Politecnico di Milano

Department of Electronics, Information and Bioengineering

Master of Science in Electronics Engineering

Statistical Models of Program/Verify Algorithm in Resistive Memory Arrays for Neural Network Accelerators

Master Thesis of:

Francesco ANZALONE

Student ID Number 898648

Supervisor: Prof. Daniele IELMINI

Co-supervisors: Dr. Elia AMBROSI, Dr. Valerio MILO

Academic Year 2018–2019

Abstract

In the last decades, the performances of computing systems have remarkably grown, driven by the scaling of electronic devices predicted by Moore's law. However, as the scaling process has been brought forward, some physical limitations of transistors operation, such as the increase of leakage current, have emerged. This has led to the flattening of the operating frequency increase, in order to limit the power dissipation and the consequent excessive temperature increase of the chip. Therefore, as speed cannot be further increased at the device level, other solutions have to be found in order to improve the performances.

In addition, current computing systems are based on the von Neumann architecture, where processing (CPU) and memory units are physically separated. This separation causes a remarkable inefficiency, since the data have to be continuously transferred between memory, where they are stored, and CPU, where they are processed, and vice versa. This operation consumes large amount of energy and make memory access time much slower than processor's speed, issue commonly referred to as memory wall.

These limitations are becoming more and more evident, since emerging data-intensive tasks, such as the internet of things and artificial intelligence, need for storage and processing huge amount of data, typically unstructured. For this reason, new computing paradigms such as neuromorphic computing, which aims at replicating the dense neural networks of the human brain, composed of neurons and synapses, have been deeply investigated in recent years. Indeed, thanks to synapse plasticity, biological neural networks are able to both store and process information with high speed and extremely low power consumption, thus achieving a huge parallelism, making them suitable for typical machine learning tasks, such as image and speech recognition.

Emerging memory technologies are very promising candidates to achieve efficient computing approaches like neuromorphic computing, since they can complement the memory hierarchy of current computing systems, enabling an improvement in scaling and speed. Moreover, thanks to their unique physical properties and structure, they can be assembled in crosspoint architecture, which allows the hardware implementation of neuromorphic systems. Among all types of emerging memories, resistive switching memory (RRAM) stands out thanks to its excellent scalability, high switching speed, good endurance, low fabrication cost and the possibility to program the cell at different resistive values, i.e. achieving multilevel operation. In particular, the latter feature is crucial for neural networks implementation in hardware, as it allows to obtain analog programming behavior.

However, RRAM devices show serious reliability issues, such as resistance fluctuations over time and programming variability, which strongly limit their adoption for emerging data-centric applications. For this reason, advanced programming techniques, such as program/verify algorithm, have been proposed in order to limit the RRAM programming variability.

This thesis focuses on the study of multilevel programming variability of a 4-kbit aluminum-doped hafnium oxide (HfAlO) RRAM array under program/verify algorithm and proposes a statistical model able to explain and predict the cycle-to-cycle (C2C) and device-to-device (D2D) variability of experimental data.

After an extensive introduction on the challenges that the semiconductor industry is nowadays facing, chapter 1 presents an overview of the most important emerging memory technologies, namely RRAM, phase change memory (PCM), spin transfer torque magneto-resistive memory (STT-MRAM) and ferroelectric memory (FeRAM). Then, the main solutions adopted for the physical implementation of neural networks in resistive switching devices crosspoint arrays are addressed.

Chapter 2 reports a detailed description of the physical properties of RRAMs, focusing, in particular, on the phenomena accounting for the switching mechanism and programming variability.

In chapter 3, after the description of the 4 kbit HfAlO RRAM array used in this work, the program/verify algorithm used to program it and obtain multilevel operation is presented. In the second part of the chapter, the variability data of an endurance experiment performed on the array are shown, analyzed, and a physical explanation of results is given.

Chapter 4 presents the statistical model developed to predict the programming variability of the HfAlO RRAM array. The model is able to simulate the programming characteristic under external program/verify conditions. The variability is then reproduced in simulation thanks to the introduction of a statistics in the model's parameters, by the Monte Carlo method. The second part of the chapter illustrates the steps followed to tune the model on the experimental data.

Chapter 5 presents the software implementation of a 5-level multilayer neural network using HfAlO RRAM devices as synaptic devices. First, different program/verify techniques are proposed and simulated via the statistical model in order to investigate their impact on the multilevel programming variability. Finally, after the description of the phases used to design and train the network, the network ability to classify handwritten digit images under different programming techniques is tested and discussed.

Sommario

Negli ultimi decenni le prestazioni dei sistemi di computazione sono notevolmente aumentate, grazie alla riduzione della dimensione dei dispositivi elettronici pronosticata dalla legge di Moore. Questa riduzione ha però fatto emergere delle limitazioni fisiche sul funzionamento dei transistor, come per esempio l'aumento della corrente parassita nello stato di off. Per questo motivo la frequenza alla quale i processori operano non è stata aumentata negli ultimi anni. Ciò ha consentito di limitare la dissipazione di potenza, che altrimenti avrebbe fatto crescere eccessivamente la temperatura dei chip, mettendone a rischio l'affidabilità e il funzionamento. Di conseguenza, non essendo possibile aumentare la velocità di calcolo a livello del dispositivo, è necessario trovare delle nuove soluzioni per aumentare le prestazioni.

Inoltre i sistemi di computazione attuali sono basati sull'architettura di von Neumann, caratterizzata dalla separazione fisica tra unità di computazione (CPU) e di memoria. Questa separazione è motivo di una notevole inefficienza, in quanto i dati devono essere continuamente trasferiti dalla memoria, dove sono immagazzinati, alla CPU, dove sono processati, e viceversa. Questa operazione consuma grandi quantità di energia e mette in risalto la differenza fra il più lento tempo di accesso alla memoria e la velocità del processore, problema che prende il nome di memory wall.

Negli ultimi anni l'emergere di nuove applicazioni che richiedono la gestione e la memorizzazione di grandi quantità di dati non strutturati, come l'intelligenza artificiale o l'internet delle cose, ha ulteriormente enfatizzato le suddette limitazioni dei sistemi di calcolo. Per questo motivo sono stati studiati con attenzione nuovi paradigmi di computazione come il neuromorphic computing, il quale mira a replicare le dense reti neurali, composte da neuroni e sinapsi, che caratterizzano il cervello umano. Infatti le reti neurali biologiche, grazie alla plasticità delle sinapsi, sono in grado sia di memorizzare che di processare le informazioni velocemente e con il consumo di modeste quantità di potenza. Ciò rende possibile ottenere un alto livello di parallelismo, requisito fondamentale per la realizzazione delle tipiche attività di machine learning, quali riconoscimento di immagini o suoni.

Le memorie emergenti sono degli ottimi candidati per realizzare questi approcci computazionali più efficienti, perché completano la gerarchia di memoria che caratterizza i sistemi di calcolo attuali, migliorandone la velocità e la possibilità di ridurne le dimensioni. Inoltre, grazie alla loro struttura e alle loro particolari proprietà fisiche, possono essere assemblate in architetture chiamate crosspoint, le quali permettono l'implementazione di sistemi neuromorfici. Tra i vari tipi di memorie emergenti quelle a switching resistivo (RRAM) si distinguono per eccellente scalabilità, alta velocità di switching, ottima endurance, scarso costo di fabbricazione e possibilità di programmare le celle in più di due stati resistivi, ottenendo così un funzionamento multilivello. In particolare, quest'ultima caratteristica è fondamentale per l'implementazione in hardware delle reti neurali, perché permette di realizzare una programmazione analogica delle celle.

Sfortunatamente, però, i dispositivi RRAM presentano dei seri problemi di affidabilità, come la fluttuazione della resistenza nel tempo e la variabilità di programmazione, che limitano fortemente l'utilizzo di tali memorie in applicazioni in cui devono essere gestite grandi quantità di dati. Per questo motivo al fine di limitare la variabilità di programmazione, sono state proposte delle avanzate tecniche di programmazione, come l'algoritmo di program/verify.

In questa tesi viene trattata la variabilità di programmazione multilivello di un array da 4 kbit composto da RRAM di ossido di afnio drogato con alluminio (HfAlO), sotto le condizioni di un particolare algoritmo di program/verify. Inoltre viene proposto un modello statistico in grado di spiegare e predire la variabilità ciclo a ciclo (C2C) e device a device (D2D) dei dati sperimentali.

Dopo un'introduzione sulle attuali sfide del mondo dell'elettronica, il capitolo 1 presenta un riepilogo delle memorie emergenti più importanti, ossia RRAM, memorie a cambiamento di fase (PCM), memorie magnetiche a spin transfer torque (STT-MRAM) e memorie ferroelettriche (FeRAM). In seguito sono affrontate le principali soluzioni adottate per implementare le reti neurali in array crosspoint di memorie resistive.

Il capitolo 2 riporta una descrizione dettagliata delle proprietà fisiche delle RRAM, concentrandosi in particolare sui fenomeni che determinano il meccanismo di switching e la variabilità di programmazione.

Nel capitolo 3, dopo la descrizione dell'array da 4 kbit di RRAM in HfAlO studiato in questa tesi, viene presentato l'algoritmo di program/verify utilizzato per programmare i dispositivi e ottenere un funzionamento multilivello. In seguito vengono mostrati i dati di variabilità ricavati da un esperimento di endurance e ne viene data una spiegazione fisica.

Il capitolo 4 presenta il modello statistico sviluppato per predire la variabilità di programmazione dell'array di RRAM. Il modello è in grado di simulare la caratteristica di programmazione sotto le condizioni di program/verify applicate esternamente. La variabilità è poi riprodotta in simulazione grazie all'introduzione di una statistica nei parametri del modello, secondo il metodo Monte Carlo. Nella seconda parte del capitolo, inoltre, vengono illustrati i passaggi seguiti per tarare il modello sui dati sperimentali.

Il capitolo 5 presenta l'implementazione in software di una rete neurale multistrato a 5 livelli, che utilizza le RRAM come dispositivi sinaptici. In un primo tempo sono proposte e simulate attraverso il modello statistico diverse tecniche di program/verify, al fine di analizzare come queste impattino sulla variabilità di programmazione multilivello. Infine, dopo la descrizione delle fasi di progettazione e training della rete, viene proposto un confronto fra le accuratezze in classificazione della rete, ottenute applicando le tecniche di programmazione selezionate.

Contents

Chapter 1

Emerging memory devices and architectures for neuromorphic computing

This chapter presents the challenges and developments that the semiconductor industry is facing nowadays. As the data to be managed by computers continue to increase to sustain emerging applications such as artificial intelligence or the internet of things, current computing systems and devices are showing their limits in terms of time and energy efficiency and scalability. Thanks to their unique physical properties, emerging memory devices represent a promising solution to overcome such limitations, as they can replace classical memories in the memory hierarchy and can be used to develop new computing paradigms, called neuromorphic computing systems, which aim at replicating the more energy-efficient structure of the human brain. After a more exhaustive introduction of the context, an overview of the most important emerging memory technologies is given, along with the description of new architectures where they can be assembled. Finally, the solutions adopted to implement in hardware a neuromorphic system are addressed.

1.1 Introduction

The technological development of the last decades has radically changed peoples' habits about communication, interaction, learning and many other aspects of everyday life. The internet has strongly modified how information is retrieved and shared, by making it available to everyone who has a connection. The advent of social media has even reinforced this trend, exponentially increasing the number of pictures, videos, texts that are daily shared between users. Moreover, smartphones, laptops and tablets have been manufactured following a frenetic rhythm, and the era of internet of things (IoT), where not only people, but also objects are fully connected, has already begun. Ultimately, emerging applications like artificial intelligence (AI) are exploding, as machines are able to recognize objects or sounds, learn patterns and take autonomous decisions.

In this scenario, it is evident that the amount af data which has to be managed by computers has become huge. In particular, such data are mostly *unstructured*, meaning that they are not ordered and cataloged, but appear in very different formats, from text files to audio, video and images. An efficient management and analysis of such data is therefore needed in order not to limit the remarkable technological development of the last decade. However, current computing systems, even the most advanced ones, are highly inefficient in such management and, since the amount of data will inevitably increase, novel and innovative systems have to be developed.

The reasons for such inefficiency can be cataloged into two different aspects, namely the limitation on device scaling, often referred to as *end of Moore's law* and the architecture of current computing systems, known as *von Neumann architecture*, which is characterized by the physical separation between the processing unit (CPU) and the memory unit. In particular, this separation emphasizes two issues: the growing performance gap between computational speed and memory access time, also known as *memory wall* and the energy dissipation due to the continue transfer of data between CPU and memory, also known as *von Neumann bottleneck*. In the following these problems are quickly addressed.

Since its early years of development, electronics has been matched with Moore's law, which states that transistor count per integrated circuit would double every two years [1]. The semiconductor industry evolved alongside this law, first thanks to Dennard's scaling rules [2] (a period that was called geometric era [3]), then thanks to advances in both process integration and parallel design which helped to overcome some physical limits that industry was facing (effective scaling era [3]). However, in order to match the increasing requirements on static power consumption and limit the stand-by leakage current, the operating voltage was not scaled under the limit of 1V [4, 5]. As a consequence, also the clock frequency increase was stopped, in order to limit the rise of areal power density, which would have reached unsustainable levels, thus compromising circuit operation and other critical features such as battery life [4, 6]. This is why clock frequencies have reached a plateau and cannot be further increased, meaning that the performance improvement must be achieved by following other paths.

On the other hand, the aforementioned physical separation between CPU and memory, typical of the von Neumann architecture, is the other issue limiting the current computing systems performance. In the last decades, the rate of improvement in CPU speed largely exceeded the improvements in memory access speed [7]. This led to the creation of a performance gap known as memory wall. It means that accessing memory to read or write data takes so long, that the whole computing system speed is determined by the memory. In addition, the separation does not affect performances only in terms of time, but also in terms of energy consumption, since data must be continuously moved from CPU, where they are processed, to off-chip memory, where they are stored, and vice versa. This problem, commonly referred to as von Neumann bottleneck, is expected to be exacerbated as applications become more data centric, where computing tasks consists of machine-learning operations such as object, image, and speech recognition [3].

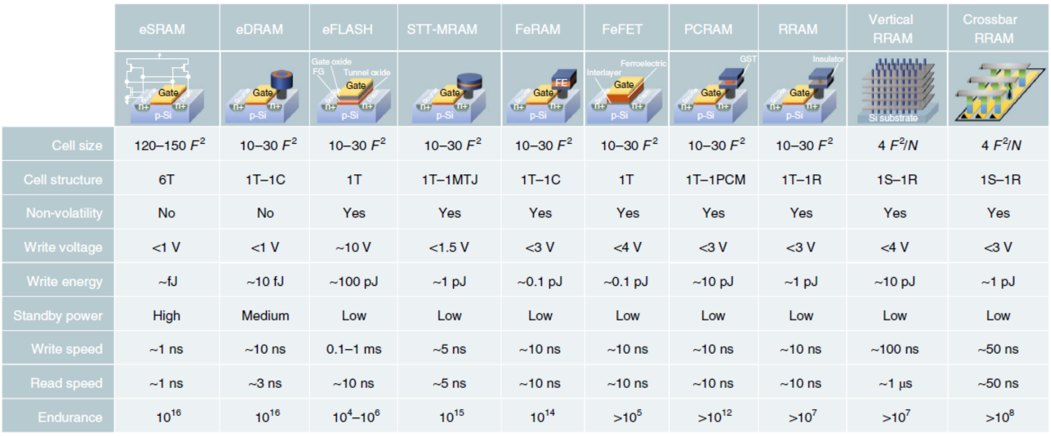
Historically, in order to mitigate the bottleneck, the most promising approach was to bring the memory closer to the processing unit, even on the same chip, exploiting a hierarchy of volatile and non-volatile data storage devices [8]. Cache memories, implemented by static RAM (SRAM) technology, are directly integrated on the processor chip, while the main memory, implemented by dynamic RAM (DRAM) technology is located on a separate chip. Unlike the latter ones, which are volatile, non-volatile memories like hard disks (HDD) or Flash are used for data storage. However, despite the efforts in the introduction of such hierarchy, it is clear that the separation between the main memory and the CPU still represents the main limit in time and energy performance.

Unlike digital computers, the human brain processes and stores information encoded by brief spikes within neurons and synapses of dense neural networks, which enables very high processing speed and extremely low power consumption of only 20 Watts [9,10]. For these reasons, the research focused on neuromorphic computing in the last years. It is a paradigm which aims at emulating the neural architecture of the human brain in terms of structure and operation to realize compact, real-time, and energy-efficient computing systems. The idea to realize such systems is to carry out calculations in situ, inside the memory itself, offering a clear advantage by totally removing the latency and energy burdens of the von Neumann bottleneck [11].

Many applications which are now widely used such as face or voice recognition, are achieved by running neural networks with many layers of neurons and synapses, also known as *deep neural networks*, which are capable of extracting information from very large datasets using deep learning techniques [12, 13]. However, they are still implemented on current computing systems, i.e. on typical von Neumann architectures, which strongly limits their performances in terms of time and power consumption. The important step that has to be made through neuromorphic computing is implementing such networks in a more efficient way, avoiding the limitations introduced by the bottleneck.

To achieve such result, researchers focused on a new class of emerging nonvolatile memory devices in the last years. These are generally grouped under the name *resistive switching devices*, and have unique storage principles which are not based on charge, as in conventional Flash memory, SRAM and DRAM. The storage concept relies instead on the physics of the active materials and the device where they are integrated. These memories, which are all two-terminal devices, include resistive switching RAM (RRAM), phase change memory (PCM), spin transfer torque magneto-resistive RAM (STT-MRAM), and ferroelectric RAM (FeRAM) [11]. Moreover, not only new devices are required for implementing novel computing concepts such as neuromorphic computing, but also new architectures that fully exploit the aforementioned features of emerging memories. Crosspoint array is the main solution to arrange in a smart and compact way such memories [11].

Emerging memories, thanks to their two-terminal structure and storage concept, have unique advantages in density and access time with respect to classical memories (DRAM, Flash, HDD). For these reasons they are good candidates to overcome the incoming end of Moore's law and replace the classical memories in storage applications. However, one of the most promising field related to emerging memories and new architectures is certainly neuromorphic computing, as it can radically change the way computing systems are structured and operated, getting rid of the von Neumann bottleneck. In any case, the development of such structures is still at a first stage, due to some


Gate
n+
p-Si
Source
Drain
Gate oxide
FG
Tunnel oxide
FE
Interlayer
Ferroelectric
GST
Insulator
Si substrate

	eSRAM	eDRAM	eFLASH	STT-MRAM	FeRAM	FeFET	PCRAM	RRAM	Vertical RRAM	Crossbar RRAM
Cell size	120–150 F^2	10–30 F^2	10–30 F^2	10–30 F^2	10–30 F^2	10–30 F^2	10–30 F^2	10–30 F^2	4 F^2/N	4 F^2/N
Cell structure	6T	1T–1C	1T	1T–1MTJ	1T–1C	1T	1T–1PCM	1T–1R	1S–1R	1S–1R
Non-volatility	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Write voltage	<1 V	<1 V	~10 V	<1.5 V	<3 V	<4 V	<3 V	<3 V	<4 V	<3 V
Write energy	~fJ	~10 fJ	~100 pJ	~1 pJ	~0.1 pJ	~0.1 pJ	~10 pJ	~1 pJ	~10 pJ	~1 pJ
Standby power	High	Medium	Low	Low	Low	Low	Low	Low	Low	Low
Write speed	~1 ns	~10 ns	0.1–1 ms	~5 ns	~10 ns	~10 ns	~10 ns	~10 ns	~100 ns	~50 ns
Read speed	~1 ns	~3 ns	~10 ns	~5 ns	~10 ns	~10 ns	~10 ns	~10 ns	~1 µs	~50 ns
Endurance	10^{16}	10^{16}	10^4–10^6	10^{15}	10^{14}	$>10^5$	$>10^{12}$	$>10^7$	$>10^7$	$>10^8$

Figure 1.1: Comparison of different memory technologies. Emerging memories (RRAM, PCM, FeRAM, STT-MRAM) fill the memory gap between DRAM and Flash and are easier to be vertically stacked reducing the cell size in a vertical or crossbar array configuration. Reprinted with permission from [3]. Copyright 2018 Nature Springer, license number 4795990052879.

emerging memories drawbacks, firstly programming variability. This work will be therefore strongly focused on this aspect.

In this chapter, after an overview of the most important resistive memory devices and architectures, the realization of neural networks capable of neuromorphic computing into crosspoint arrays is addressed.

1.2 Emerging non-volatile memory devices

Figure 1.1 compares the most important memory technologies, including both classical and emerging device concepts. The latter ones, namely resistive switching random access memory (RRAM), phase change memory (PCM), ferroelectic memory (FeRAM) and spin-transfer torque magnetic memory (STT-MRAM), have comparable values in speed, write voltage and size. Moreover, they have particular features that make them suitable for filling the performance gap between DRAM and Flash or HDD. Indeed, they are nonvolatile, but have much higher write speed ($\sim 5 - 10ns$) and smaller write voltage ($<$ 3V) than Flash, which brings them closer to the DRAM specifications [3]. Moreover they are more scalable and, since they use a different set of materials and require different device fabrication processes from Flash, they can be easily monolithically integrated on-chip with the microprocessor cores, enabling high-bandwidth data traffic between them [8]. Last two columns on the right illustrate the great advantage in scalability that could be achieved using RRAMs or PCMs assembled in 3D architectures. Indeed, by exploiting the third dimension and vertically stacking several devices, it is possible to reduce the effective cell size down to the minimum value $\frac{4F^2}{N}$, where F is the minimum lithographic feature size and N is the number of levels stacked.

Most importantly, figure 1.1 summarizes the generic specifications of emerging memories and their advantages with respect to classical memories. Now a brief description of the physical principles that account for their operation is given.

1.2.1 Phase Change Memory (PCM)

Phase change memories are based on the so-called phase change materials, namely substances which release or absorb sufficient energy to encounter phase transition. In particular, regarding PCMs, such transition is from amorphous to crystalline phase. Many materials exist in an amorphous phase and a crystalline phase, however, a very small subset of these materials have simultaneously all the properties that make them useful for data storage technologies [14]. These are the chalcogenides, such as $Ge_2Sb_2Te_5$ (GST), whose most important feature is the strong difference in terms of electrical conductivity between the amorphous phase (high resistance) and the crystalline phase (low resistance), which can be of to five orders of magnitude [14].

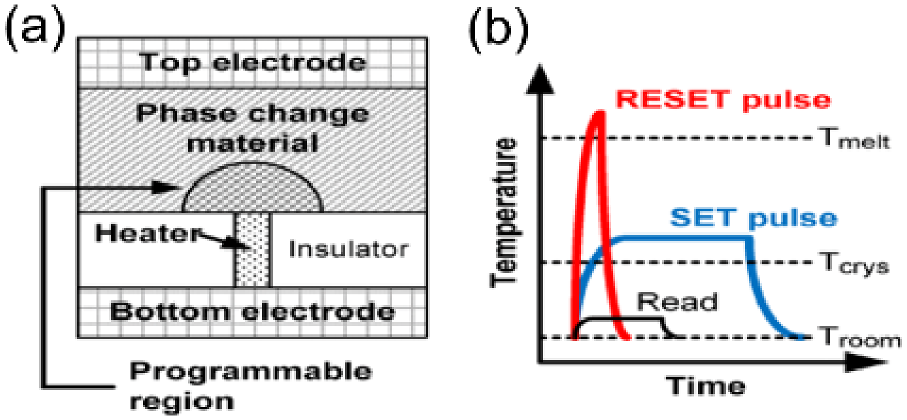


Figure 1.2: (a) Cross section of the typical mushroom structure, composed by a pillar-like bottom electrode which acts as a heater and by a hemispherical shape in which the amorphous material develops. (b) Time evolution of temperature when electrical pulses are applied to set, reset and read the cell. To obtain the amorphous reset state, a very short pulse with high amplitude is applied at the top electrode leading to overcome the melting temperature, which results in amorphization of the active material. To achieve the set state, instead, a longer pulse with low amplitude allows to reach the crystallization temperature, thus leading to active material crystallization. Reprinted with permission from [14]. Copyright 2010 IEEE.

Figure1.2 shows a cross section of a conventional PCM structure (a) and the temperature as a function of time for the different electrical pulses that are applied to set, reset and read the cell (b). The structure shown in (a) is usually named mushroom type, since the bottom electrode has a pillar-like shape that act as a heater and the amorphous region usually develops itself in a hemispherical shape at the bottom electrode level.

The pristine device is always in the crystalline phase, since the processing temperature of the back end of the line (BEOL), where it is fabricated, is sufficient to crystallize the phase change material. To reset the PCM cell into the amorphous phase, the programming region is first melted and then quenched rapidly by applying a large electrical current pulse for a short time period. This leads to a region of amorphous, highly resistive active material in the PCM cell. To set the PCM cell into the crystalline phase, an electrical current pulse of medium amplitude is applied to anneal the programming region at a temperature between the crystallization temperature and the melting temperature for a time period long enough to crystallize.

PCMs present set and reset distributions with high resistance window, i.e. the ratio between high-resistance state (HRS) and low-resistance state (LRS), which allows the storage (and retaining over time) of more than 1 bit of data per cell, i.e. multilevel operation, the ability to read/sense the resistance states without perturbing them, high endurance ($> 10^{12}$ as shown in figure 1.1), and long data retention (usually specified as 10 year data lifetime at some elevated temperature) [15]. On the other hand, a disadvantage in PCM operation is the limitation in operating speed given by the set programming time. Indeed, the set transition is not immediate, since it takes finite time to fully crystallize the amorphous region [14], unlike what happens during reset transition where a very short time is sufficient to achieve the phase change, as shown in figure 1.2 (b). Moreover, PCMs are affected by the drift phenomenon, which consists in the increase in the electrical resistance of the amorphous state with time at room temperature. Such issue, which is caused by the amorphous material structural relaxation, i.e. the thermally-activated, atomistic rearrangement of the amorphous structure [16], is unwanted because it is not controllable and might undermine the multilevel operation of the memory making the different levels indistinguishable [14].

1.2.2 Resistive-switching RAM (RRAM)

Figure 1.3 shows the schematic structure and operation of an RRAM device. The device consists of an insulating layer, usually a metal oxide (MeO_x),

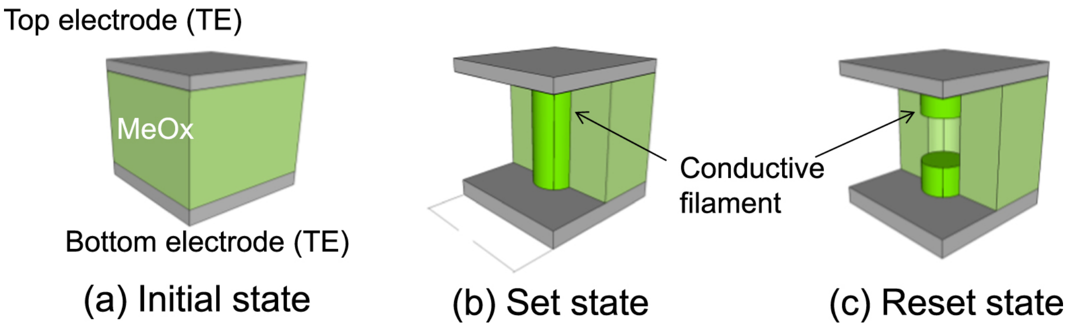


Figure 1.3: RRAM device structure and operation. (a) Pristine device: the simple metal-insulator-metal (MIM) typical structure is underlined. (b) LRS: after the electroforming operation, a conductive filament (CF) connecting BE and TE is created, largely increasing the device conductance. (c) HRS: after the reset operation, the CF is dissolved leaving however some clusters close to the two electrodes. Reprinted with permission from [17]. Copyright 2016 IOP publishing.

interposed between a top electrode (TE) and a bottom electrode (BE), both generally realized by metallic layers or stacks (figure 1.3 (a)). The device is initially subjected to the operation of electroforming, where a conductive filament (CF) is formed by dielectric breakdown (figure 1.3 (b)). After forming, the device exhibits a significant increase in electrical conductivity, as the CF connects the TE and BE, thus resulting in the LRS of the RRAM. The reset operation can then be carried out to disconnect the CF, resulting in the HRS, as shown in figure 1.3 (c). Alternating the set and reset operation, the CF can be repeatedly connected/disconnected, thus allowing multiple transition cycles between HRS and LRS [17]. It is important to underline that during forming and set operation the current is typically limited by a compliance system or a series resistor/transistor, in order to control the size of the CF and avoid the destructive (hard) breakdown of the switching layer [17].

There are two main methods of resistive switching: unipolar, in which both set and reset are obtained through the application of voltage pulses

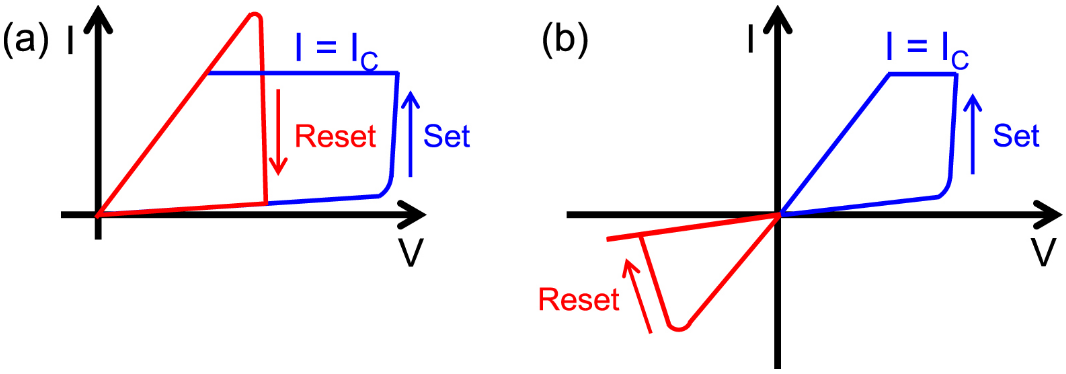


Figure 1.4: (a) Unipolar typical IV characteristic. Both set and reset operations happen at voltages of same polarity (positive in the figure). (b) Bipolar IV characteristic, showing that set transition is achieved at positive voltage, while reset at negative. Both cases show the importance of a limitation in current (compliance current I_C) during the set, crucial to avoid the oxide hard breakdown. Reprinted with permission from [17]. Copyright 2016 IOP publishing.

with the same polarity and bipolar, where the set transition generally occurs at positive voltage and reset at negative voltage. Figure 1.4 illustrates the typical I-V characteristic for both operations. While unipolar switching is based on the purely thermal acceleration of red-ox transitions [18], bipolar switching relies on ionic migration assisted by the temperature and the electric field [19]. During the reset, ionized defects, within the CF migrate toward the TE, which is negatively biased, thus depleting the CF in correspondence to the highest temperature region. The displaced defects are re-injected into the depletion region in the subsequent set operation, conserving the total number of defects [17, 19]. For this reason, bipolar devices usually show better endurance properties and uniformity than unipolar ones [20]. In chapter 2, a deeper explanation of the switching mechanism in bipolar RRAMs will be addressed, focusing in particular on the difference between the graduality of set and reset transitions, shown in figure 1.4 (b).

Note that the type of defects which constitutes the CF can be different, but generally they are divided into two families, which represent two different types of devices:

- *OxRAM* where the metal oxide and top electrode are made of transition metals. In this case oxygen vacancies are introduced by the TE in the dielectric and move towards the BE, thus creating the CF.
- *CBRAM* (conductive bridge RAM) where TE is made of active metals (like Cu or Ag), whose high-mobility cations migrate under the electric field leading to the generation of CF.

The main difference between these two types of RRAMs is the resistive window, as CBRAMs display a factor of about 10^4, while OxRAMs show 10^2 [17]. Despite this, both devices show very similar electrical properties [21,22], suggesting a common classification in advantages and drawbacks of such memories. The promising features are excellent scalability [23], high switching speed ($\sim 10ns$), low current operation, excellent endurance ($> 10^7$), and good CMOS compatibility [8]. However, RRAM suffers from severe reliability issues, such as resistance fluctuations and switching variability, which will be addressed in chapter 2.

1.2.3 Spin transfer torque magnetoresisistive RAM (STT-MRAM)

STT-MRAM is based on the magnetic tunnel junction (MTJ), which is a device consisting of two ferromagnetic thin films separated by a tunnel oxide barrier as shown in figure 1.5 (a), (b). The magnetization direction of one ferromagnetic layer, called pinned layer (PL), is fixed, while the magnetization direction of the other ferromagnetic layer, called free layer (FL), can freely rotate.

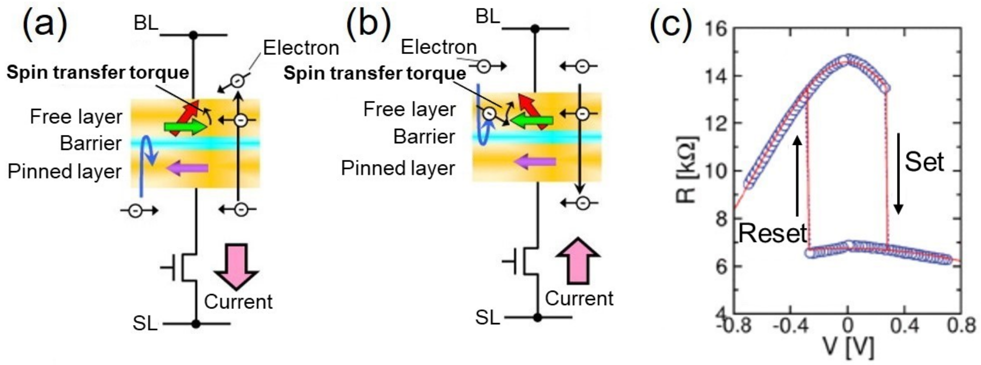


Figure 1.5: (a) Set transition in STT-MRAM. Only electrons with spin parallel to the PL magnetization tunnel through the barrier and reach the FL triggering the magnetization inversion by STT. (b) Reset transition. The electrons whose spin is anti-parallel to the FL magnetization, are reflected at the PL-barrier interface and are injected back to the FL, inducing the transition to HRS. Reprinted with permission from [24]. Copyright 2012 Elsevier. (c) Typical R-V characteristic of a STT-MRAM. Noteworthy features are the abrupt transition of both set and reset and the limited resitive window. Reprinted with permission from [25]. Copyright 2019 IEEE.

The resistance depends on the relative orientation of magnetization in the two ferromagnetic layers. In case of parallel orientation (P), the resistance is low (LRS), while it is high (HRS) in the case of anti-parallel (AP) configuration. In set transition (AP to P, figure 1.5 (a)), electrons flow from the pinned layer to the free layer. As electrons pass through the pinned layer, only the ones with the same spin direction corresponding to the pinned layer magnetization are able to tunnel, whereas the others are reflected back towards the electrode. This spin-polarized current exerts STT on the magnetization of the free layer, and when the amount of spin-polarized current exceeds the threshold value, the magnetization of the free layer is switched. On the contrary, in the reset transition (P to AP, figure 1.5 (b)) polarity is reversed and current flows from the FL to the PL. In this case, the electrons reflected at the PL barrier interface, i.e. the electrons whose spin is opposite to the pinned layer magnetization, are re-injected into the FL, triggering the FL magnetization switching [24].

The great advantages of STT-MRAM are the fast read/write access time [26], and the extremely high cycling endurance ($> 10^{15}$), due to the fact that no atom moves during the set and reset operations, and only a breakdown in the tunnel oxide barrier can limit it. However, unlike PCM and RRAM, the use of STT-MRAM in neuromorphic applications is less immediate, since they are less suitable to analogue programming and multilevel operation. Indeed, the set and reset transitions are typically binary and the ratio between HRS and LRS is small in STT-MRAM, as shown in figure 1.5 (c) [25].

1.2.4 Ferroelectric RAM (FeRAM)

Ferroelectric random access memory (FeRAM) is a nonvolatile memory, which relies on the polarization switching in a ferroelectric (FE) material, such as a perovskite material (PZT, $PbZrTiO_3$, or SBT, $SrBi_2Ta_2O_9$) [27] or doped-HfO$_2$ [28]. The typical structure consists of a MIM stack, where the insulator layer is made with a ferroelectric material, as shown in figure 1.6 (a).

The application of an external bias determines the orientation of the electric dipoles within the FE material, as shown in the polarization-voltage (P-V) characteristic of figure 1.6 (b), where the polarization is the electric dipole moment per unit volume. In particular, when the external bias is brought back to 0, the FE material exhibits a residual polarization (P_r), whose polarity is the same as the one of the external bias applied. On the other hand, the minimum voltage needed to achieve a polarization switching is the coercive voltage (V_c), as shown in figure 1.6 (b).

Note that FE switching does not impact on the MIM resistance, as the latter is not sensitive to the FE polarization itself, meaning that FeRAMs cannot be used as resistive memories. On the other hand, the displacement

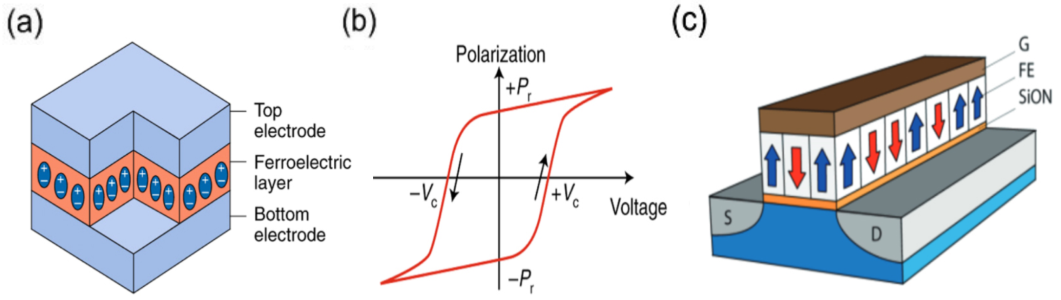


Figure 1.6: (a) Illustrative sketch of a FeRAM MIM structure (b) Polarization-voltage characteristic, typical of FE materials, where are highlighted the residual polarization P_r, i.e. P when external bias is brought back to 0, and the coercive voltage V_c, i.e. the minimum voltage needed to invert polarization polarity. (c) Ferroelectric field-effect transistor (FeFET), where the FE polarization of the FE dielectric layer dictates the threshold voltage, thus serving as a nonvolatile memory and synaptic weight element. Adapted with permission from [29]. Copyright 2019 IOP Publishing.

current induced by the polarization switching can be sensed externally to probe the FE state, thus providing for the read operation [27]. However, this operation is destructive of the pre-existing state, since it is necessary to switch the polarization in order to read the information stored. For this reason, the readout operation is expensive in terms of time and energy.

The advantage of FeRAM is that they can achieve high-speed read/write operations ($\sim 10ns$) comparable to that of DRAM, without losing data when the power is turned off [30]. On the other hand, the aforementioned issue related to the readout operation, along with the fact that the resistance is not changed by the polarization switching, make FeRAM unsuitable for application in the field of neuromorphic computing [11]. However, it is possible to gain a resistance change by FE switching, thus enabling the use of a FE-based device as a synaptic element, by adopting a different structure, namely a ferroelectric field-effect transistor (FeFET) structure [31]. This is a MOS transistor where the gate dielectric is a FE layer, as shown in figure 1.6 (c). By controlling the polarization state through the gate voltage, it is possible to affect the threshold voltage, thus determining the channel resistance and providing a non-destructive read methodology [32].

1.3 Crosspoint array and selectors

Excluding FeFETs, all emerging memories described in section 1.2 are two-terminal devices. Such important feature, along with the possibility to fabricate those devices in the back end of the line, allows to organize them in a more compact architecture known as crosspoint array. It consists of a matrix of vertical and horizontal metal lines on two different planes, whose intersection points are the location for the memory devices. These are fabricated vertically and link the two metal lines, obtaining a random access array, as shown in figure 1.7 (a) . The benefit from the integration density point of view is evident, since the space required for each device is $4F^2$ (where F is the minimum litographic feature size) and in principle, no additional space is needed for the structure itself. The same density could not be achieved with any other architecture involving a MOSFET. Moreover, an extension to the third dimension is easy to obtain by simply vertically stacking more crosspoint arrays. In this case, the density would further increase, as the effective cell dimension would be $\frac{4F^2}{N}$ where N is the number of levels introduced [3].

In particular, such matrix structure is very promising for the implementation of neuromorphic applications, since the synaptic weights of a neural networks can be easily organized by a matrix configuration that can be then directly implemented in a crosspoint array [33]. A further discussion on the subject is addressed in the following section.

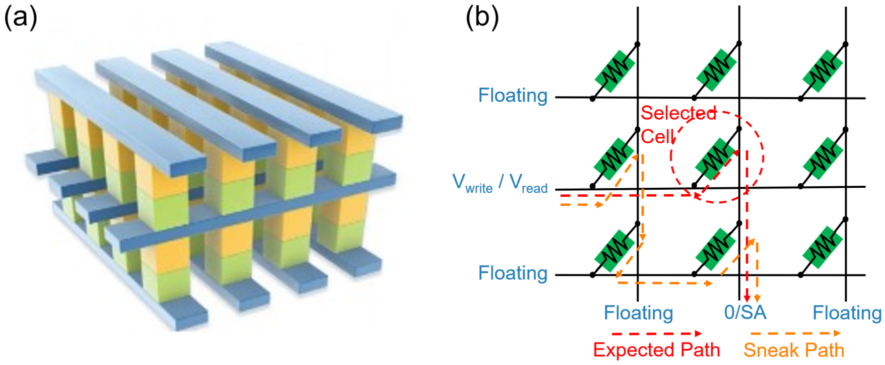


Figure 1.7: (a) Crosspoint structure. The possibility of 3D stacking in order to increase the integration density is highlighted, since two layers are represented. The 1S1R structure is underlined by the two different colors used, representing respectively the selector and the memory. (b) Schematic representation of a crosspoint array, highlighting the sneak current paths which are unavoidable if a selector is not inserted. Adapted with permission from [35] and [36]. Copyright 2017 IEEE.

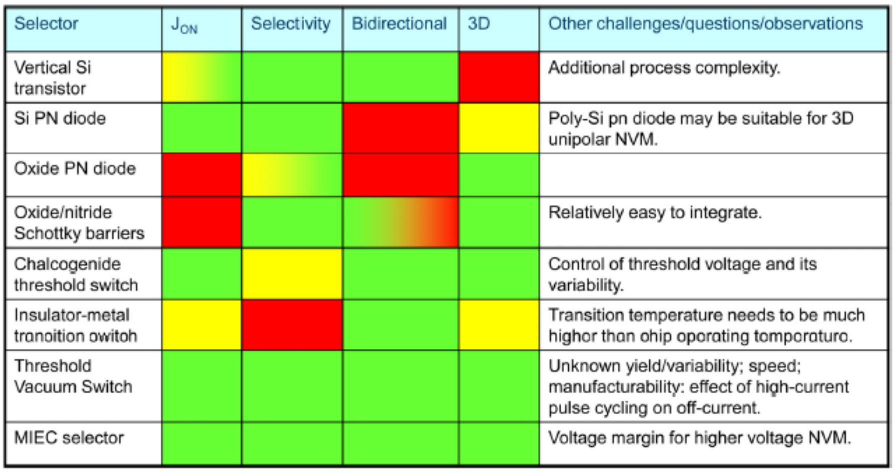
However, the crosspoint structure presents some drawbacks. Any read/write operation is achieved through the application of a set of voltages at the edge of the array, in order to select a specific cell. Unfortunately, in a NxN array, for a single selected cell, there are $2(N-1)$ *half-selected* cells (belonging to the same row or column as the selected one), and $(N-1)$x$(N-1)$ *un-selected* cells (all the others, not belonging to the selected row or column). Instead, one would like to access the selected device at will, leaving all other cells completely unperturbed, avoiding any additional power dissipation [34].

Such operation is impossible to achieve, because current sneak paths will form (figure 1.7 (b)), causing misreading on the selected cell and miswriting on the un-selected cells [36]. A possible solution to avoid such perturbations is using different biasing schemes [37, 38] which limit the voltage drop on the un-selected cells by applying a certain voltage to the corresponding lines instead of leaving them floating.

However, the application of such biasing schemes is not very efficient when the resistance of the metal lines is not negligible compared with the resistance of the memory device, which is the case for advanced technology nodes, where wires get thinner, increasing line resistance [34]. Moreover, biasing the un-selected lines significantly increases power consumption. These are the key reasons why the introduction of a *selector*, i.e. a two-terminal device with a strong non-linear characteristic, comparable to the one of a diode, is necessary in any crosspoint array.

A good selector must have an off-state resistance which is higher than the memory HRS, in order to limit the leakage current flowing in the serial structure when the cell is un-selected, while the on-state resistance must be smaller than the memory LRS, so that most of the voltage applied to the serial structure drops on the memory when the cell is selected. This is equivalent to say that the on-state current density must be very high not to limit the current required to the memory for set or reset operations. Moreover, an important feature for selectors is endurance. It must be much higher than the one of the memory cell, since selectors need to switch from off-state to on-state at every read operation, unlike memories which switch only when they are written. They also need to show bipolar characteristic and be compatible to integration in the back end of the line [34].

Many different kinds of selectors have been studied and proposed in the last years. They include silicon-based devices like vertical transistors [39] or NPN diodes [40], oxide diodes based on semiconducting oxide heterojunctions [41] or metal-oxide Schottky barrier [42], threshold switching devices like Ovonic theshold switching (OTS) [43], metal-insulator transition (MIT) [44]



Selector	J_{ON}	Selectivity	Bidirectional	3D	Other challenges/questions/observations
Vertical Si transistor	🟨🟩	🟩	🟩	🟥	Additional process complexity.
Si PN diode	🟩	🟩	🟥	🟨	Poly-Si pn diode may be suitable for 3D unipolar NVM.
Oxide PN diode	🟥	🟨🟩	🟥	🟩	
Oxide/nitride Schottky barriers	🟥	🟩	🟩🟥	🟩	Relatively easy to integrate.
Chalcogenide threshold switch	🟩	🟨	🟩	🟩	Control of threshold voltage and its variability.
Insulator-metal transition switch	🟨	🟥	🟩	🟨	Transition temperature needs to be much higher than chip operating temperature.
Threshold Vacuum Switch	🟩	🟩	🟩	🟩	Unknown yield/variability; speed; manufacturability: effect of high-current pulse cycling on off-current.
MIEC selector	🟩	🟩	🟩	🟩	Voltage margin for higher voltage NVM.

Figure 1.8: Summary table of selector device types. 4 different categories are compared, namely on-state current density (J_{ON}), selectivity, i.e. ratio between on-state and off-state resistance, bidirectionality, i.e. the capability of operate in both polarities, and possibility of vertical stacking. For J_{ON}, green indicates values $\sim 10MA/cm^2$, yellow $\sim 1MA/cm^2$ and red $< 1MA/cm^2$. For selectivity, instead, green means ON-OFF ratio $> 10^6$, yellow $> 10^4$ and red $< 10^4$. For 3-D integration, green indicates full 400 °C BEOL compatibility, while yellow is related to higher temperature processes. Adapted with permission from [34]. Copyright 2014 American Vacuum Society.

and threshold vacuum switching (TVS) [45] and Copper containing mixed-ionic-electronic-conduction (MIEC) devices [46].

Figure 1.8 summarizes the different families of selectors, highlighting the performances for each type. Excluding TVS and MIEC selectors, whose knowledge is still too preliminary, OTS devices are surely the most promising for integration in crosspoint arrays due to the larger familiarity about the materials involved, mostly chalcogenides as in PCMs, the large on-state current density, the bidirectionality and most of all the great compatibility with back end of the line processes, which allows 3D stacking [47].

1.4 Resistive switching devices for neuromorphic computing

1.4.1 Neural networks

As introduced in section 1.1, neuromorphic computing aims at developing circuits that replicate the operation of the human brain. An essential feature of any neuromorphic circuit is the neural network architecture, where data are sent by neuronal terminals through a highly parallel net of synaptic paths [29]. Neural networks can be divided into two fundamental classes referred to as *artificial neural networks* (ANNs) and *spiking neural networks* (SNNs) [48], respectively. The main difference between the two types is the training methodology. ANNs, which are also called deep neural networks (DNNs) due to many hidden layers used to process information, typically adopt supervised learning tecniques such as the backpropagation, where the error between effective output and the desired response called label is iteratively minimized during training process [12, 49]. On the other hand, SNNs aim at replicating brain functionalities implementing brain-inspired Hebbian-type learning schemes widely investigated in biological experiments such as the spike-timing dependent plasticity (STDP) [50, 51] and spike-rate dependent plasticity (SRDP) [52].

Most of the success that has been recently achieved by neural networks in fundamental machine learning tasks such as face recognition, speech recognition and image classification has been mainly due to the implementation of ANNs. For these reasons, the investigation will focus only on this subject in the following.

A schematic representation of a deep neural network is illustrated in figure 1.9. The structure is composed by an input layer of neurons, providing the raw data which have to be processed by the network, one or more hidden layers, where the intermediate solutions are computed, and an output layer, which

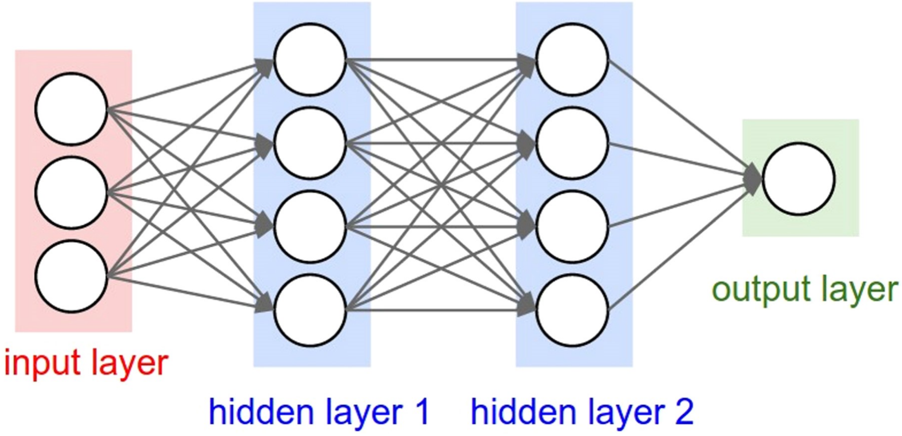


Figure 1.9: Fundamental neural network structure, composed by an input layer, one or more hidden layers and the output layer. Neurons receive signals from previous layers (or raw data for input layer) and compute a summation or integration. The result is transmitted towards the following layer via synaptic connections which multiply the signal by a proper synaptic weight. Adapted with permission from [29]. Copyright 2019 IOP Publishing.

provides the final solution to the problem [29]. The neurons are responsible for implementing a summation or integration of all the incoming signals and applying a non-linear operator to it, providing the result as an output to be transmitted to the following layer. On the other hand, synapses represent the weight by which any signal is multiplied in the connection between two neurons.

Such structure, likewise the human brain, is composed of much more synapses than neurons. Therefore, the synaptic element should be extremely small and energy efficient. Such features are well fitted by the emerging memories described in section 1.2, which are therefore considered the most promising devices for artificial synapses implementation. Moreover, in order to present good performances, the number of weights used in such networks must be so large that the transferring of such values from memory to processors in classical CPUs or GPUs would be highly inefficient in terms of energy and time [53]. Instead, implementing such networks in large crosspoint arrays would allow to avoid such limitation.

In the next paragraph, the explanation of backpropagation scheme used to train ANNs will be addressed.

1.4.2 Backpropagation algorithm for neural network training

Backpropagation is the main type of supervised learning algorithm used to train ANNs to perform machine learning tasks [12, 54]. This algorithm consists of computing the gradient of an objective (or cost) function with respect to the synaptic weights, by applying the chain rule for derivatives [12, 54]. The objective function in this case is the error function, i.e. the difference between the network's output and the expected result known as data label. The key concept behind the method is that the output of each neuron can be written as a function of its inputs. Therefore, as the derivative of the objective function is computed with respect to the output of a neuron, it is possible to refer it to its inputs, using the chain rule for derivatives. Since such inputs are at the same time the ouptuts of the previous layer, this procedure can be repeated for each layer starting from the output all the way to the input layer. This approach clarifies the origin of the algorithm name.

Figure 1.10, illustrates the implementation of backpropagation rule in an ANN. The network is a multilayer perceptron (MLP) with the input layer, 2 hidden layers and the output layer, which is trained on handwritten digit images from the Modified National Institute of Standards and Technology (MNIST) dataset [54] for an image classification task. The images from training dataset are forward propagated through the network, providing an x_i value for each neuron and a classification guess y_j for the images. Such guess is represented by the output of the last layer neurons and is compared to the correct answer g_j,

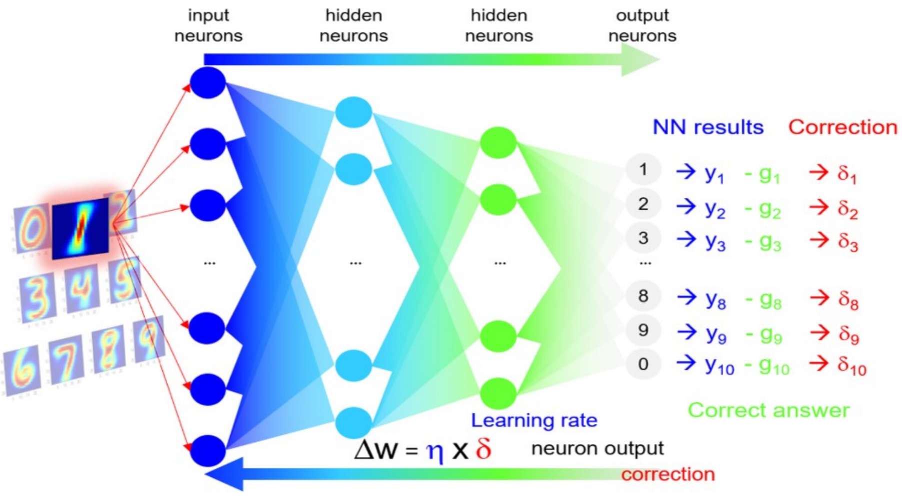

input neurons
hidden neurons
hidden neurons
output neurons
NN results
Correction
1 → y_1 - g_1 → δ_1
2 → y_2 - g_2 → δ_2
3 → y_3 - g_3 → δ_3
8 → y_8 - g_8 → δ_8
9 → y_9 - g_9 → δ_9
0 → y_{10} - g_{10} → δ_{10}
Correct answer
Learning rate
$\Delta w = \eta \times \delta$
neuron output
correction

Figure 1.10: Backpropagation algorithm execution within a multilayer fully-connected neural network. Input patterns are forward propagated, then the output results y_j are compared with the correct answers g_j. The errors $\delta_j = y_j - g_j$ are backpropagated to previous layer and used to calculate the synaptic weight update in the network based on equation 1.1. Reprinted with permission from [29]. Copyright 2019 IOP Publishing.

which is the data label. By subtracting the two quantities, an error $\delta_j = y_j - g_j$ is obtained and the error function is calculated as $C = \frac{1}{2}\sum_j^N \delta_j^2$, where N is the number of output neurons. δ_j is then backpropagated through the entire network, allowing for the calculation of the error for the neurons of any layer. Finally, the synaptic weights w_{ij} are updated according to the formula:

$$\Delta w_{ij} = \eta \cdot x_i \cdot \delta_j \tag{1.1}$$

where η is the learning rate. This procedure is then repeated for every training image, and the entire training set is iteratively presented for many training cycles, called epochs. After this procedure is completed, all the synaptic weights of the ANN are optimized on training dataset and the classification ability of the network is evaluated using unseen images from the test dataset.

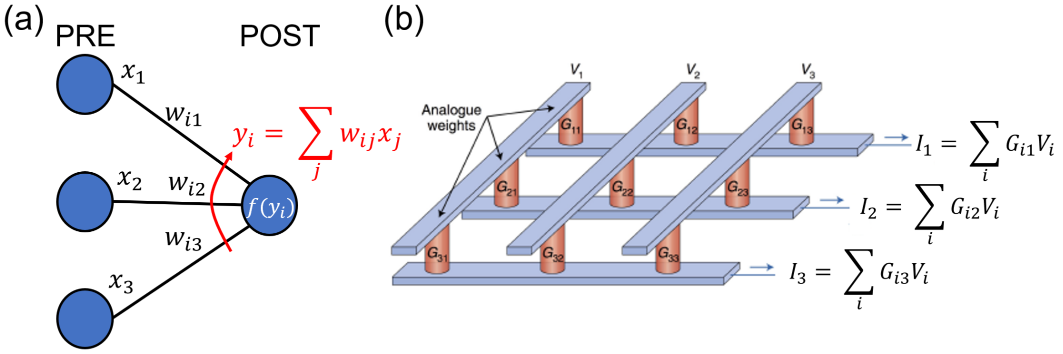


Figure 1.11: (a) Schematic representation of pre-synaptic (PRE) and post-synaptic (POST) neurons. At the POST level, all the signals coming from the PRE are multiplied by the corresponding synaptic weight and summed. (b) Physical implementation of a neural network in a 3x3 crosspoint array. The current flowing in each row represents the POST signal related to a specific neuron. It is indeed given by the weighted sum of the voltages applied to each column. Adapted with permission from [55].

This additional phase following the training process is known as inference or classification phase.

1.4.3 ANNs implementation in crosspoint arrays

As mentioned in section 1.4.1, in neural networks, neurons compute the sum or integral of all the incoming signals, while synapses represent the weight of each connection between neurons. For each synapse, it is therefore possible to define a pre-synaptic (PRE) and a post-synaptic (POST) neuron, as shown in figure 1.11 (a). The signal at the input of the POST neuron (y_i) can be written as a function of all the output signals emitted by PRE neurons (x_j), as shown in equation 1.2.

$$y_i = \sum_j w_{ij} \cdot x_j \tag{1.2}$$

where w_{ij} is the synaptic weight connecting the j-th PRE with the i-th POST.

Given the crosspoint structure introduced in section 1.3, the hardware implementation of the multiply-and-accumulate (MAC) operation via synaptic weights expressed by equation 1.2 is straightforward. Indeed, the summation can be implemented by Kirchhoff's law by summing all the currents for a single raw, while the multiplication by the synaptic weight is performed through Ohm's law, by considering the weights as the device conductance [11]. In this way, equation 1.2, can be rewritten as:

$$I_i = \sum_j G_{ij} \cdot V_j \tag{1.3}$$

The PRE signals are therefore described by column voltages V_j, the conductances G_{ij} are those related to the devices connecting the i-th row to j-th column, and the POST signal is the current flowing along the i-th row. The physical implementation of this fundamental concept for neural network applications is illustrated in figure 1.11 (b) via a 3x3 crosspoint array, where the physical implementation of the summation is highlighted.

Note that equations 1.3 and 1.2 can also be seen as a matrix vector multiplications (MVM), where V_j (x_j) represents the element of the constant term vector, G_{ij} (w_{ij}) the matrix element and I_i (y_i) the element of the resulting vector. Unlike what happens in digital computers, which are based on the von Neumann architecture, solving such operation in crosspoint arrays allows to achieve a very high parallelism and reduce time and energy consumption. Indeed, once the input voltages V_j are applied, the currents I_i, i.e. the results of the computational process, are obtained in just one clock cycle.

However, weights calculated in software during training can be either positive or negative values, while G is only positive. The solutions adopted to solve this problem are illustrated in figure 1.12. In the more compact one, shown in (a), the total value is obtained as the difference between a tunable conductance G_{ij} and a fixed reference conductance G_r, and the overall synaptic weight is thus given by $w_{ij} = G_{ij} - G_r$. The other solution, (b), implies instead two tunable

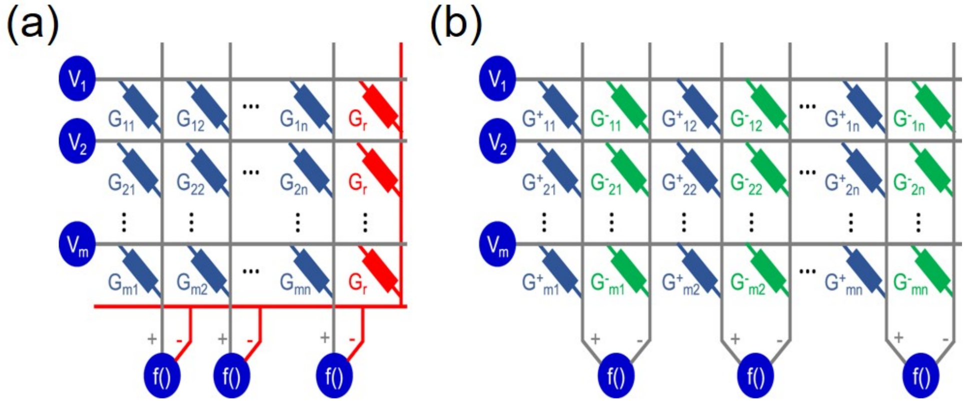


Figure 1.12: Schematic representation of a crosspoint array implementing a neuromorphic circuit. (a) Synaptic weight is represented by a differential pair consisting of an array memory G_{ij}, which is programmed at need, and a reference memory G_r, which is instead constant. (b) Synaptic weight is represented by a differential pair consisting of two memory devices G_{ij}^+ and G_{ij}^-, both programmable. Reprinted with permission from [29]. Copyright 2019 IOP Publishing.

conductance G_{ij}^+ and G_{ij}^- to implement each synaptic weight as $w_{ij} = G_{ij}^+ - G_{ij}^-$. The first method is used when the device conductance can be both increased and decreased by analog programming, while the second one is preferred when the analog tuning can be performed only in one direction [29].

As stated in section 1.4.2 an efficient and accurate backpropagation-based training process can be obtained by adjusting the weight correction Δw_{ij} to be linearly dependent on the product of x and δ, as expressed in equation 1.1. For this purpose, a high degree of linearity between voltage applied and conductance response is crucial in the weight updates of the memory devices.

Unfortunately, the conductance response of any of the aforementioned non-volatile memory do not naturally exhibit the required linearity. Moreover, other typical resistive switching device non-idealities, such as programming variability, stochasticity and asymmetry between increasing/decreasing responses may strongly impact the network performances [56]. For these reasons, it is very difficult to obtain a fully analog behavior, where ideally, infinite conductance states are possible. However, thanks to some advanced techniques, it is possible to program the device conductance in more than two states (HRS and LRS), thus obtaining multilevel operation. Indeed, increasing the number of possible states strongly improves the network performance.

The problem related to multilevel programming is intrinsic in its definition. Since the resistive window, i.e. the ratio between HRS and LRS, for a particular device is given, adding intermediate states decreases the relative distance in conductance among them. Therefore, issues such as programming variability (especially for RRAMs) and drift (in PCMs) have much more impact, as they blur the distinction in conductance levels, thus limiting the bit precision [33]. A method which is used to limit variability and consequently increase linearity is the program/verify algorithm [57]. It is a scheme which alternates the application of programming and reading pulses; during the latter ones, which have much smaller amplitude in order to keep the device in its state, the current conductance level is measured and compared to a target value. The algorithm stops delivering programming pulses as soon as the measured conductance is larger than the target. Such closed loop system ensures a very high programming accuracy [58], at the cost of the degradation of speed performance. It is therefore very useful in applications with offline training, where it is fundamental to precisely write the weights calculated in software into the synaptic devices.

In conclusion, ANNs hardware implementation in crosspoint architectures is a very promising solution to improve energy efficiency, by avoiding the von Neumann bottleneck. Indeed, the crucial computational step for ANNs is the MVM (eq. 1.2), which can be performed very efficiently in terms of energy and time in crosspoint arrays, exploiting the physical properties of the emerging memory element. However, such devices show some drawbacks in stability, variability and linearity, which can degrade the network performances. In the case of RRAM arrays, programming variability is the main issue, but it can be limited through program/verify algorithms. Chapters 2, 3, 4 will be focused on such topics.

Chapter 2

Resistive Random Access Memory (RRAM)

This chapter presents a detailed description of resistive random access memories. After a brief introduction, the switching mechanism is addressed, introducing an analytical model able to describe the formation and disruption of the conductive filament during set and reset transitions. In the second part of the chapter, the physical mechanisms causing cycle-to-cycle variability are investigated, focusing in particular on the dependence of the standard deviation on the device resistance.

2.1 Introduction

Resistive switching memory (RRAM) devices are among the most promising candidates for next generation memory, thanks to the low power and high speed operation, high cycling endurance, and low fabrication costs [17]. Furthermore, the scaling perspectives are very encouraging, since the two-terminal structure and the compatibility with back end of the line (BEOL) process allows an efficient 3D integration in crosspoint arrays [41, 59], enabling the possibility to implement neuromorphic computing systems that can overcome the von Neumann bottleneck [8].

On the other hand, RRAMs present some drawbacks, namely programming variability [60], noise and states fluctuations [61–63], which can affect device reliability. Among these, variability is surely the most problematic, because it may prevent the multi-level operation of such devices, which is instead a promising solution for neural network implementation in emerging memory arrays.

In this chapter, first the switching mechanism in RRAMs is described and, then, the variability issue is addressed, focusing on the microscopic mechanism that may explain it.

2.2 Switching mechanism

RRAMs operation relies on the creation and disruption of a conductive filament (CF) inside a dielectric material, usually a metal oxide, which is interposed between two metallic electrodes forming a metal-insulator-metal (MIM) structure. As introduced in section 1.2.2, two operations are possible in RRAMs: unipolar and bipolar. The latter, where set and reset transitions occur under different polarity, is the preferred one due to the higher endurance and uniformity [20]. For these reasons, such mechanism, based on the migration of defects inside the dielectric, will be addressed in the following. Bipolar RRAMs can be further classified into oxide-based RRAM (OxRAM) and conductive bridge RAM (CBRAM), depending on the type of migrating defects: oxygen vacancies for the former ones and cations supplied by Ag or Cu-based metallic cap at the top electrode for the latter ones. Since the study presented in following chapters is on devices based on HfO, which is a typical material used to realize OxRAMs, the investigation in this chapter is limited on such family.

In OxRAMs the dielectric switching layer consists of a transition metal oxide such as HfO_x, TiO_x and TaO_x. The top electrode is often made of a reactive metal, so that some defects are introduced during deposition [17]. This leads to the facilitation of electroforming operation, i.e. the first time the filament is formed starting from the pristine state, and the determination of switching polarity, e.g. set at positive voltage, reset at negative voltage. The distinctive feature of bipolar OxRAMs is that the total number of oxygen vacancies is conserved during set and reset transitions, as they only migrate back and forth from one electrode to the depleted gap to form and disrupt the filament [17]. The electroforming operation is therefore very important, since it creates a locally degraded region, through a soft electrical breakdown of the oxide, and determines the total concentration of defects [64].

Even though defect migration is the common mechanism in the CF formation and disruption, the way it is triggered presents substantial discrepancies between set and reset transitions. For this reason, the reduction in resistance during set is associated to the increase of the cross-sectional area in a continuous CF, whereas the resistance increase during reset corresponds to the increase of a gap length in an interrupted CF [65]. Figure 2.1 (a) schematically illustrates such difference, as ϕ denotes the CF diameter and Δ the gap length. The difference between set and reset dynamics is observed also in the typical IV characteristic shown in figure 2.1 (b), obtained by successive set and reset operations on the

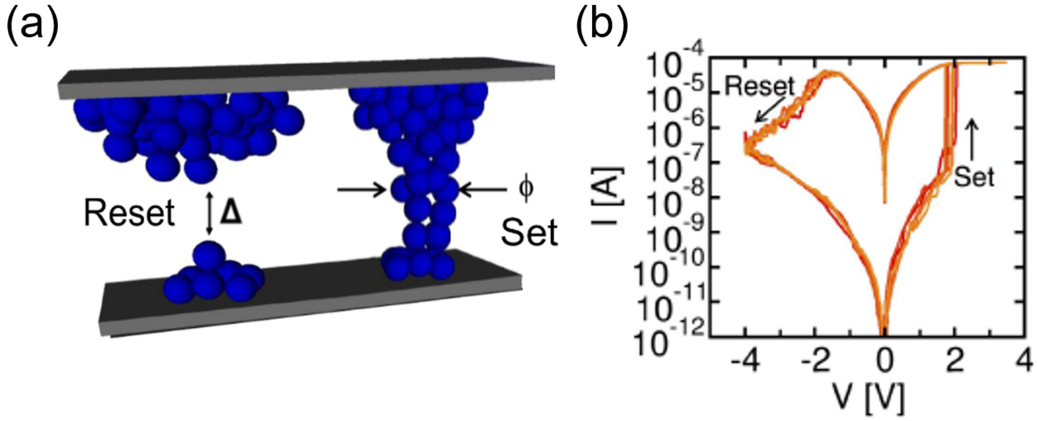


Figure 2.1: (a) Schematic representation of reset and set mechanism. The former relies on the increase of gap length Δ which is formed in the switching layer. The latter is related to the increase of ϕ, filament diameter. (b) Typical IV curve in bipolar OxRAM memory. Set transition is very abrupt, while reset is more gradual. Adapted with permission from [55] and [17]. Copyright 2016 IOP Publishing.

device. Indeed, set transition is very abrupt, whereas reset transition is more gradual.

A qualitative explanation for such different dynamics is linked to the triggering of positive or negative feedback of field, temperature and defect distribution along the CF [19, 66]. In fact, defects migrate in response to the large electric field across the depleted gap and large temperatures at the defect reservoir during set transition. As defect migration starts to take place, the depleted gap length decreases, thus the local electric field and temperature increase, which further accelerates defect migration. Such positive feedback effect would result in a destructive failure of the device; however, current limitation (compliance) systems introduce an external negative feedback which allows to reduce the voltage across the device during set transition, thus preventing destructive breakdown and enabling a detailed control of the final CF diameter ϕ and resistance [67].

On the other hand, defect migration during reset transition is triggered by a relatively low electric field across the CF, since it is continuous in the initial state. As the depleted gap Δ starts to form, the electric field decreases in the CF regions where defects are located, thus decreasing the temperature and slowing down the migration kinetics. As a result of such negative feedback effect, the voltage must be increased to further sustain the reset transition, resulting in the gradual increase of resistance [64].

Different models [19,66], both analytical and numerical, have been proposed to describe the switching mechanism previously mentioned. The approach generally adopted is to describe separately the evolution of ϕ during set and Δ during reset.

2.2.1 Set modeling

Figure 2.2 (a) shows the CF schematic evolution when an increasing positive voltage ramp V_A is applied to the device. Moving from left to right in the figure, the starting point is the reset state characterized by the depletion gap Δ. The vacancies migrate from the top stub (edge z_2) towards the gap, thus forming the filament. As V_A keeps increasing, the CF diameter increases following the rate equation 2.1.

$$\frac{d\phi}{dt} = Ae^{-\frac{E_A}{kT(z_2)}} \tag{2.1}$$

where E_A is the energy barrier for ion migration, k is the Boltzmann constant and the temperature is calculated at the injecting edge z_2. The voltage applied to the device has an impact on the barrier, as it is reduced according to equation 2.2

$$E_A = E_{A0} - \alpha qV \tag{2.2}$$

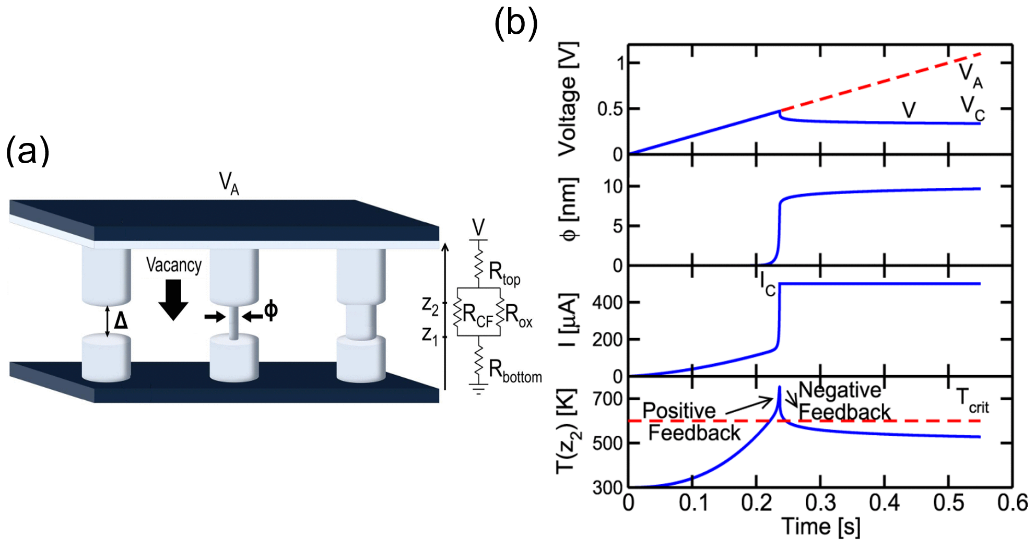


Figure 2.2: (a) Schematic representation of the CF evolution during set operation. Starting from the reset state, where the filament is interrupted by gap Δ, as V_A is increased, first the filament is created and then it rapidly grows in size, providing an highly conductive path represented by R_{CF}. The equivalent circuit is shown on the right. (b) Simulated trends of voltage V across the device, filament diameter ϕ, current I and temperature T as a function of time. The positive feedback responsible for the abrupt transition is evident, as well as the negative feedback externally forced from the compliance current. Reprinted with permission from [66]. Copyright 2014 IEEE.

where α is the barrier lowering factor and E_{A0} is the barrier at zero field. Moreover, temperature is obtained by solving the one-dimensional Fourier steady state equation 2.3

$$k_{th}\frac{d^2T}{dz^2} + J^2\rho_{CF} = 0 \tag{2.3}$$

where k_{th} is the thermal conductivity, J is the current density and ρ_{CF} is the filament resistivity. The circuit at the right end of figure 2.2 (a) shows the electrical model associated to the device during set, where R_{CF} is related to ϕ by $R_{CF} = \frac{4\rho_{CF}L}{\pi\phi^2}$, indicating the CF resistance in the set state.

Figure 2.2 (b) shows the simulated trends of the voltage V across the device, filament diameter ϕ, current I and temperature at the injecting edge $T(z_2)$ as a function of time, calculated by integrating equation 2.1. Initially ϕ is equal to zero. As V_A rises, it triggers the conduction in the gap, which results in a strong increase in temperature, and in field across the gap. As soon as $T(z_2)$ overcomes the critical temperature for defect migration T_{crit}, ϕ is formed and rapidly increases as the positive feedback is activated. In fact, the larger the filament, the smaller R_{CF} and the larger J and $T(z_2)$ with it, which further enhances the defect migration. As soon as $I = I_c$, ϕ stops increasing and therefore V (voltage across the device) must stabilize at the value $V_c = RI_c$, as shown in the top plot in figure 2.2 (b). This explains the inverse proportionality between LRS and compliance current, which is a property exploited to obtain multilevel operation in RRAMs [68]. Note that the external negative feedback introduced by I_c is visible also in the temperature trend, which rapidly decreases under T_{crit}, as shown in the plot at the bottom of figure 2.2 (b).

2.2.2 Reset modeling

Figure 2.3 (a) schematically illustrates the filament evolution during reset, when V_A is a decreasing voltage staircase. The initial state is the set state, characterized by the continuous filament shown at the left end. As V_A becomes more negative, positively charged vacancies, driven by electric field and temperature, starts to migrate towards the top electrode, thus creating two stubs, one at the top where vacancies are accumulated and one at the bottom, from where vacancies are released. Such process starts in the filament middle point, where temperature is maximum in the initial state, according to equation 2.3. The continuous decrease of V_A, reinforces the process and results in the increase of the gap length Δ. The rate of such increase is given by equation 2.4:

$$\frac{d\Delta}{dt} = Ae^{-\frac{E_A}{kT(z_1)}} \tag{2.4}$$

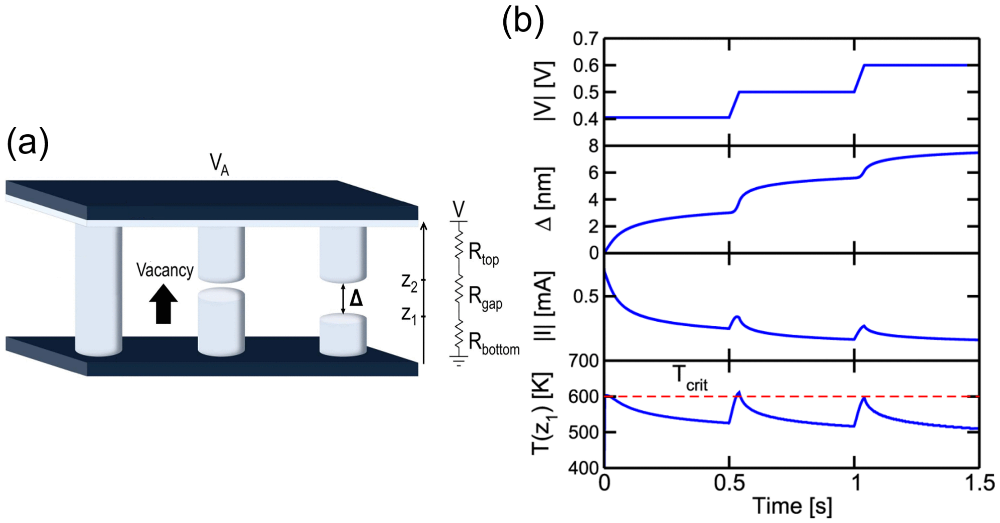


Figure 2.3: (a) Schematic representation of the gap evolution during reset operation. Starting from the set state, where the filament is continuous, as V_A is decreased, first the filament is disrupted and then a gap of length Δ is created, resulting in a high resistive path represented by R_{gap}. The equivalent circuit is shown on the right. (b) Simulated trends of voltage |V| across the device, gap length Δ, current $|I|$ and temperature T as a function of time. As soon as T overcomes the critical temperature for vacancy migration T_{crit}, the negative feedback is triggered. In fact, Δ increases, leading to current and temperature decrease, which limits the migration, resulting in the gradual decrease of $|I|$. Reprinted with permission from [66]. Copyright 2014 IEEE.

where E_A is given by equation 2.2 and T is calculated through eq. 2.3 and evaluated in z_1, i.e. at the vacancy-injecting stub edge.

Figure 2.3 (b) shows the simulated trends of voltage across the device $|V|$, gap length Δ, current $|I|$ and temperature in z_1 as a function of time during the reset transition. As $|V|$ is increased, a current flows in the filament, leading to the rise of temperature $T(z_1)$. As $T(z_1)$ overcomes the critical temperature for vacancy migration T_{crit}, defects start to migrate, thus creating the gap. However, when the high resistive gap is created, the device resistance is increased and therefore the current decreases. This results in the temperature reduction under T_{crit}, which stops vacancy migration. Such process highlights the natural negative feedback typical of reset transition, which is therefore a self-limiting mechanism. Only the further increase in $|V|$ is able to reactivate the process, enlarging Δ. The final result is therefore the gradual decrease of $|I|$ shown in figure 2.3 (b).

2.3 Switching variability

Unlike well-established Flash NAND technology, where variability mainly shows up as device-to-device (D2D), RRAM devices additionally display a cycle to cycle (C2C) statistical variability, due to their different operating mechanism [69]. This phenomenon strongly limits the possibility to operate these devices at low current, which is mandatory for operating large crosspoint arrays to avoid excessive voltage drop across the high-resistance wordlines and bitlines [70]. Moreover, given the relatively small resistance window of oxide-based RRAM, multi-level operation is strongly affected by variability, thus limiting the possibility of achieving analog programming of synaptic weights.

The main reason causing C2C variability is the fact that the number of defects involved in the formation and rupture of the CF is discrete. In fact, during each set/reset process, the CF assumes different conformations, since defects change in number and geometrical arrangement in a stochastic way, each time affecting the resistive state of the device.

Several studies have been carried out on this aspect, showing common results regarding the C2C variability dependence on typical parameters, like the compliance current I_c, the stop voltage V_{stop} and pulse duration t_{pulse} [60, 69, 71]. In particular, figure 2.4 shows the experimental results of two different works [69, 71] about the I_c dependence. Figure 2.4 (a) displays the

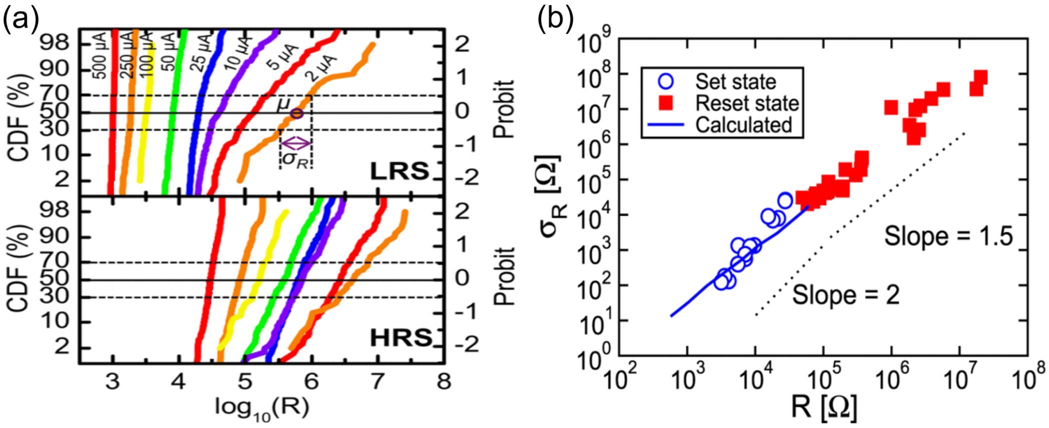


Figure 2.4: (a) C2C cumulative distributions of LRS (top) and HRS (bottom) for different compliance currents I_c. The distributions bending increases as I_c is reduced, suggesting that variability is larger when the number of injected defects is smaller, since the natural variation in defect number has more impact. Adapted with permission from [69]. Copyright 2013 IEEE. (b) Standard deviation σ_R as a function of median R. The two quantities are directly proportional in log scale, but display different slopes between set and reset states. The solid line shows a slope consistent only with reset states, since it is calculated via a statistical model which accounts for the variable number of injected defects, neglecting the geometrical shape variations of the CF. Adapted with permission from [71]. Copyright 2014 IEEE.

resistance C2C cumulative distributions for different values of I_c, both for set state (LRS, top) and reset state (HRS, bottom). As previously mentioned, the curves are centered at larger resistance values for decreasing I_c, meaning that small compliance currents correspond to small diameter of the CF, leading to larger resistances and vice versa. However, the distinctive feature of the distributions is the increasing bending as I_c decreases, suggesting that the standard deviation σ_R, whose graphical meaning is highlighted in the figure, increases as the compliance current is reduced. The distributions therefore show the increase of standard deviation σ_R with median resistance μ_R. Such behavior is observable both for the LRS and the HRS, but in the latter case the distribution tilting is less marked, possibly indicating a difference between set and reset process dynamics under low current operation [69].

These outcomes are confirmed by another study [71], whose results are shown in figure 2.4 (b). It illustrates the trend of C2C standard deviation (σ_R) as a function of median resistance R for set and reset states. As expected, σ_R increases with R, highlighting a direct proportionality in the logarithmic scale. However, it is noticeable that the slope of such dependence shows different values for set states with respect to reset states, namely about 2 for the former ones and 1.5 for the latter ones. This observation is in accordance to what highlighted in figure 2.4 (a), suggesting that this result can be considered a common feature regarding OxRAMs.

A general understanding of the aforementioned log-scale direct proportionality between σ_R and R lies in the impact of the stochastic variation of the number of discrete defects on the total number of defects injected. The smaller the total number of defects, the larger the impact, leading to more variability. A model accounting for such statistical fluctuations has been proposed to explain variability [71]. It is based on equations 2.1 and 2.4, previously introduced to explain the discrete migration of ionized defects during set and reset transitions. The approach used to address variability is introducing a statistics in E_A, so that it is possible to associate a different value of energy barrier to each defect, via the typical Monte Carlo method. The CF or gap growths, therefore, follow a sequence of discrete defect events, each characterized by a random value of E_A and a corresponding migration rate. This operation allows to describe the structural change of the HfO_x material in the gap region, due to change of the composition profile resulting from the growth of the CF during set transition and the growth of a depleted gap during reset transition.

The main result of the simulations, which were performed for different I_c values, is displayed in the solid line of figure 2.4 (b). The simulation captures the increase of σ_R with median R and is consistent with the Poisson statistics, predicting the variability on the number of localized defects in the gap after reset transition. In fact, the simulated slope equal to about 1.5 can be simply derived considering the Poisson statistics as follows. The conduction in the reset state is expected to follow the Poole-Frenkel (PF) mechanism, where the current is proportional to the density of localized states, which act as centers for thermally activated emission of carriers [72]. Assuming that injected defects all contribute to PF current, the reset-state resistance R can thus be written as:

$$R = B\frac{e^{\frac{E_C}{kT}}}{A_{CF}n_D} = Be^{\frac{E_C}{kT}}\frac{\Delta}{N_D} \tag{2.5}$$

where B is a preexponential constant, A_{CF} is the CF cross section area, E_C is the PF energy barrier controlling the activation energy for conduction in the reset state, n_D is the defect density, and N_D is the defect number in the gap region of length Δ, which controls R in the reset state. Since N_D is affected by Poisson fluctuations with spread $\sigma_{N_D} = N_D^{0.5}$, the spread of the resistance can be obtained as:

$$\sigma_R = R\frac{\sigma_{N_D}}{N_D} = RN_D^{-0.5} \propto R^{1.5} \tag{2.6}$$

However, figure 2.4 (b) shows that the simulation does not explain the larger slope shown by the experimental data in the set states. This is because the statistical model only accounts for variable number of injected defects as a result of the random E_A, while such larger slope is due to the additional contribution of random position of defects within the gap region, which is not considered in the model. In fact, in the case of low resistive states, the CF can be considered to be composed by a large number of defects, as it displays a large diameter ϕ after the set transition. In this case, the resistance variability is not dominated by a variation in the number of defects, but rather by a slight

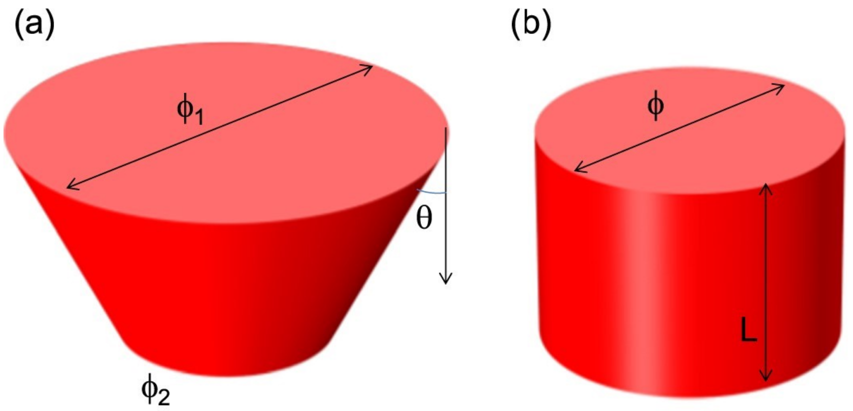


Figure 2.5: Schematic representation of the geometrical variation model for LRS resistance statistics. The variation of LRS resistance values can be estimated by assuming a variable truncated-cone geometry of the CF with a variable angle θ (a). Considering a minimum CF resistance for a cylindrical geometry ($\theta = 0$, (b)), calculations indicate that $\sigma_R \propto R^2$, in agreement with experimental results. Adapted with permission from [17]. Copyright 2016 IOP Publishing.

variation in the position of the defects affecting the shape of the CF, hence its resistance.

A simplified model which accounts for the geometrical shape variation is illustrated in figure 2.5 [17]. The CF shape can be modeled as a truncated cone as shown in figure 2.5 (a). The related resistance is given by:

$$R = \frac{4\rho L}{\pi\phi_1\phi_2} = \frac{4\rho L}{\pi(\phi + L\theta)(\phi - L\theta)} \tag{2.7}$$

where L is the effective length of the CF, ϕ_1 is the minimum diameter of the truncated cone, ϕ_2 is the maximum diameter and θ is the angle defining the inclination of the lateral cone surface. The case in which equation 2.7 is minimum is shown in figure 2.5 (b) where the truncated cone is reduced to a cylinder, as $\theta = 0$. The resistance variation can thus be estimated as the difference between the resistance of the cone-shaped CF and the minimum resistance of the cylinder-shaped CF, which results in:

$$\sigma_R = \frac{4\rho L}{\pi\phi^2}\left(\frac{1}{1-\left(\frac{L\theta}{\phi}\right)^2} - 1\right) \approx R\left(\frac{L\theta}{\phi}\right)^2 \tag{2.8}$$

where the approximation holds for $\theta << 2\phi/L$, i.e. small cone angles. Substituting $\phi^2 = 4\rho L/(\pi R)$ in equation 2.8, it becomes:

$$\sigma_R \approx \frac{\pi L\theta^2}{4\rho}R^2 \tag{2.9}$$

which provides evidence for the larger slope (~ 2) observed in the set states in figure 2.4 (b), as the standard deviation σ_R depends on the square of resistance R.

The different slopes of figure 2.4 (b) are therefore explained by the different nature of the CF at different resistance values. For reset states, the CF is depleted and variability is dominated by the defect number fluctuation controlled by Poisson statistics which determines the slope equal to 1.5. On the other hand, for set states the CF is continuous and variability is driven by geometrical shape variations, accounting for the larger slope [17].

Chapter 3

Study of RRAM variability from array-level experiments

This chapter presents the experimental variability data and the related analysis. Firstly, the experimental setup is described, specifying both the chip structure, consisting of 4 kbit arrays of HfO-based RRAMs, and the program/verify technique used to program it. Secondly, the results of an endurance experiment are shown, highlighting the particular inverse proportionality between the resulting median resistance and standard deviation, which stands out from the cycle to cycle distribution shape. Finally, a physical explanation of such results is given thanks to an analysis of set graduality. This quantity is found to be a distinctive feature of each device, being the primary source of variability.

3.1 Structure of HfO-based RRAM

The experimental data discussed in this chapter are collected from two 4 kbit arrays of HfO-based RRAM devices with one-transistor/one-resistor (1T1R) structure capable of multilevel operation. Figure 3.1 shows the microphotograph (a) and the simplified block diagram (b) of the 4 kbit array test structure [73]. This consists of four architectural blocks, namely the array of 4096 1T1R RRAM cells, a wordline (WL) address decoder (XDC MUX), a bitline (BL) address decoder (YDC MUX) and an operation control circuitry (Mode) [73]. Figure 3.1 (c) shows in detail the 1T1R structure, consisting of a NMOS transistor, manufactured in 0.25 μm BiCMOS technology (W = 1.14 μm, L = 0.24 μm), connected in series to the RRAM, which is integrated on top of metal line 2 of the CMOS process. The memory device has a metal-insulator-metal (MIM) structure consisting of a stack of area 600 x 600 nm^2 with a 150 nm TiN top and bottom electrode layers deposited by magnetron sputtering, a 7nm Ti intermediate layer and a 6 nm HfO -based layer. [57, 74].

HfO is a typical material used in RRAM devices thanks to the well consolidated know-how on deposition and control of structural properties for high-k gate dielectric applications. However, the requirements for analogue applications are different from those needed in binary memory devices, therefore it is important to deeply investigate the structural and electrical properties of such materials. This is the reason why two different processes of HfO deposition are used in such arrays, leading to different compositions, namely HfO and HfAlO. Both of them are obtained through Atomic Layer Deposition (ALD), the former at 150 °C, the latter at 300 °C by doping with $\sim$ 10 % Al content.

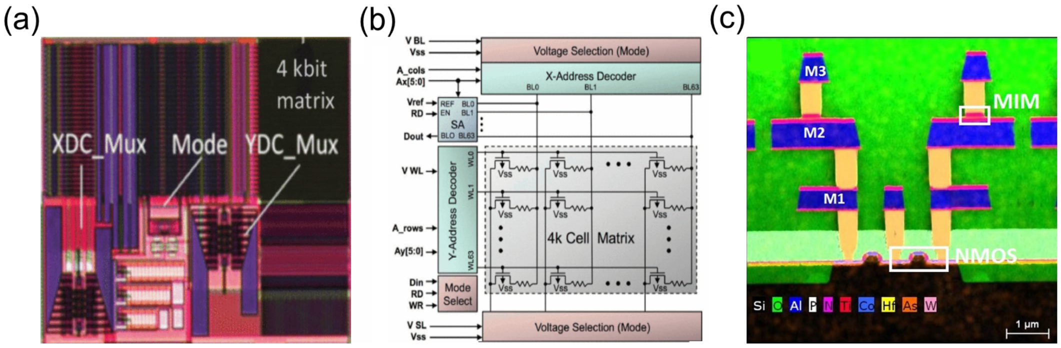


Figure 3.1: (a) Microphotograph of the 4 kbit array test structure with indication of the constituting blocks. (b) Simplified block diagram of the memory array. Reprinted with permission from [73]. Copyright 2014 IEEE. (c) STEM-EDX image of the 1T1R integrated structure. The transistor is fabricated in the front end, while the RRAM device is fabricated in the back end, on top of metal M2. The top and bottom electrodes are madee of TiN, with a Ti cap to induce oxygen scavenging between the TiN top electrode and the HfO-based layer. Adapted with permission from [74]. Copyright 2019 AIP Publishing LLC.

3.2 ISPVA Algorithm

The Incremental Step Pulse with Verify Algorithm (ISPVA) is the program/verify technique used to program the array introduced in section 3.1. It works in similar ways for both reset and set operations. In the former case, the typical waveform of program/verify algorithms, characterized by the alternation of programming (P) and read-out (V) pulses, is applied to the source side of the transistor while the drain terminal connected to the MIM resistor (top electrode) is grounded. The programming pulses have increasing amplitude from 0.5V to 2V with steps of 0.1V and fixed duration of 10 μs while the read-out (V) ones have constant amplitude 0.2V and same duration, as illustrated in figure 3.2 (a).

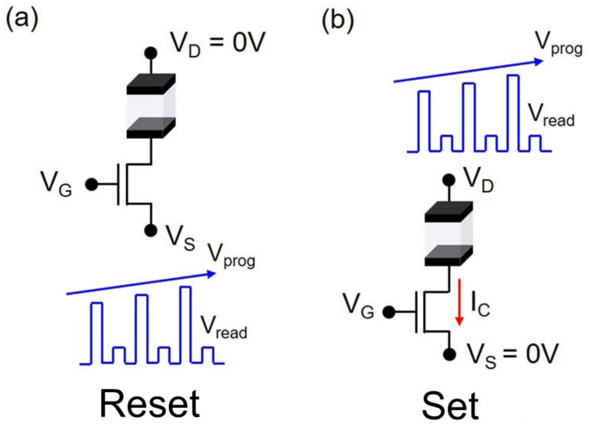


Figure 3.2: Schematic representation of (a) reset operation, used to bring back the device to the HRS, and (b) set operation, used to program the device in the 4 LRSs. Such levels, characterized by decreasing resistance, are achieved through the application of increasing compliance current I_c or gate voltage V_G. Reprinted with permission from [74]. Copyright 2019 AIP Publishing LLC.

On the other hand, for set operation the same waveform is applied at the top electrode, while the source terminal is grounded, as shown in figure 3.2 (b). The voltage applied to the gate during set operation is chosen in order to obtain the desired compliance current and LRS in correspondence to the programming pulse (P), while it is 2.7 V during the reset operation because it is necessary not to limit the current flowing in the device since the reset is a self compliant mechanism. During the read-out phase (V) instead, 1.7 V are applied to the gate in both set and reset operations in order to minimize the channel resistance in the series.

The key of a good multi-level approach in RRAM devices is in the accurate control of the multiple conductive states. The approach used in this work is to define 1 HRS and 4 LRS states. As already mentioned, one simple way to achieve multi-level operation is changing the compliance current through the gate voltage. Therefore it is necessary to associate a proper V_G value to

	R_{trg} [$k\Omega$]	V_G [V]
L_0	40	2.7
L_1	20	1
L_2	10	1.2
L_3	6.6	1.4
L_4	5	1.6

Table 3.1: Recap of R_{trg} and V_G pairs used in the ISPVA. Target is a lower limit for L_0, while it is an upper limit for $L_1 - L_4$.

each LRS. Moreover, a crucial step in every program/verify algorithm is the definition of the targets, i.e. the desired resistance values. In this work 4 $< R_{trg}, V_G >$ pairs were adopted, namely $< 20, 1 >$, $< 10, 1.2 >$, $< 6.6, 1.4 >$, $< 5, 1.6 >$ $< k\Omega, V >$, corresponding to levels $L_1 - L_4$.

On the other hand, the R_{trg} chosen for the HRS level L_0 in the reset operation is $40k\Omega$. Table 3.1 summarizes the pairs adopted for both set and reset operations.

In the ISPVA, the programming pulses (P) are applied to the device until the resistance value measured during the verify pulses (V) is smaller than the targets for $L_1 - L_4$ case, or larger for L_0 case. As a consequence, the targets represent an upper limit to the resistance values for set and a lower limit for reset. Indeed, during set process the device resistance is decreased (from HRS to LRS) by the application of the pulses until the target is overcome, while during reset it is increased (from LRS to HRS).

The crucial resistance value among the ones measured in every verify pulse (V) is the first one read after the target crossing. This is due to the fact that programming pulses (P) are not anymore applied to the device after that happens. Such values will be called *after switching* in the following.

3.3 Cycle to cycle distributions

In order to assess the variability issue, an endurance experiment is carried out, performing 1000 switching cycles, i.e. consecutive set and reset operation according to ISPVA, for a set of 1000 devices per level. The *after switching* values are extracted for every cycle and device.

Since the aim of this study is the investigation and modeling of RRAM multilevel operation, the data analyzed in the following are extracted only from the set operation part of the whole endurance experiment. The results of such experiment, carried out on the HfAlO array, are shown in figure 3.3. In (a), the cycle to cycle (C2C) cumulative distributions of *after switching* resistance are displayed: from this plot it is possible to observe how the programmed state is distributed along the cycles for every device. This means that every curve, which is related to a single device, is composed by 1000 points, each representing the resistance measured in every cycle.

First of all, it is straightforward to identify four different groups of distributions, which are associated to levels $L_1 - L_4$. The targets are highlighted by the dashed vertical lines and help to delimit the maximum acceptable range to distinguish the different levels. From the figure, it is easy to better understand that the targets are upper limits to the resistance values of every state. In fact the distributions tend to bend as they get closer to the desired value and they never cross it. This is a typical result when a program/verify algorithm is adopted: as long as the resistance measured in the verify pulses is larger than the target, a programming pulse is applied, consequently decreasing the resistance until it is lower than desired value.

The ideal case would show overlapping and vertical distributions for every level, suggesting the absence of variability among devices and cycles. The results of figure 3.3, (a) display instead the variability of measured RRAM devices. In fact, the distribution spreading is evident for all levels. This means

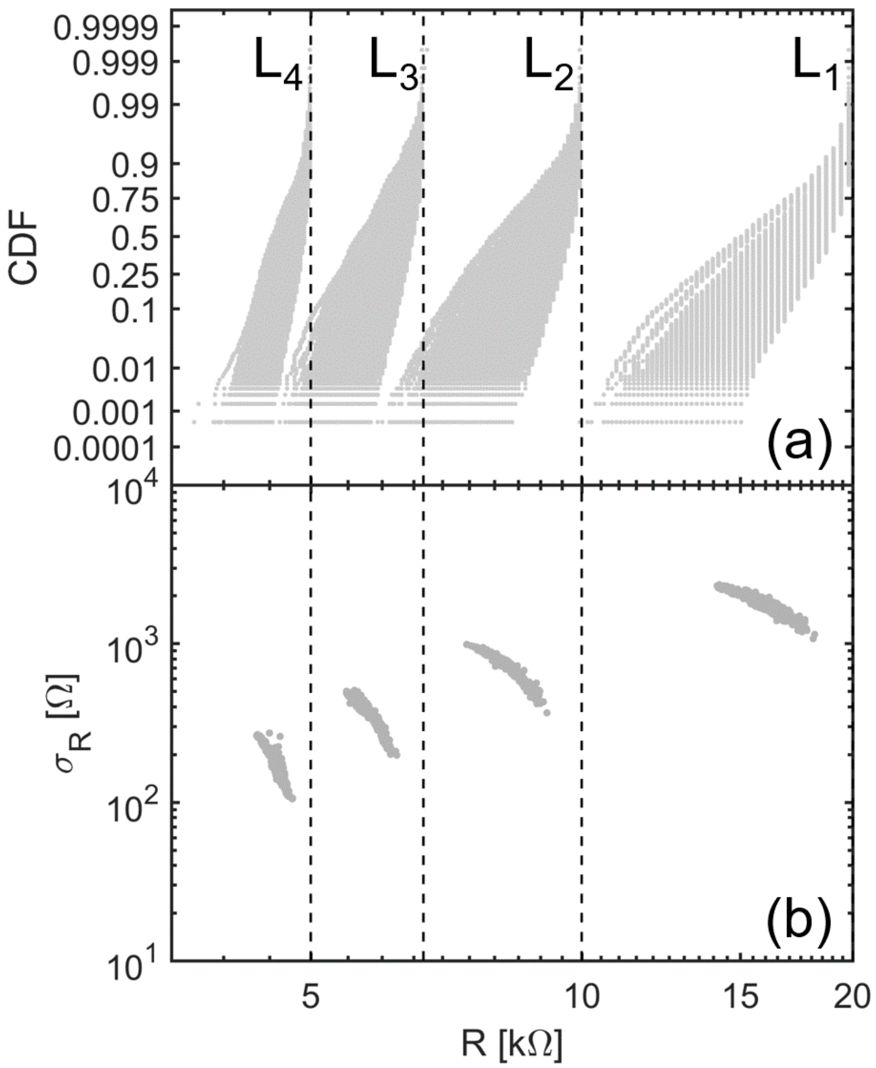


Figure 3.3: (a) Cycle to cycle (C2C) cumulative distributions of after switching resistance in HfAlO 1T1R devices. Each curve denotes one of the 1000 cells tested in the experiment; every curve is composed by 1000 points, each representing the resistance measured in every cycle. The distributions shape shows an inverse proportionality between C2C standard deviation σ_R and median μ_R, which is highlighted in (b), where the correlation between the two quantities is plotted. The comma-like shape underlines the behavior of such C2C distributions, which is found to be typical of the ISPVA.

.

that different devices are programmed on average within a different range of resistances, ranging from values very close to the targets to quite smaller ones. Moreover, it is noticeable that there are tails overlapping the lower target, both for L_1 and L_2, for example. This is unwanted since it represents the possibility that a device is programmed in a different state from the desired one. The presence, although limited, of such variability suggests that some improvements in the ISPVA might be necessary to improve the programming precision, thus allowing to increase the number of LRS levels in HfAlO RRAM cells.

The distinctive feature of all the four groups of distributions lies in the shape they acquire as a whole. It is easy to note that the more the curves have smaller values than the target, the more they are horizontally bent. This sentence can be translated in more technical terms by claiming that it is possible to find an inverse proportionality between the resistance median value (μ_R) and its standard deviation (σ_R), both calculated along cycles. σ_R represents the bending: the larger σ_R, the more horizontal the distribution. On the other hand, μ_R gives information about the average distance between measured *after switching* R and the target: the smaller μ_R, the larger such average distance. Note that μ_R can be extracted from a CDF plot like the one in figure 3.3 (a) by taking the R value in correspondence to the 50% probability level.

Such inverse proportionality is evidenced in 3.3 (b) where σ_R is plotted against μ_R. It is observable that the smaller μ_R, the larger σ_R and vice versa, which results in the comma-like shape visible in the figure. It is important to highlight that such trend is common to all the four levels. However, when data of different levels are compared, the expected trend of direct proportionality (in log-scale) between σ_R and μ_R described in chapter 2, is observed. Both considerations suggest that the comma-like shape is due to the programming algorithm rather than a specific device feature.

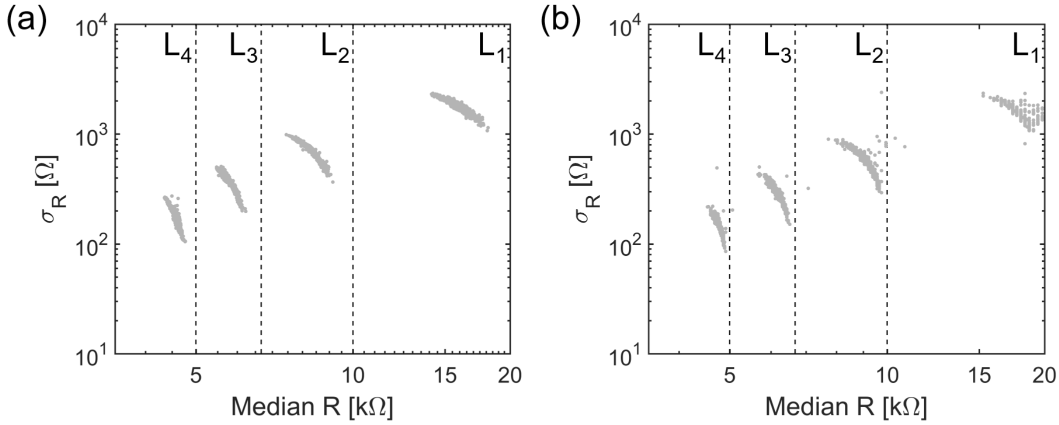


Figure 3.4: Standard deviation σ_R as a function of the median resistance μ_R for HfAlO (a) and HfO (b) RRAM cells. Since the trends are very similar, it can be stated that they depend entirely on the ISPVA.

A comparison between HfAlO and HfO RRAM cells is illustrated in figure 3.4. The comma-like shape is found also for HfO data, suggesting that such result is totally related to the program/verify algorithm. Based on this kind of analysis, it is difficult to determine which material is the best from the point of view of variability, since the data are almost overlapping. However, other works [68, 75, 76] show how Al doping in HfO_2 is able to strongly improve the device performance in terms of resistance window, switching variability and intrinsic retention. The general understanding of such improvement lies in the Al atoms ability to generate and localize the oxygen vacancies (V_O) chains, decreasing their diffusivity and consequently forming a more stable conductive filament in the switching process [76]. Accordingly to these arguments, this work will focus only on HfAlO data in the following.

3.4 Analysis of V_{set} and set graduality

In the previous section, the variability data on which this work is based were presented. Now the goal is to better investigate those data in order to understand what are the physical explanations of the results shown in figure 3.3.

Such further investigations were possible because important additional data were available. Those are the programming characteristics, i.e. the sequence of conductance G values measured for every verify pulse (V) of the ISPVA, for every cycle and device of the endurance experiment. Thanks to that, it was possible to reconstruct the conductance trend as a function of the voltage applied to the top electrode.

Two parameters were extracted from data:

- V_{set}
- Set graduality

Both quantities were then correlated with σ_R in order to investigate a possible link between them and the C2C variability of *after switching* resistance.

3.4.1 V_{set} extraction

In this analysis, V_{set} is defined as the top electrode voltage for which the conductance increase with respect to the previous value is higher than 30 μS, as can be noted in figure 3.5 (a). This method was used in order to be highly robust to noise, since it was found that especially for L_1, i.e. the level with smaller target in conductance ($G_{trg} = 50\mu S$), a simple approach based on the crossing of a single threshold was strongly affected by the conductance fluctuations in the HRS.

A typical C2C cumulative distribution of V_{set} is illustrated in figure 3.5 (b). This plot shows how the V_{set} values are distributed for a single device along the 1000 switching cycles performed during the endurance experiment. Due to the programming characteristic discrete nature, such CDF is discrete as well, meaning that possible values for V_{set} are all separated by the step increase

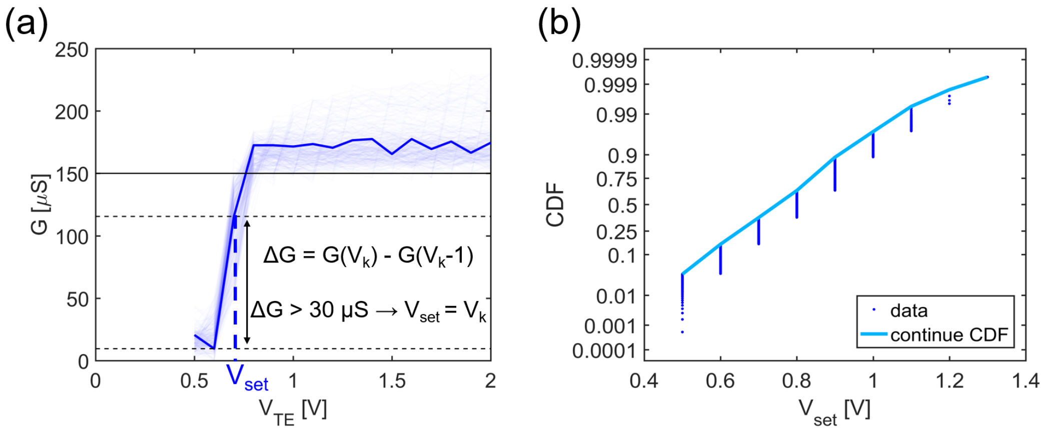


Figure 3.5: (a) Typical programming characteristics, i.e. conductance G measured during verify pulses (V) as a function of the amplitude of the last programming pulse (P) applied. The noise-robust method used to extract V_{set} distributions marks V_{set} as the top electrode voltage able to induce a conductance increase ΔG larger than 30 μS. (b) Typical V_{set} C2C cumulative distribution. Due to the programming characteristic discrete nature, such CDF is discrete as well. However, according to the CDF definition, a continue curve is extracted by connecting the upper points of each different group.

of the top electrode voltage (0.1 V). However, only the upper point for each group of cycles with same V_{set} is the relevant one, since, according to the CDF definition, it represents the probability to have cycles whose V_{set} is smaller or equal to that particular value. Therefore, in order to simplify the visualization in the following, a continue curve is extracted by linking all the upper points.

The cumulative distributions of V_{set} for all LRS levels are shown in figure 3.6 (a). Most of the curves exhibit a certain degree of overlap, thus suggesting a limited variability among devices. L_1 and L_2 show a slight distribution bending for probability larger than 99 %; this means that there is $\sim$ 1 % of the cycles in which the device experiences the resistance switch at very large voltages. Those values can be interpreted as cycles where the device either does not set properly or the V_{set} extraction method does not reject the HRS noise in the best way.

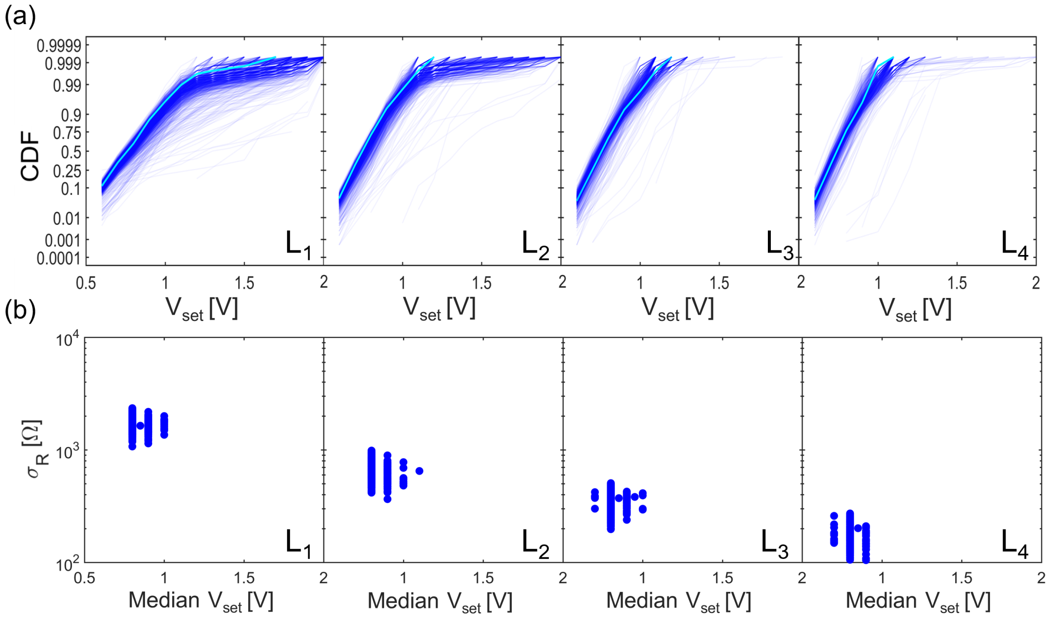


Figure 3.6: (a) C2C cumulative distributions of V_{set} for all LRS levels. The light blue curves represent the median device. The variability among devices is limited, since the distributions are mostly overlapping. (b) *After switching* σ_R as a function of median V_{set}. No correlation is found between those quantities, therefore the investigation on the nature of C2C resistance CDF introduced in section 3.3 must be extended to the set graduality.

However, due to the very limited number of such cycles in the experiment, they can be neglected in the evaluation of C2C median and standard deviation. The median device behavior is highlighted in light blue for all levels. Its parameters were found to be: $\mu_{V_{set}} = 0.75V$ and $\sigma_{V_{set}} = 0.1V$.

On the other hand, figure 3.6 (b) shows *after switching* σ_R (discussed in section 3.3) as a function of the median V_{set} for all LRS levels. No particular correlation is visible in such plots. This might be explained by what is observed in the V_{set} CDFs, where the variability among devices is limited. It is therefore impossible to describe the different behavior of devices discussed in section 3.3 by means of only V_{set}. Therefore, it is necessary to investigate not only the voltage at which the transition from HRS to LRS takes place, but also the graduality of such transition.

3.4.2 Set graduality

In order to define the set graduality a new parameter is introduced; it is called V_{trg} and represents the first value of top electrode voltage for which the conductance is larger than the target. Therefore, the two quantities V_{set} and V_{trg} describe properly the set graduality in the form of the difference between them. The higher such difference, the larger the number of programming pulses necessary to cross the target after the set event, the lower such voltage difference, the smaller such number, down to the limit of 0 difference, meaning that the set event corresponds with the target crossing.

Figure 3.7 displays the C2C median R μ_R (a) and σ_R (b) as a function of the difference of C2C median V_{trg} ($\mu_{V_{trg}}$) and median V_{set} ($\mu_{V_{set}}$) for all levels $L_1 - L_4$. Unlike the V_{set} case of figure 3.6, (b), figure 3.7 shows a relevant correlation between the quantities. In particular, an increasing trend of μ_R and a decreasing one of σ_R are evident for all levels. This observation suggests that the less gradual the set on average(small $\mu_{V_{trg}}$ - $\mu_{V_{set}}$), the smaller μ_R and the larger σ_R, while the more gradual the set (large $\mu_{V_{trg}}$ - $\mu_{V_{set}}$), the larger μ_R and the smaller σ_R. The decrease of one quantity, related to the increase of the other, is an important clue which confirms the inverse proportionality between μ_R and σ_R already found in section 3.3.

A qualitative explanation to the results is the following: the devices exhibiting a more abrupt programming characteristic, are less controllable since they get closer to the target with higher derivative in the characteristic. Therefore, they are more likely to experience larger conductance jumps in the first pulses after the set, which brings them to overcome the target with larger gaps and stabilize at largest median conductance, or, equally, at lowest μ_R in every

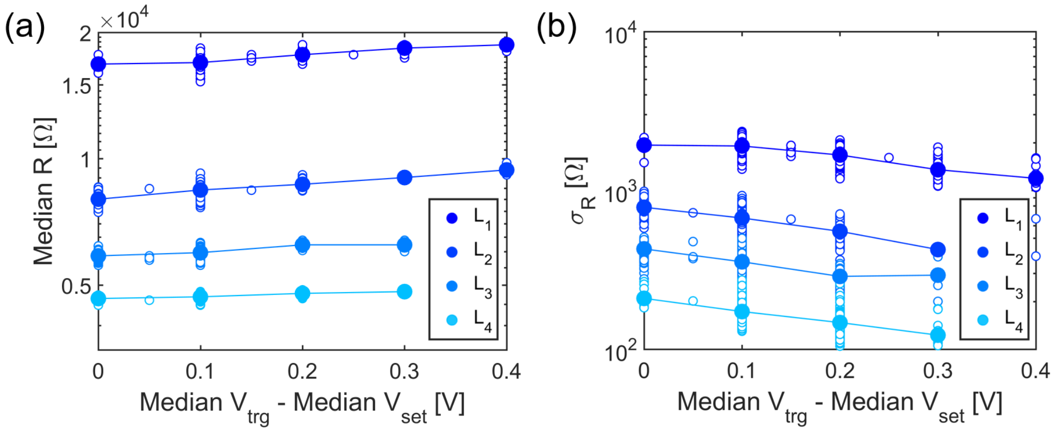


Figure 3.7: (a) C2C median R μ_R as a function of the difference of median V_{trg} and median V_{set}. μ_R increases with $\mu_{V_{trg}}$ - $\mu_{V_{set}}$, suggesting that devices with more abrupt programming characteristic reach on average smaller resistances and vice versa. (b) C2C standard deviation as a function of the difference of median V_{trg} and median V_{set}. In this case instead, σ_R decreases with $\mu_{V_{trg}}$ - $\mu_{V_{set}}$. The more abrupt devices show larger variability along cycles because of the larger derivative with which the programming characteristic crosses the target and vice versa.

level. Consequently, those devices will show a larger range of *after switching* resistance and their σ_R will be the largest.

On the other hand, more gradual devices are more controllable because they cross the target with a smaller derivative. In other words, the conductance jumps in correspondence to the crossing are smaller in this case and therefore the devices will stabilize at smallest conductance, very close to the target level. Consequently, their μ_R is the largest and σ_R is the smallest.

Figure 3.8 (a) shows an important confirm to the proposed physical explanation. Among all the C2C cumulative distributions for all levels, different devices with decreasing graduality are highlighted, going from red to green to blue, representing devices having $\mu_{V_{trg}}$ - $\mu_{V_{set}}$ respectively equal to 0, 0.1, 0.2 for $L_2 - L_4$ and 0.1, 0.2, 0.3 for L_1. As expected, the most abrupt devices (red

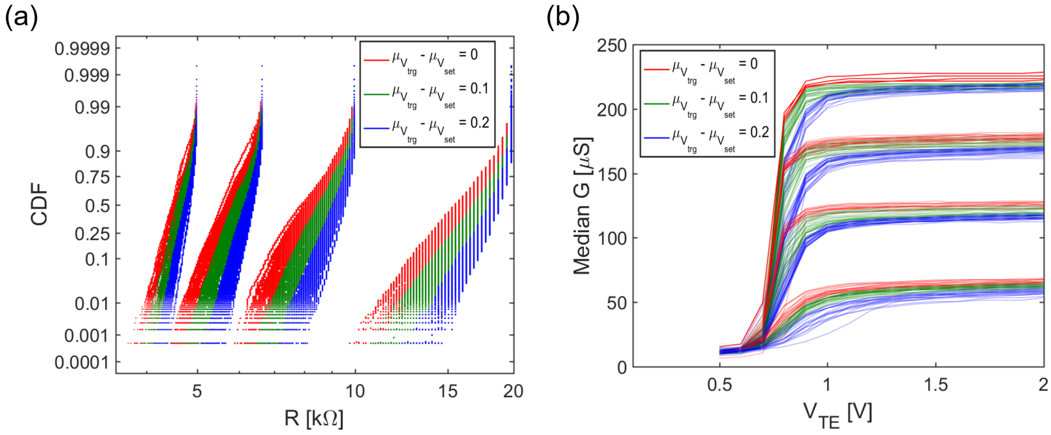


Figure 3.8: (a) C2C cumulative distributions for all levels, highlighting the devices having different $\mu_{V_{trg}}$ - $\mu_{V_{set}}$ representing the set graduality. The most abrupt ones, in red, occupy the top portion, meaning that they have largest σ_R and smallest μ_R. The most gradual, in blue, instead are concentrated in the bottom part, having smallest σ_R and largest μ_R. In green, the intermediate cases are shown. (b) median G as a function of top electrode voltage for the set of devices previously extracted through $\mu_{V_{trg}}$ - $\mu_{V_{set}}$. The order in graduality, illustrated by colors going from blue to green to red, is respected, since the blue curves have smallest μ_G (largest μ_R), while the red ones have the largest μ_G (smallest μ_R)

curves) occupy always the top part of the whole shape, meaning that they have the largest σ_R and the smallest μ_R. The most gradual devices (blue curves) instead, are always in the bottom part of the distributions, having the smallest σ_R and the largest μ_R. Finally, the green ones stay in the middle in accordance to the intermediate value of $\mu_{V_{trg}}$ - $\mu_{V_{set}}$ which characterizes them.

On the other hand, figure 3.8, (b) displays the median programming characteristic trend, i.e. the median conductance G as a function of the voltage of pulses applied to top electrode. The same approach is used, highlighting the programming characteristic of the same devices whose C2C distributions are shown in figure 3.8, (a). Again, the dependence on the graduality is evident, as the blue curves representing the most gradual devices stabilize at smallest G (largest μ_R), the red ones reach the largest G values (smallest μ_R), while the green ones stay in the middle.

Chapter 4

Modeling of statistical variability of 4 kbit HfAlO RRAM array

This chapter presents the statistical model developed in order to predict the programming variability of the HfAlO RRAM array introduced in the previous chapter. The model takes as inputs the programming conditions imposed by the program/verify algorithm and gives as outputs the simulated programming characteristic. The programming variability is reproduced through the introduction of a statistic in some parameters, via the Monte Carlo method. The model is then tuned on the experimental data in order to identify the parameter values that represent the array variability. After the description of the differential equation and the parameters on which the model is based, the strategy adopted to tune them on experimental data is presented, first focusing on the reproduction of cycle-to-cycle variability and then extending to the device-to-device one.

4.1 Introduction

In this section, a compact model capable of describing the variability in multi-level programming of 1T1R RRAM devices via program/verify algorithm is proposed. The core of this model will not be the attempt to achieve a precise modeling of the creation and disruption of the conductive filament within the oxide under the ISPVA particular conditions. On the contrary, a simpler and more compact approach in describing the switching mechanism is adopted, paying particular attention in facilitating the model integration with Monte Carlo methods used to simulate the programming variability.

The logic steps used in this chapter are the following. Firstly, the simple model which explains the switching mechanism is studied, with the goal of reproducing the nominal programming characteristic, i.e. the conductance G evolution as a function of the top electrode voltage, introduced in section 3.4. Secondly, a variability is incorporated in some model parameters in order to investigate their impact on the characteristic graduality and on the final resistance value (R *after switching*). Finally, the parameters are tuned with the aim of reproducing all experimental results, via Monte Carlo simulation.

4.2 Model description

4.2.1 Differential Equation

Unlike the case addressed in chapter 2, the equation 4.1 used in this model in order to describe the switching mechanism takes into consideration only the electrical effect on the device, neglecting any temperature dependence.

$$\frac{dG}{dt} = Ae^{\alpha V_R} \tag{4.1}$$

Figure 4.1 (a) shows the 1T1R structure where V_R indicates the voltage across the device. Equation 4.1, when integrated, allows to describe the time evolution

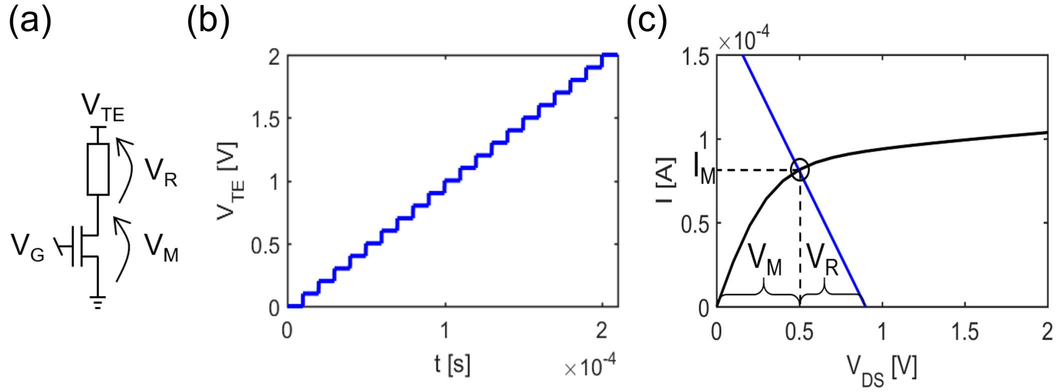


Figure 4.1: (a) 1T1R structure; V_R, voltage across the resistive device, is the value inserted in the differential equation 4.1 which describes the time evolution of conductance G. (b) Waveform applied to top electrode ($V_{TE}(t)$) which sets a condition for the integration of equation 4.1. Only the programming pulse (P) sequence of the real waveform used in the ISPVA is represented, because the model offers directly the conductance G as an output, with no need to measure it in the readout pulses (V), unlike what happens in the ISPVA. (c) V_R computation method: it is obtained thanks to the identification of the circuit working point, i.e. the intersection point between the MOSFET characteristic (in black) and the device load curve (in blue).

of the conductance G as a function of A and α. A is a parameter indicating the evolution velocity as in any Arrhenius-type equation. The activation energy is somehow included in such parameter, since the temperature dependence is neglected in this case. On the other hand, α is the barrier lowering factor as introduced in chapter 2.

The conditions in which the equation is solved, are those typical of the ISPVA, introduced in section 3.2. The main constrictions are the application of a particular waveform to the top electrode (V_{TE}) and a constant voltage applied to the gate (V_G) and consequently a fixed MOSFET voltage-current characteristic. Regarding the top electrode voltage, it is important to underline that the V_{TE} waveform used in this section is different from the one described in 3.2. It has the shape of a pulsed ramp, as shown in figure 4.1, (b) and represents only the programming part of the whole waveform used in the experiment. In fact, it is not crucial to perfectly replicate it, since the model was developed in order to have the conductance G programming characteristic as an output. The verify part of the experiment can easily be performed by software on the model output.

Due to the V_{TE} dependence on time, equation 4.1 integration is not straight forward and cannot be solved analytically, because V_R will have a dependence on time, too. Moreover, it is not simple to extract the V_R value to be put in equation 4.1, since the conductance G itself is varying during the programming.

Therefore an iterative method is adopted, executing the following operations at every time step:

- Computing the device load curve $I_R^{(k)}(V_{DS}) = G^{(k-1)}V_R = G^{(k-1)}(V_{TE} - V_{DS})$ with the G value at the previous step. Such curve represents the current which flows in the device as a function of V_{DS}, namely the voltage across the MOSFET channel.

- Computing $V_R^{(k)}$ using $I_R(V_{DS})$ and the MOSFET characteristic $I_M(V_{DS})$ as shown in figure 4.1, (c). Due to the series configuration, the intersection point between the two curves will represent the circuit working point in terms of current I_M and V_M. Given V_M, it is straightforward to compute $V_R = V_{TE} - V_M$.

- Computing the conductance update $\Delta G^{(k)} = Ae^{\alpha V_R^{(k)}} \cdot \Delta t$

- Computing the conductance value $G^{(k)} = G^{(k-1)} + \Delta G^{(k)}$

The initial G value G_0 is chosen to be 10 μS. Such procedure is repeated as long as G is smaller than the target. Moreover, it is crucial to assure the solution convergence through the right time step of integration (Δt) selection. A deeper analysis of such issue will be carried out in the following. As a first guess, it is possible to define an upper limit for Δt: since the V_{TE} steps have duration $10\mu s$ as shown in figure 4.1, (b), it will be smaller than such value.

4.2.2 A and α tuning: V_{set} addition

At this point it is crucial to perform a first coarse tuning of A and α in order to well fit the conductance characteristic extracted from the experimental data. The most important features to fit are the top electrode voltage for which the conductance starts to evolve towards larger values, defined as V_{set}, and the transition derivative, i.e. the conductance evolution graduality after V_{set}. Both features depend on the combination of A and α.

From this considerations, two conditions on $\frac{dG}{dt}$ can be introduced. Firstly, it must be very small (for example $1\frac{S}{s}$) up to $V_{TE} \simeq V_{set}$ in order to well fit V_{set}. Secondly, $\frac{dG}{dt}$ must be higher than the derivative of the set transition extracted from data, for $V_{TE} > V_{set}$. However, regarding this last point, only an approximation for the $\frac{dG}{dt}$ value can be extracted, because data have a large $dt = 10\mu s$, while a smaller dt needs to be used in the model for convergence issues. A reasonable value in this case is $\frac{dG}{dt} = 50\frac{S}{s}$.

The equation system 4.2 summarizes the two conditions:

$$\begin{cases} \frac{dG}{dt} < 1\frac{S}{s} & V_{TE} < V_{set} \\ \frac{dG}{dt} > 50\frac{S}{s} & V_{TE} \geq V_{set} \end{cases} \tag{4.2}$$

The first condition can be rewritten as $A < e^{-\alpha V_R}$ with V_R corresponding to the voltage across the device when $V_{TE} = V_{set}$ (0.7V in the example of figure 4.2), while the second one results in $A > 50e^{-\alpha V_R}$ with V_R corresponding to the voltage across the device when $V_{TE} > V_{set}$ (0.8V in the example).

Figure 4.2 (a) illustrates the A trends as a function of α in a semilogarithmic plot. The blue curve represents the first condition upper limit, while in red the

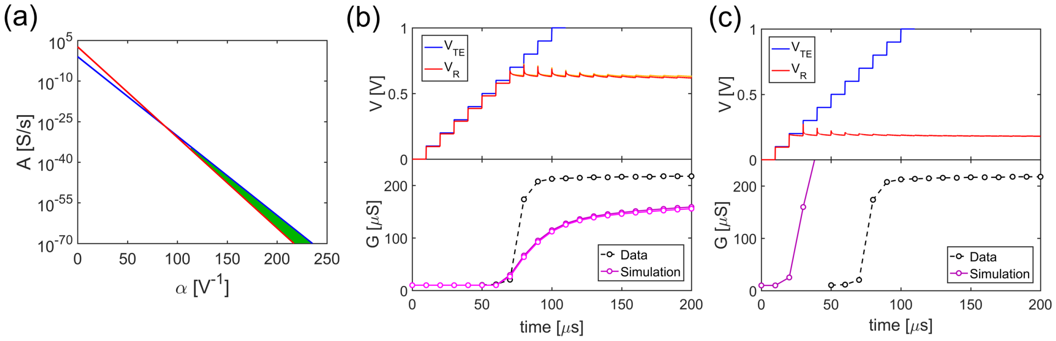


Figure 4.2: (a) Graphical representation of the two conditions put on $\frac{dG}{dt}$ in order to determine the order of magnitude of possible $< A, \alpha >$ pairs able to well fit the experimental data. The green area identifies the region of A - α plane where both conditions are satisfied. (b) Simulated V_R and G trends as a function of time for two different $< A, \alpha >$ pairs selected inside the green area. V_{set} is well fitted in both cases, but the resulting G is not enough abrupt. (c) Simulated V_R and G trends as a function of time for a $< A, \alpha >$ pair outside the green area. With respect to case (b), α is the same, while A is smaller by orders of magnitude. In this case the G transition graduality is comparable with data, but V_{set} is now very different. These considerations bring to the introduction of a third parameter called V_{set}, which allows to decouple the requirements on A

second requirement lower limit is shown. This study leads to the definition of a portion in the A - α plane, highlighted in green in the figure, where each pair of the two parameters satisfies the requirements of system 4.2. The intersection point between the two curves is at $< A = 2 \cdot 10^{-27} \frac{S}{s}, \alpha = 87V^{-1} >$ and represents the upper boundary for A and lower for α.

Figure 4.2 (b) illustrates the simulated V_R and conductance G trends as a function of time, comparing the latter with experimental data for two $< A, \alpha >$ pairs extracted from the green region of figure 4.2 (a), namely $< 2 \cdot 10^{-28}, 100 >$ and $< 10^{-42}, 150 >$. Such simulations are overlapping and show that V_{set} is well fitted, but that the difference in graduality is striking. Many other $< A, \alpha >$ pairs extracted from the same region were tested, leading to the same results of figure 4.2 (b).

A possible explanation may be derived from the V_R trend. It is noticeable that V_R starts to decrease when G increases, since an external negative feedback is triggered by the current limitation introduced by the transistor as discussed in chapter 2. However, as V_R approaches values slightly smaller than V_{set} ($\sim 0.65V$ in figure), it tends to saturate to that level and does not further decrease. This is due to the fact that for that lower V_R value, $\frac{dG}{dt}$ is forced to be very small by the first condition on A shown in system 4.2. Therefore, as soon as V_R approaches $\sim 0.65V$, $\frac{dG}{dt}$ is much smaller than what needed to well fit the data transition graduality and the simulated G trend is more gradual.

This inconsistency might be due to the coarse approximation in setting the upper limit of the second inequality of system 4.2 ($50\frac{S}{s}$). However, as it is shown in the following, other values for such limit were tested, with no particular improvement.

It is evident that a change must be made to the model. It is necessary to somehow decouple the two different conditions that G must satisfy, which cannot be achieved with the same A value. In fact, A must be very small for $V_{TE} < V_{set}$ in order to select the right V_{set}, as previously explained. On the other hand it was shown that this small A value does not satisfy the requirement on set graduality.

Figure 4.2, (c) illustrates an additional confirm to this argument. It shows the same trends for a $< A, \alpha >$ pair outside the green region, namely $< 10^{-12}, 150 >$. In particular, the same value for α as in figure 4.2 (b) ($150V^{-1}$) is used, but a much larger A ($10^{-12}\frac{S}{s}$) is chosen. In this case, G transition is more abrupt and comparable to the data, but V_{set} is much smaller and does not fit the experimental curve.

To summarize, it was found to be impossible to well fit the experimental data only with two parameters A and α, since the transition abruptness and the V_{set} value cannot be achieved at the same time.

The solution to this issue consists of the introduction of a third parameter called V_{set}. It is defined as the top electrode voltage (V_{TE}) until which no update of conductance G is performed. It means that $\frac{dG}{dt}$ is externally forced to 0 if $V_{TE} < V_{set}$. On the contrary, if $V_{TE} > V_{set}$, the aforementioned iterations are carried out.

This change allows to choose the $< A, \alpha >$ pair, taking into account only the requirement on graduality and not caring about the condition on V_{set}, which is determined externally through the new parameter.

The introduction of a new parameter might appear to make the model tuning more complex since a third degree of freedom is added. However, it is a basic parameter to be defined and identified from data and therefore it is possible to extract a range of values for such quantity. In fact some analysis on the subject were already mentioned in section 3.4.

Thanks to this addition, it is much more straightforward to fit the experimental data. Therefore, this more complete model is a starting point for the additional investigations which are shown in the following, starting from a study of convergence.

4.2.3 Convergence study

As previously mentioned, it is important to define a proper time interval which allows the iterative method to converge to the solution of equation 4.1. The finer Δt, the closer to the true solution the result, despite a larger computational effort, i.e a higher number of iterations. Based on this trade-off, it is necessary to choose a value which ensures the convergence with a limited number of iterations.

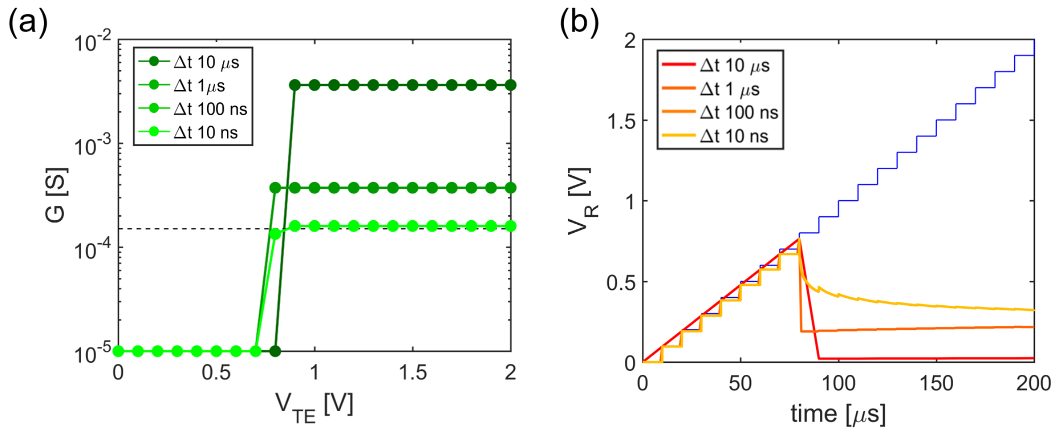


Figure 4.3: (a) Simulated conductance G as a function of the top electrode voltage for different Δt. When Δt is too large, the jump at $V_{TE} = V_{set}$ diverges to very large values. (b) Simulated V_R trend as a function of time for different time steps. For large Δt, it decreases to small values and shows a numerical overshoot for the $1\mu s$ case. From both plots it is noticeable that $\Delta t = 100ns$ and $\Delta t = 10ns$ cases are overlapping, meaning that the method is converging to the solution. This study allows to choose $\Delta t = 100ns$ as a good trade-off between convergence and number of iterations.

Figure 4.3 illustrates the simulated conductance evolution as a function of V_{TE}, (a), and the simulated V_R trend as a function of time, (b), for different values of Δt. The triplet of parameters chosen for this simulation is not well tuned on the experimental data yet, but it provides a good and sufficient understanding for the current purpose of the convergence study. Such parameter values are: $< A = 10^{-3}, \alpha = 15V^{-1}, V_{set} = 0.7V >$. Moreover, the conditions used in this example are: $G_{trg} = 150\mu S$ and $V_G = 1.4V$, therefore related to the programming of L_3. Note that the conductance curve is not left evolving for the whole V_{TE} waveform duration, but as soon as the target is crossed, the first value higher than the target is reused for larger V_{TE} values.

From figure 4.3, it is observable that for the largest $\Delta t = 10\mu s$, the solution diverges to values larger than $1mS$ and consequently V_R goes very close to 0 and remains constant at such very small value. Regarding $\Delta t = 1\mu s$ instead, the jump corresponding to $V_{TE} = V_{set}$ is lower, but still too big. In fact, the V_R trend shows the typical numerical overshoot: it instantly goes to the minimum value ($\sim 0.2V$) and then increases getting closer to the right solution. This mechanism obviously has not physical meaning and therefore this Δt value cannot be used.

The curves calculated for $\Delta t = 100ns$ and $\Delta t = 10ns$ are completely overlapping, meaning that the iteration method is now converging to the solution. V_R trend is in fact monotonous and G takes some steps to reach the target. Since a good result is already achieved with $\Delta t = 100ns$, such value is chosen as time step in the following, in order not to excessively increase the computational steps.

4.3 Parameters tuning and data fitting

Up to now, the description focused on the parameters and the integration of the differential equation which explains the device conductance evolution with time under particular external conditions. As mentioned, the model takes some constraints, in particular V_{TE} waveform and V_G value, as inputs and returns the simulated G programming characteristic as an output, from which it is possible to determine the *after switching* conductance or resistance values.

In this phase instead, the goal is to tune the three parameters of the model, A α and V_{set}, so that the simulated conductance programming characteristic and *after switching* R C2C distributions well fit the experimental data.

As addressed in chapter 3, the device set transition graduality is the crucial feature in determining the *after switching* resistance C2C distribution in terms of median R (μ_R) and standard deviation (σ_R). Therefore, in the following simulations, the programming characteristic graduality will be varied by tuning the model parameters. However, it is important to define which of them have more impact to this objective.

The strategy adopted is to associate a single $< A, \alpha >$ pair to each device in order to describe the set transition graduality with these two parameters only. Regarding the third parameter V_{set}, figure 3.6 (a) shows C2C distributions that are almost overlapping and quite bent, suggesting that V_{set} variability is mainly on cycles rather than on devices. Moreover, figure 3.6 (b) shows the absence of correlation between $\mu_{V_{set}}$ and μ_R, meaning that it is not necessary to characterize a single device with a particular $\mu_{V_{set}}$. Based on this result, every device will be associated to the same $\mu_{V_{set}}$ in the following simulations.

Given this strategy, it is important to underline that it is desirable to fit the programming characteristic and the C2C distributions related to a single device with the same $< A, \alpha >$ pair for all the four LRSs. In fact, it is reasonable to affirm that the set graduality is an intrinsic device characteristic which does not depend on the level to be programmed, i.e. on the $< V_G, G_{trg} >$ pair.

It is therefore necessary to classify the different device behaviors regarding graduality in the same way for all levels $L_1 - L_4$. Figure 4.4 shows how this is achieved. In (a) the same C2C distributions introduced in figure 3.3 are re-proposed. From these, the median value for each curve is extracted, i.e. the *after switching* resistance value at 50%. Figure 4.4 (b) then illustrates how such median values are distributed for each level.

Thanks to this analysis, it is possible to extract the median resistance value corresponding to different percentiles for each level. Moreover, given those values, the median programming characteristic of the selected device can be derived. Figure4.4 (b) illustrates the 5 percentile values used, namely 99, 90, 50, 10 , 1.

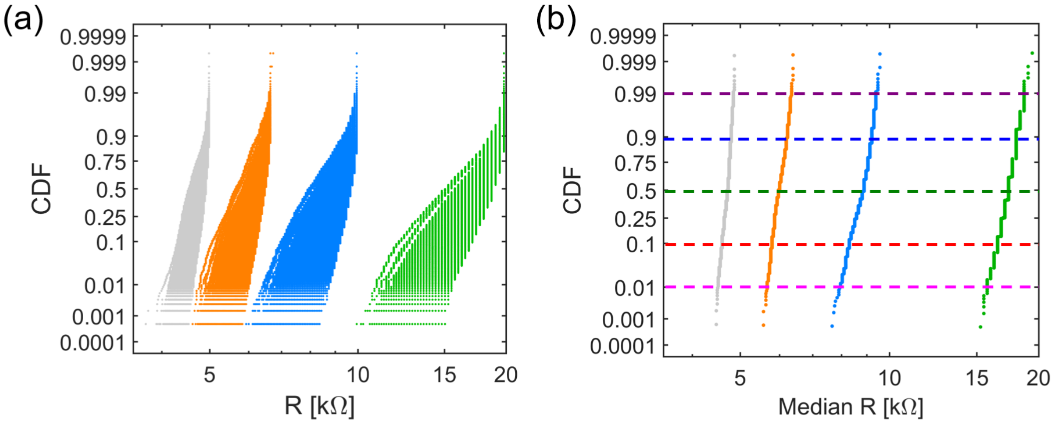


Figure 4.4: (a) C2C distributions of *after switching* R for all LRSs. (b) median R on cycles distributions for all LRSs. From those, the values corresponding to the highlighted percentiles are sampled for each level in order to identify 5 representative device, from the one presenting larger μ_R and graduality (99%) to the one characterized by smaller μ_R and graduality (1%).

To summarize, the operation previously described consists firstly of identifying 5 representative devices which differ in median R and then of deriving the specific median programming characteristic on cycles and C2C R distribution for each of them. Such specific quantities are the ones to be fitted through A and α tuning.

Figure 4.5 shows such results. First of all, more than one curve is plotted for every percentile, because a small group of 10-15 devices showing the same μ_R is always considered in order to have a larger sample. It is noticeable that the device at 99% is the one with larger μ_R and therefore the corresponding C2C distribution (figure 4.5 (a)) and median G characteristic on cycles (figure 4.5 (b)) lie in the bottom part of the whole. On the contrary, the devices at 1% show the smallest μ_R and occupy the top parts, showing the most abrupt trends of median G. The devices at intermediate percentiles systematically distribute themselves following the same order.

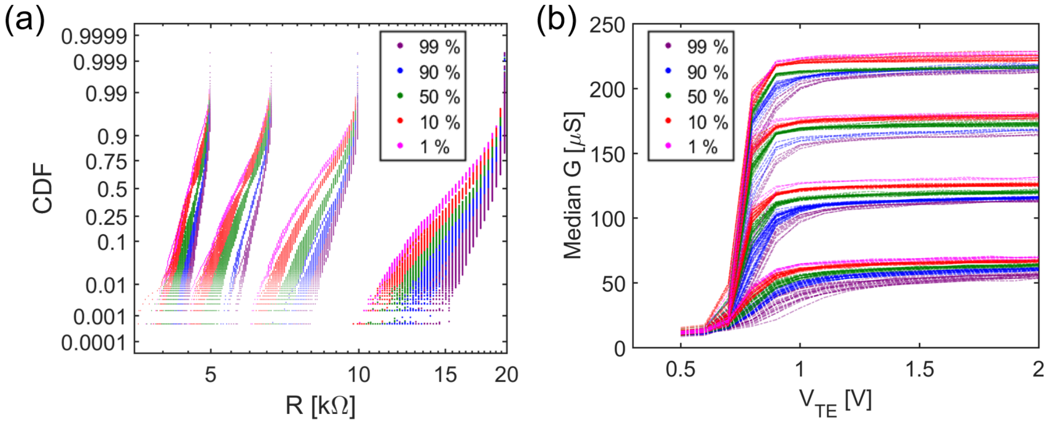


Figure 4.5: C2C R distributions (a) and median G on cycles as a function of V_{TE}, i.e. median programming characteristics on cycles (b), displaying the ensemble of devices selected for each percentile.

The data presented in figure 4.5 are the basis on which the simulation results will be shaped. As already mentioned, the strategy applied is to associate a $< A, \alpha >$ pair to each device, starting at first from the 5 previously defined. Such operation aims at obtaining 5 $< A, \alpha >$ combinations from which a set of 1000 pairs, which describe the whole device set, will be built. These 5 combinations are therefore found by simulating the endurance experiment, i.e. 1000 programming cycles, until the best fit of experimental data of figure 4.5 is achieved.

The 1000 programming cycles are simulated by introducing a certain variability in V_{set} and by adding an intrinsic read noise expressed as σ_G to the conductance G. This is the typical Monte Carlo method approach which consists in solving equation 4.1 1000 times, each time with a different V_{set} value and with the introduction of a random noise which is extracted from a gaussian distribution with 0 median value and σ_G as standard deviation. Such noise represents both the unavoidable instrument noise which affects the measurement and the intrinsic device noise due to the conductive filament stochastic nature [71]. During the A and α tuning process, the simulations are carried out

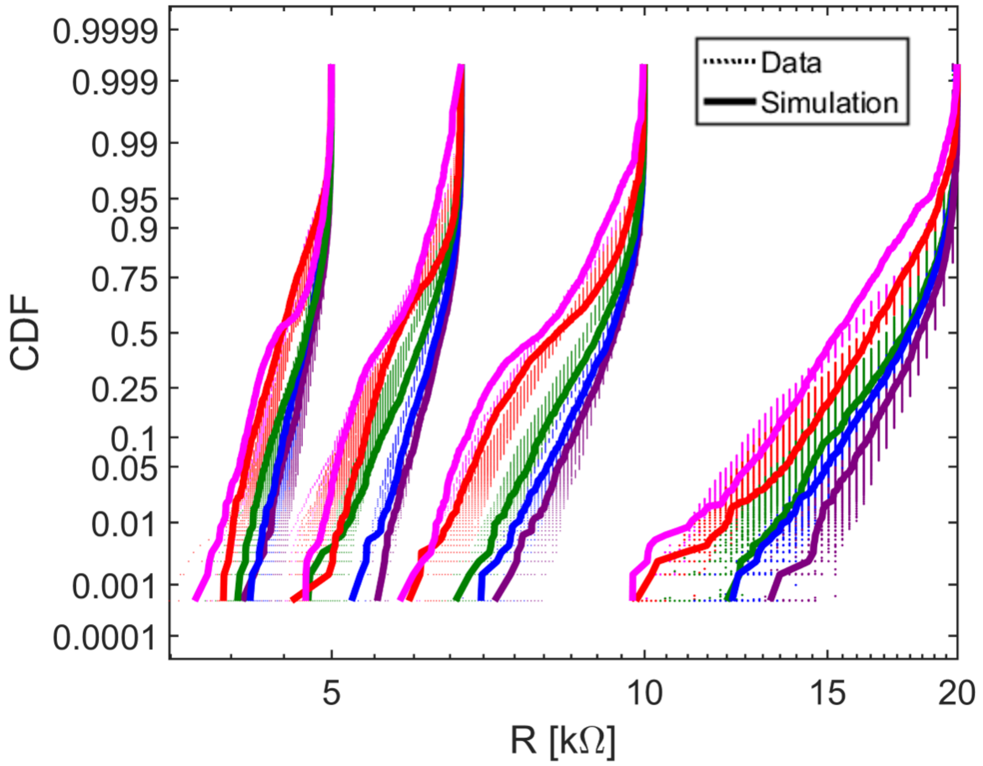


Figure 4.6: Simulated C2C R distributions for the 5 representative devices. One $< A, \alpha >$ pair is associated to each device, whose programming is simulated for all 4 LRSs by simply varying V_G and G_{trg}. The statistic variability is introduced through the V_{set} parameter ($\mu_{V_{set}} = 0.75V$, $\sigma_{V_{set}} = 0.05V$) and the addition of a read noise characterized by $\sigma_G = 1\mu S$.

by setting $\mu_{V_{set}}$, $\sigma_{V_{set}}$ and σ_G respectively to $0.75V, 0.05V, 1\mu S$, and by tuning A and α until the best fit of the experimental data is found.

Figures 4.6 and 4.7 show the simulation results. Note that the same $< A, \alpha >$ pair is used in the simulation to fit all levels for a single representative device. The result is remarkable since with a simple equation as eq. 4.1, by introducing a statistic in one parameter (V_{set}) and a read noise, it is possible to well fit the data, both in the median programming characteristic and in the distribution of the *after switching* resistance. Moreover, such results evidence the model

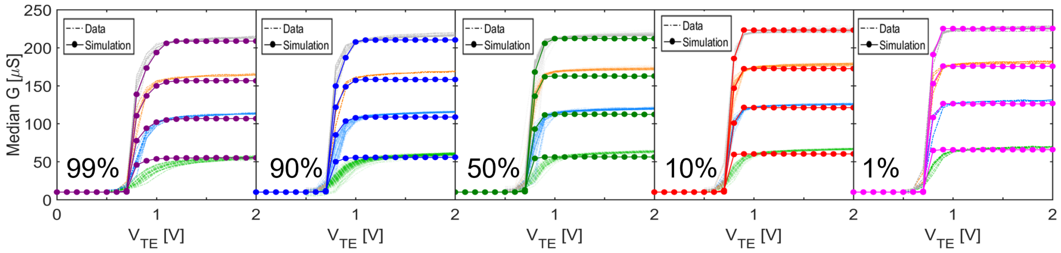


Figure 4.7: Simulated median G as a function of top electrode voltage V_{TE} for the 5 representative devices. All medians are computed from the 1000 simulated programming characteristics. The figure underlines the model compactness, since the different levels are fitted with the same parameters $< A, \alpha >$ for each device.

compactness, because the outputs related to a single device are obtained for all 4 levels immediately, by simply varying the $< V_G, G_{trg} >$ pair.

Table 4.1, collects the 5 $< A, \alpha >$ pairs linked to the 5 fitted representative devices. A and α show an inverse proportionality. A ranges more than one order of magnitude, while α's variation is much more limited, since it is an exponential factor. Therefore, a variation in both parameters strongly affects the simulated programming characteristic. This might be the reason for the aforementioned inverse proportionality: the variation in the representative devices programming characteristic is quite limited (figure 4.5 (b)). Therefore, to well fit such curves, the change of just one parameter must be somehow compensated by the other one. In this way, such little variations are achieved.

Knowing the percentile to which they are related is a further advantage for the following steps. In fact, from these 5 values it is straightforward to derive μ_A, μ_α, σ_A and σ_α which will be used to generate a distribution of 1000 pairs representative of the whole array. Such discussion will be investigated in the following sections.

	$A[\frac{S}{s}]$	$\alpha[V^{-1}]$
1%	$8 \cdot 10^{-2}$	11.3
10%	$3.8 \cdot 10^{-2}$	12.2
50%	$1.2 \cdot 10^{-2}$	13.5
90%	$4 \cdot 10^{-3}$	14.75
99%	$1.4 \cdot 10^{-3}$	16

Table 4.1: A and α corresponding to the 5 representative devices. An inverse proportionality is found between the two quantities. However, A varies more than one order of magnitude, whereas α's range is more limited since it is an exponential factor.

4.4 Simulation of an endurance experiment with RRAM statistical model

In the previous section, $5 < A, \alpha >$ pairs were found in association to 5 representative devices. Now, in order to completely replicate the experimental data, it is necessary to simulate the endurance experiment for the complete set of 1000 devices per each level. Supposing that the two parameter distributions will be gaussian, the next step consists therefore of building a bivariate normal distribution in the $A - \alpha$ plane and of extracting from that $1000 < A, \alpha >$ pairs. Note that a bivariate normal distribution is the bi-dimensional generalization of a one-dimensional normal distribution.

Equation 4.3 describes the probability density function of a multi-variate normal distribution (the most general case) [77]

$$f_X(x_1, ..., x_k) = \frac{exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))}{\sqrt{(2\pi)^k \det \Sigma}} \tag{4.3}$$

where x is a real k-dimensional vector and Σ is the covariance matrix, i.e. a square matrix giving the covariance between each pair of elements of a given

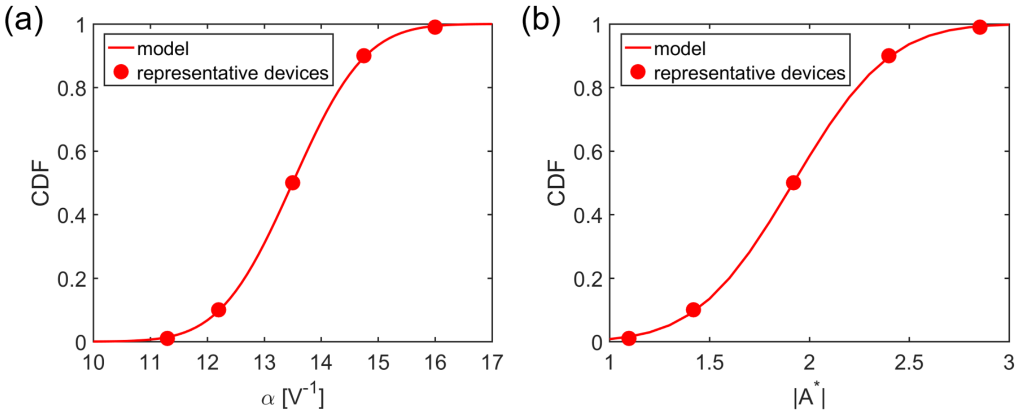


Figure 4.8: Resulting α (a) and $|A^*|$ (b) cumulative distribution functions extracted from the values associated to the 5 representative devices fitted in the previous section. The parameters featuring these distributions are $\mu_\alpha = 13.5V^{-1}$, $\sigma_\alpha = 1V^{-1}$, $\mu_{A^*} = -1.92$ and $\sigma_{A^*} = 0.38$.

random vector. In the matrix diagonal there are variances, i.e., the covariance of each element with itself.

As shown in table 4.1, A ranges of more than one order of magnitude. It is therefore simpler to create the bivariate distribution as a function of $A^* = \log_{10}(A)$.

In the bivariate case, equation 4.3, already referred to the variables A^* and α, becomes:

$$f(\alpha, A^*) = \frac{1}{2\pi\sigma_\alpha\sigma_{A^*}\sqrt{1-\rho^2}} \cdot \tag{4.4}$$
$$\cdot \exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(\alpha-\mu_\alpha)^2}{\sigma_\alpha^2} + \frac{(A^*-\mu_{A^*})^2}{\sigma_{A^*}^2} - \frac{2\rho(\alpha-\mu_\alpha)(A^*-\mu_{A^*})}{\sigma_\alpha\sigma_{A^*}}\right]\right)$$

where ρ is the correlation between α and A.

In this case $\boldsymbol{\mu} = \begin{pmatrix} \mu_\alpha \\ \mu_{A^*} \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_{A^*} \\ \rho\sigma_\alpha\sigma_{A^*} & \sigma_{A^*}^2 \end{pmatrix}$.

In order to create such bivariate distribution, it is therefore necessary to know ρ, σ_α and σ_{A^*}. The two standard deviations are easy to extract from the values obtained in the previous section and are equal to $1V^{-1}$ and

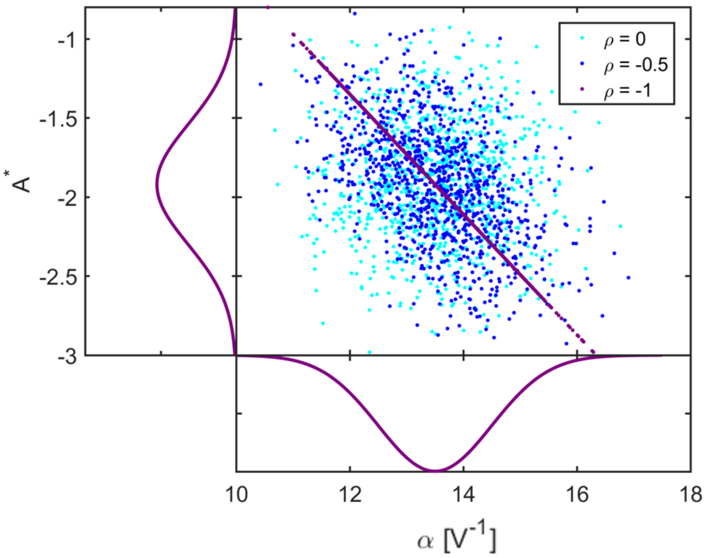


Figure 4.9: Correlation plot for 1000 $< A^*, \alpha >$ pairs extracted from the bivariate normal distribution in equation 4.4. The effect of different correlation coefficients ρ is compared. As expected, when the perfect correlation is forced, all values distribute on a straight line, instead when A and α are completely uncorrelated the values are much more spread. For simplicity $\rho = -1$ is chosen in the following.

$0.38 S/s$ respectively. Figure 4.8 shows the related cumulative density functions, highlighting the 5 values used as a basic structure. Regarding ρ instead, for sake of simplicity, it is at first set equal to -1, suggesting a perfect correlation between the two parameters. Such statement is not entirely precise, since a perfect correlation between two separate quantities is not usually physically expected. However, since A and α does not directly refer to specific physical quantities, ρ cannot be measured or extracted from data and is used as external parameter that can be modified to verify the impact on the simulations.

Figure 4.9 shows the correlation plot of 1000 $< A^*, \alpha >$ values extracted from the bivariate distribution described by equation 4.4 with different correlation coefficients ρ. As expected, in the $\rho = -1$ case all values distribute themselves

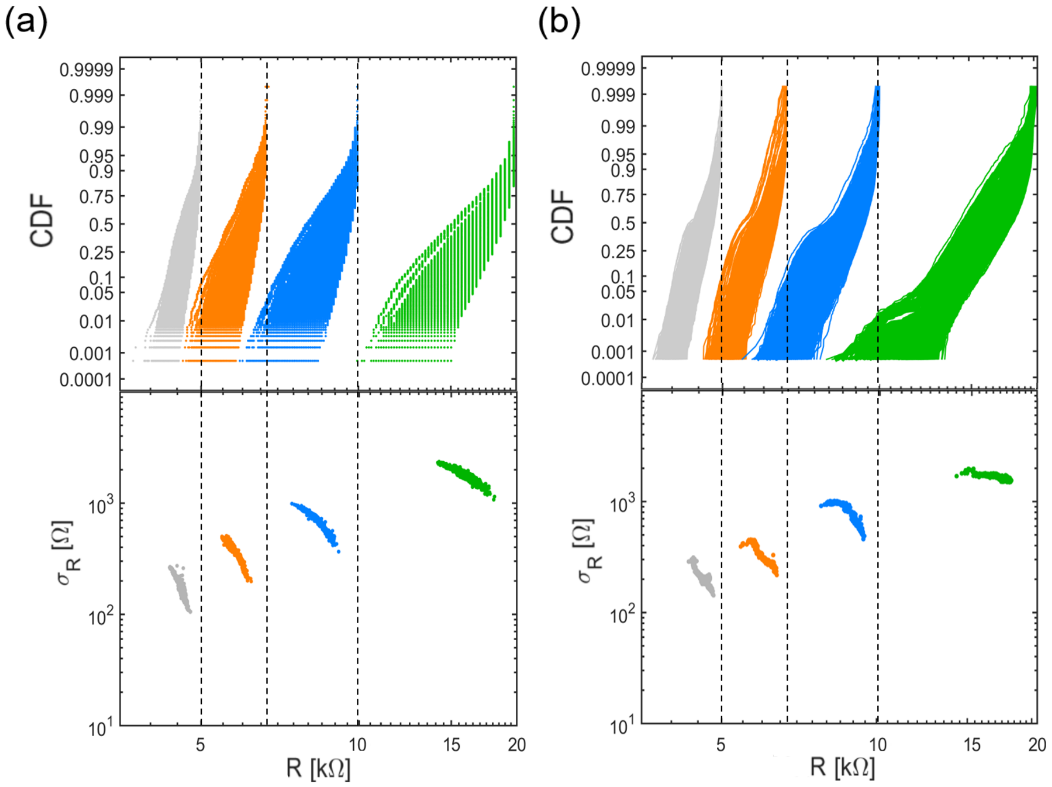


Figure 4.10: Comparison between experimental data (a) and simulation (b) for C2C *after switching* R distributions and relative σ_R as a function of μ_R. Calculations well reproduce the experimental data, thus supporting the validity of the statistical model.

on a straight line representing the aforementioned proportionality between A and α. When $|\rho| < 1$ instead, it can be noted that the values spread on a larger region of the $A - \alpha$ plane. In any case, the important achievement of this step is that the set of 1000 pairs is extracted from a normal distribution, meaning that the values have different probability to be extracted, which is largest next to the median values and decreases approaching the extremes. According to these considerations, $\rho = -1$ is chosen in the following.

Given the 1000 $< A, \alpha >$ pairs of figure 4.9 representing the whole array in terms of D2D variability, it is possible to simulate the whole endurance

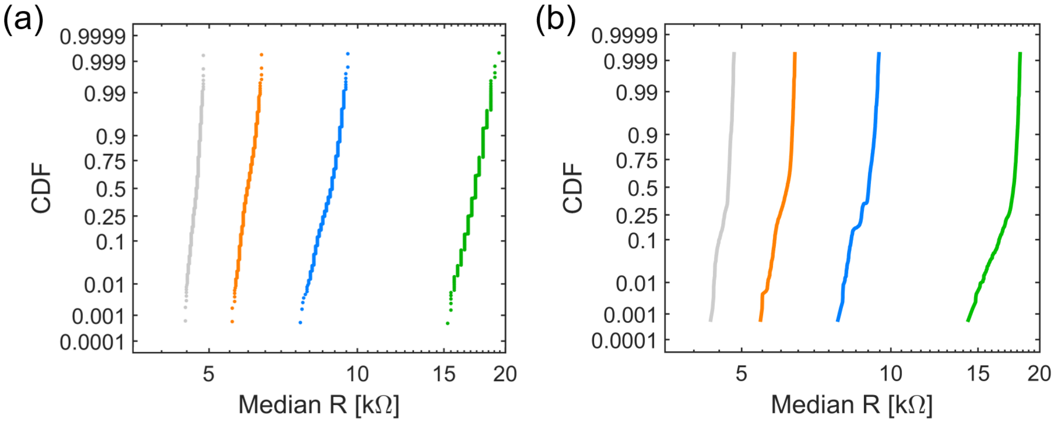


Figure 4.11: Comparison between data (a) and simulation (b) regarding the median R distributions, which display the D2D variability.

experiment in order to replicate the results of figure 4.4. For each pair, the 1000 programming cycles are simulated in the same way used in section 4.3, i.e. by introducing a variability in V_{set} and adding a read noise σ_G.

Figures 4.10 and 4.11 show the results related to the complete experiment simulation. The similarity between data and simulation is remarkable in every plot, in terms of C2C cumulative distributions, μ_R and σ_R. These results support the model efficacy. Starting from a simple differential equation describing the programming characteristic, the set of model parameters related to the median device behavior is found. Then, introducing a statistic in such parameters, it is possible to predict both the C2C and the D2D variability.

Chapter 5

Neural network simulation with RRAM statistical model

This chapter presents the implementation of a multilayer neural network with the HfAlO RRAM array, focusing in particular on the impact of programming variability on the network classification accuracy. For this reason, in the first part of the chapter the statistical model introduced in the previous chapter is used to simulate the array programming variability under different program/verify techniques, showing that the device conductance tuning is finer with respect to the classical ISPVA approach. In the second part of the chapter instead, the steps used to design and train the network in the array are described, highlighting the advantages introduced by an incremental quantization technique. Finally, the network is tested by taking into account the variability and the expected accuracy achieved under the different programming techniques is compared.

5.1 Introduction

In the previous chapter, the statistical model developed to assess the programming variability of the 4 kbit HfAlO RRAM array was introduced and tuned on the experimental data. At this point, the average device behavior and its variability range are well defined by the parameters and therefore the model well represents the array in terms of variability. For these reasons, some applications can now be introduced.

Recent works [78–80] have investigated the possibility to implement a neural network into HfO-based RRAM arrays. In this chapter, such application is addressed. As introduced in chapter 1, multi-level operation is crucial to achieve synaptic weights with analog behavior, enabling neural network training with high precision. Indeed, the higher the number of levels, the larger the accuracy. Therefore, the common goal is to refine multi-level operation to increase the number of levels. However, programming variability represents the main obstacle to this purpose, as it blurs the distinction among different levels. In this scenario, the statistical model can be used as a very powerful tool in order to predict the variability under different programming condition, in order to find the one which is expected to minimize variability.

In the first part of the chapter, two new programming techniques, which are different from the ISPVA, are presented. Then, their application to the array is simulated via the statistical model, in order to investigate if the expected variability is decreased with respect to the experimental data found using ISPVA approach.

In the second part of the chapter, the implementation of a multilayer neural network with HfAlO RRAM devices is presented. First, the multi-layer perceptron (MLP) network is designed and trained in software for inference demonstration tasks. Then, its implementation in the array is simulated.

Finally, the network inference accuracy is tested, allowing to evaluate how the variability can affect the network performances.

5.2 Simulations under different program/verify techniques

As previously introduced, in this section the statistical model is used in order to investigate the impact of different program/verify techniques on HfAlO RRAM variability. Their configuration is derived through the application of some modification to the ISPVA, which is used as a starting point. The two different programming conditions adopted are the following:

- *Finer* V_{TE}: the same technique as the ISPVA is used, but with a finer top electrode voltage V_{TE} step ($\Delta V_{TE} = 0.01V$ instead of $0.1V$)
- *Hybrid*: firstly the ISPVA approach up to a value (called G_{ph_1}) smaller than the target is used. Then, a pulsed ramp V_G is applied to the gate at fixed V_{TE} until the nominal target is reached.

Both techniques are supposed to improve the programming variability. By applying the first technique, the device receives more gradual steps and the filament shape might be more controllable. The Hybrid approach, instead, is introduced in order to understand if it is possible to obtain a better control on the device programming by operating through the compliance current variation rather than on the top electrode voltage. It is designed in a simple way, since the main goal is to simulate a programming technique which can be truly used in an experimental setup. Figure 5.1 shows a simulated programming characteristic in the two phases of the Hybrid technique. In phase 1, the ISPVA approach is maintained, but an intermediate $< V_{G_1}, G_{ph_1} >$ pair is associated to each nominal G_{trg}. The idea is to provide a coarse programming via the typical

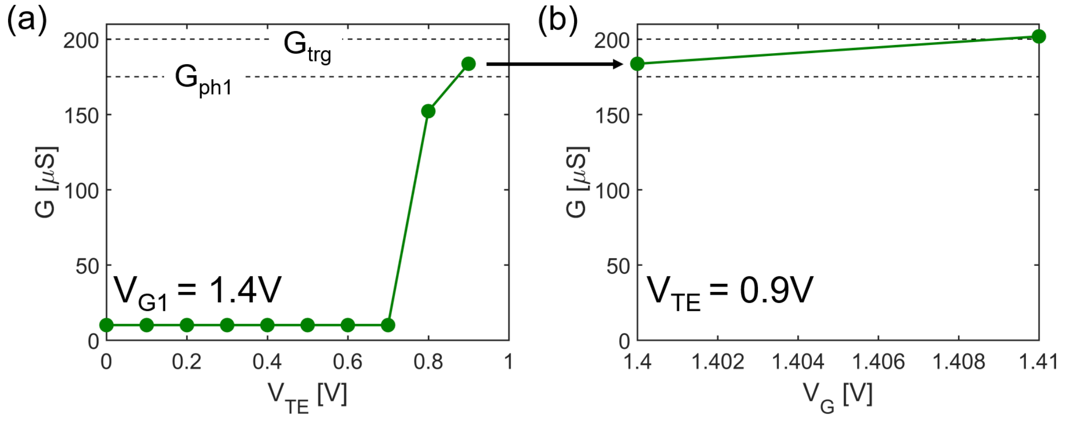


Figure 5.1: Simulated programming characteristic adopting the Hybrid technique. The ISPVA approach, where V_G is constant and V_{TE} is increased by steps, is used in phase 1 (a) up to a target G_{ph_1} which is lower than the nominal one. Then, during phase 2 (b), V_{TE} is kept constant to the last value used in phase 1, and V_G is increased with steps of $0.01V$ until G_{trg} is reached.

ISPVA approach up to a conductance value which is still lower than the desired one. In phase 2, instead, the top electrode voltage is kept constant to the last value used in phase 1, and the V_G is increased from V_{G_1} to gradually change the compliance current and perform a finer programming, controlling the filament creation in a better way. It is therefore expected that the variability in this case will be lower.

Table 5.1 shows the parameters used in simulating the Hybrid technique. The V_G step used in phase 2 is $\Delta V_G = 0.01V$.

In a first moment, the simulation is carried out only on the 5 representative devices introduced in section 4.3, since the goal is to understand how the median device (50 %) and the tail devices (1% and 99%) are expected to behave in the new programming conditions.

Figure 5.2 shows the simulation results against the experimental data. Firstly, it is observable that in both approaches the most gradual device (99%) has C2C distributions which well replicate the experimental data, meaning

	G_{trg} [μS]	G_{ph_1}[μS]	V_{G_1} [V]
L_1	50	25	0.8
L_2	100	50	1
L_3	150	100	1.2
L_4	200	150	1.4

Table 5.1: Recap of G_{trg}, G_{ph_1} and V_{G_1} used in the Hybrid approach. As soon as G_{ph_1} is reached, V_{TE} is kept constant to the last value of phase 1 and V_G is increased by steps of 0.01V until G_{trg} is overcome.

that no improvement is introduced. In fact the gradual devices, as addressed in chapter 3, are the most controllable ones in terms of programming even with the classical ISPVA approach.

The Finer V_{TE} approach illustrated in figure 5.2 (a) brings benefits for the programming of more abrupt devices in levels L_1, L_2, L_3 where it is visible that the simulated curves are less bent and are closer to each other, suggesting a reduced device-to-device (D2D) variability. For level L_4, instead, it is noticeable that the simulations do not strongly differ from the experimental data.

On the other hand, the Hybrid approach illustrated in figure 5.2 (b) shows the best results since even the cycle-to-cycle (C2C) distribution related to the most abrupt representative device (1%) lies very close to the others in every levels, suggesting that the D2D variability is strongly limited using this approach.

A deeper analysis and comparison between the two simulated programming techniques and the experimental data is illustrated in figure 5.2 (c), which shows the standard deviation σ_R as a function of the median R μ_R. One common positive result is that μ_R is generally closer to the target in both simulations with respect to the experimental data. As anticipated, the σ_R calculated in the Finer V_{TE} approach (blue circles) shows comparable values with respect to the

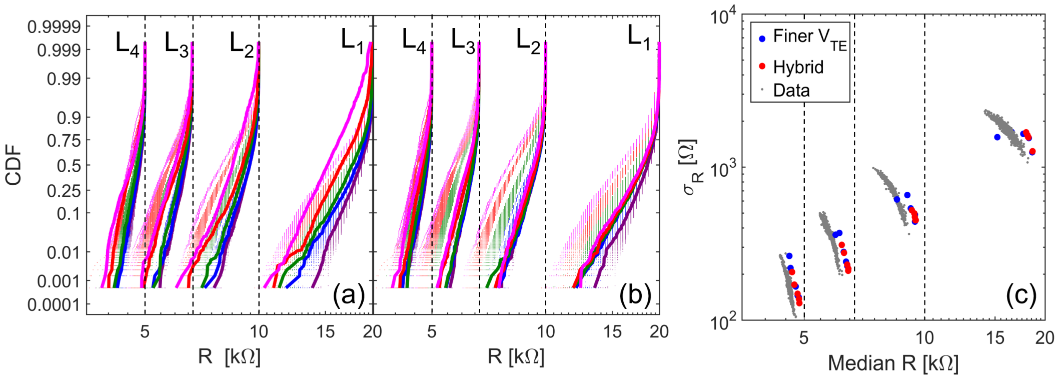


Figure 5.2: Comparison of the 5 representative devices C2C R distributions in the Finer V_{TE} (a) and Hybrid (b) approaches. In the former case some improvements are achieved for levels L_1, L_2, L_3, but the simulated variability for level L_4 is comparable with the experimental data. The latter case is instead much better in limiting the D2D programming variability. (c) Standard deviation σ_R as a function of median R μ_R for the two simulated programming techniques and the experimental data. This figure confirms the fact that both techniques reduce variability and in particular the Hybrid approach is the most efficient for all levels.

experimental data (in gray) for L_4, suggesting that this approach is not able to sufficiently reduce the variability for all levels. On the other hand, the Hybrid approach shows the best results, since the red circles are more concentrated than the blue and gray ones for each level, meaning that the programming variability is strongly reduced.

5.3 Design and simulation of a neural network for image classification

Figure 5.3 (a) shows a schematic representation of the neural network designed for inference demonstration. It is an MLP network composed by the

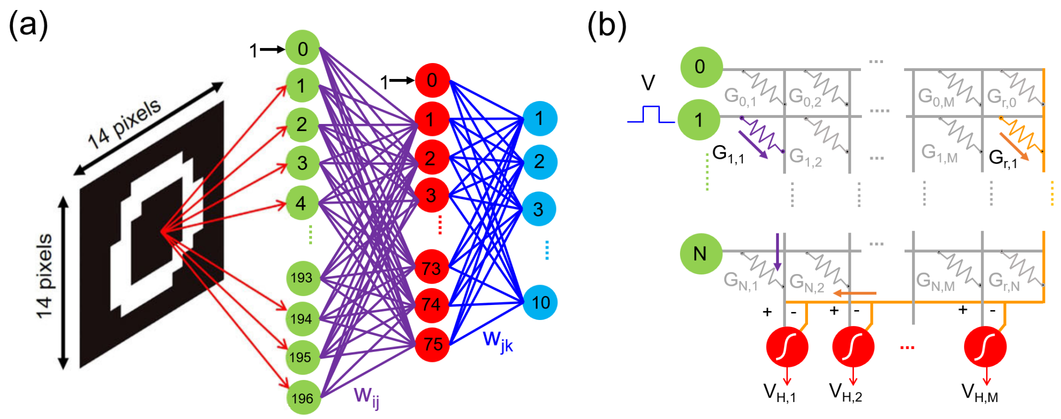


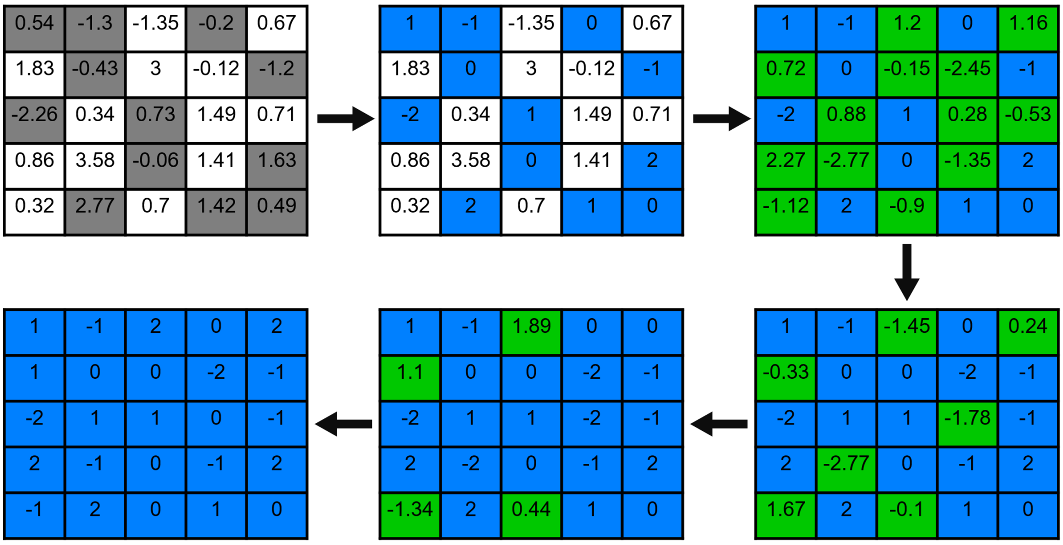
Figure 5.3: (a) Schematic representation of the MLP network designed for inference demonstration, composed by 197 input neurons, 76 hidden neurons and 10 output neurons. A downscaled version of the original MNIST images is used for size limitations. (b) Representation of the matrix of weights connecting input and hidden layers written in the RRAM array. Each weight is implemented as the difference of a programmable conductance G_{ij} and a reference conductance G_r. Each hidden neuron collects the currents of each column and transforms them through a sigmoid function into a voltage output V_{H_j} which becomes the input signal for the output neurons. Note that the voltage V used in inference phase is 0.2V, i.e. the readout voltage used in the ISPVA, as introduced in section 3.2. Adapted with permission from [74].

input layer, consisting of 197 neurons, one hidden layer with 76 neurons and the output layer with 10 neurons. Note that such size is not compatible with the implementation in the 4 kbit array, since the number of weights is too large. However, this choice is made since this part of the work is an extension of a previous work [74] which presented a smaller size network which fitted in the array at disposal. In this case, a larger size array (16 kbit) with the same variability features as the real one is considered. Thanks to this size extension, a better classification accuracy can be achieved.

The process used to train and simulate this network is the following:

- The network is trained in software on the Modified National Institute of Standards and Technology (MNIST) database of handwritten digits via the backpropagation algorithm [54].
- The weights obtained are quantized with the Incremental Network Quantization (INQ) from 64-bit floating point to 5 evenly spaced discrete levels centered around 0, representing the 5 resistive levels which can be programmed in the array.
- The 5 discrete levels centered around 0 are mapped into the 5 conductance values representing the HRS and the 4 LRSs targets.
- The programming variability affecting each level is introduced, by selecting the conductance value representing the synaptic weight, from a distribution centered around the 5 descrete levels and with standard deviation extracted from data or simulated via the model.

As mentioned, the first step is the network supervised training on the MNIST training dataset with 60000 handwritten digit images by backpropagation. The signal emitted by input neurons in response to image submission is forward propagated toward the output layer leading to generation of an error signal which is calculated as a difference of effective output signal and expected network response. Such error is backpropagated toward the input layer and exploited to update synaptic weights according to equation 1.1. After this operation is iteratively performed on the entire training dataset for 20 epochs, by halving the learning rate η every 5 epochs using 1 as initial value, the final weight matrix is obtained. Note that a downscaled version (14x14 pixels) of the original MNIST dataset (28x28) images is used, because, despite the increase in size, a 16 kbit array is still too small to contain a network that is able to deal with the original size images. After the training procedure, the network is tested on 10000 unseen images from MNIST test dataset resulting in an



0.54	-1.3	-1.35	-0.2	0.67
1.83	-0.43	3	-0.12	-1.2
-2.26	0.34	0.73	1.49	0.71
0.86	3.58	-0.06	1.41	1.63
0.32	2.77	0.7	1.42	0.49

1	-1	-1.35	0	0.67
1.83	0	3	-0.12	-1
-2	0.34	1	1.49	0.71
0.86	3.58	0	1.41	2
0.32	2	0.7	1	0

1	-1	1.2	0	1.16
0.72	0	-0.15	-2.45	-1
-2	0.88	1	0.28	-0.53
2.27	-2.77	0	-1.35	2
-1.12	2	-0.9	1	0

1	-1	2	0	2
1	0	0	-2	-1
-2	1	1	0	-1
2	-1	0	-1	2
-1	2	0	1	0

1	-1	1.89	0	0
1.1	0	0	-2	-1
-2	1	1	-2	-1
2	-2	0	-1	2
-1.34	2	0.44	1	0

1	-1	-1.45	0	0.24
-0.33	0	0	-2	-1
-2	1	1	-1.78	-1
2	-2.77	0	-1	2
1.67	2	-0.1	1	0

Figure 5.4: Schematic explanation of INQ algorithm on a 5x5 matrix. The top row shows the three basic operations: weight partition (left), group-wise quantization (center) and re-training (right). The bottom row show the results after the second (right), third (center) and final (left) iteration, resulting in the incremental quantization of the original matrix.

inference accuracy of 96.2%. Note that the network size extension with respect to the previous work [74], allows to achieve a remarkably higher classification accuracy (96.2 % vs 92 %).

At this point, the real-valued weight matrix of the optimized MLP has to be written in the array, by mapping each weight to the conductance value of the devices. Since the array that has to be programmed is larger than the real one, the implementation steps are simulated in software, in order to have a first idea of the network performances. First of all, RRAMs can be programmed using only 5 levels, therefore the synaptic weights must be converted from 64-bit floating point precision to 5 discrete values, namely -2, -1, 0, 1, 2.

This operation is performed through a particular technique called Incremental Network Quantization (INQ) [81]. Figure 5.4 schematically shows the algorithm operation on a 5x5 weight matrix, for sake of simplicity. It is divided into three steps, namely weight partition, group-wise quantization and re-training. First, the whole matrix is equally divided into two groups of values (top left matrix in figure). Then, the first ones are quantized with a rounding strategy (top center matrix). Finally, the second group is re-trained keeping the first group values constant to the quantized value (top right matrix). This procedure is then repeated on the re-trained values, so that the quantization is achieved in an incremental way, as shown in the bottom row of figure 5.4.

This quantization approach is very efficient, since the accuracy changes from 96.2 % (real-valued weights) to 93.5 % (5-level weights). Moreover, the original algorithm can be further improved by applying a little modification to the weight partition step. Instead of separating the two groups by choosing the matrix elements in a random way, a pre-processing is performed by calculating the quantization error for the whole matrix and then selecting the elements causing the largest error. These values are those to be quantized, while the other ones are re-trained. Such modification allows to strongly improve the network accuracy, as it rises to 95.1 %, regaining about 1.5 % with respect to the random INQ approach, which means that additional 150 digit images are correctly classified using this enhanced quantization algorithm.

At this point, the 5 discrete levels have to be mapped into the 5 conductance targets representing the HRS and the 4 LRSs. However, since levels have both positive and negative sign and the device conductance can only be positive, it is necessary to implement each synaptic weight as the difference of two conductances $w_{ij} = G_{ij} - G_r$, where G_r is a reference value programmed at the intermediate level, as shown in figure 5.3 (b) [29]. After these steps, the weight matrix based on the 5 conductance values is obtained.

Given the structure shown in figure 5.3 (b), the total number of weights to write in the array is given by $197 \cdot 75 + 75 \cdot 10 = 15535$, corresponding to the tunable conductances G_{ij}, which is added to $197 + 76 = 273$, corresponding to

Case study	Inference accuracy
Real-valued weights	96.2%
5-level-valued discrete weights (INQ random)	93.53%
5-level-valued discrete weights (INQ max error)	95.1%

Table 5.2: Classification accuracy achieved on MNIST test dataset with real-valued weights, 5-level weights with random quantization, and 5-level weights with maximum-error-based quantization, respectively.

the reference conductances G_r, thus resulting in 15808 total weights, which is compatible with the implementation in a 16 kb array, as previously mentioned.

Table 5.2 summarizes the network performance in classification accuracy on MNIST test dataset. The adoption of the INQ algorithm, modified in the weight partition step by selecting the matrix elements exhibiting the largest quantization error, is very convenient as the accuracy with respect to the real-valued case is reduced by about 1%.

The last step consists of introducing the programming variability, by associating a distribution to each discrete level. Specifically, the conductance value for each RRAM device is extracted with a certain probability from a distribution centered around the 5 discrete levels with a standard deviation determined by the experimental data or by the simulations performed via the statistical model. The larger the standard deviation, the higher the variability and the lower the expected network accuracy.

5.4 Comparison of neural network performance for different programming algorithms

At this point, it is necessary to extract the standard deviation which has to be associated to each level in order to replicate the array programming

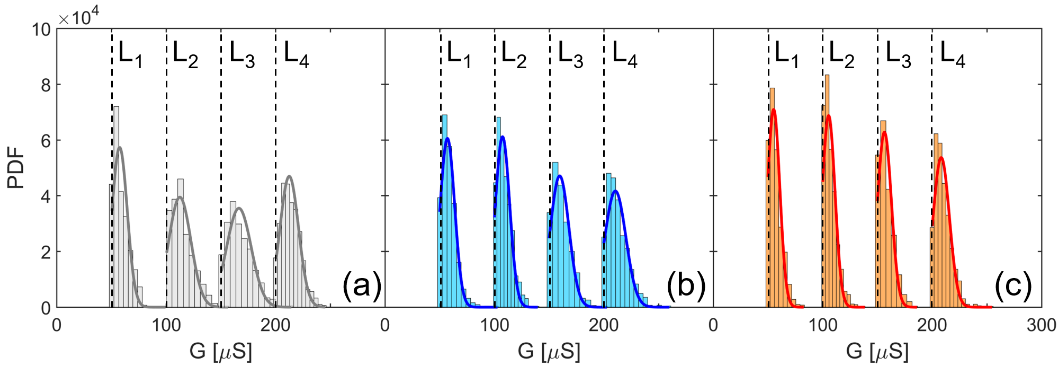


Figure 5.5: PDF of *after switching* G for simulated single programming event of 1000 devices for each level. The bars are the results of the simulations, while the solid lines are the gaussian PDFs calculated with the extracted μ_G and σ_G, shown in table 5.3. Three different programming techniques are compared, namely classic ISPVA (a), Finer V_{TE} (b) and Hybrid approach (c). The three cases are ordered by decreasing σ_G, meaning that the Hybrid approach, which involves the conductance control through the compliance current variation, is the best in limiting the programming D2D variability.

variability and investigate its impact on the network classification accuracy. The idea is therefore to replicate a single programming event for 1000 devices per each LRS level ($L_1 - L_4$) and provide the D2D distribution of the final conductance value. Such operation is carried out for experimental data, i.e. programming with ISPVA, and for the Finer V_{TE} and Hybrid approaches. Note that the HRS used in all the simulations is extracted by experimental data and exhibits $\mu_G = 10\mu S$ and $\sigma_G = 10\mu S$.

Regarding the experimental data, given the C2C distributions for a set of 1000 devices per level, the standard deviation σ_G is extracted by selecting the *after switching* conductance of a random cycle among the 1000 cycles performed in the endurance experiments. This operation is repeated several times and the average behavior is extracted. On the other hand, regarding the Finer V_{TE} and Hybrid approaches, the standard deviation σ_G is obtained by simulating

Programming technique	$L_1[\mu S]$		$L_2[\mu S]$		$L_3[\mu S]$		$L_4[\mu S]$	
	μ_G	σ_G	μ_G	σ_G	μ_G	σ_G	μ_G	σ_G
ISPVA (exp. data)	57.5	6.96	112.5	10.39	166.5	11.24	212.5	8.5
Finer V_{TE}	57.04	6.59	107.4	6.53	159	8.4	210	9.57
Hybrid	55.15	5.63	105.4	5.81	156.75	6.35	208.3	7.44

Table 5.3: Median G μ_G and standard deviation σ_G extracted or simulated for the three different programming techniques considered. Moving from the ISPVA to the Hybrid, σ_G decreases, suggesting a better control on variability as shown in figure 5.5

the programming event of 1000 devices per level, by selecting only one value of the parameter V_{set} (responsible for the C2C variability in the statistical model, as addressed in section 4.3) from the corresponding distribution, with no need to replicate the 1000 endurance cycles.

Figure 5.5 illustrates the probability density functions obtained in the aforementioned way for data (classic ISPVA approach (a)), Finer V_{TE} (b), and Hybrid (c). Moving from left to right, it is noticeable that distributions get thinner, meaning that σ_G becomes smaller and, therefore, that the different programming approaches improve the control on the final conductance value with respect to the classic ISPVA expressed by the experimental data. The Hybrid scheme is thus expected to be the best approach in limiting the D2D variability.

Table 5.3 summarizes the median G μ_G and standard deviation σ_G related to the distributions of figure 5.5. As expected, σ_G decreases for the Finer V_{TE} and Hybrid approaches.

Given the results shown in figure 5.5 and table 5.3, the network is tested on 10000 unseen MNIST images for 1000 times, each time by drawing a different value from conductance distributions shown in figure 5.5. The average inference accuracy calculated is 93.7%, 94.14% and 94.2%, respectively for

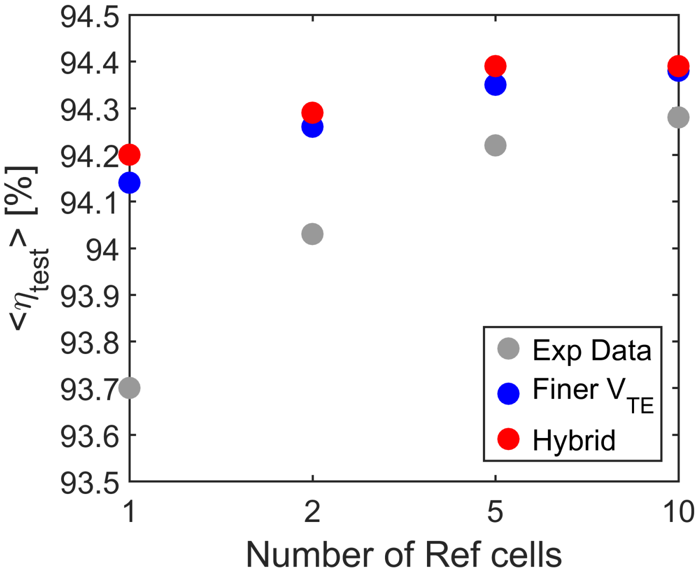


Figure 5.6: Average inference accuracy achieved as a result of the network test on 10000 unseen MNIST images including the expected programming variability as a function of the number of cells used to implement the reference values. Since Finer V_{TE} and Hybrid approaches show lower variability with respect to ISPVA, the corresponding accuracy is larger for all numbers of reference cells. Moreover, accuracy improves as such number increases, because the variability on the final G_{ref} is reduced.

ISPVA (experimental data), Finer V_{TE} and Hybrid. Such accuracy can be further improved by implementing the reference weight as the parallel of two or more conductances, i.e. by using one or more extra columns on the right of figure 5.3 (b). Indeed, the reference conductance G_{ref} is programmed into one of the 5 levels and show the related variability. Averaging two or more conductance values allows to achieve a more accurate final G_{ref}, thus obtaining a better control on the implementation of every weight. Figure 5.6 shows the simulated average inference accuracy as a function of the number of cells used to implement G_{ref}, for the ISPVA, Finer V_{TE} and Hybrid approaches.

As expected, the accuracy improves as the number of cells implementing the reference value increases for all the three techniques, since the variability on the final G_{ref} value is reduced. However, the curve for Hybrid case reaches a plateau around 94.4 %, which can be due to the less controllable HRS (L_0) variability.

Moreover, accuracy improves moving from ISPVA to Finer V_{TE} and to Hybrid, since the variability is lower in those cases, as shown in table 5.3. Note that the value for Hybrid in the best case of 10 cells used to implement the reference value (94.4 %) is very close to the classification rate achieved with 5 ideal discrete-valued weights with no variability (95.1 %) indicated in table 5.2. This means that variability is so limited in this case that it has an almost negligible impact on the network performances. For this reason, an Hybrid-like approach, where the device programming is tuned by the compliance current at a fixed top electrode voltage, is very promising to significantly mitigate the HfAlO RRAM variability issues during programming, thus making this RRAM device suitable for harwdare implementation of synaptic weights for neural network accelerators. Moreover, such technique can enable device programming with extra levels, which could additionally improve the network inference, getting closer to the ideal case of real-valued weights.

Conclusions

In this thesis, a new statistical model to mitigate multilevel programming variability of a 4kbit HfAlO RRAM array under program/verify algorithm was developed.

First, the physical reasons causing variability in RRAMs were addressed. Since the number of vacancies involved in the filament formation and disruption is discrete, the stochastic fluctuations on such number have a strong impact on the filament conductive properties. This results in the resistance variability and, in particular, in the increase of standard deviation (σ_R) with the resistive state (μ_R), in accordance with the Poisson statistics.

The work focused on the study of variability data measured during an endurance experiment carried out by applying a program/verify algorithm on the 4 kbit array of HfAlO RRAM devices with 1T1R structure. As expected, the increase of cycle-to-cycle (C2C) σ_R with μ_R, i.e. the 4 LRSs levels, was observed, but the distinctive feature of such data was found to be an inverse proportionality between σ_R and μ_R related to the same resistance state or level. To explain this result, set graduality, defined as the slope of the programming characteristic, was introduced. A correlation was found between the graduality of set transition and σ_R and μ_R, which allowed to understand that resistive cells with large set graduality showed a smaller variability and vice versa, thus suggesting that device-to-device (D2D) variability was caused by this distinctive parameter.

After understanding this physical dependence, a statistical model able to finely predict RRAM variability into the array was developed. The conditions imposed externally by the program/verify algorithm were replicated and used as inputs, while the simulated programming characteristic was chosen to be the model output. The model parameters were defined in a simple way, so that their impact on the simulated programming characteristic graduality was easy to determine. Then, the model was tuned on the experimental data. The variability was modeled by introducing a statistics in the model parameters, using a Monte Carlo approach. The model was tested by demonstrating to faithfully replicate, under the conditions imposed by the program/verify algorithm, both the C2C and the D2D programming variability of experimental data. In particular, a remarkable result was the model capability to well fit the data related to the different programming levels with no need to change some internal parameter, but only changing the external programming conditions, as it happens with the real array.

Finally, the statistical model of HfAlO RRAM was used to simulate the RRAM devices serving as synaptic connections into a multilayer neural network designed for classification of MNIST handwritten digit images. The network was first trained in software via the backpropagation algorithm, so that real-valued weights were computed. Then, such full-precision weights were quantized using 5 levels via an incremental quantization algorithm, thus achieving 5-level-valued weights. The adoption of such algorithm was very advantageous, since the network inference accuracy decreased by only $\sim 1\%$ with respect to the real-valued case. As a last step, 5-level weights were mapped into 5 resistance states with variability distributions predicted by the statistical model, to simulate the real performance of the neural network, which is expected to decrease as the variability increases. For this reason, different program/verify strategies were tested via the statistical model, in order to reduce device variability with respect to the experimental data. Finally, the image classification accuracy calculated under different programming techniques was compared. In particular, the programming approach which tunes the compliance current rather than the top electrode voltage was proved to be more efficient in reducing the variability and therefore in increasing the classification accuracy, bringing it very close to the ideal 5-level-valued case.

To conclude, the statistical model developed in this thesis is a useful tool to predict the variability of the 4 kbit HfAlO RRAM array. For this reason, it will be helpful to improve the programming accuracy of such array or larger size ones, which is a crucial step to enable the hardware demonstration of neural network accelerators with very high performances.

List of Figures

List of Tables

Bibliography

[1] G. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, pp. 114–117, 1965.

[2] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideovt, E. Bassous, and A. R. LeBlanc, “Design of ion-implanted MOSFET with very small physical dimensions,” *Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[3] S. Salahuddin, K. Ni, and S. Datta, “The era of hyper-scaling in electronics,” *Nature Electronics*, vol. 1, 2018.

[4] T. N. Theis and H.-S. P. Wong, “The end of Moore’s law: A new beginning for information technology,” *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[5] M. Horowitz, “1.1 computing’s energy problem (and what we can do about it),” in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[6] T. N. Theis and P. M. Solomon, “In quest of the "next switch": prospects for greatly reduced power dissipation in a successor to the silicon field-effect transistor,” *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2005–2014, 2010.

[7] W. A. Wulf and S. A. McKee, “Hitting the memory wall: implications of the obvious,” *ACM SIGARCH computer architecture news*, vol. 23, no. 1, pp. 20–24, 1995.

[8] H.-S. P. Wong and S. Salahuddin, “Memory leads the way to better computing,” *Nature Nanotechnology*, vol. 10, no. 3, p. 191, 2015.

[9] E. R. Kandel and J. H. Schwartz, *Principles of Neural Science: 1985*. Elsevier, 1985.

[10] D. Kuzum, S. Yu, and H. P. Wong, “Synaptic electronics: materials, devices and applications,” *Nanotechnology*, vol. 24, no. 38, 2013.

[11] D. Ielmini and H.-S. P. Wong, “In-memory computing with resistive switching devices,” *Nature Electronics*, vol. 1, no. 6, pp. 333–343, 2018.

[12] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] H.-S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, “Phase change memory,” *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010.

[15] G. W. Burr, M. J. Brightsky, A. Sebastian, H.-Y. Cheng, J.-Y. Wu, S. Kim, N. E. Sosa, N. Papandreou, H.-L. Lung, H. Pozidis *et al.*, “Recent progress in phase-change memory technology,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 2, pp. 146–162, 2016.

[16] D. Ielmini, S. Lavizzari, D. Sharma, and A. L. Lacaita, “Physical interpretation, modeling and impact on phase change memory (PCM) reliability of resistance drift due to chalcogenide structural relaxation,” in *2007 IEEE International Electron Devices Meeting (IEDM).* IEEE, 2007, pp. 939–942.

[17] D. Ielmini, “Resistive switching memories based on metal oxides: mechanisms, reliability and scaling,” *Semiconductor Science and Technology*, vol. 31, no. 6, 2016.

[18] D. Ielmini, R. Bruchhaus, and R. Waser, “Thermochemical resistive switching: materials, mechanisms, and scaling projections,” *Phase Transitions*, vol. 84, no. 7, pp. 570–602, 2011.

[19] S. Larentis, F. Nardi, S. Balatti, D. C. Gilmer, and D. Ielmini, “Resistive switching by voltage-driven ion migration in bipolar RRAM - Part II: Modeling,” *IEEE Transactions on Electron Devices*, vol. 59, no. 9, pp. 2468–2475, 2012.

[20] F. Nardi, S. Balatti, S. Larentis, and D. Ielmini, “Complementary switching in metal oxides: Toward diode-less crossbar RRAMs,” in *2011 IEEE International Electron Devices Meeting (IEDM).* IEEE, 2011, pp. 1–31.

[21] A. Belmonte, W. Kim, B. T. Chan, N. Heylen, A. Fantini, M. Houssa, M. Jurczak, and L. Goux, “A thermally stable and high-performance 90-nm Al_2O_3/Cu-Based 1T1R CBRAM Cell,” *IEEE Transactions on Electron Devices*, vol. 60, no. 11, pp. 3690–3695, 2013.

[22] H. Lee, P. Chen, T. Wu, Y. Chen, C. Wang, P. Tzeng, C. Lin, F. Chen, C. Lien, and M.-J. Tsai, “Low power and high speed bipolar switching with a thin reactive Ti buffer layer in robust $\mathrm{HfO_2}$ based RRAM,” in *2008*

IEEE International Electron Devices Meeting (IEDM). IEEE, 2008, pp. 1–4.

[23] V. V. Zhirnov, R. Meade, R. K. Cavin, and G. Sandhu, "Scaling limits of resistive memories," *Nanotechnology*, vol. 22, no. 25, 2011.

[24] T. Kawahara, K. Ito, R. Takemura, and H. Ohno, "Spin-transfer torque RAM technology: Review and prospect," *Microelectronics Reliability*, vol. 52, no. 4, pp. 613–627, 2012.

[25] R. Carboni, E. Vernocchi, M. Siddik, J. Harms, A. Lyle, G. Sandhu, and D. Ielmini, "A physics-based compact model of stochastic switching in spin-transfer torque magnetic memory," *IEEE Transactions on Electron Devices*, vol. 66, no. 10, pp. 4176–4182, 2019.

[26] A. D. Kent and D. C. Worledge, "A new spin on magnetic memories," *Nature Nanotechnology*, vol. 10, no. 3, p. 187, 2015.

[27] T. Mikolajick, C. Dehm, W. Hartner, I. Kasko, M. Kastner, N. Nagel, M. Moert, and C. Mazure, "FeRAM technology for high density applications," *Microelectronics Reliability*, vol. 41, no. 7, pp. 947–950, 2001.

[28] T. Böscke, J. Müller, D. Bräuhaus, U. Schröder, and U. Böttger, "Ferroelectricity in hafnium oxide thin films," *Applied Physics Letters*, vol. 99, no. 10, 2011.

[29] D. Ielmini and S. Ambrogio, "Emerging neuromorphic devices," *Nanotechnology*, vol. 31, no. 9, p. 092001, 2019.

[30] Y. Arimoto and H. Ishiwara, "Current status of ferroelectric random-access memory," *Mrs Bulletin*, vol. 29, no. 11, pp. 823–828, 2004.

[31] M. Trentzsch, S. Flachowsky, R. Richter, J. Paul, B. Reimer, D. Utess, S. Jansen, H. Mulaosmanovic, S. Müller, S. Slesazeck *et al.*, "A 28nm

HKMG super low power embedded NVM technology based on ferroelectric FETs," in *2016 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2016, pp. 5–11.

[32] H. Mulaosmanovic, J. Ocker, S. Müller, M. Noack, J. Müller, P. Polakowski, T. Mikolajick, and S. Slesazeck, "Novel ferroelectric FET based synapse for neuromorphic systems," in *2017 Symposium on VLSI Technology*. IEEE, 2017, pp. 176–177.

[33] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature Materials*, vol. 18, no. 4, pp. 309–323, 2019.

[34] G. W. Burr, R. S. Shenoy, K. Virwani, P. Narayanan, A. Padilla, B. Kurdi, and H. Hwang, "Access devices for 3D crosspoint memory," *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena*, vol. 32, no. 4, 2014.

[35] I. Micron Technology, "3D XPoint™ Technology." [Online]. Available: https://www.micron.com/products/advanced-solutions/3d-xpoint-technology

[36] H. Wu, X. H. Wang, B. Gao, N. Deng, Z. Lu, B. Haukness, G. Bronner, and H. Qian, "Resistive random access memory for future information processing system," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1770–1789, 2017.

[37] C.-L. Lo, T.-H. Hou, M.-C. Chen, and J.-J. Huang, "Dependence of read margin on pull-up schemes in high-density one selector–one resistor crossbar array," *IEEE Transactions on Electron Devices*, vol. 60, no. 1, pp. 420–426, 2012.

[38] C.-W. S. Yeh and S. S. Wong, "Compact one-transistor-N-RRAM array architecture for advanced CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 5, pp. 1299–1309, 2015.

[39] X. Wang, Z. Fang, X. Li, B. Chen, B. Gao, J. Kang, Z. Chen, A. Kamath, N. Shen, N. Singh *et al.*, “Highly compact 1T-1R architecture ($4F^2$ footprint) involving fully CMOS compatible vertical GAA nano-pillar transistors and oxide-based RRAM cells exhibiting excellent NVM properties and ultra-low power operation,” in *2012 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2012, pp. 6–20.

[40] V. Srinivasan, S. Chopra, P. Karkare, P. Bafna, S. Lashkare, P. Kumbhare, Y. Kim, S. Srinivasan, S. Kuppurao, S. Lodha *et al.*, “Punchthrough-diode-based bipolar RRAM selector by Si epitaxy,” *IEEE Electron Device Letters*, vol. 33, no. 10, pp. 1396–1398, 2012.

[41] I. Baek, D. Kim, M. Lee, H.-J. Kim, E. Yim, M. Lee, J. Lee, S. Ahn, S. Seo, J. Lee *et al.*, “Multi-layer cross-point binary oxide resistive memory (OxR-RAM) for post-NAND storage application,” in *2005 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2005, pp. 750–753.

[42] W. Y. Park, G. H. Kim, J. Y. Seok, K. M. Kim, S. J. Song, M. H. Lee, and C. S. Hwang, “A Pt/TiO2/Ti Schottky-type selection diode for alleviating the sneak current in resistance switching memory arrays,” *Nanotechnology*, vol. 21, no. 19, pp. 195–201, 2010.

[43] S. R. Ovshinsky, “Reversible electrical switching phenomena in disordered structures,” *Physical Review Letters*, vol. 21, no. 20, p. 1450, 1968.

[44] M. Imada, A. Fujimori, and Y. Tokura, “Metal-insulator transitions,” *Reviews of Modern Physics*, vol. 70, no. 4, p. 1039, 1998.

[45] C. Ho, H.-H. Huang, M.-T. Lee, C.-L. Hsu, T.-Y. Lai, W.-C. Chiu, M. Lee, T.-H. Chou, I. Yang, M.-C. Chen *et al.*, “Threshold vacuum switch (TVS) on 3D-stackable and $4F^2$ cross-point bipolar and unipolar resistive random

access memory," in *2012 IEEE International Electron Devices Meeting (IEDM).* IEEE, 2012, pp. 2–8.

[46] K. Gopalakrishnan, R. Shenoy, C. Rettner, K. Virwani, D. Bethune, R. Shelby, G. Burr, A. Kellock, R. King, K. Nguyen *et al.*, "Highly-scalable novel access device based on mixed ionic electronic conduction (MIEC) materials for high density phase change memory (PCM) arrays," in *2010 Symposium on VLSI Technology.* IEEE, 2010, pp. 205–206.

[47] D. Kau, S. Tang, I. V. Karpov, R. Dodge, B. Klehn, J. A. Kalb, J. Strand, A. Diaz, N. Leung, J. Wu *et al.*, "A stackable cross point phase change memory," in *2009 IEEE International Electron Devices Meeting (IEDM).* IEEE, 2009, pp. 1–4.

[48] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[49] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[50] G.-q. Bi and M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of Neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 1998.

[51] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, no. 5297, pp. 213–215, 1997.

[52] P. J. Sjöström, G. G. Turrigiano, and S. B. Nelson, "Rate, timing, and cooperativity jointly determine cortical synaptic plasticity," *Neuron*, vol. 32, no. 6, pp. 1149–1164, 2001.

[53] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, and G. W. Burr, “Recent progress in analog memory-based accelerators for deep learning,” *Journal of Physics D: Applied Physics*, vol. 51, no. 28, 2018.

[54] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[55] E. Ambrosi, “Characterization of resistive switching devices for memory and computing,” Ph.D. dissertation, Politecnico di Milano, 2019.

[56] G. W. Burr, R. M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti *et al.*, “Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element,” *IEEE Transactions on Electron Devices*, vol. 62, no. 11, pp. 3498–3507, 2015.

[57] E. Pérez, C. Zambelli, M. K. Mahadevaiah, P. Olivo, and C. Wenger, “Toward reliable multi-level operation in RRAM arrays: Improving post-algorithm stability and assessing endurance/data retention,” *IEEE Journal of the Electron Devices Society*, vol. 7, pp. 740–747, 2019.

[58] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, “High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm,” *Nanotechnology*, vol. 23, no. 7, 2012.

[59] Y. Sasago, M. Kinoshita, T. Morikawa, K. Kurotsuchi, S. Hanzawa, T. Mine, A. Shima, Y. Fujisaki, H. Kume, H. Moriya *et al.*, “Cross-point phase change memory with $4F^2$ cell size driven by low-contact-resistivity poly-Si diode,” in *2009 Symposium on VLSI Technology.* IEEE, 2009, pp. 24–25.

[60] S. Balatti, S. Ambrogio, D. C. Gilmer, and D. Ielmini, “Set variability and failure induced by complementary switching in bipolar RRAM,” *IEEE Electron Device Letters*, vol. 34, no. 7, pp. 861–863, 2013.

[61] D. Ielmini, F. Nardi, and C. Cagli, “Resistance-dependent amplitude of random telegraph-signal noise in resistive switching memories,” *Applied Physics Letters*, vol. 96, no. 5, 2010.

[62] S. Yu, R. Jeyasingh, Y. Wu, and H.-S. P. Wong, “Characterization of low-frequency noise in the resistive switching of transition metal oxide HfO_2,” *Physical Review B*, vol. 85, no. 4, 2012.

[63] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, “Statistical fluctuations in HfO_x resistive-switching memory: Part II - Random telegraph noise,” *IEEE Transactions on Electron Devices*, vol. 61, no. 8, pp. 2920–2927, 2014.

[64] D. Ielmini and V. Milo, “Physics-based modeling approaches of resistive switching devices for memory and in-memory computing applications,” *Journal of Computational Electronics*, vol. 16, no. 4, pp. 1121–1143, 2017.

[65] F. Nardi, S. Larentis, S. Balatti, D. C. Gilmer, and D. Ielmini, “Resistive switching by voltage-driven ion migration in bipolar RRAM-Part I: Experimental study,” *IEEE Transactions on Electron Devices*, vol. 59, no. 9, pp. 2461–2467, 2012.

[66] S. Ambrogio, S. Balatti, D. C. Gilmer, and D. Ielmini, “Analytical modeling of oxide-based bipolar resistive memories and complementary resistive switches,” *IEEE Transactions on Electron Devices*, vol. 61, no. 7, pp. 2378–2386, 2014.

[67] D. Ielmini, “Modeling the universal set/reset characteristics of bipolar RRAM by field-and temperature-driven filament growth,” *IEEE Transactions on Electron Devices*, vol. 58, no. 12, pp. 4309–4317, 2011.

[68] E. Pérez, A. Grossi, C. Zambelli, P. Olivo, R. Roelofs, and C. Wenger, “Reduction of the cell-to-cell variability in Hf_{1-x} Al_x O_y based RRAM arrays by using program algorithms,” *IEEE Electron Device Letters*, vol. 38, no. 2, pp. 175–178, 2016.

[69] A. Fantini, L. Goux, R. Degraeve, D. Wouters, N. Raghavan, G. Kar, A. Belmonte, Y.-Y. Chen, B. Govoreanu, and M. Jurczak, “Intrinsic switching variability in HfO_2 RRAM,” in *2013 5th IEEE International Memory Workshop*. IEEE, 2013, pp. 30–33.

[70] S. Ambrogio, S. Balatti, A. Cubeta, and D. Ielmini, “Statistical modeling of program and read variability in resistive switching devices,” in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 2029–2032.

[71] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, “Statistical fluctuations in HfO_x resistive-switching memory: part I-set/reset variability,” *IEEE Transactions on Electron Devices*, vol. 61, no. 8, pp. 2912–2919, 2014.

[72] R. Ongaro and A. Pillonnet, “Synthetic theory of Poole and Poole-Frenkel (PF) effects,” *IEE Proceedings A (Science, Measurement and Technology)*, vol. 138, no. 2, pp. 127–137, 1991.

[73] C. Zambelli, A. Grossi, P. Olivo, D. Walczyk, T. Bertaud, B. Tillack, T. Schroeder, V. Stikanov, and C. Walczyk, “Statistical analysis of resistive switching characteristics in ReRAM test arrays,” in *2014 International*

Conference on Microelectronic Test Structures (ICMTS). IEEE, 2014, pp. 27–31.

[74] V. Milo, C. Zambelli, P. Olivo, E. Pérez, M. K. Mahadevaiah, O. G. Ossorio, C. Wenger, and D. Ielmini, "Multilevel HfO_2-based RRAM devices for low-power neuromorphic networks," *APL Materials*, vol. 7, no. 8, 2019.

[75] A. Fantini, L. Goux, S. Clima, R. Degraeve, A. Redolfi, C. Adelmann, G. Polimeni, Y. Y. Chen, M. Komura, A. Belmonte *et al.*, "Engineering of $Hf_{1-x}Al_x$ O_y amorphous dielectrics for high-performance RRAM applications," in *2014 IEEE 6th International Memory Workshop (IMW).* IEEE, 2014, pp. 1–4.

[76] S. Yu, B. Gao, H. Dai, B. Sun, L. Liu, X. Liu, R. Han, J. Kang, and B. Yu, "Improved uniformity of resistive switching behaviors in HfO_2 thin films with embedded Al layers," *Electrochemical and Solid State Letters*, vol. 13, no. 2, p. 36, 2009.

[77] Y. L. Tong, *The multivariate normal distribution.* Springer Science & Business Media, 2012.

[78] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.

[79] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang *et al.*, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature Communications*, vol. 9, no. 1, pp. 1–8, 2018.

[80] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong *et al.*, "Face classification using electronic synapses," *Nature Communications*, vol. 8, no. 1, pp. 1–8, 2017.

[81] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless CNNs with low-precision weights," in *International Conference on Learning Representations (ICLR)*, 2017.