, thus the RRAM devices, can process data under digital and analogue mode of operation.More specifically, the combined logic circuit is implemented as a reconfigurable RRAM-based memory-dependent resistive network employing both pull-up and pull-down 1T1R-based memory arrays with additional neural emulation circuitry for the readout operation. Similarly to the TLG design, a CMOS-based sensor part is used as a ”compare and remember” unit that classifies the input and saves the result (i.e. as a binary state in the CMOS-based latch). An adversarial RRAM-based voltage divider circuit is used to provide a biasing/thresholding values against which the intermediate node voltage of the RRAM logic topology compares against. This hardware feature is in agreement with the design and operation of the proposed data processor designs for IMC architectures.

The combined RRAM-based circuit design for MAC and TL (MAC-TL) operations was centred around the need to develop a low-complexity reconfigurable computing structure capable of being used as versatile building blocks of larger IMC systems. Given that MAC and TL operations can be considered the basis of many neuro-inspired algorithms the design of a programmable circuit for performing both of these fundamental operations is of high importance. In this section, I am showcasing simulation results of the proposed combined gate. The design and testing of this circuit can provide important insights towards designing a RRAM-based reconfigurable system capable of being easily integrated into the data-path of conventional electronics.

The combined gate presented here is based on the RRAM-based MAC circuit and TLG designs showcased in Chapters 3 and 4 and is shown in Fig. 5.1. More specifically, the combined circuit showcased here incorporates two MAC circuits (shown in Fig. 3.2 of Chapter 3, Section 3.3) as the differential pair of input and threshold arrays of a RRAM-based TLG (shown as the MCMTLG in Fig. 4.4 and Fig. 4.6 of Chapter 4, Section 4.3). Essentially, I am integrating the two neuro-inspired designs as part of a single reconfigurable primitive circuit capable of enabling different data processing modes on-the-fly for IMC systems. The configuring process of the computing modes, thus operating as analogue MAC operations or digital TL, can be achieved with a small

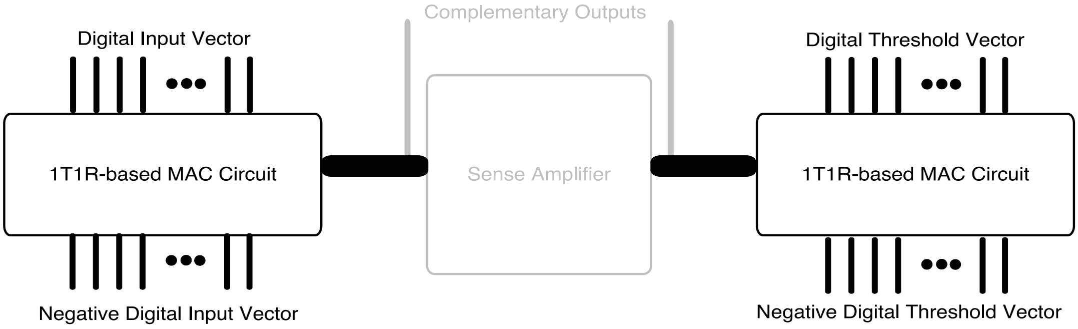

FIGURE 5.2: Schematic of a combined near-memory primitive circuit configured to perform MAC operations. The sense amplifier that can perform the TL operation (current-mode comparison between the two arrays) is deactivated. Thus, each 1T1R array is capable of performing a separate MAC computation in parallel. The negative digital vectors are active and a simultaneous mapping of input and/or threshold biasing (per array) can be performed. Hence, negative input-weight and negative threshold-weight combinations are available for the MAC operations.

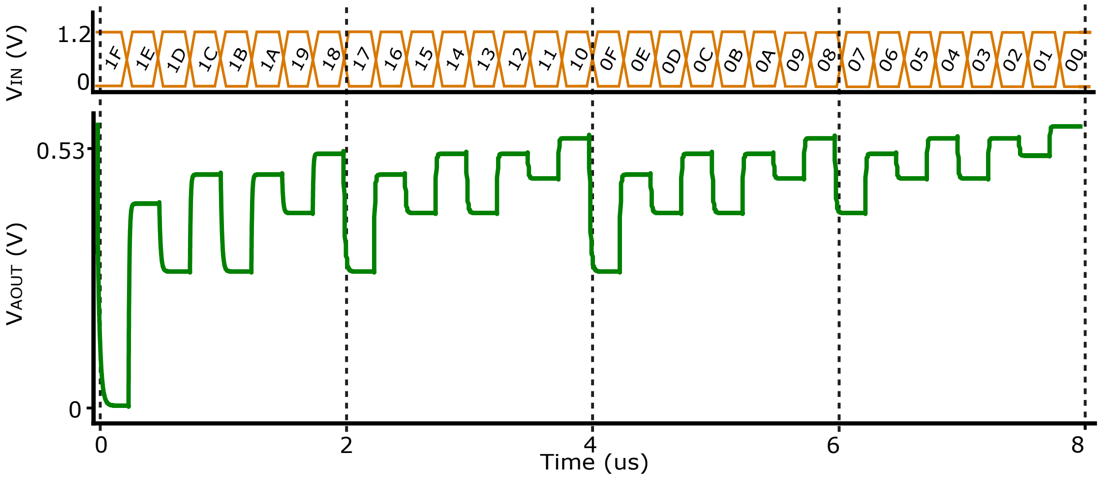

FIGURE 5.3: Simulated waveform when the combined primitive circuit has been configured to perform analogue MAC operations. The V_{IN} is shown in a hexadecimal format as MSB-$(000V_{IN5})$ $(V_{IN4}$ V_{IN3} V_{IN2} $V_{IN1})$-LSB. A pull-down resistive weight of 900kΩ is also used for the combinatorial MAC mode. The integrating capacitor is disconnected from the circuit.

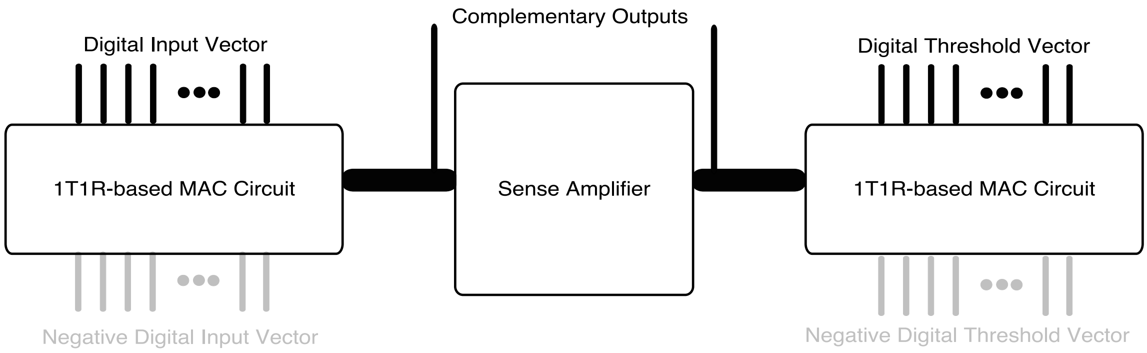

FIGURE 5.4: Schematic of a combined near-memory primitive circuit configured to perform TL operations. The negative digital input and threshold vectors and thus the pull-down 1T1R arrays are deactivated. The two 1T1R arrays form a differential pair for mapping input and threshold analogue weights. The two adversarial currents generated by the reading operation of the two 1T1R arrays are compared through the sense amplifier circuit. Due to the design of the sense amplifier as a SRAM-based latching element complementary outputs can be obtained as the result of the comparison process.

pass-gate network controlling the pull-down 1T1R arrays (alongside the capacitor used for the MAC circuit's analogue integrator mode) and the connection to the sense amplifier. Thus, a small area overhead is required for the controlled switching between the different modes of operation of two pass-gates per memory array of the differential network.

To control the reconfiguration operation of the combined RRAM-based circuit, appropriate signals need to be generated. For the case of the MAC circuit configuration (shown in Fig. 5.2), the pull-down 1T1R network need to be enabled with the additional capacitor for the case of MAC integrator mode. Simultaneously, the output of the 1T1R computing arrays are disconnected from the sense amplifier. The two 1T1R arrays shown in the example configuration of Fig. 5.2 can operate independently when in MAC circuit mode. In Fig. 5.3, simulated results from an example MAC circuit configuration are exhibited. The resistive memory configuration, for the example case study showcase in Fig. 5.2, is set to: {212kΩ, 408kΩ, 403kΩ, 415kΩ, 417kΩ}. The RRAM model instances used are based on fitting parameters extracted by real RRAM devices

For the case of the RRAM-based current mode TLG circuit configuration, the pull-down 1T1R-based networks and accumulating capacitors need to be disconnected from the main circuit, as shown in Fig. 5.4. More specifically, the outputs of the current mode 1T1R-based memory arrays are connected to the sense amplifiers (otherwise used for memory read operations) which performance the comparison between the analogue values of the differential arrays. As discussed in Chapter 4, Section 4.3, for the differential TL operation, two 1T1R arrays are required to create a differential pair (one for

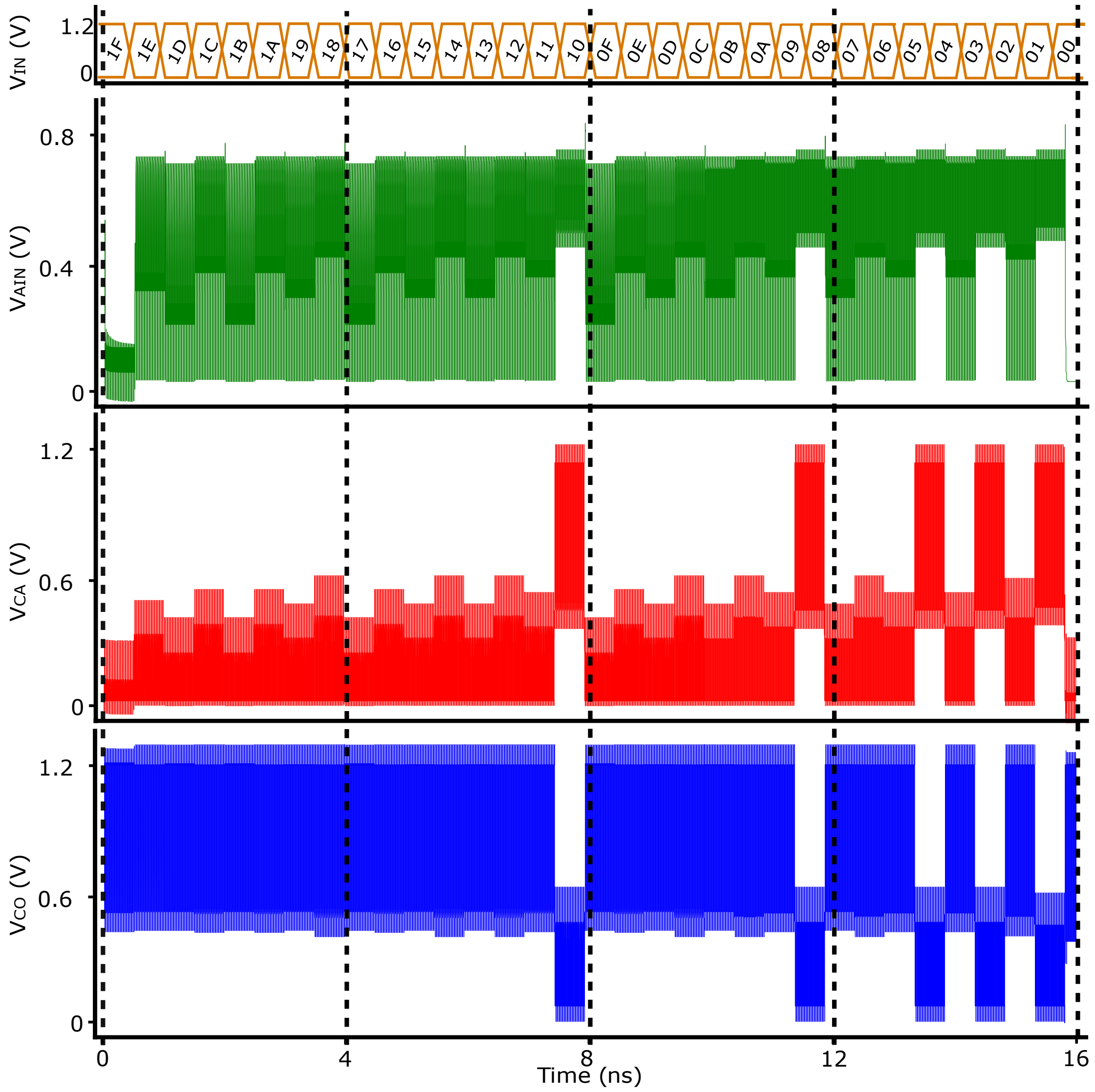

FIGURE 5.5: Simulated waveform when the combined primitive circuit has been configured to perform TL operations. The V_{IN} is shown in a hexadecimal format as *MSB*-$(000V_{IN5})$ $(V_{IN4}$ V_{IN3} V_{IN2} $V_{IN1})$-*LSB*. The TLG weights are configured as {212kΩ, 408kΩ, 403kΩ, 415kΩ, 417kΩ,; 97kΩ}. The V_{AIN} is the voltage of the analogue output of the input-dependent 1T1R-based array. The V_{CA} and V_{CO} are the complementary outputs of the sense amplifier.

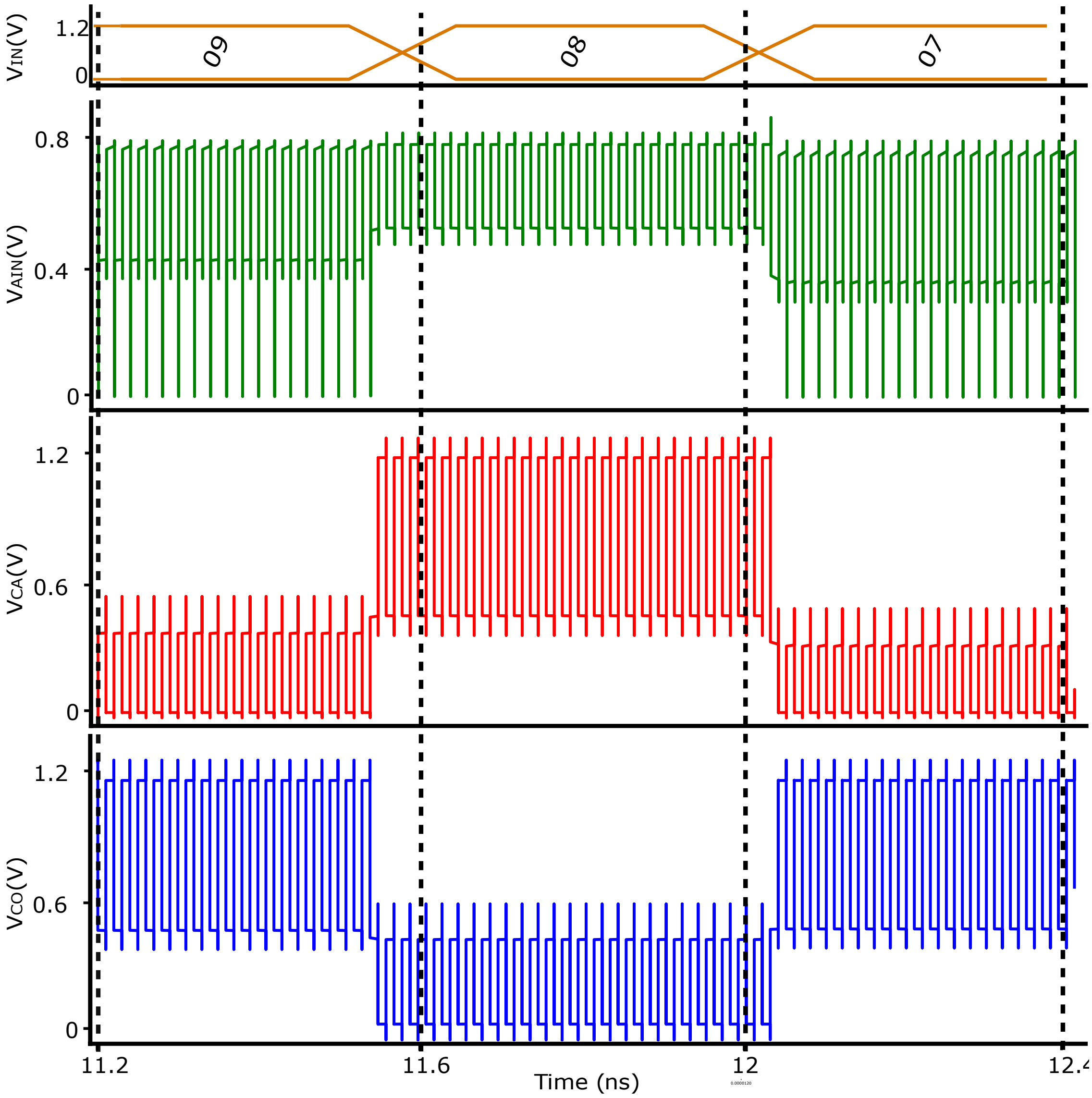

FIGURE 5.6: A close-up look on the simulated waveform presented on Fig. 5.5. The V_{IN} is shown in a hexadecimal format as MSB-(000V_{IN5}) (V_{IN4} V_{IN3} V_{IN2} V_{IN1})-LSB. The V_{AIN} is the voltage of the analogue output of the input-dependent 1T1R-based array. The V_{CA} and V_{CO} are the complementary outputs of the sense amplifier.

the input network and one for the bias network). In Fig. 5.5 and 5.6, simulated results from a TL operation mode example are shown. The RRAM resistive states are the same as for the example case study of the MAC computing configuration with the additional use of a single threshold RRAM device (through the differential threshold 1T1R-based array): {212kΩ, 408kΩ, 403kΩ, 415kΩ, 417kΩ,; 97kΩ}.

Since the combined circuit is based on the aforementioned neuro-inspired circuits and is designed as an extension of the MAC and TLG designs, it is itself easily integrated into IMC-based circuits and systems. The pull-up and pull-down arrays can be assigned to either the same 1T1R computing array or different arrays (one for positive current contributions and one for negative current contributions with the two output nodes connected), while the sense amplifier is a necessary part of the IMC since it is employed for normal memory read operations. Thus, following the rules of the IMC system design (as discussed in Section 2, Section 2.3), the reconfigurable RRAM-based combined logic circuit can be used as a primitive logic circuit for MAC-TL operations inside a memory-centric architecture without the need for extensive reconfiguration hardware. Scaling capability of neuro-inspired circuits and easy organisation into larger reconfigurable sea-of-gates structures is an important design parameter. Most neuro-inspired fundamental computing operation need to be capable in arranging their operations in a massively parallel manner towards providing acceleration for applications, such as ANNs models, for example as shown by Xi et al. (2020). Based on the findings and discussion of this section, the design of the combined RRAM-based logic circuit proposed here can accommodate sea-of-gate architectures.

5.2 RRAM-based Wake-Up-Circuit Design

The evolution of Internet-of-Things (IoTs) and the rise of edge computing necessitates the introduction of new systems that are capable of operating under different performance modes. This is especially true for the cases where a continuous search for a specific input pattern needs to be in place for detection before any meaningful data processing can be done. Thus, the design of low-performance and low-power circuit designs aimed for sensing operation with regards to specific stimulus in the environment, is of high interest, as discussed by Meyer et al. (2019). When an appropriate input pattern is detected then a high-performance computing system is activated inside the edge processor to performed a detailed data processing of the input pattern, as shown by Meyer et al. (2019).

Different modes of operation are required when the power consumption are of the essence, such as in the case of edge computing. Hence, the close integration of low-power event triggered sensing circuits that can ”understand”/classify patterns from their environment and control the high-performance and, usually, high-power part of the edge computing architecture is a necessity. Duty-cycling is an important technique widely used under the edge computing paradigm. In most case of duty-cycling, a low-power system is employed to perform a low-precision pre-processing of a input signal, as shown by Gupta et al. (2019). If specific requirements are matched when the pre-processing is complete, then the received signal includes all the necessary identifiers for an event triggering of a high-performance and high-power system that thoroughly process the input signal in a high resolution manner. The circuit used as the pre-processor usually is referred to as the Wake-Up Circuit (WUC) since the main computing operation performed is the generation of a wake-up enabling signal to the high-performance high-power edge computing if it detect an pre-defined pattern in the input signal, as highlighted by Rovere et al. (2018a,b). WUC are usually placed between a main high-performance computing system and a sensing system that converts and communicates information from the environment to the edge computer. Usually the sensor-WUC systems are capable of ”observing”/monitoring the environment around the edge computer and detect specific patterns that can be then processed by the high-performance part of the edge computer. In some cases, the edge computer is responsible for only a partial processing before transmitted the partially processed data to another computer for even more high resolution processing.

In this section, I am showcasing an expansion of the neuro-inspired design methodology exhibited on see Chapter 3 and 4 (see Sections 3.4 and 4.5) as well as of the combined RRAM-based logic gate shown in Section 5.1. More specifically, an implementation of a RRAM-based WUC, an increasingly interesting circuit for computing at the edge as discussed by Meyer et al. (2019); Rovere et al. (2018a,b), is shown. The RRAM-based WUC base its operating capabilities on the computing RRAM-based 1T1M array configured in an analogue input and output (fully analogue MAC circuit) mode of operation. The proposed computing concepts combines the design of Current-Mode RRAM-based TLG and the RRAM-based MAC circuit and is another step towards highlighting the flexibility of RRAM-based arrays, as shown by Serb et al. (2017, 2018a,b); Papandroulidakis et al. (2018, 2019b,a). Essentially, I am adapting an extended version of the combined MAC-TLG circuit to perform a simple voltage signal pattern recognition. For the analogue MAC circuit gate and operation as an analogue associative memory. The RRAM-based primitive WUC is based on the previous set of primitive circuits and, through this case study, I am highlighting the potential in developing complex computational networks using low-complexity logic blocks based on the RRAM technology. The specific WUC design showcased here can be easily realised inside a memory architecture design and thus can be implemented as a smaller configured circuit of a larger reconfigurable IMC system. The system in simulated using commercially available 65nm technology node for the transistor devices and the RRAM model presented by Messaris et al. (2018) with model instances extracted from real RRAM devices. For the simulations, the Cadence’s Virtuoso Spectre simulation environment is used.

As discussed in previous chapters (discussed in Chapter 2, Section 2.3), RRAM-based IMC systems are capable of massively parallel neuro-inspired operation, such as TL computing, and can be efficiently operated in cross-domain data processing solutions due to the use of RRAM devices that are naturally analogue devices. Through cross-domain data processing circuits, it is possible to interpret specific formats of data and convert them into another format. If this conversion can be performed during the data processing then, for some cases, the need for an additional data processing step (thus additional Analogue to Digital or Digital to Analogue data conversion) can be eliminated, thus designing systems with lower power dissipation and lower chip area. Due to the computational flexibility of the 1T1R-based computing array, the sensor generating the stimulus could be place near the IMC system. Depending on the application, the 1T1R-based circuit can be used without converting the MOSFET-controlling vector to the digital information domain but instead using analogue signals to control the IMC arrays (similarly to what shown in Fig. 4.7 hardware measurement in Chapter 4). In that case, the circuit operates as an analogue(-input) RRAM-based MAC circuit, similarly to what is shown in Chapter 3 (the output remains in analogue format as in the initial circuit). Similarly to the previous digital-input MAC circuits showcased in Chapter 3, the analogue MAC circuit can be effectively used to implement a primitive analogue associative memory. The memory contents of RRAM devices are fully analogue while in this specific use case the input voltage set controlling the transistors of the array are also analogue. The result of the MAC computing is provided in analogue form, similarly to the digital-input MAC circuit showcased in Chapter 3.

Using what is essentially an extended version of the combined RRAM-based gate, as shown in Fig. 5.1, the output of the input 1T1R-based array is compared against two voltage threshold levels simultaneously. Comparison against additional voltage levels could be introduced but by limiting the comparison against only two other arrays, the circuit can be easily mapped into a crossbar-based system without any additional reconfiguration network. The proof-of-concept application in this sections is centred around the identification of a simple pattern of a voltage signal evolution through time. The voltage could be generated through any kind of sensor that provides stimulus for the computing 1T1M array. To identify a time-constrained pattern appropriate counting circuits need to be introduced additionally to the circuitry used for checking the voltage level of the input signal. A general block diagram description of the RRAM-based WUC discussed and designed in this section is shown in Fig. 5.7. In Fig. 5.7, I make the concept assumption that an optical sensor is detecting a specifi event from the environment and performs an appropriate conversion of the sensory data to a set of input signals. This, of course, can be generalised to any similar sensor system that detects specific events from its environment and translates them into appropriate electrical signals to be proposed by the RRAM-based WUC. The input vector is then introduced to the RRAM-based WUC, which is part of a generic IMC system, and more specifically to the gates of the accompanying MOSFET devices of the 1T1R arrays. As discussed, the input signals can either be in analogue or digital form since the combined hybrid RRAM-MOSFET MAC-TL circuit can operate under both input stimulus. The main operation of comparing the input stimuli with the appropriate voltage levels is performed in-memory through the use of appropriate weights in the 1T1R components. The comparison is performed by the sense amplifier at the end of the RRAM-based memory banks.

The analogue MAC circuit receives a set of signals as input stimuli (could be either digital or analogue) and generates an analogue voltage value mapped from the array

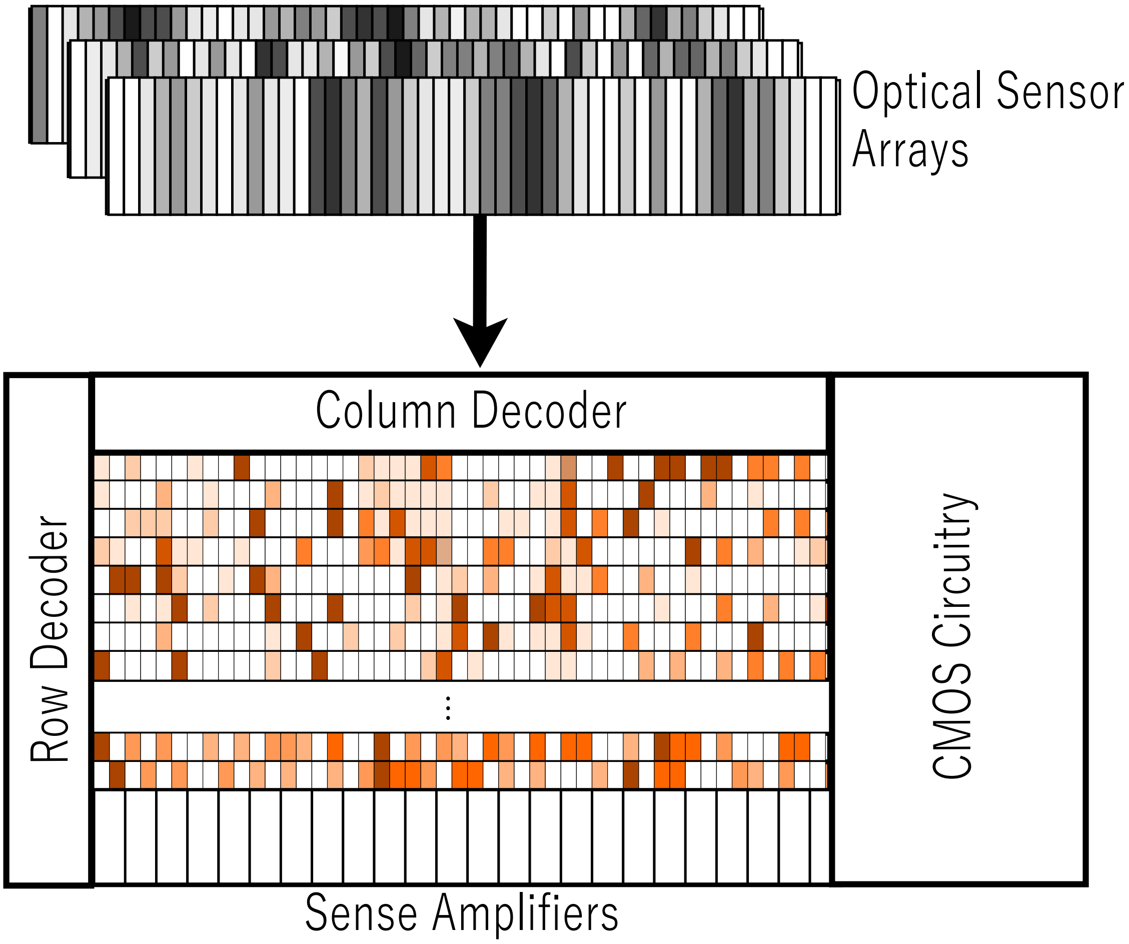

FIGURE 5.7: In this figure, a general block diagram description of the main components of the RRAM-based Wake-Up System is showcased. The optical sensor is an example considered for this concept and other types of sensing systems could be employed to convert an environmental (with regards to the computer) observation to an appropriate set of electrical signals. The output of the sensor send the input stimulus to the IMC system with the different coloured cells representing different RRAM-based weights. I assume a DRAM-like crossbar-based structure for the IMC system. the main memory and computing unit is found in the form of the 1T1R array. The main WUC operation are performed inside the memory array (through the 1T1R arrays and the sense amplifiers) with some additional peripheral circuitry necessary for the whole pattern detection system to be complete, thus making this system a Near-Memory Computing (NMC) architecture. If a pattern is matched then a wake-up signal (pattern match flag signal) is generated to enable a high-performance system towards processing the environmental stimulus further. The actual high-performance computer is out of scope for the WUC design of this section. Similar system configurations that incorporate a event-detecting systems' duty-cycling is considered important especially for applications at the edge where the continuous operation of high-performance systems can be hard to implement.

based on the input vector and the weighting information of the RRAM contents. The comparison operation is essentially based on the RRAM-based TLG circuit, showcased in Chapter 4 and shown by Papandroulidakis et al. (2018), and on the combined gate showcased in Section 5.1. The extended version of the combined gates used for WUC is shown in Fig. 5.8 where essentially two RRAM-based combined MAC-TL circuits, placed inside an IMC architecture, are simultaneously operated to perform at the same time two comparisons. The RRAM-MOSFET TLG designs can be easily integrated into a crossbar array topology of hybrid 1T1R-based computing-memory word-lines and peripheral sense amplifiers, thus a design well-fitted for IMC systems. Some additional peripheral circuitry is necessary for the specific application presented in this work. Circuit such as timing circuits (counters), shift registers and some additional control logic necessary for generating the appropriate control signals for the peripheral and main circuitry are build using CMOS technology. The RRAM arrays are used as a form of miniaturised associative memory with capabilities in programmable logic operations (since the comparisons are based on the memory contents of the arrays, a concept discussed thoroughly in my work with the RRAM-based TLG design showcased in Chapter 4 (also referred to as MCMTLG), as showcased by Papandroulidakis et al. (2018)).

The reference voltage levels for each phase of the pattern matching operation are defined by appropriately programmed 1T1R components of the threshold arrays (arrays to the left and right of the analogue MAC circuit as shown by the leftmost and rightmost 1T1R array in Fig. 5.8, respectively). The outputs of the sense amplifiers, that perform the comparison operation, are connected into simple Boolean digital logic. More specifically, the circuit is a Boolean AND function with inputs the CA of one sense amplifier and the CO of the second sense amplifier. Through that AND gate it is possible to check if the output of the analogue MAC circuit is between the two voltage thresholds. The generated control signal (output of the AND gate) is connected to CMOS-based counters. If a proper number of sequential comparison TL-based evaluation is measured, through the counter, then the matching to a specific pattern phase (controlled by the set of threshold values programmed in the RRAM devices of threshold 1T1R arrays) is confirmed. If a specific phase is matched then the circuit starts the evaluation of the next phase of the pattern matching operation by enabling the next set of threshold RRAM weights. The counters are initialised for them to count the next potential matching count operation. In the case that during the counting the evaluated signal is found outside the reference voltage band (thus there is a mismatch of the ongoing pattern check operation), then a simple Boolean logic circuit (i.e. a XOR/XNOR gate to compare the digital outputs of the sense amplifiers) generates a terminate matching operation signal for the RRAM-based WUC to start again from the initial phase of the pattern matching operation. This is based on the assumption that a specific pattern need to be detected without any breaks in between the different phases of the pattern.

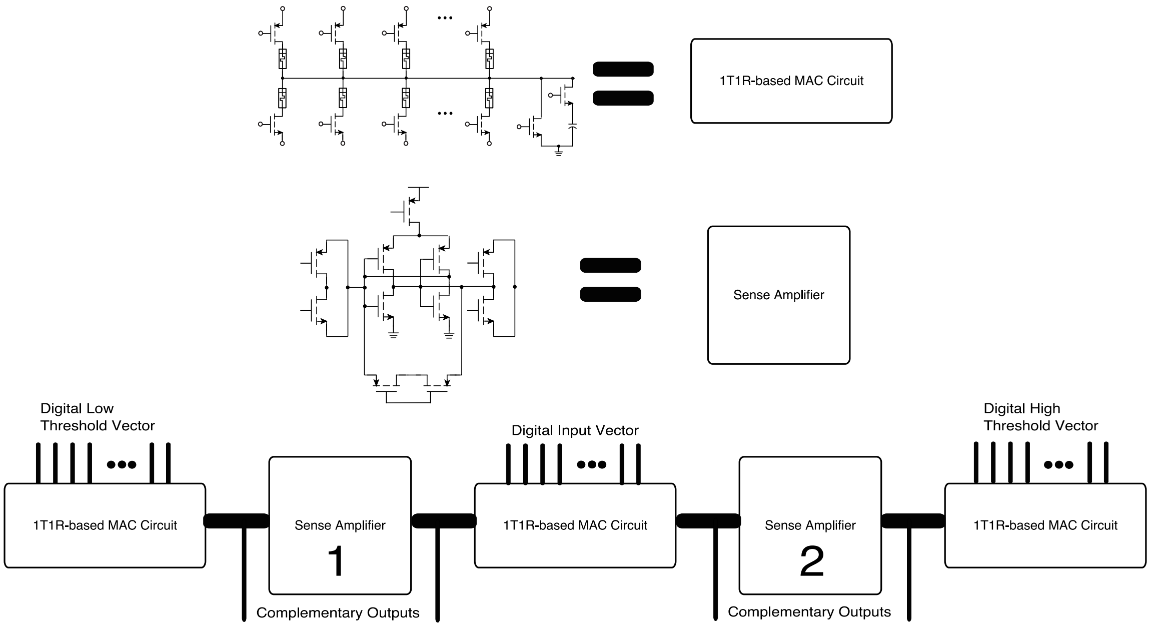

FIGURE 5.8: Schematic of the circuit-level system representation is showcased. The main computing circuits are the memory array itself used as a miniaturised associative memory to map information of a vector of analogue voltages into a single analogue value. In order to complete the computation additional circuitry is necessary thus categorising the specific RRAM-based WUC design under the near-memory computing (NMC) techniques. It is worth noting that the main computing circuits where the data mapping is performed such as the 1T1R arrays and the sense amplifiers are traditionally fundamental parts of a typical memory architecture, such as DRAM. Hence, essentially the overhead is of the shift registers, counters and additional Boolean logic gates used to enable the tracking of the signal and the generation of appropriate enabling/disabling signals in case of a pattern match/mismatch. In the block diagram of the system, the digital input vector is controlling the input 1T1R array while the digital low and high threshold vector is controlling the low and high thresholds, respectively. For this specific case study the negative vectors (controlling the pull-down network of the MAC circuit) are not active. The first and second sense amplifier performs the comparison of the input against the low and high threshold, respectively.

5.3 WUC Simulation and Results

The RRAM-based WUC is tested through simulation on Cadence's Virtuoso Spectre simulation environment. The simulated system is based on RRAM-based TLG design presented in Chapter 4, Section 4.3 and presented by Papandroulidakis et al. (2018), as well as the combined gate shown in Section 5.1. Two CMOS-based sense amplifiers are connected simultaneously to this computing array with two other 1T1M arrays that are being used to store two sets for the reference threshold voltage level, as shown in Fig. 5.8. The reference voltages are used to define the analogue voltage band necessary to the pattern matching operation. A set of digital voltages is controlling the selection lines of the RRAM-based MAC circuit pull-up network. The output of the sense amplifiers, after a computing step of Boolean-based digital processing (thus with CMOS

Boolean logic gates), is used to control the CMOS counter operation which will time the matching event per phase of the whole simulation. In the case that the analogue output of the analogue MAC circuit moves outside of the reference voltage band, this event need to be detected, and the process of the pattern matching terminated and re-initialised from the first phase. The simulations that showcase this proof-of-concept RRAM-based WUC are based on a circuit-level description of the aforementioned system parts.

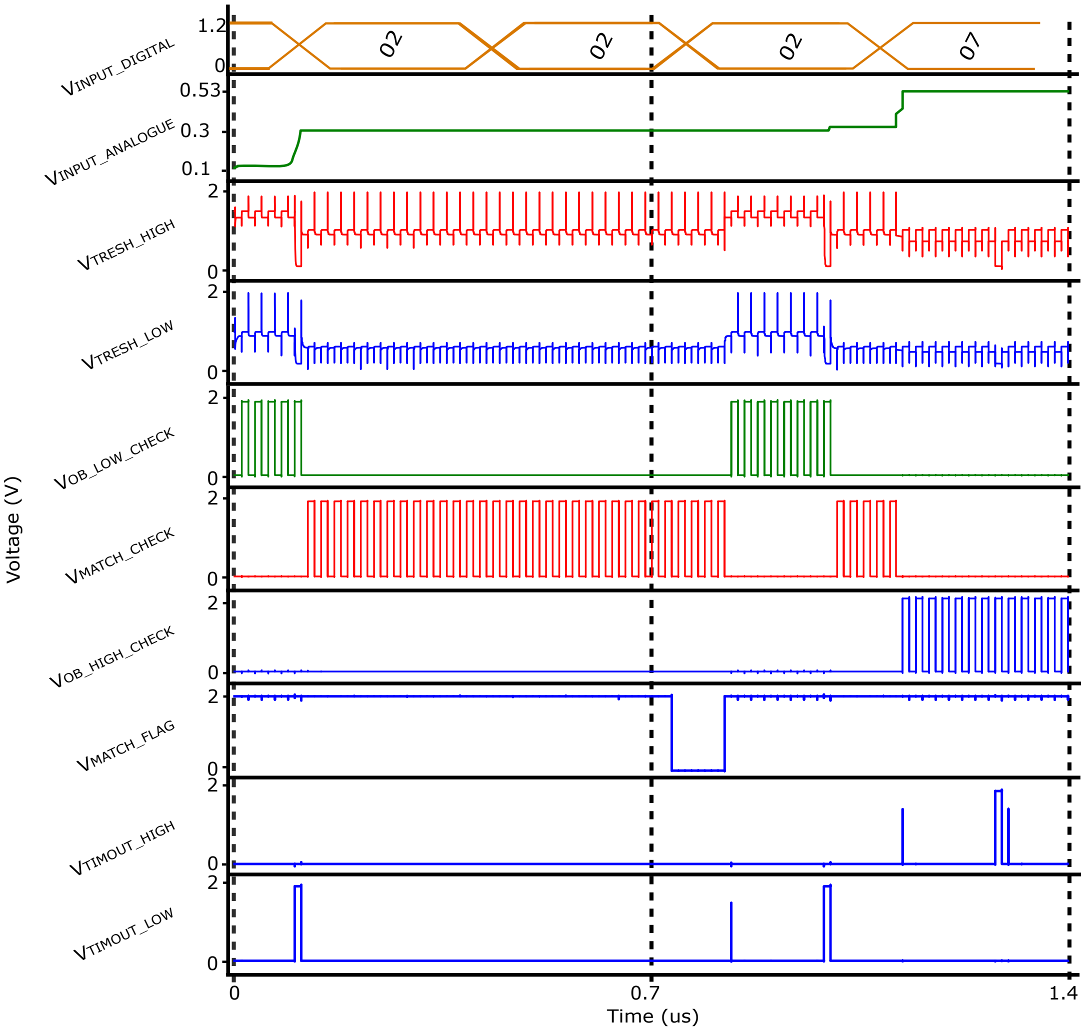

FIGURE 5.9: In this figure, a proof-of-concept RRAM-based WUC example of a pattern mismatch is exhibited. After confirming a match for the first phase of this two-phase pattern (V_{MATCH_FLAG} goes LOW), the signal generated by the input 1T1R array is outbound for the second phase of the pattern. This result in the firing of the signal $V_{OB_LOW_CHECK}$ and the generation of a total pattern match reset with the V_{TIMOUT_LOW} signal. It is possible to configure the timing constraints by increasing the size of the CMOS DFF-based counters.

The simulations of the proof-of-concept RRAM-based WUC are exhibited in Fig. 5.9 and Fig. 5.10. The simulations are employing a commercially available 65nm technology node for the MOSFET components and the RRAM device model showcased by

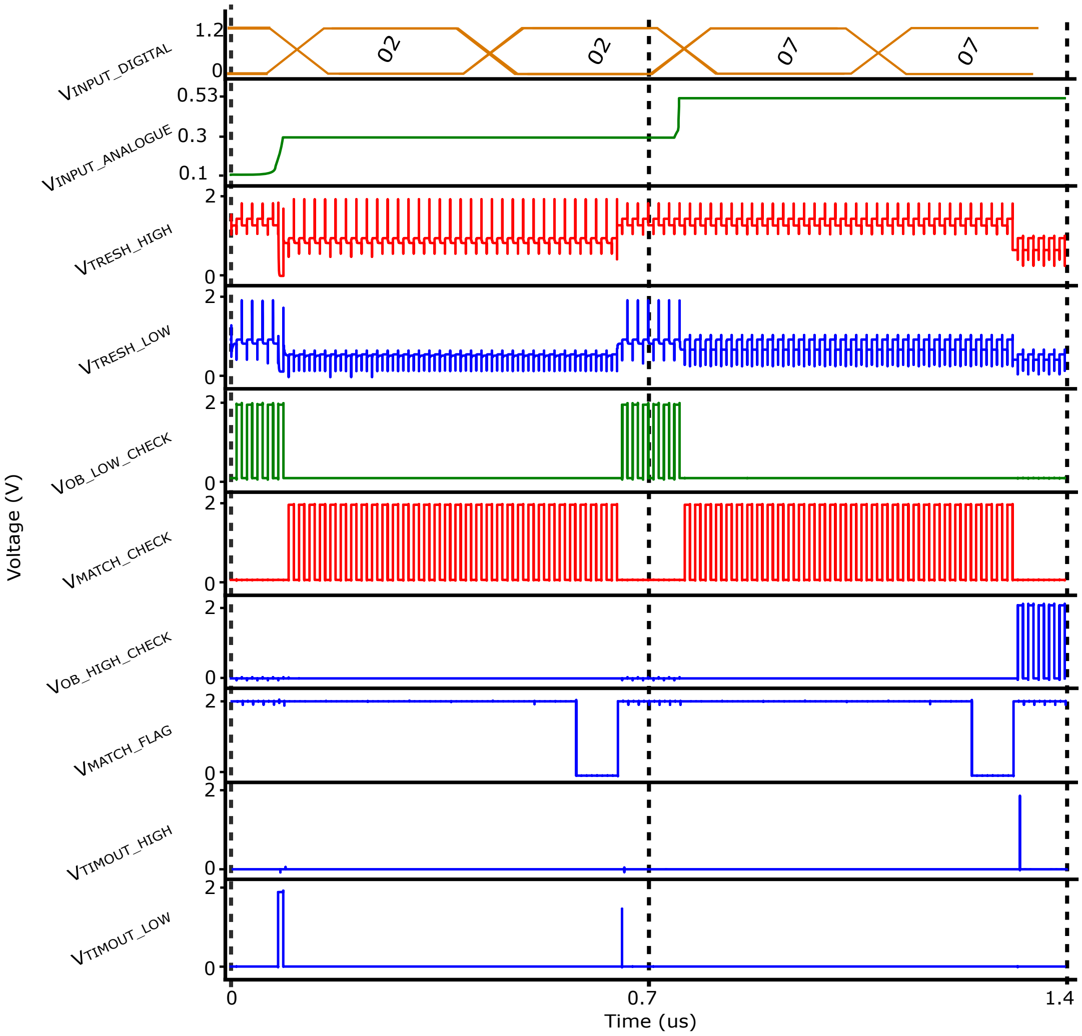

FIGURE 5.10: In this figure another example of the proof-of-concept WUC operation is showcased. In this simulation results a two-phase pattern match is confirmed.The $V_{MATCH_{F}LAG}$ signal is confirming both phases of the example two-phase pattern. Thus, with additional circuitry (i.e. a counter or shift register) the appropriate wake-up signal, enabling the high-performance and high-precision computer, can be generated.

Messaris et al. (2018). RRAM model instances have generated through a parameter fitting process (described by Messaris et al. (2018)) which is based on the parameter extraction from real RRAM devices. The measurement of the real RRAM devices was performed with the use of a probe-station and the measurements board ArC One (ArC Instruments, UK), similarly to the process described in Section 4.2. In Fig. 5.9 an example operation where a two-phase pattern mismatch is exhibited while in Fig. 5.10 an example operation where a two-phase pattern match is showcased. The $V_{MATCH_{C}HECK}$ signal is used to control the counting of phase matching per pattern phases. If the matching is occurring for a satisfactory amount of time (as defined arbitrarily by the pattern matching requirements), thus if enough $V_{MATCH_{C}HECK}$ pulses have occurred without any outbound flag reset, then a flag signal $V_{MATCH_{F}LAG}$ is generated. The timing is defined by using the appropriate size of CMOS-based counters, thus it is useful to integrate CMOS counter that can be used for timing the maximum possible phase duration and then through a multiplexing network the appropriate outputs of the counter will control the $V_{MATCH_{F}LAG}$ signal. The $V_{MATCH_{F}LAG}$ is used as a clock signal for a shift register which, in turn, is used to control the accompanying MOSFET devices of the threshold 1T1R memory arrays (the threshold voltage levels are shown as the $V_{THRESH_{L}OW}$ and $V_{THRESH_{H}IGH}$ signal in Fig. 5.9 and Fig. 5.10). Two additional outbound flag signals $V_{OB_{L}OW_{C}HECK}$ and $V_{OB_{H}IGH_{C}HECK}$ are firing when the signal is lower than the pattern phase voltage band and when the signal is higher that the reference voltage band, respectively. I use additional CMOS D-Flip-Flop (DFF) -based counters to control the time of outbound signals and if a pre-defined time passed with the signal being outbound then a total reset of the pattern matching operations is performed. The pattern match reset is controlled by the time-out flag signals $V_{TIMOUT_{L}OW}$ or $V_{TIMOUT_{H}IGH}$. Similarly to the CMOS counter for phase match, the area overhead of the outbound check counters depend on the maximum available timing for measuring the pattern mismatch of the signal. Additional simple CMOS-based Boolean AND/OR logic gates are necessary to generate the match/mismatch signals that control the counter circuits and can send the flag signal to the high performance computing system in the case of the full pattern match occurrence.

The timing constraints can be implemented in the form of CMOS-based counters. The outputs of the sensors are used to control the counting operation. More specifically, when the input voltage signal is outside the threshold voltage band then no counting is occurring. In the case that the input voltage is within the limits of the threshold voltage band constraint then a counting is occurring and when a specific amount of time has passed then the next phase of the pattern matching operation is selected. For the next phase of the pattern matching to occur the next set of threshold weights of the threshold 1T1R arrays are being enabled. Similarly to the first phase, the voltage output of the gate needs to be confined to the new reference voltage band for a specific time. If all the programmed phases are matched without any mismatch in between then a final pattern has been found and the high performance systems waiting for the wake-up signal can be enabled. Other than the DFF-based counters that define the timing of matching/mismatching phases the main operation performed by the CMOS peripheral circuitry is detection of whether the signal is within (or not) the phase voltage window. The detection is performed through a XNOR Boolean logic gate. The input are connected to two sense amplifiers used in the WUC scheme while two other 1T1R arrays mapping the threshold values are connected as the adversarial connections for each sense amplifier. More specifically, I am using the canonical output of the first sense amplifier (checking the low threshold) and the complementary output of the second sense amplifier (checking the high threshold) as inputs in a XNOR gate to check when only one of the inputs (the canonical output of the low threshold sensor) is HIGH while the other LOW. In the cases where both inputs are LOW or HIGH then a phase mismatch occurs. If the phase mismatch exceeds a specific timing constraint (depending on the pattern detection requirements) then a full pattern mismatch occurs and the RRAM-based WUC is initialised to recheck from the first phase of the pattern.

It is useful to note that the main advantage of implementing a system like the RRAM-based WUC is our capability to integrate primitive pattern matching circuits into a programmable computing RRAM-based IMC system. It is also worth highlighting that the hardware cost in system's area and complexity can be considered low. The 1T1R arrays and the sense amplifiers are necessary computing and/or memory components for implementing a simple memory architecture. The additional circuitry required for this proof-of-concept system are the CMOS DFF-based counter, the XNOR gates and some additional Boolean AND/OR gates for the generation of appropriate signals controlling the operation of WUC and enabling or disabling the high-performance systems that awaits the full pattern match flag signal. The AND/OR gate can easily be mapped into RRAM-based TLG circuits in a subsequent IMC bank. The non-linearly separable XNOR could also be mapped in multiple arrays of RRAM-based TLGs. The capability to map even supplemental logic functions into the IMC system can help reducing the area and power cost of the CMOS peripheral circuitry.

Since the RRAM-based WUC implementation is build around the IMC-centric circuits of 1T1R-based MAC computing arrays and the TLG design, it is possible to consider this RRAM-based WUC as a Near-Memory Computing (NMC) approach, since there are CMOS-based circuits placed in the periphery of the memory array that are necessary for the WUC operation (i.e. such as CMOS counters, shift registers, simple Boolean logic gates etc.). As discussed previously, the RRAM-based MAC circuit and TLG can be easily integrated into a memory system architecture, similarly to a DRAM memory bank but instead of a capacitor a RRAM device is used as the memory element. In such designs, the sense amplifiers are connected to two memory arrays with the read memory operation being used to charge/discharge the bit-line enough for the sense amplifier to perform a memory read per bit-cell of a memory word. This is a common design for DRAM due to the storage of complementary information in the two adjacent arrays towards mitigating the sensing classification delay occurred due to the usually large bit-lines. A similar approach in designing a RRAM-based IMC system centred around the MAC circuit and the TLG can be followed especially since, in most cases, the IMC system is required to be able to operate also as simple memory and not only as an optimised accelerator for computing specific operations, thus combining multiple modes of operation depending on each algorithm's requirements. Thus, the IMC architecture that is considered as the topology to implement parallel RRAM-based WUC has already the connectivity in place to perform what is essentially a dual MAC-TL operation for comparing a single input signal against two voltage level threshold. This can be achieved by activating three adjacent 1T1R-based memory arrays and the two sense amplifiers connected to these memory arrays.

5.4 Conclusions

In this chapter, I showcased the design of a combined RRAM-based MAC-TLG circuit for IMC systems. I employed the RRAM-based circuits exhibited in Chapter 3 and 4 to design circuits of an increased level of reconfiguration capabilities fitted for IMC systems. Firstly, I am showcasing simulation results of a combined RRAM-based gate based on the 1T1R-based MAC circuit and the RRAM-based current mode TLG. The combined circuit is implemented by introducing a programmable pull-down RRAM-based network for the use as a MAC circuit. Additional circuitry such as an integrating capacitor per 1T1R computing array can be also added to the design towards enabling the different modes of operation or the array (i.e. integrating mode, combinatorial mode, etc.). Secondly, I am showcasing a proof-of-concept simulation of a RRAM-based Wake-Up Circuit based on the combined circuit. The design is allowing the implementation of a simple RRAM-based WUC for identifying simple voltage signal patterns inside an IMC accelerator architecture.

The circuit designs showcased in this chapter exhibit the computational flexibility of the previous primitive circuits and how a larger network of such primitives logic blocks can be used to effectively map in hardware more complex logic functions. The main computation is performed inside the memory system while a small number of low-complexity CMOS-based components are employed as part of a peripheral circuitry that helps with the generation of appropriate control signals to enable the simple voltage level pattern matching operation of the proof-of-concept WUC system. Thus, the design seems feasible to be integrated in IMC without a large hardware area overhead.

One of the main design decisions in IMC systems is the design of the main primitive component for in-memory acceleration operations as well as the topological connectivity of those circuits. Through this proof-of-concept RRAM-based WUC, I am highlighting how primitive RRAM-based circuits can be used into a programmable IMC computing substrate towards performing complex operations. Additionally, the showcased results indicate that implementing WUC-based detection of simple signal pattern can be designed in RRAM-based IMC systems essentially without any modification of the computing-memory arrays that form the sea-of-gate design. This enables the design of programmable IMC with simple pattern classification capabilities inside the same accelerator architecture. This can have potentially the result of implementing a homogeneous IMC system where the duty-cycling is enabling different parts of the IMC-based accelerator.

Chapter 6

Conclusions and Discussion

In this section, a discussion and concluding remarks are presented with regards to the findings from this thesis. More specifically, I am discussing the results of this thesis with regards to the hardware realisation of RRAM-based circuits and systems capable of performing mixed signal computation in-situ. Additionally, I am discussing the potential organisation of these RRAM-based (also referred to as memristor-based or memristive) logic into digital and analogue reconfigurable sea-of-gates networks with some preliminary findings (as found through Cadence Virtuoso Spectre simulations) showcasing promising results. It is important for future implementations of similar circuits and systems that the competitive traits of RRAM-enhanced neuro-inspired circuits and systems are appropriately understood and the findings presented in this thesis can be used as an important stepping stone in the design and methods development required to further expand the investigation of the hybrid RRAM-MOSFET circuits.

In this work, I showcased, designed and practically implemented circuits and systems based on RRAM devices towards identifying the inner workings of primitive RRAM-based neuro-inspired circuits and designing computationally flexible building blocks for RRAM-enhanced In-Memory Computing (IMC) designs. The circuits were designed and studied in Spectre simulations (through Cadence's Virtuoso simulation environment and the RRAM model proposed by Messaris et al. (2018)) and practically realised in hardware, using real RRAM devices (based on the work by Stathopoulos et al. (2017)) and off-the-shelf MOSFET components, towards providing practical evidence regarding their actual behaviour as part of a future IMC system. Through the exhibited results, a clearer picture of the importance of RRAM devices can be showcased for hybrid RRAM-CMOS reconfigurable electronics and in silico classification engines.

More specifically, I showcase the design of low-complexity primitive RRAM-based circuits capable of performing fundamental computer arithmetic and logic operations, such as Multiply-Accumulate (MAC) (shown in Chapter 3) and Threshold Logic (TL) (shown in Chapter 4), analogue signal comparison, digital vector classification, etc. Through the implementations and testing (both in simulation and in hardware experiments) of primitive RRAM-based circuits a novel hardware IC solution methodology for IMC accelerator architecture in future post-von Neumann systems is showcased. The design and testing of the RRAM-based circuits has been approached through the scope of introducing minimal additional circuits other than the ones necessary for the operation of RRAM-based memory operation, thus requiring a small area of integration for easy configuration into a dense IMC architecture.

The operation of the primitive circuits were validated through practical realisations in hardware exhibiting strong evidence towards implementing RRAM-based classification systems in silico. The practical implementation through hardware is important since it provides further evidence on the actual operation of these circuits. The use of real experimental RRAM devices on a hardware prototype circuit, as presented and discussed throughout the thesis, showcased that primitive circuits for in silico classifiers inside IMC systems are feasible. Additional simulations using state-of-art RRAM device model is also providing additional evidence with regards to the expected performance of such circuits and systems.

Furthermore, I showcased higher-level circuits and systems where the hybrid RRAM-MOSFET primitive circuits can be employed as part of a larger IMC system. Towards validating the role of these low-complexity primitive circuits performing cornerstone arithmetic and logic operations, I am showcasing configuration of proof-of-concept circuits and systems, such as the Winner-Take-All network (WTA) based on the RRAM-based MAC circuit (shown in Chapter 3) and Wake-Up Circuit (WUC) based on the composite RRAM-based MAC-TLG circuit (shown in Chapter 5). Through these proof-of-concept systems, I showcase a methodology for integrating RRAM-based primitive circuits into higher level circuits for neuro-inspired computing inside the memory architecture.

Through my findings, I support the research area of integrating RRAM technologies into computer architectures. Although many RRAM-based circuits are aimed towards replacing specific circuits of conventional computer architectures, such as logic gates, memory arrays, etc., there is an even greater effort at investigating how the novel RRAM devices can be employed towards implementing novel RRAM-MOSFET post-von Neumann designs, such as the next generation of RRAM-based IMC systems. The proof-of-concept circuits and systems showcased in this thesis are based around the use of RRAM-based memory arrays that can be employed additionally as computing systems. More importantly, the circuit design is approached from the start as part of an IMC accelerator architecture that bases its main computation on RRAM devices. Although IMC-based computing concepts have been explored in the last decades, as discussed in Chapter 2 (Section 2.3 and 2.7), the findings shown in this thesis are adding more information towards maturing such computing paradigms. This is achieved by providing both proof-of-concept hardware experiments results in addition to extensive simulation testing and by employing state-of-art RRAM devices. Although the circuits designs tested (under both experimental setups and simulation) are exploring a similar computational concept they are showcasing important distinct case studies of the IMC-based primitive circuit category of RRAM-based logic. Thus, each experiment is covering a separate need for testing and exploring the capabilities of primitive RRAM-based logic gates. Different levels of integration inside an IMC system are investigated (both at memory array level and higher system level). The findings for each case can be used together towards providing a more accurate extrapolation of the potential performance of such circuits. For the case of the experimental measurement, the full functionality of the circuit under test, using the current state-of-art in hardware emerging components (RRAM devices) and of-the-shelf components, is important to establish the type of circuit behaviour that should be expected in future high-level IC implementations. At the same time, through simulations it is highly useful to employ the best of state-of-art emerging and conventional technologies available to test an indicative performance of the proposed designs. The combined experimental and simulated results can showcase the potential advantages of integrating RRAM technology in novel computer architectures which, with further maturity of the RRAM technology and even closer co-design of RRAM-MOS circuits, could be considered one of the most promising candidates for future accelerators. RRAM devices can provide a wide variety of beneficial traits in circuits and systems especially for the family of neuro-inspired IMC computer architectures and this is supported by the results shown throughout this thesis.

Many routes of exploring the computational capabilities of hybrid RRAM-CMOS systems can be followed as future work from the showcased findings of this thesis. More specifically, the implementation of hybrid RRAM-CMOS IC designs to test the integration capabilities and performance of the proposed systems in higher-level hardware experiments could be an interesting next step for this research area. Testing a system-level hardware implementation of the a RRAM-based IMC paradigm can be considered an important part of future work since it would be beneficial in paving the road for the maturity of the RRAM-MOS circuits and systems co-design. The hybrid circuit implementations that can be used to explore the design and operation requirements of larger IC systems is broad. It can include primitive computing gates, such as the ones showcased in this thesis (thus enabling the testing of a variety of complex operations), or larger and application-specific circuits. Although any circuit design that can be adapted for integration for IMC-based architectures can be used to test a higher-level system and/or architecture, the simpler circuit designs can be employed to substantially enhance the reconfiguration capabilities of a larger system. Additionally, a system build based on low-complexity primitive circuits, such as the ones showcased in this thesis, can more easily allocate the appropriate number of computational resources for each task maximising performance for minimum power dissipation, since the part of the accelerator not used can be powered down. Applications such as Deep Neural Networks (DNNs) that require a large number of MAC operations could be highly accelerate with the proposed hybrid MAC-TL circuit. Additional interesting case studies can be explored when integrating primitive RRAM-MOSFET circuit in IMC-like architectures such as the employment of different areas of the same memory system (thus different memory arrays) for different operations in terms of type of data processing and the further exploration of the performance of analogue LUTs.

Since the completion of this thesis, further findings have been showcased and new solutions have been suggested by researchers on the topic of hybrid RRAM-CMOS circuits and systems. Some of the most interesting results, that signify the effort to design and implement RRAM-based circuits, are those showcasing two distinct methods of computing with RRAM: the conventional and unconventional computing. The first paradigm is continuing to provide evidence of increasing maturity of IMC-like computing. Findings shown by Murali et al. (2021) support the further advancements on three-dimensional architecture of IMC-based computing cores for the design of mixed-signal accelerator design. Additionally, results highlighted by Yin et al. (2020) showcase new evidence on the maturity of hybrid RRAM-MOS IC designs for low-complexity binary ANNs. Furthermore, advancements of hybrid Non-Volatile SRAM (NVSRAM), based on a hybrid RRAM-MOS design, can be employed to accelerate the integration of different memory technologies as well as to create novel memory and computing systems, as shown by Bazzi et al. (2021). Another important paradigm that shows promising results is the incorporation of RRAM technology in unconventional methods of computing. This is especially interesting since it can provide evidence of otherwise unwanted parts of the RRAM behaviour, such as resistive state stochasticity and current-voltage non-linearity, being exploited to introduce computations in a truly unconventional manner. For example, RRAM-based systems for hardware cryptographic engines showcase high interest. RRAM devices, due to their intrinsic behaviour, have been already showcased as highly useful in designing powerful Physically Unclonable Functions (PUFs) and True Random Number Generators (TRNGs), as shown by Uddin et al. (2019); Uddin and Rose (2019); Cambou et al. (2020). The advancements on hardware cryptography designs and performance can potentially greatly benefit the security of computing systems with highly efficient PUF and TRNG systems employed another reconfiguration mode of IMC accelerator architecture. Furthermore, another great example of the computationally flexibility of such hybrid RRAM-CMOS system could be the employment of such designs in probabilistic and stochastic computing, as shown by Le Gallo et al. (2018); Krestinskaya and James (2018); Rahimi et al. (2015). The design and implementation in hardware of these systems seem, in many cases, feasible through the existing technologies and design methodologies. More importantly, the main circuit and systems architecture can remain the same while, by employing different programming and/or reading schemes as well as using different readout circuits and methods, we can employ all the aforementioned unconventional ways of computing with the same main computing RRAM-based arrays that we use for the conventional methods. Considering the hardware results shown in this thesis it is possible to extrapolate that a higher-level and higher-scope RRAM-based sea-of-gates system could exhibit interesting results by designing a hybrid RRAM-CMOS crossbar-based system to test novel RRAM-enhanced IMC with conventional and/or unconventional computing capabilities.

At the same time, some challenges are important to be considered towards designing and testing large scale systems that base their operation in the simpler circuits explored in this thesis. Although the RRAM technology can showcase some very promising performance and stable behaviour, further maturity of this family of technologies need to be achieved to enhance and stabilise the behaviour of the devices in a large scale system where thousands (or even a higher order of magnitude) of RRAM devices should work under a specific range of requirements. Of course, the requirements could be alleviated to some degree if we consider alternative methods of computing (i.e. neuromoprhic computing, etc.), but for all purposes of modern IMC accelerator designs and operations improvements in uniformity of operation need to advance further. Additionally, although in studies such as mine, specific details of the RRAM-MOS circuits response can be observed and useful insights can be gathered, a larger sustained effort in developing fully operational larger systems and architectures (especially fully custom fabricated hybrid ICs) that incorporate RRAM devices for memory and computation in hardware and not only in simulation is necessary. Such larger scale hardware-based studies could shed light on further issues for improvement that need to be address before a potential commercialisation of the RRAM technology. Additionally, the observation of the RRAM-MOS circuits and systems as a closely integrated system in silico could provide new levels of understanding for such designs and potentially novel design methodologies.

All in all, computers need to change in order to provide solutions for high-performance and low-power computing at the edge, enhancing IoT-based fog computing, as well as a multitude of other emerging approaches in distributing computing. RRAM technologies have showcased that they can be an integral part of the effort to design future computers and adapt them to the ever-increasing needs of the big data and decentralised edge computing era. RRAM and circuit design advances can have impact not only on how novel memory systems are designed but more importantly on how memory systems can be used to perform a wide variety of computing. With the advent of such emerging devices, computing concepts such as IMC and reconfigurable computing can gain new levels of performance.

Appendix A

List of publications

Journal:
1. **G. Papandroulidakis**, A. Serb, A. Khiat, G. V. Merrett and T. Prodromakis, “Practical Implementation of Memristor-Based Threshold Logic Gates”, in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 8, August 2019

Conference paper:
1. J. Szypicyn, C. Papavasiliou, **G. Papandroulidakis**, A. Serb, S. Stathopoulos, Geoff V. Merrett and T. Prodromakis, ”Reconfigurable Memristor Integrated Circuits”, International Conference on Electronics, Information, and Communication (ICEIC2020), 2020

2. **G. Papandroulidakis**, A. Serb, A. Khiat, Geoff V. Merrett and T. Prodromakis, “Practical Implementation of digital in-analogue out memristor-based logic circuit” in International Conference on Memristive Materials, Devices and Systems (MEMRISYS) 2019, 2019.

3. **G. Papandroulidakis**, L. Michalas, A. Serb, A. Khiat, Geoff V. Merrett and T. Prodromakis “A Digital-In-Analogue-Out Logic Gate Based on Metal-Oxide Memristor Devices”, in IEEE International Symposium on Circuits and Systems (ISCAS) 2019, 2019

4. A. Serb, **G. Papandroulidakis**, A. Khiat, and T. Prodromakis, “Plane-Splitting Logic Techniques using Hyrbid CMOS-Memristor Circuits and Systems,” in International Conference on Memristive Materials, Devices and Systems (MEMRISYS) 2018, 2018.

5. **G. Papandroulidakis**, A. Khiat, A. Serb, S. Stathopoulos, L. Michalas, and T. Prodromakis, “Desing and Practical Implementation of Memristor-based Threshold Logic Gate,” in International Conference on Memristive Materials, Devices and Systems (MEMRISYS) 2018, 2018.

6. A. Serb, **G. Papandroulidakis**, A. Khiat, and T. Prodromakis, “Processing big-data with Memristive Technologies: Splitting the Hyperplane Efficiently,” in IEEE International Symposium on Circuits and Systems (ISCAS) 2018, 2018.

7. **G. Papandroulidakis**, A. Khiat, A. Serb, S. Stathopoulos, L. Michalas, and T. Prodromakis, “Metal Oxide-enabled Reconfigurable Memristive Threshold Logic Gates,” in IEEE International Symposium on Circuits and Systems (ISCAS) 2018, 2018.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, 2016. ISSN 0270-6474. URL `http://arxiv.org/abs/1603.04467`.

G. C. Adam, B. D. Hoskins, M. Prezioso, F. Merrikh Bayat, B. Chakrabarti, and D. B. Strukov. Highly-uniform multi-layer ReRAM crossbar circuits. *European Solid-State Device Research Conference*, 2016-Octob:436–439, 2016. ISSN 19308876. .

Shyam Prasad Adhikari, Changju Yang, Hyongsuk Kim, and Leon O. Chua. Memristor bridge synapse-based neural network and its learning. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1426–1435, 2012. ISSN 2162237X. .

A. Afifi, A. Ayatollahi, and F. Raissi. Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits. *ECCTD 2009 - European Conference on Circuit Theory and Design Conference Program*, pages 563–566, 2009. .

Amogh Agrawal, Akhilesh Jaiswal, Chankyu Lee, and Kaushik Roy. X-SRAM: Enabling in-memory boolean computations in CMOS static random access memories. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(12):4219–4232, 2018. ISSN 15498328. .

F Alibart, L Gao, and D B Strukov. A Reconfigurable FIR Filter with Memristor-Based Weights. *arXiv:1608.05445 [cs.ET]*, 2016.

Yousra Alkabani, Mario Miscuglio, Volker J Sorger, and Tarek El-ghazawi. OE-CAM : A Hybrid Opto-Electronic Content Addressable Memory. *arXiv:1912.02220 [physics.app-ph]*, xx(xx):1–14, 2019.

Stefano Ambrogio, Pritish Narayanan, Hsinyu Tsai, Robert M. Shelby, Irem Boybat, Carmelo Di Nolfo, Severin Sidler, Massimo Giordano, Martina Bodini, Nathan C.P. Farinha, Benjamin Killeen, Christina Cheng, Yassine Jaoudi, and Geoffrey W. Burr. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature*, 558(7708):60–67, 2018. ISSN 14764687. . URL `http://dx.doi.org/10.1038/s41586-018-0180-5`.

Janeen D.W. Anderson, Carver A. Mead, Timothy P. Allen, and Michael F Wall. CMOS Winner Take All Circuit with Offset Adaptation, 1992.

K. L. Baishnab, Mustafijur Rahaman, and F. A. Talukdar. A 200μv resolution and high speed VLSI winner-take-all circuit for self-organising neural network. In *Proceedings of International Conference on Methods and Models in Computer Science, ICM2CS09*, pages 5–8, 2009. ISBN 9789380043579. .

K. L. Baishnab, P. K. Paul, Naushad Manzoor Laskar, Sourav Nath, and Paramita Sarkar. Modelling and optimization of CMOS winner-takes-all circuit for improved slew rate using swarm intelligence based techniques. *Journal of Information and Optimization Sciences*, 38(6):841–856, 2017. ISSN 0252-2667. .

F. Merrikh Bayat, M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva, and D. Strukov. Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nature Communications*, 9(1):1–7, 2018. ISSN 20411723. . URL `http://dx.doi.org/10.1038/s41467-018-04482-4`.

Farnood Merrikh Bayat, Mirko Prezioso, Bhaswar Chakrabarti, Irina Kataeva, and Dmitri Strukov. Memristor-based perceptron classifier: Increasing complexity and coping with imperfect hardware. In *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, pages 549–554, 2017. ISBN 9781538630938. .

Hussein Bazzi, Adnan Harb, Hassen Aziza, Mathieu Moreau, and Abdallah Kassem. RRAM-based non-volatile SRAM cell architectures for ultra-low-power applications. *Analog Integrated Circuits and Signal Processing*, 106(2):351–361, feb 2021. ISSN 15731979. .

V. Beiu. Ultra-fast noise immune CMOS threshold logic gates. *Midwest Symposium on Circuits and Systems*, 3:2–5, 2000. .

V. Beiu. Threshold logic implementations: the early days. In *2003 46th Midwest Symposium on Circuits and Systems*, pages 1379–1383, 2003. ISBN 0-7803-8294-3. . URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1562552`.

V Beiu, J M Quintana, and M J Avedillo. VLSI implementations of threshold logic-a comprehensive survey. *IEEE transactions on neural networks / a publication of the*

IEEE Neural Networks Council, 14(5):1217–43, 2003a. ISSN 1045-9227. . URL http://www.ncbi.nlm.nih.gov/pubmed/18244573.

V. Beiu, J. M. Quintana, M. J. Avedilo, and R. Andonie. Differential implementations of threshold logic gates. *SCS 2003 - International Symposium on Signals, Circuits and Systems, Proceedings*, 2:489–492, 2003b. .

V. Beiu, J. M. Quintana, M. J. Avedilo, and R. Andonie. Differential implementations of threshold logic gates. *SCS 2003 - International Symposium on Signals, Circuits and Systems, Proceedings*, 2:489–492, 2003c. .

Valeriu Beiu, Jose M Quintana, Maria J Avedillo, Mawahib Sulieman I, Edificio Cica, and Avda Reina. Threshold Logic: From Vacuum Tubes to Nanoelectronics. In *2003 46th Midwest Symposium on Circuits and Systems*, pages 930–935, 2003d. ISBN 0780382943. . URL https://ieeexplore.ieee.org/document/1562439/.

D. Biolek, Z. Kolka, V. Biolkova, and Z. Biolek. Memristor models for SPICE simulation of extremely large memristive networks. *Proceedings - IEEE International Symposium on Circuits and Systems*, 2016-July(14):389–392, 2016. ISSN 02714310. .

Dalibor Biolek, Viera Biolkova, and Zdenek Kolka. Spice models of memristive devices forming a model of Hodgkin-Huxley axon. *2013 18th International Conference on Digital Signal Processing, DSP 2013*, 2013. .

Zdeněk Biolek, Dalibor Biolek, and Viera Biolková. SPICE model of memristor with nonlinear dopant drift. *Radioengineering*, 18(2):210–214, 2009. ISSN 12102512.

S. Bobba and N Hajj. Current-Mode Threshold Logic Gates. In *Computer Design, 2000. Proceedings. 2000 International Conference on*, pages 235–240, Austin, TX, USA, 2000. IEEE. ISBN 0769508014. .

Geoffrey W. Burr, Robert M. Shelby, Abu Sebastian, Sangbum Kim, Seyoung Kim, Severin Sidler, Kumar Virwani, Masatoshi Ishii, Pritish Narayanan, Alessandro Fumarola, Lucas L. Sanches, Irem Boybat, Manuel Le Gallo, Kibong Moon, Jiyoo Woo, Hyunsang Hwang, and Yusuf Leblebici. Neuromorphic computing using non-volatile memory. *Advances in Physics: X*, 2(1):89–124, 2017. ISSN 2374-6149. . URL https://www.tandfonline.com/doi/full/10.1080/23746149.2016.1259585.

Fuxi Cai, Justin M. Correll, Seung Hwan Lee, Yong Lim, Vishishtha Bothra, Zhengya Zhang, Michael P. Flynn, and Wei D. Lu. A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. *Nature Electronics*, 2 (7):290–299, jul 2019. ISSN 25201131. . URL https://www.nature.com/articles/s41928-019-0270-x.

Bertrand Cambou, David Hély, and Sareh Assiri. Cryptography with Analog Scheme Using Memristors. *ACM Journal on Emerging Technologies in Computing Systems*, 16(4), oct 2020. ISSN 15504840. .

Kristofor D. Carlson, Michael Beyeler, Nikil Dutt, and Jeffrey L. Krichmar. GPGPU accelerated simulation and parameter tuning for neuromorphic applications. *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, pages 570–577, 2014. ISSN 00189219. .

B. Chakrabarti, M. A. Lastras-Montaño, G. Adam, M. Prezioso, B. Hoskins, K.-T. Cheng, and D. B. Strukov. A multiply-add engine with monolithically integrated 3D memristor crossbar/CMOS hybrid circuit. *Scientific Reports*, 7(September 2016):42429, 2017. ISSN 2045-2322. . URL http://www.nature.com/articles/srep42429.

Yi Chung Chen, Wenhua Wang, Hai Li, and Wei Zhang. Non-volatile 3D stacking RRAM-based FPGA. *Proceedings - 22nd International Conference on Field Programmable Logic and Applications, FPL 2012*, pages 367–372, 2012. . URL http://www.mendeley.com/research/nonvolatile-3d-stacking-rrambased-fpga.

Kwang-Ting Cheng and Dmitri B Strukov. 3D CMOS-Memristor Hybrid Circuits: Devices, Integration, Architecture, and Applications. In *ACM International Symposium on Phsyical Design (ISPD) 2012*, Napa, California, USA, 2012.

Leon Chua. If it's pinched it's a memristor. *Semiconductor Science and Technology*, 29(10): 1–42, 2014. ISSN 0268-1242. .

Leon Chua. Everything you wish to know about memristors but are afraid to ask. *Radioengineering*, 24(2):319–368, 2015. ISSN 12102512. .

Loai Danial, Nicolas Wainstein, Shraga Kraus, and Shahar Kvatinsky. Breaking Through the Speed-Power-Accuracy Tradeoff in ADCs using a Memristive Neuromorphic Architecture. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(5):396–409, 2018. . URL https://asic2.group/wp-content/uploads/2018/06/FINAL-VERSION-Loai-1.pdf.

Loai Danial, Evgeny Pikhay, Eric Herbelin, Nicolas Wainstein, Vasu Gupta, Nimrod Wald, Yakov Roizin, Ramez Daniel, and Shahar Kvatinsky. Two-terminal floating-gate transistors with a low-power memristive operation mode for analogue neuromorphic computing. *Nature Electronics*, 2(12):596–605, dec 2019. ISSN 25201131. .

Chandra Babu Dara, Themistoklis Haniotakis, and Spyros Tragoudas. Low Power and High Speed Current-Mode Memristor-Based TLGs. In *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DTFS)*, pages 89–94, New York City, NY, USA, 2013. IEEE. ISBN 9781479915859.

Chandra Babu Dara, Themistoklis Haniotakis, and Spyros Tragoudas. Delay Analysis for Current Mode Threshold Logic Gate Designs. *IEEE Transactions on Very Large Scale Integration Systems*, 25(3):1063–1071, 2017. .

N. Dastanova, S. Duisenbay, O. Krestinskaya, and A.P. James. Bit-plane Extracted Moving-object Detection using Memristive Crossbar-CAM Arrays for Edge Computing Image Devices. *IEEE Access*, 6(March), 2018. ISSN 21693536. .

Jonas Doevenspeck, Robin Degraeve, Stefan Cosemans, Philippe Roussel, Bram Ernst Verhoef, Rudy Lauwereins, and Wim Dehaene. Analytic variability study of inference accuracy in RRAM arrays with a binary tree winner-take-all circuit for neuromorphic applications. *European Solid-State Device Research Conference*, 2018-Septe: 62–65, 2018. ISSN 19308876. .

Rodney Douglas, Misha Mahowald, and Carver Mead. Neuromorphic Analogue VLSI. *Annual Review of Neuroscience*, 18(1):255–281, 1995. ISSN 0147006X. . URL `http://neuro.annualreviews.org/cgi/doi/10.1146/annurev.neuro.18.1.255`.

Idongesit E. Ebong and Pinaki Mazumder. CMOS and memristor-based neural network design for position detection. *Proceedings of the IEEE*, 100(6):2050–2060, 2012. ISSN 00189219. .

Arthur H. Edwards, Hugh J. Barnaby, Kristy A. Campbell, Michael N. Kozicki, Wei Liu, and Matthew J. Marinella. Reconfigurable memristive device technologies. *Proceedings of the IEEE*, 103(7):1004–1033, 2015. ISSN 00189219. .

Tarek El-Chazawi, Esam El-Araby, and Miaoqing Huang. The Promise of High-Performance Reconfigurable Computing. *IEEE Computer Society, Research Feature*, 2008. .

A.S. Emara, A.H. Madian, H.H. Amer, S.H Amer, and M.B. Abdelhalim. Testing of Memristor Ratioed Logic (MRL) XOR Gate. *IEEE ICM 2016*, 2016. .

B. Rasitha Fernando, Raqibul Hasan, and M. Tarek Taha. Low Power Memristor Crossbar Based Winner Takes All Circuit. *Proceedings of the International Joint Conference on Neural Networks*, 2018-July:1–6, 2018. .

Steve Furber. Large-scale neuromorphic computing systems. *Journal of Neural Engineering*, 13(051001), 2016. . URL `http://iopscience.iop.org/1741-2552/13/5/051001`.

Pierre Emmanuel Gaillardon, Xifan Tang, Gain Kim, and Giovanni De Micheli. A Novel FPGA Architecture Based on Ultrafine Grain Reconfigurable Logic Cells. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(10):2187–2197, 2015. ISSN 10638210. . URL `Gaillardon2015`.

Manuel Le Gallo, Abu Sebastian, Roland Mathis, Matteo Manica, Tomas Tuma, Costas Bekas, Alessandro Curioni, and Evangelos Eleftheriou. Mixed-Precision 'Memcomputing'. *arXiv:1701.04279v3 [cs.ET]*, pages 1–7, 2017. URL `http://arxiv.org/abs/1701.04279`.

Ligang Gao, Fabien Alibart, and Dmitri B Strukov. Programmable CMOS / Memristor Threshold Logic. *IEEE Transactions on Nanotechnology*, 12(2), 2013a. ISSN 1536-125X. .

Ligang Gao, Fabien Alibart, and Dmitri B. Strukov. Programmable CMOS/memristor threshold logic. *IEEE Transactions on Nanotechnology*, 12(2):115–119, 2013b. . URL http://www.mendeley.com/research/programmable-cmosmemristor-threshold-logic.

Sujan K. Gonugondla, Mingu Kang, Yongjune Kim, Mark Helm, Sean Eilert, and Naresh Shanbhag. Energy-Efficient Deep In-memory Architecture for NAND Flash Memories. *Proceedings - IEEE International Symposium on Circuits and Systems*, 2018-May:0–4, 2018. ISSN 02714310. .

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):1–1, 2017. ISSN 0278-0070. . URL http://ieeexplore.ieee.org/document/7876763/.

Zhenyuan Guo, Jun Wang, and Zheng Yan. Global exponential synchronization of two memristor-based recurrent neural networks with time delays via static or dynamic coupling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2):235–249, 2015. ISSN 10834427. .

Sarthak Gupta, Pratik Kumar, Tathagata Paul, André van Schaik, Arindam Ghosh, and Chetan Singh Thakur. Low Power, CMOS-MoS2 Memtransistor based Neuromorphic Hybrid Architecture for Wake-Up Systems. *Scientific Reports*, 9(1):1–9, dec 2019. ISSN 20452322. . URL https://doi.org/10.1038/s41598-019-51606-x.

Basma Hajri, Mohammad M Mansour, and Ali Chehab. Oxide-based RRAM Models for Circuit Designers : A Comparative Analysis. In *2017 12th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*. IEEE, 2017. ISBN 9781509063772.

Basma Hajri, Hassen Aziza, Mohammad M Mansour, and Senior Member. RRAM Device Models : A Comparative Analysis With Experimental Validation. *IEEE Access*, 7:168963–168980, 2020. .

Yasmin Halawani, Student Member, Baker Mohammad, Senior Member, Muath Abu Lebdeh, Mahmoud Al-qutayri, Senior Member, and Said F Al-sarawi. ReRAM-Based In-Memory Computing for Search Engine and Neural Network Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2):388–397, 2019.

Said Hamdioui, Mottaqiallah Taouil, Hoang Anh Du Nguyen, Adib Haron, Lei Xie, and Koen Bertels. Memristor: The enabler of computation-in-memory architecture for big-data. *2015 International Conference on Memristive Systems, MEMRISYS 2015*, pages 9–11, 2016. .

Raqibul Hasan and Tarek M. Taha. Memristor crossbar based winner take all circuit design for self-organization. *ACM International Conference Proceeding Series*, 2017-July: 1–4, 2017. .

Zhezhi He and Deliang Fan. Energy efficient reconfigurable threshold logic circuit with spintronic devices. *IEEE Transactions on Emerging Topics in Computing*, 5(2):223–237, 2017. ISSN 21686750. .

J A Hidalgo-Lopez, J C Tejero, J Fernandez, and A Gago. New types of digital comparators. *Proceedings of the 1995 IEEE International Symposium on Circuits and Systems-ISCAS 95. Part 3 (of 3)*, 1:29–32, 1995. ISSN 02714310. URL `http://www.scopus.com/inward/record.url?eid=2-s2.0-0029205032{&}partnerID=40{&}md5=9a29fba90afb4009c3c56f0bb5449807`.

Veeresh Hongal, Raghavendra Kotikalapudi, and Minsu Choi. Design, test, and repair of MLUT (Memristor Look-Up Table) based asynchronous nanowire reconfigurable crossbar architecture. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 4(4):427–437, 2014. ISSN 21563357. .

Miao Hu, Hai Li, Yiran Chen, Qing Wu, Garrett S. Rose, and Richard W. Linderman. Memristor crossbar-based neuromorphic computing system: A case study. *IEEE Transactions on Neural Networks and Learning Systems*, 25(10):1864–1878, 2014. ISSN 21622388. .

Miao Hu, John Paul Strachan, Zhiyong Li, Emmanuelle Merced Grafals, Noraica Davila, Catherine Graves, Sity Lam, Ning Ge, R Stanley Williams, Jianhua Yang, and Hewlett Packard Labs. Dot-Product Engine for Neuromorphic Computing: Programming 1T1M Crossbar to Accelerate Matrix-Vector Multiplication. In *IEEE Design Automation Conference*, pages 1—-6, 2016a. ISBN 9781450311991. .

Miao Hu, John Paul Strachan, Zhiyong Li, R. Stanley, and Williams. Dot-product engine as computing memory to accelerate machine learning algorithms. In *Proceedings - International Symposium on Quality Electronic Design, ISQED*, pages 374–379, Santa Clara, CA, USA, 2016b. ISBN 9781509012138. .

Miao Hu, Catherine E. Graves, Can Li, Yunning Li, Ning Ge, Eric Montgomery, Noraica Davila, Hao Jiang, R. Stanley Williams, J. Joshua Yang, Qiangfei Xia, and John Paul Strachan. Memristor-Based Analog Computation and Neural Network Classification with a Dot Product Engine. *Advanced Materials*, 1705914:1–10, 2018. ISSN 15214095. .

Yipeng Huang, Ning Guo, Mingoo Seok, Yannis Tsividis, and Simha Sethumadhavan. Evaluation of an Analog Accelerator for Linear Algebra. *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*, pages 570–582, 2016. .

Yu-cherng Hung, Chung-yang Tsai, and Bin-da Liu. 1-V Rail-to-Rail Analog CMOS Programmable Winner-Take-All Chip With Two-Side Searching Capability. In *IEEE Int. Conf. Neural Networks & Singal Processing 2003*, pages 337–340, 2003. ISBN 0780377028.

Daniele Ielmini and H.-S. Philip Wong. In-memory computing with resistive switching devices. *Nature Electronics*, 1(6):333–343, 2018. ISSN 2520-1131. . URL http://www.nature.com/articles/s41928-018-0092-2.

Giacomo Indiveri. A current-mode hysteretic Winner-take-all network, with excitatory and inhibitory coupling. *Analog Integrated Circuits and Signal Processing*, 28(3):279–291, 2001. ISSN 09251030. .

Giacomo Indiveri and Shih Chii Liu. Memory and Information Processing in Neuromorphic Systems, 2015. ISSN 00189219.

Giacomo Indiveri, Elisabetta Chicca, and Rodney Douglas. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Transactions on Neural Networks*, 17(1):211–221, 2006. ISSN 10459227. .

Giacomo Indiveri, Bernabé Linares-Barranco, Tara Julia Hamilton, André van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, Johannes Schemmel, Gert Cauwenberghs, John Arthur, Kai Hynna, Fopefolu Folowosele, Sylvain Saighi, Teresa Serrano-Gotarredona, Jayawan Wijekoon, Yingxue Wang, and Kwabena Boahen. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5(MAY):1–23, 2011. ISSN 16624548. .

Hiroshi Inokawa, Akira Fujiwara, and Yasuo Takahashi. A Multiple-Valued Logic and Memory With Combined Single-Electron and Metal–Oxide–Semiconductor Transistors. *IEEE TRANSACTIONS ON ELECTRON DEVICES*, 50(2), 2003.

A. P. James, L. R. V. J. Francis, and D. Kumar. Resistive Threshold Logic. *arXiv:1208.0090v1 [cs.ET]*, pages 1–12, 2013. . URL http://arxiv.org/abs/1308.0090{%}0Ahttp://dx.doi.org/10.1109/TVLSI.2012.2232946.

Alex Pappachen James. Memristor Threshold Logic : An Overview to Challenges and Applications. In *International Conference on Contemporary Computing and Informatics*, pages 13–16, 2016.

Alex Pappachen James, Anusha Pachentavida, and Sherin Sugathan. Edge detection using resistive threshold logic networks with CMOS flash memories. *International Journal of Intelligent Computing and Cybernetics*, 7(1):79–94, 2014. ISSN 1756-378X. . URL http://www.emeraldinsight.com/10.1108/IJICC-06-2013-0032.

Alex Pappachen James, Dinesh S. Kumar, and Arun Ajayan. Threshold Logic Computing: Memristive-CMOS Circuits for Fast Fourier Transform and Vedic Multiplication. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(11):2690–2694, 2015a. ISSN 10638210. .

Alex Pappachen James, Dinesh S. Kumar, and Arun Ajayan. Threshold Logic Computing: Memristive-CMOS Circuits for Fast Fourier Transform and Vedic Multiplication. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(11):2690–2694, 2015b. . URL `http://www.mendeley.com/research/threshold-logic-computing-memristivecmos-circuits-fast-fourier-transform-vedic-mult`

Supreet Jeloka, Naveen Bharathwaj Akesh, Dennis Sylvester, and David Blaauw. A 28 nm Configurable Memory (TCAM/BCAM/SRAM) Using Push-Rule 6T Bit Cell Enabling Logic-in-Memory. *IEEE Journal of Solid-State Circuits*, 51(4):1009–1021, 2016. ISSN 00189200. .

Yeonjoo Jeong and Wei Lu. Neuromorphic computing using memristor crossbar networks: A focus on bio-inspired approaches. *IEEE Nanotechnology Magazine*, 12(3): 9–18, 2018. ISSN 19427808. .

J Joshua Yang, Dmitri B Strukov, and Duncan R Stewart. Memristive devices for computing. *Nature Nanotechnology*, 8, 2013. .

Pilin Junsangsri, Jie Han, and Fabrizio Lombardi. Design and Comparative Evaluation of a PCM-Based CAM (Content Addressable Memory) Cell. *IEEE Transactions on Nanotechnology*, 16(2):359–363, 2017. ISSN 1536125X. .

Mingu Kang and Naresh R. Shanbhag. In-Memory Computing Architectures for Sparse Distributed Memory. *IEEE Transactions on Biomedical Circuits and Systems*, 10(4):855–863, 2016. ISSN 19324545. .

Mingu Kang, Sujan K. Gonugondla, Ameya Patil, and Naresh R. Shanbhag. A Multi-Functional In-Memory Inference Processor Using a Standard 6T SRAM Array. *IEEE Journal of Solid-State Circuits*, 53(2):642–655, 2018. ISSN 00189200. .

Robert Karam, Ruchir Puri, Swaroop Ghosh, and Swarup Bhunia. Emerging Trends in Design and Applications of Memory-Based Computing and Content-Addressable Memories. *Proceedings of the IEEE*, 103(8):1311–1330, 2015. ISSN 15582256. .

Devinder Kaur. Associative RAM-net memory neural target classifier. *Optical Engineering*, 37(7):2043, 1998. ISSN 0091-3286. .

Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu. The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices. In *Proceedings - International Symposium on High-Performance Computer Architecture*, pages 194–207. IEEE, 2018. ISBN 9781538636596. .

Kyung Min Kim, Nuo Xu, Xinglong Shao, Kyung Jean Yoon, Hae Jin Kim, R. Stanley Williams, and Cheol Seong Hwang. Single-Cell Stateful Logic Using a Dual-Bit Memristor. *Physica Status Solidi - Rapid Research Letters*, 13(3):1–8, 2019. ISSN 18626270. .

Jack Koenig, David Biancolin, Jonathan Bachrach, and Krste Asanovic. A Hardware Accelerator for Computing an Exact Dot Product. *Proceedings - 24th IEEE Symposium on Computer Arithmetic, ARITH 2017*, pages 114–121, 2017. .

Zdenek Kolka, Viera Biolkova, and Dalibor Biolek. Simplified SPICE Model of TiO 2 Memristor. In *2015 International Conference on Memristive Systems, MEMRISYS*, number 3, pages 9–10, 2015. ISBN 9781467392099.

Olga Krestinskaya and Alex Pappachen James. Approximate Probabilistic Neural Networks with Gated Threshold Logic. *arXiv:1808.00733 [cs.ET]*, pages 10–13, 2018. URL `http://arxiv.org/abs/1808.00733`.

Olga Krestinskaya and Alex Pappachen James. Feature extraction without learning in an analog Spatial Pooler memristive-CMOS circuit design of Hierarchical Temporal Memory. *arXiv:1803.05131v1 [cs.ET]*, 2018.

Olga Krestinskaya, Alex Pappachen James, and Leon O. Chua. Neuro-memristive Circuits for Edge Computing: A review. *arXiv:1807.00962 [cs.ET]*, pages 1–17, 2018a. URL `http://arxiv.org/abs/1807.00962`.

Olga Krestinskaya, Akshay Kumar Maan, and Alex Pappachen James. Programmable Memristive Threshold Logic Gate Array. *arXiv:1809.00419v1 [cs.ET]*, 2018b.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012. ISSN 10495258. .

Niranjan Kulkarni, Jinghua Yang, Jae Sun Seo, and Sarma Vrudhula. Reducing Power, Leakage, and Area of Standard-Cell ASICs Using Threshold Logic Flip-Flops. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(9):2873–2886, 2016. ISSN 10638210. .

Raushan Kumar, Prashant Kumar, and Sahadev Roy. Efficient minimization Techniques for Threshold Logic Gate. *International Research Journal of Engineering and Technology (IRJET)*, 3(4):1375–1382, 2016.

T Nandha Kumar, Haider A F Almurib, and Fabrizio Lombardi. A Novel Design of a Memristor-Based Look-Up Table (LUT) For FPGA. In *Circuits and Systems (APCCAS), 2014 IEEE Asia Pacific Conference on*, Ishigaki, Japan, 2014. IEEE.

Akshay Kumar Maan, Deepthi Anirudhan Jayadevi, and Alex Pappachen James. A Survey of Memristive Threshold Logic Circuits. *IEEE Transactions on Neural Networks and Learning Systems*, 28(8):1734 – 1746, 2016. .

Ian Kuon and Jonathan Rose. Measuring the Gap between FPGAs and ASICs. In *FPGA*, 2006. ISBN 1595932925. .

Shahar Kvatinsky, Avinoam Kolodny, Uri C. Weiser, and Eby G. Friedman. Memristor-based IMPLY logic design procedure. In *Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors*, number 864, pages 142–147. IEEE, 2011. ISBN 9781457719523. .

Shahar Kvatinsky, Nimrod Wald, Guy Satat, Avinoam Kolodny, Uri C. Weiser, and Eby G. Friedman. MRL - Memristor Ratioed Logic. In *International Workshop on Cellular Nanoscale Networks and their Applications*, 2012. ISBN 9781467302890. .

Shahar Kvatinsky, Guy Satat, Nimrod Wald, Eby G. Friedman, Avinoam Kolodny, and Uri C. Weiser. Memristor-based material implication (IMPLY) logic: Design principles and methodologies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(10):2054–2066, 2014. ISSN 10638210. .

F. Lalchhandama, Brojo Gopal Sapui, and Kamalika Datta. An improved approach for the synthesis of boolean functions using memristor based IMPLY and INVERSE-IMPLY Gates. *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, 2016-Septe:319–324, 2016. ISSN 21593477. .

Miguel Angel Lastras-Montaño, Bhaswar Chakrabarti, Dmitri B Strukov, and Kwang-Ting Cheng. 3D-DPE: A 3D high-bandwidth dot-product engine for high-performance neuromorphic computing. *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1257–1260, 2017. .

Manuel Le Gallo, Abu Sebastian, Giovanni Cherubini, Heiner Giefers, and Evangelos Eleftheriou. Compressed Sensing with Approximate Message Passing Using In-Memory Computing. *IEEE Transactions on Electron Devices*, 65(10):4304–4312, 2018. ISSN 00189383. .

Jun Young Lee and Sun Nam Hwang. A high-gain boost converter using voltage-stacking cell. *Transactions of the Korean Institute of Electrical Engineers*, 57(6):982–984, 2008. ISSN 19758359. .

Samuel Leshner. *Modeling and Implementation of Threshold Logic Circuits and Architectures*. PhD thesis, Arizona State University, PhD Thesis, 2010.

Samuel Leshner, Niranjan Kulkarni, Sarma Vrudhula, and Krzysztof Berezowski. Design of a robust, high performance standard cell threshold logic family for DSM technology. In *Proceedings of the International Conference on Microelectronics, ICM*, pages 52–55, 2010. ISBN 9781612841519. .

Bo Li, Yonglei Zhao, and Guoyong Shi. A novel design of memristor-based bidirectional associative memory circuits using Verilog-AMS. *Journal of Neurocomputing (Elsevier)*, 330:437–448, 2019. .

Boxun Li, Yuzhi Wang, Yu Weng, Yiran Chen, and Huazhong Yang. Training itself: Mixed-signal training acceleration for memristor-based neural network. *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, pages 361–366, 2014. .

Kai Shin Li, Ming Taou Lee, Min Cheng Chen, Cho Lun Hsu, J. M. Lu, C. H. Lin, C. C. Chen, B. W. Wu, Y. F. Hou, C. Yi Lin, Y. J. Chen, T. Y. Lai, M. Y. Li, I. Yang, C. S. Wu, Fu Liang Yang, and W. K. Yeh. Study of sub-5 nm RRAM, tunneling selector and selector less device. *Proceedings - IEEE International Symposium on Circuits and Systems*, 2015-July:385–388, 2015. ISSN 02714310. .

Shuangchen Li, Cong Xu, Qiaosha Zou, Jishen Zhao, Yu Lu, and Yuan Xie. Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-volatile Memories. In *Proceedings of the 53rd Annual Design Automation Conference on - DAC '16*, pages 1–6, Austin, TX, USA, 2016. ISBN 9781450342360. . URL http://dl.acm.org/citation.cfm?doid=2897937.2898064.

Shuangchen Li, Dimin Niu, Krishna T. Malladi, Hongzhong Zheng, Bob Brennan, and Yuan Xie. DRISA: A DRAM-based reconfigurable in-situ accelerator. *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, Part F1312:288–301, 2017. ISSN 10724451. .

E. Linn, S. Menzel, S. Ferch, and R. Waser. Compact modeling of CRS devices based on ECM cells for memory, logic and neuromorphic applications. *Nanotechnology*, 24(38), 2013. ISSN 09574484. .

Chenchen Liu, Qing Yang, Bonan Yan, Jianlei Yang, Xiaocong Du, Weijie Zhu, Hao Jiang, Qing Wu, Mark Barnell, and Hai Helen Li. A memristor crossbar based computing engine optimized for high speed and accuracy. *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, 2016-Septe:110–115, 2016. ISSN 21593477. .

Qi LIu, Bin Gao, Peng Yao, Dong Wu, Junren Chen, Yachuan Pang, Wenqiang Zhang, Yan Liao, Cheng-Xin Xue, Wei-Hao Chen, Jianshi Tang, Yu Wang, Meng-Fan Chang, He Qian, and Huanqiang Wu. A Fully Integrated Analog ReRAM Based 78.4TOPS/W Compute-In-Memory Chip with Fully Parallel MAC Computing. In *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*. IEEE, 2020. ISBN 9781728132051.

Xiaoxiao Liu, Mengjie Mao, Beiye Liu, Hai Li, Yiran Chen, Boxun Li, Yu Wang, Hao Jiang, Mark Barnell, Qing Wu, and Jianhua Yang. RENO: A High-efficient Reconfigurable Neuromorphic Computing Accelerator Design. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, 2015. .

Akshay Kumar Maan, Dinesh Sasi Kumar, Sherin Sugathan, and Alex Pappachen James. Memristive Threshold Logic Circuit Design of Fast Moving Object Detection. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(10):2337–2341, 2015. ISSN 10638210. .

Akshay Kumar Maan, Deepthi Anirudhan Jayadevi, and Alex Pappachen James. A survey of memristive threshold logic circuits. *IEEE Transactions on Neural Networks and Learning Systems*, 28(8):1734–1746, 2016. ISSN 21622388. .

Advait Madhavan, Timothy Sherwood, and Dmitri Strukov. Race Logic: A hardware acceleration for dynamic programming algorithms. *Proceedings - International Symposium on Computer Architecture*, pages 517–528, 2014. ISSN 10636897. .

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. ISSN 00074985. .

Sally A McKee. Reflections on the Memory Wall. *Proceedings of the 1st Conference on Computing Frontiers*, pages 162–167, 2004. . URL http://doi.acm.org/10.1145/977091.977115.

Carver A. Mead and Timothy P. Allen. Adaptable CMOS Winner Take All Circuit, 1991.

A. Medina-Santiago, Mario Alfredo Reyes-Barranca, Ignation Algredo-Badillo, Alfornso Martinez Cruz, Kelsey Alejandra Ramirez Gutierrez, and Adrian Eleazar Cortes-Barron. Reconfigurable-Arithmetic-Logic-Unit-Designed-With-Threshold-Logic-Gates.pdf. *Institute of Engineering and Technology (IET) Journal*, 1(46):81–91, 2018.

Adnan Mehonic, Abu Sebastian, Bipin Rajendran, Osvaldo Simeone, Eleni Vasilaki, and Anthony J. Kenyon. Memristors—From In-Memory Computing, Deep Learning Acceleration, and Spiking Neural Networks to the Future of Neuromorphic and Bio-Inspired Computing. *Advanced Intelligent Systems*, page 2000085, 2020. ISSN 2640-4567. .

F. Merrikh-Bayat, M. Prezioso, X. Guo, B. Hoskins, D. B. Strukov, and K. K. Likharev. Memory Technologies for Neural Networks. *2015 IEEE 7th International Memory Workshop, IMW 2015*, 2015. ISSN 2159-483X. .

Farnood Merrikh-Bayat and Saeed Bagheri Shouraki. Memristive neuro-fuzzy system. *IEEE Transactions on Cybernetics*, 43(1):269–285, 2013. ISSN 21682267. .

Farnood Merrikh-Bayat, Saeed Bagheri Shouraki, and Farshad Merrikh-Bayat. Memristor crossbar-based hardware implementation of fuzzy membership functions. *Proceedings - 2011 8th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011*, 1:645–649, 2011. .

Farnood Merrikh-Bayat, Xinjie Guo, Michael Klachko, Mirko Prezioso, Konstantin K. Likharev, and Dmitri B. Strukov. High-Performance Mixed-Signal Neurocomputing With Nanoscale Floating-Gate Memory Cell Arrays. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–9, 2017. ISSN 21622388. .

I. Messaris, A. Ascoli, G. S. Meinhardt, R. Tetzlaff, and L. O. Chua. Mem-Computing CNNs with Bistable-Like Memristors. *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2019. . URL https://ieeexplore.ieee.org/document/8702414/.

Ioannis Messaris, Alexandrou Serb, and Themis Prodromakis. A Compact Verilog-A Memristor Switching Model. *arXiv:1703.01167*, 2017. . URL https://arxiv.org/ftp/arxiv/papers/1703/1703.01167.pdf.

Ioannis Messaris, Alexander Serb, Spyros Stathopoulos, Ali Khiat, Spyridon Nikolaidis, and Themistoklis Prodromakis. A Data-Driven Verilog-A ReRAM Model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, TBA(TBA): 1–12, 2018. ISSN 02780070. .

Matthias Meyer, Timo Farei-Campagna, Akos Pasztor, Reto Da Forno, Tonio Gsell, Jérome Faillettaz, Andreas Vieli, Samuel Weber, Jan Beutel, and Lothar Thiele. Event-triggered Natural Hazard Monitoring with Convolutional Neural Networks on the Edge. *IPSN 2019 - Proceedings of the 2019 Information Processing in Sensor Networks*, pages 73–84, 2019. .

L Michalas, A Khiat, S Stathopoulos, and T Prodromakis. Metal - TiO2 contacts: An electrical characterization study. Technical report, 2017. URL http://arxiv.org/abs/1712.04218https://arxiv.org/ftp/arxiv/papers/1712/1712.04218.pdf.

S. M. Mirhoseini, M. J. Sharifi, and D. Bahrepour. New RTD-based general threshold gate topologies and application to three-input XOR logic gates. *Journal of Electrical and Computer Engineering*, 2010(1):1–5, 2010. ISSN 20900147. .

Yiannis Moisiadis, Ilias Bouras, Angela Arapoyanni, and Lampros Dermentzoglou. A static differential double edge-triggered flip-flop based on clock racing. *Microelectronics Journal*, 32(8):665–671, 2001. ISSN 00262692. .

D. Moro-Frias, M. T. Sanz-Pascual, and C. A. De La Cruz Blas. A novel current-mode winner-take-all topology. In *2011 20th European Conference on Circuit Theory and Design, ECCTD 2011*, pages 134–137. IEEE, 2011. ISBN 9781457706189. .

Seyed Nima Mozaffari and Spyros Tragoudas. A Generalized Approach to Implement Efficient CMOS-Based Threshold Logic Functions. *IEEE Transactions on Circuits and Systems I: Regular Papers*, PP(99):1–14, 2017. .

Seyed Nima Mozaffari and Spyros Tragoudas. Maximizing the Number of Threshold Logic Functions Using Resistive Memory. *IEEE Transactions on Nanotechnology (TNANO)*, 17(5):897 – 905, 2018. .

Seyed Nima Mozaffari, Spyros Tragoudas, and Themistoklis Haniotakis. More Efficient Testing of Metal-oxide Memristor-based Memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 0070(c):1–1, 2016. ISSN 0278-0070. . URL `http://ieeexplore.ieee.org/document/7565639/`.

Seyed Nima Mozaffari, Spyros Tragoudas, and Themistoklis Haniotakis. A new method to identify threshold logic functions. In *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 934–937, 2017. ISBN 9783981537086.

Seyed Nima Mozaffari, Spyros Tragoudas, and Themistoklis Haniotakis. A Generalized Approach to Implement Efficient CMOS-Based Threshold Logic Functions. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(3):946–959, 2018. ISSN 15498328. .

Gauthaman Murali, Xiaoyu Sun, Shimeng Yu, and Sung Kyu Lim. Heterogeneous Mixed-Signal Monolithic 3-D In-Memory Computing Using Resistive RAM. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(2):386–396, feb 2021. ISSN 15579999. .

Onur Mutlu, Justin Meza, and Lavanya Subramanian. The Main Memory System: Challenges and Opportunities. *Communications of the Korean Institute of Information Scientists and Engineers*, pages 16–41, 2015.

Augusto Neutzling, Jody Maick Matos, Alan Mishchenko, Andre Reis, and Renato P. Ribas. Effective Logic Synthesis for Threshold Logic Circuit Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, XX(X):1–12, 2018. ISSN 02780070. .

Nishant S. Nukala, Niranjan Kulkarni, and Sarma Vrudhula. Spintronic Threshold Logic Array (STLA) - A compact, low leakage, non-volatile gate array architecture. *Journal of Parallel and Distributed Computing*, 2014. .

Christian Pacha, Karl Goser, Andreas Brennemann, and Werner Prost. A threshold logic full adder based on resonant tunneling transistors. In *European Solid-State Circuits Conference*, number September, pages 428–431, 1998. .

Biao Pan, Kang Wang, Xing Chen, Jinyu Bai, Jianlei Yang, Youguang Zhang, and Weisheng Zhao. SR-WTA: Skyrmion racing winner-takes-all module for spiking neural computing. *Proceedings - IEEE International Symposium on Circuits and Systems*, 2019-May:1–5, 2019. ISSN 02714310. .

G. Papandroulidakis, I. Vourkas, A. Abusleme, G.C. Sirakoulis, and A. Rubio. Crossbar-Based Memristive Logic-in-Memory Architecture. *IEEE Transactions on Nanotechnology*, 16(3), 2017. ISSN 1536125X. .

G Papandroulidakis, A Khiat, A Serb, S Stathopoulos, L Michalas, and T Prodromakis. Metal Oxide-enabled Reconfigurable Memristive Threshold Logic Gates. In *IEEE International Symposium on Circuits and Systems (ISCAS) 2018*, 2018. ISBN 9781538648810.

G. Papandroulidakis, L. Michalas, A. Serb, A. Khiat, G.V. Merrett, and T. Prodromakis. A digital in-analogue out logic gate based on metal-oxide memristor devices. In *Proceedings - IEEE International Symposium on Circuits and Systems*, volume 2019-May, 2019a. ISBN 9781728103976. .

G. Papandroulidakis, A. Serb, A. Khiat, G.V. Merrett, and T. Prodromakis. Practical Implementation of Memristor-Based Threshold Logic Gates. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(8), 2019b. ISSN 15580806. .

Melika Payvand, Manu V Nair, Lorenz K. Muller, and Giacomo Indiveri. A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation. *Faraday Discussions Royal Society of Chemistry*, 2018. ISSN 1359-6640. . URL http://arxiv.org/abs/1807.05128{%}0Ahttp://dx.doi.org/10.1039/C8FD00114F.

Hector Pettenghi, Mar??a J. Avedillo, and Jos?? M. Quintana. Using multi-threshold threshold gates in RTD-based logic design: A case study. *Microelectronics Journal*, 39 (2):241–247, 2008a. ISSN 00262692. .

Héctor Pettenghi, María J. Avedillo, and José M. Quintana. A novel contribution to the RTD-based threshold logic family. *Proceedings - IEEE International Symposium on Circuits and Systems*, pages 2350–2353, 2008b. ISSN 02714310. .

Shuang Pi, Peng Lin, and Qiangfei Xia. Cross point arrays of 8 nm × 8 nm memristive devices fabricated with nanoimprint lithography. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, 31(6):06FA02, 2013. ISSN 21662746. . URL http://scitation.aip.org/content/avs/journal/jvstb/31/6/10.1116/1.4827021.

T Prodromakis and C Toumazou. A Review on Memristive Devices and Applications. In *2010 17th IEEE International Conference on Electronics, Circuits and Systems*, 2010.

Themistoklis Prodromakis, Christos Papavassiliou, and Christofer Toumazou. A Versatile Memristor Model With Nonlinear Dopant Kinetics. *IEEE TRANSACTIONS ON ELECTRON DEVICES*, 58(9), 2011. .

Li Qu, Xiaole Cui, Xiaoyan Xu, Xiaoxin Cui, and Ye Ma. The multi-input MRL logic gate and its application. *2019 IEEE International Conference on Electron Devices and Solid-State Circuits, EDSSC 2019*, pages 2019–2020, 2019. .

J M Quintana and A Rueda. Low-power CMOS threshold-logic gate. *Electronics Letters*, 31(25):2157–2159, 1995. . URL `https://ieeexplore.ieee.org/document/481016/?arnumber=481016`.

Abbas Rahimi, Amirali Ghofrani, Kwang-ting Cheng, Luca Benini, and Rajesh K Gupta. Approximate Associative Memristive Memory for Energy-Efficient GPUs. In *Design, Automation, and Test in Europe Conference & Exhibition (DATE)*, number 2, pages 1497–1502, 2015. ISBN 9783981537048. .

Ehsan Rahiminejad, Mehdi Saberi, Reza Lotfi, Mohammad Taherzadeh-Sani, and Frederic Nabki. A Low-Voltage High-Precision Time-Domain Winner-Take-All Circuit. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(1):1–1, 2019. ISSN 1549-7747. .

L. F. Rahman, F. A. Rudha, M. B. I. Reaz, and M. Marufuzzaman. The Evolution of Digital to Analog Converter. In *2016 International Conference on Advances in Electrical, Electronic and System Engineering*, pages 14–16, 2016. ISBN 9781509028894.

Jeyavijayan Rajendran, Harika Manem, Ramesh Karri, and Garrett S. Rose. Memristor based programmable threshold logic array. *Proceedings of the 2010 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2010*, pages 5–10, 2010. .

Jeyavijayan Rajendran, Ramesh Karri, and Garrett S. Rose. Parallel memristors: Improving variation tolerance in memristive digital circuits. *Proceedings - IEEE International Symposium on Circuits and Systems*, pages 2241–2244, 2011. ISSN 02714310. .

Jeyavijayan Rajendran, Harika Manem, Ramesh Karri, and Garrett S. Rose. An Energy-Efficient Memristive Threshold Logic Circuit. *IEEE Transactions on Computers*, 61(4): 474–487, 2012. ISSN 0018-9340. .

J. Fernández Ramos, J. A.Hidalgo López, M. J. Martin, J. C. Tejero, and A. Gago. A threshold logic gate based on clocked coupled inverters. *International Journal of Electronics*, 84(4):371–382, 1998. ISSN 13623060. .

John Reuben, Rotem Ben-hur, Nimrod Wald, Nishil Talati, Ameer Haj Ali, Pierre-emmanuel Gaillardon, and Shahar Kvatinsky. Memristive Logic: A Framework for Evaluation and Comparison. *IEEE International Symposium on Power and Timing Modeling, Optimization and Simulation*, (Section III), 2017.

John Reuben, Dietmar Fey, and Christian Wenger. A modeling methodology for Resistive RAM based on Stanford-PKU model with extended multilevel capability. *IEEE Transactions on Nanotechnology*, 18(June):1–1, 2019. ISSN 1536-125X. .

Raul Rojas. Threshold Logic. In *Neural Networks*, chapter 2 (from Ne, pages 29–41. Springer-Verlag, 1996.

Giovanni Rovere, Schekeb Fateh, and Luca Benini. A 2.1 μW event-driven wake-up circuit based on a level-crossing ADC for pattern recognition in healthcare. *2017 IEEE Biomedical Circuits and Systems Conference, BioCAS 2017 - Proceedings*, 2018-Janua:1–4, 2018a. .

Giovanni Rovere, Schekeb Fateh, and Luca Benini. A 2.2-μ W Cognitive Always-On Wake-Up Circuit for Event-Driven Duty-Cycling of IoT Sensor Nodes. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(3):543–554, 2018b. ISSN 21563357. .

Giulia Santoro, Giovanna Turvani, and Mariagrazia Graziano. New Logic-In-Memory Paradigms: An Architectural and Technological Perspective. *Micromachines*, 10(6): 368, 2019. ISSN 2072-666X. . URL `https://www.mdpi.com/2072-666X/10/6/368`.

Abu Sebastian, Manuel Le Gallo, Geoffrey W. Burr, Sangbum Kim, Matthew Brightsky, and Evangelos Eleftheriou. Tutorial: Brain-inspired computing using phase-change memory devices. *Journal of Applied Physics*, 124(11), 2018. ISSN 00236438. .

Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-aljameh, Evangelos Eleftheriou, Manuel Le Gallo, Riduan Khaddam-aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15(7):529–544, 2020. ISSN 1748-3395. . URL `http://dx.doi.org/10.1038/s41565-020-0655-z`.

A. Serb, G. Papandroulidakis, A. Khiat, and T. Prodromakis. Processing big-data with Memristive Technologies: Splitting the Hyperplane Efficiently. *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018a. . URL `https://ieeexplore.ieee.org/document/8351773/`.

Alexander Serb, Giacomo Indiveri, Themis Prodromakis, Hesham Mostafa, Christian G. Mayr, and Ali Khiat. Implementation of a spike-based perceptron learning rule using TiO2x memristors. *Frontiers in Neuroscience*, 9(October):1–11, 2015. .

Alexander Serb, Christos Papavassiliou, and Themistoklis Prodromakis. A memristor-CMOS hybrid architecture concept for on-line template matching. *Proceedings - IEEE International Symposium on Circuits and Systems*, pages 1–4, 2017. ISSN 02714310. .

Alexantrou Serb, Radu Berdan, Ali Khiat, Christos Papavassiliou, and Themistoklis Prodromakis. Live demonstration: A versatile, low-cost platform for testing large

ReRAM cross-bar arrays. In *Proceedings - IEEE International Symposium on Circuits and Systems*, page 441. Institute of Electrical and Electronics Engineers Inc., 2014. ISBN 9781479934324. .

Alexantrou Serb, Ali Khiat, and Themistoklis Prodromakis. Seamlessly Fused Digital-Analogue Reconfigurable Computing using Memristors. *Nature Communications*, 9 (2170):16–18, 2018b. . URL https://eprints.soton.ac.uk/421791/.

Teresa Serrano-Gotarredona and Bernabé Linares-Barranco. A modular current-mode high-precision winner-take-all circuit. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(2):132–134, 1995.

Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A Kozuch, Onur Mutlu, Phillip B Gibbons, Todd C Mowry, and Hasan Has-San. Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology. In *In Proceedings of Annual IEEE/ACM International Symposium on Microarchitecture*, 2017. ISBN 9781450340342. . URL http://dx.doi.org/10.1145/3123939.3124544.

Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R. Stanley Williams, and Vivek Srikumar. ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars. In *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*, 2016. ISBN 9781467389471. .

Patrick M. Sheridan, Chao Du, and Wei D. Lu. Feature Extraction Using Memristor Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11):2327–2336, 2016. ISSN 21622388. .

Xin Si, Win San Khwa, Jia Jing Chen, Jia Fang Li, Xiaoyu Sun, Rui Liu, Shimeng Yu, Hiroyuki Yamauchi, Qiang Li, and Meng Fan Chang. A Dual-Split 6T SRAM-Based Computing-in-Memory Unit-Macro with Fully Parallel Product-Sum Operation for Binarized DNN Edge Processors. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(11):4172–4185, 2019. ISSN 15580806. .

Xin Si, Student Member, Jia-jing Chen, Yung-ning Tu, Wei-hsing Huang, Jing-hong Wang, Yen-cheng Chiu, Wei-chen Wei, Ssu-yen Wu, Xiaoyu Sun, and Student Member. A Twin-8T SRAM Computation-in-Memory Unit-Macro for Multibit CNN-Based AI Edge Processors. *IEEE Journal of Solid-State Circuits*, 55(1):189–202, 2020.

Anne Siemon, Dirk Wouters, Said Hamdioui, and Stephan Menzel. Memristive Device Modeling and Circuit Design Exploration for Computation-in-Memory. *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2019. . URL https://ieeexplore.ieee.org/document/8702600/.

Spyros Stathopoulos, Ali Khiat, Maria Trapatseli, Simone Cortese, Alexantrou Serb, Ilia Valov, and Themis Prodromakis. Multibit memory operation of metal-oxide Bi-layer memristors. *Scientific Reports*, 7(1), 2017. ISSN 20452322. .

R Strandberg and J Yuan. Single input current-sensing differential logic (SCSDL). *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, 1:764–767 vol.1, 2000. ISSN 02714310. .

D B Strukov, W Robinett, G Snider, J P Strachan, W Wu, Q Xia, J Joshua Yang, and R Stanley Williams. Hybrid CMOS / Memristor Circuits. *New York*, pages 1967–1970, 2010. ISSN 9781424453085. . URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5537020.

Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart, and R. Stanley Williams. The missing memristor found. *Nature*, 459(7250):1154–1154, 2009. ISSN 0028-0836. . URL http://www.nature.com/doifinder/10.1038/nature08166.

Jian Wei Su, Xin Si, Yen Chi Chou, Ting Wei Chang, Wei Hsing Huang, Yung Ning Tu, Ruhui Liu, Pei Jung Lu, Ta Wei Liu, Jing Hong Wang, Zhixiao Zhang, Hongwu Jiang, Shanshi Huang, Chung Chuan Lo, Ren Shuo Liu, Chih Cheng Hsieh, Kea Tiong Tang, Shyh Shyuan Sheu, Sih Han Li, Heng Yuan Lee, Shih Chieh Chang, Shimeng Yu, and Meng Fan Chang. A 28nm 64Kb Inference-Training Two-Way Transpose Multibit 6T SRAM Compute-in-Memory Macro for AI Edge Chips. *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 2020-Febru:240–242, 2020. ISSN 01936530. .

Zhong Sun, Elia Ambrosi, Alessandro Bricalli, and Daniele Ielmini. Logic Computing with Stateful Neural Networks of Resistive Switches. *Advanced Materials*, 30(38):1–8, 2018a. ISSN 15214095. .

Zhong Sun, Elia Ambrosi, Alessandro Bricalli, and Daniele Ielmini. Logic Computing with Stateful Neural Networks of Resistive Switches. *Advanced Materials*, 30(38): 1802554, 2018b. ISSN 09359648. .

Mahdi Tarkhan and Mohammad Maymandi-Nejad. Design of a memristor based fuzzy processor. *AEU - International Journal of Electronics and Communications*, 84(August 2017):331–341, 2018. ISSN 16180399. . URL https://doi.org/10.1016/j.aeue.2017.10.039.

Suryanarayana Tatapudi and Valeriu Beiu. Split-Precharge Differential Noise-Immune Threshold Logic Gate (SPD-NTL). In *International Work-Conference on Artificial Neural Networks, IWANN 2003: Artificial Neural Nets Problem Solving Methods pp 49-56*, 2003.

M. Teimoori, A. Ahmadi, S. Alirezaee, and M. Ahmadi. A novel hybrid CMOS-memristor logic circuit using Memristor Ratioed Logic. *Canadian Conference on Electrical and Computer Engineering*, 2016-Octob(1):1–4, 2016. ISSN 08407789. .

Thanh Tran, Adrian Rothenbuhler, Elisa H Barney Smith, Vishal Saxena, and Kristy A. Campbell. Reconfigurable Threshold Logic Gates using memristive devices. *2012 IEEE Subthreshold Microelectronics Conference, SubVT 2012*, pages 174–193, 2012. ISSN 2079-9268. .

Venkatesh Mani Tripathi, Sandeep Mishra, Jyotishman Saikia, and Anup Dandapat. A Low-Voltage 13T Latch-Type Sense Amplifier with Regenerative Feedback for Ultra Speed Memory Access. *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)*, pages 341–346, 2017. . URL http://ieeexplore.ieee.org/document/7884801/.

Son Ngoc Truong, Khoa Van Pham, Wonsun Yang, and Kyeong Sik Min. Sequential Memristor Crossbar for Neuromorphic Pattern Recognition. *IEEE Transactions on Nanotechnology*, 15(6):922–930, 2016a. ISSN 1536125X. .

Son Ngoc Truong, Khoa Van Pham, Wonsun Yang, Kyeong Sik Min, Yawar Abbas, Chi Jung Kang, Sangho Shin, and Ken Pedrotti. Ta2O5-memristor synaptic array with winner-take-all method for neuromorphic pattern matching. *Journal of the Korean Physical Society*, 69(4):640–646, 2016b. ISSN 19768524. .

Mesbah Uddin and Garrett S. Rose. A Practical Sense Amplifier Design for Memristive Crossbar Circuits (PUF). *International System on Chip Conference*, 2018-Septe:209–214, 2019. ISSN 21641706. .

Mesbah Uddin, Aysha S. Shanta, Md Badruddoja Majumder, Md Sakib Hasan, and Garrett S. Rose. Memristor Crossbar PUF based Lightweight Hardware Security for IoT. *2019 IEEE International Conference on Consumer Electronics, ICCE 2019*, pages 1–4, 2019. .

Zahid Ullah, Kim Ilgon, and Sanghyeon Baeg. Hybrid partitioned SRAM-based ternary content addressable memory. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(12):2969–2979, 2012. ISSN 15498328. .

Hossein Valavi, Peter J Ramadge, Eric Nestler, and Naveen Verma. A 64-Tile 2.4-Mb In-Memory Computing CNN Accelerator Employing Charge-Domain Compute. *IEEE Journal of Solid-State Circuits*, 54(6):1789–1799, 2019. .

Naveen Verma, Hongyang Jia, Hossein Valavi, Yingi Tang, Murat Ozalay, Lung-Yen Chen, Bonan Zhang, and Peter Deaville. In-Memory Computing: Advances and prospects. *IEEE Solid-State Circuits Magazine*, pages 43–55, 2019. .

Mario Vestias and Horacio Neto. Trends of CPU, GPU and FPGA for high-performance computing. *Conference Digest - 24th International Conference on Field Programmable Logic and Applications, FPL 2014*, 2014. . URL http://www.mendeley.com/research/trends-cpu-gpu-fpga-highperformance-computing.

I. Vourkas, G. Papandroulidakis, G.C. Sirakoulis, and A. Abusleme. 2T1M-based double memristive crossbar architecture for in-memory computing. *International Journal of Unconventional Computing*, 12(4), 2016. ISSN 15487202.

Ioannis Vourkas and Georgios Ch Sirakoulis. Emerging Memristor- Based Logic Circuit Design Approaches : A Review. *IEEE Circuits and Systems Magazine*, 16(3160042):15–30, 2016. .

Sarma Vrudhula, Niranjan Kulkami, and Jinghua Yang. Design of threshold logic gates using emerging devices. *Proceedings - IEEE International Symposium on Circuits and Systems*, 2015-July:373–376, 2015. ISSN 02714310. .

Qingzhou Wan, Mohammad T. Sharbati, John R. Erickson, Yanhao Du, and Feng Xiong. Emerging Artificial Synaptic Devices for Neuromorphic Computing. *Advanced Materials Technologies*, 4(4):1–34, 2019. ISSN 2365709X. .

J. J. Wang, Q. Yu, S. G. Hu, Yanchen Liu, Rui Guo, T. P. Chen, Y. Yin, and Y. Liu. Winner-Takes-All mechanism realized by memristive neural network. *Applied Physics Letters*, 115(24), 2019. ISSN 00036951. .

Shiping Wen, Xudong Xie, Zheng Yan, Tingwen Huang, and Zhigang Zeng. General memristor with applications in multilayer neural networks. *Neural Networks*, 103 (2018):142–149, 2018. ISSN 08936080. . URL `http://linkinghub.elsevier.com/retrieve/pii/S0893608018301084`.

MV Wilkes. The memory wall and the CMOS end-point. *ACM SIGARCH Computer Architecture News*, pages 1994–1996, 1995. ISSN 0163-5964. . URL `http://dl.acm.org/citation.cfm?id=218865`.

Chia Heng Wu, Ting Sheng Chen, Ding Yuan Lee, Tsung Te Liu, and An Yeu Wu. Low-latency Voltage-Racing Winner-Take-All (VR-WTA) circuit for acceleration of learning engine. *2017 International Symposium on VLSI Design, Automation and Test, VLSI-DAT 2017*, pages 1–4, 2017. .

Yue Xi, Bin Gao, Jianshi Tang, An Chen, Meng Fan Chang, Xiaobo Sharon Hu, Jan Van Der Spiegel, He Qian, and Huaqiang Wu. In-Memory Learning With Analog Resistive Switching Memory: A Review and Perspective. *Proceedings of the IEEE*, 2020. ISSN 15582256. .

Qiangfei Xia and J. Joshua Yang. Memristive crossbar arrays for brain-inspired computing. *Nature Materials*, 18(4):309–323, 2019. ISSN 14764660. . URL `http://dx.doi.org/10.1038/s41563-019-0291-x`.

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 Letters2*, 9(10):3640–3645, 2009. .

Lei Xie, Hoang Anh, Du Nguyen, Mottaqiallah Taouil, Said Hamdioui, and Koen Bertels. Interconnect Networks for Memristor Crossbar. In *Nanoscale Architectures (NANOARCH), 2015 IEEE/ACM International Symposium on*, 2015.

Lei Xie, Hoang Anh, Du Nguyen, Mottaqiallah Taouil, Said Hamdioui, Koen Bertels, and Mohammad Alfailakawi. Non-Volatile Look-up Table Based FPGA Implementations. In *Non-Volatile Look-up Table Based FPGA Implementations*, 2016. ISBN 978-1-5090-4900-4. .

Lei Xie, H. A.Du Nguyen, Jintao Yu, Ali Kaichouhi, Mottaqiallah Taouil, Mohammad Alfailakawi, and Said Hamdioui. Scouting Logic: A Novel Memristor-Based Logic Design for Resistive Computing. *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, 2017-July:176–181, 2017. ISSN 21593477. .

Jinghua Yang, Niranjan Kulkarni, Shimeng Yu, and Sarma Vrudhula. Integration of threshold logic gates with RRAM devices for energy efficient and robust operation. In *Proceedings of the 2014 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2014*, number 3, pages 39–44, 2014. ISBN 9781479963836. .

Leonid Yavits, Amir Morad, and Ran Ginosar. Computer Architecture with Associative Processor Replacing Last-Level Cache and SIMD Accelerator. *IEEE Transactions on Computers*, 64(2):368–381, 2015. ISSN 00189340. .

Shihui Yin and Zhewei Jiang. Vesti : Energy-Efficient In-Memory Computing Accelerator for Deep Neural Networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(1):48–61, 2020.

Shihui Yin, Xiaoyu Sun, Shimeng Yu, and Jae Sun Seo. High-Throughput In-Memory Computing for Binary Deep Neural Networks with Monolithically Integrated RRAM and 90-nm CMOS. *IEEE Transactions on Electron Devices*, 67(10):4185–4192, oct 2020. ISSN 15579646. .

Chengshuo Yu, Taegeun Yoo, Tony Tae Hyoung Kim, Kevin Chai Tshun Chuan, and Bongjin Kim. A 16K Current-Based 8T SRAM Compute-In-Memory Macro with Decoupled Read/Write and 1-5bit Column ADC. *Proceedings of the Custom Integrated Circuits Conference*, 2020-March, 2020. ISSN 08865930. .

Jintao Yu, Hoang Anh Du Nguyen, Lei Xie, Mottaqiallah Taouil, and Said Hamdioui. Memristive Devices for Computation-In-Memory. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019. . URL `http://arxiv.org/abs/1907.07898{%}0Ahttp://dx.doi.org/10.23919/DATE.2018.8342278`.

Yue Zha and Jing Li. Reconfigurable In-Memory Computing with Resistive Memory Crossbar. In *ICCAD '16: Proceedings of the 35th International Conference on Computer-Aided Design*, 2016. ISBN 9781450344661. .

Yue Zha and Jing Li. IMEC: A Fully Morphable In-Memory Computing Fabric Enabled by Resistive Crossbar. *IEEE Computer Architecture Letters*, 6056(c):1–1, 2017. ISSN 1556-6056. . URL http://ieeexplore.ieee.org/document/7862129/.

Askhat Zhanbossinov, Kamilya Smagulova, and Alex Pappachen James. CMOS-memristor dendrite threshold circuits. *2016 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2016*, pages 131–134, 2017. .

Jintao Zhang, Zhuo Wang, and Naveen Verma. In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array. *IEEE Journal of Solid-State Circuits*, 52(4):915–924, 2017. ISSN 00189200. .

Renyuan Zhang and Mineo Kaneko. A 16-valued logic FPGA architecture employing analog memory circuit. *Proceedings - IEEE International Symposium on Circuits and Systems*, 2016-July:718–721, 2016. ISSN 02714310. .

Wenqiang Zhang, Bin Gao, Jianshi Tang, Peng Yao, Shimeng Yu, Meng Fan Chang, Hoi Jun Yoo, He Qian, and Huaqiang Wu. Neuro-inspired computing chips, jul 2020a. ISSN 25201131. URL https://www.nature.com/articles/s41928-020-0435-7.

Yang Zhang, Zhongrui Wang, Jiadi Zhu, Yuchao Yang, Mingyi Rao, Wenhao Song, Ye Zhuo, Xumeng Zhang, Menglin Cui, Linlin Shen, Ru Huang, and J. Joshua Yang. Brain-inspired computing with memristors: Challenges in devices, circuits, and systems, mar 2020b. ISSN 19319401.

Ying Zhou, Huaqiang Wu, Bin Gao, Wei Wu, Yue Xi, Peng Yao, Shuanglin Zhang, Qingtian Zhang, and He Qian. Associative Memory for Image Recovery with a High-Performance Memristor Array. *Advanced Functional Materials*, 1900155:1–7, 2019. ISSN 16163028. .

Qiuling Zhu, Berkin Akin, H. Ekin Sumbul, Fazle Sadi, James C. Hoe, Larry Pileggi, and Franz Franchetti. A 3D-stacked logic-in-memory accelerator for application-specific data intensive computing. *2013 IEEE International 3D Systems Integration Conference, 3DIC 2013*, 2013. .

Mohammed A. Zidan, John Paul Strachan, and Wei D. Lu. The future of electronics based on memristive systems. *Nature Electronics*, 1(1):22–29, 2018. ISSN 2520-1131. . URL http://www.nature.com/articles/s41928-017-0006-8.