A study of using STT-MRAM as Memory PUF: Design, Modeling and Quality Evaluation

BY

PAOLO VINELLA

B.S., Politecnico di Torino, Turin, Italy, 2012

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2015

Chicago, Illinois

Defense Committee:

Wenjing Rao, Chair and Advisor
Zhichun Zhu
Fabrizio Bonani, Politecnico di Torino

Dedicated to my Parents,

who have always provided me with the best and unconditional moral and economical support, enabling me to achieve results I am proud of and accomplish my goals at School as well as in life. I will hardly be thankful enough for what you have done.

ACKNOWLEDGMENTS

My most truthful gratitude goes to the Academic Committee, including Staff and Faculty, at both University of Illinois at Chicago and Politecnico di Torino, for providing me with the great opportunity of being part of the joint program TOP-UIC for my Master Degree. An ambitious and challenging path that has enriched me on both cultural and academic levels.

I am grateful to Prof. Rao, my Thesis Advisor at UIC, for accepting me into her Research Group and assisting me through the entire thesis work, and for deeply believing in my potential, offering me both a Teaching Assistantship and a Research Assistantship at UIC. An additional experience that I have more than welcomed and has surely had a huge impact on further enriching and amplifying my skills.

A special thanks to Prof. Bonani, not only Thesis Advisor representing Politecnico di Torino, but also twice my Professor of two among my past Academic courses, held both in Italy and in China.

A particular mention goes also to Soroush Khaleghi, who offered me his precious consulting in part of this thesis work, especially in Section 3.7, while working on his PhD at UIC in the research group of Prof. Rao.

Last, but absolutely not least, my sincere gratitude needs to reach Miss Lynn Thomas as well, Project Coordinator of TOP-UIC for the Chicago side of the Program, and the whole Staff of the ECE Dept. at UIC for supporting me with their care and impeccable professionalism.

PV

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

FIGURE **PAGE**

LIST OF FIGURES (Continued)

FIGURE **PAGE**

LIST OF FIGURES (Continued)

SUMMARY

The contribution of the present thesis work is to offer an extensive analysis and new methodologies and guidelines on the use of **Magnetic Memory (MRAM)**, in its most advanced, recent and performing flavor (STT-MRAM) as a tool to realize a unique and unclonable binary signature out of a magnetic memory chip, eventually implanted as a part of a more complex digital system device, providing then a low-cost and robust tool to generate a signature out of each of the devices of a given manufactured production lot.

The need of generating a signature comes from the new dictations of **Hardware Security field**, a set of practices and rules devised to fight intellectual property theft and electronic chips piracy. The technique used to apply these principles to magnetic memory is called **Physically Unclonable Function**, or **PUF** for short.

Chapter 1 briefly provides a qualitative overview on MRAM and PUF, describing their respective potentials and advantages.

In **Chapter 2**, a deep background on the topics of Magnetic Memory and Hardware Security is offered. Especially when it comes to MRAM, this technology is very new and experimental: there is then no unified literature on the topic and the current research outcomes, even though very challenging and original, do not follow any common standard. Chapter 2 has then been written, after collecting information from several tenth between books and papers on the topic, to guide the reader on magnetic memories, from their main working principles towards the current applications, providing a hopefully clear and ordered path of the current experimental

trends, classifying them according to the different fields of use in the digital electronics domain. Also, some background on Hardware Security and PUF is offered.

Chapter 3 reflects the main thesis goals, achievements and contributions to the fields of MRAM and Hardware Security, containing the following results:

1. Analytical study of a proposed DC electrical model of the equivalent resistance of a magnetic memory (MRAM) cell, to verify and test its boundaries, statistical analysis and application to physical unclonable functions. Focus on the associated equivalent resistances distribution of the two parallel/antiparallel states of a magnetic memory and dependence on physical parameters (MRAM oxide thickness and process variations);
2. Extension of the previously proposed work [17] [18] on MRAM as Physical Unclonable Function, consisting of two circuit layouts, based on a string of MRAM, to an entire memory matrix – in order to use the very same main memory of a digital system to implement a PUF, assuming MRAM to become the next generation on-chip main memory;
3. Further extension with respect to point 2) using two new electrical sensing models;
4. Analysis of the results for all the four cases, evaluating the PUF quality (in terms of uniqueness and stability), and influence of the variability of some parameters of an MRAM on the PUF quality itself, using the concept of Hamming distance, including bit-error rate;
5. Augmenting the proposed PUF model for Hardware Security: STT-MRAM not only becomes a tool to generate a unique Signature ID for a given memory chip but also a way to support chip Authentication (a so called Strong PUF).

CHAPTER 1

INTRODUCTION

Magnetic memories, for short MRAM, represent a novel class of digital memory technology able to store data playing with magnetic field effects on ferromagnetic nanomaterials, devised to overcome current limitations driven by the scale down of current CMOS-based memory devices. Its most recent flavor, p-STT-MRAM, promises to become a unique, universal, fast memory technology applicable through all the memory hierarchy stack of a digital computer system, combining all the advantages of current memories at little cost and design effort. MRAM technology enables super fast speeds in both read and write operations, non-volatility, unlimited endurance, low cost and high density.

Hardware Security is a new field, applied to electronic systems, representing a set of design practices and rules that contribute in the intellectual property protection, fighting electronic chips piracy and design thefts. Its most refined implementation, called Physically Unclonable Function (PUF), enables to exploit some random physical properties of a given electronic device, subject to process variation due to fabrication tolerances, to generate a strong and unique signature out of a particular integrated circuit, for (but not limited to) security and authentication purposes.

Both MRAM and PUF are very new concepts of digital electronic design: MRAM is a new promising memory device, while PUF is a new application needed for Hardware Security reasons. They are originally not necessarily correlated, and the contribution of this thesis work is to study them both and tie them together, improving the first ideas over the study of MRAM as PUF conducted by [17] and [18] with a more extensive and rigorous analysis over MRAM process variation to validate its suitability as PUF and proposing new electrical models to generate a Physically Unclonable Function out of an MRAM chip.

The way a magnetic memory cell stores a binary information (a logic 0 or 1) is associated to its behavior as binary variable resistor (respectively low resistive state and high resistive state). It has been seen, realizing early prototypes of magnetic memory, that both states do not exhibit a nominal value only but are accompanied by a rather wide standard deviation. This peculiarity, fought for digital applications, can be used to realize a Physically Unclonable Function to generate a binary signature out of a MRAM chip.

The work done so far on STT-MRAM as PUF is very little and extremely preliminary, and finding a suitable model to describe the resistive states of MRAM both for digital and PUF applications, taking into account process variation, is something never deeply investigated before – current models have either been proposed theoretically or they come from specific fitting analysis of very narrowed types of magnetic memories only.

CHAPTER 2

BACKGROUND

The focus of the present thesis work is on employing **magnetic memory**, a non-volatile digital memory technology, in the field of **Hardware Security**, in order to verify its capabilities into securely identifying a chip design of a particular produced chip lot, by generating a unique signature for each of them, aimed at preventing devices cloning and design theft.

Different methods exist to implement hardware security features within a chip, most of which are digital (given the intrinsic digital nature of nowadays' chips). In the present thesis work, the **Physical Unclonable Function (PUF)** method is analyzed.

The physical device employed to achieve this goal is the **magnetic-tunneling junction, or MTJ**, in its most advanced flavor at the moment: the **spin-torque transfer magnetic memory (STT-MRAM)**.

The goal of this background chapter is to provide an extensive introduction on MRAM with a stronger accent on Hardware Security and PUF fields. The related notions are presented and organized as follows:

- A brief analysis on the future trends of digital memories is presented, highlighting the main reasons and current limitations loading to the transition to the next-generation memories, where 'magnetic' and 'non-volatile' are both becoming two adjective of a viable solution.
- The main working principles behind magnetic memories is introduced, focusing then on the particular memory technology used in the present thesis work: the STT-MRAM.

- Further remark is posed on STT-MRAM from an higher-level standpoint. A DC electrical model, together with the main characterization features of this technology especially for PUF applications, are introduced.
- The present chapter moves then into investigating the field of Hardware Security, typical possible solutions to achieve chip authentication and the role of PUF, presenting some among the most popular current PUF ideas and implementations.
- Finally, the current work and ideas, still preliminary, on specifically employing STT-MRAM as PUF are presented. This last part represents an introduction as well as the main connection with next Chapter and the motivations of this thesis work.

2.1 Computer memories: trend towards non-volatile technology

Digital computer memories have experienced a quite fast evolutions in the last few decades. CPU caches are reaching incredible speeds and higher densities of integration, and the same promises are carried out by volatile main memory (RAM). Above all, the most abrupt and remarkable evolution during the last ten years has interested secondary storage: great developments on Flash memories are leading to a gradual transitioning from the slower noisier hard disk drives (HDD) to solid-state drives (SDD), effectively overcoming one of the most frustrating bottlenecks in computer systems.

Solid state memories, based on advanced evolutions of MOS semiconducting technology revisited to introduce non-volatile features, have driven the development of more efficient embedded systems, after some preliminary attempts on using miniaturized hard-disk drives did not lead to any feasible result.

Nowadays **embedded system** does not mean any longer low performance super-specialized chip: recent evolutions on semiconductor technology are seriously changing this scenario. Low power devices, combined with very remarkable performance and more careful chip design, enable to realize microprocessors which can feature high-speed multicore capabilities without need of any active power dissipation and very low power requirements. Traditional personal computers are being used less and less by people in everyday's life, and transition towards handheld devices as tablet, smartphones and intermediate categories (the so called *phablets*) is something that we experience more and more. This is what an embedded systems is starting to sound like in the last few years. The future, in my opinion, will be a combination of super portable and wearable computing systems linked, through an high speed and reliable internet connection, to cloud services and remote computation. Personal computers, in their traditional definition, will gradually disappear or at least be confined into a very reduced set of specific applications.

This is the reason why, nowadays, there is less and less room into choosing between **speed and energy saving: even in embedded systems, we need them both**. Although this may sound in total opposition with respect to the traditional trade off between high performances and power consumption, researchers are pushing towards new memory classes which are able to achieve both.

One of the most promising trends is **non-volatile memory**: a particular class of digital memory devices, proposed in different flavors, that tries to combine the best aspects of any current memory technology. Magnetic memory, MRAM, is a type of non-volatile memory, and the present thesis work focuses on implementing Hardware Security techniques on top of its most advanced version: the Spin-Torque Transfer Magnetic Memory: STT-MRAM.

2.1.1 Digital memories hierarchy general considerations

Thinking computer architecture means thinking about memory organization and sub-systems. Almost every digital chip requires a certain amount of memory to save, either temporarily or permanently (with respect to time or power on/off), some data. In nowadays computer systems, the memory hierarchy is structured as: CPU registers, cache memory (typically more than one level), main memory (or RAM) and secondary storage (HDDs or, more recently, SSDs).

Although this hierarchy allows to get, at each level, a reasonable trade-off between performance, area, cost and power, the need of enabling an intercommunication among these levels implies using controllers and real-time translation between hardware and software protocols used at each stage. In the traditional approach, it is impossible to get a computer system working only with one kind of memory. For example, using RAM for everything would imply at the same time extremely slow CPU registers, as well as poor density as secondary storage and lack of non-volatility.

Performance limitations are equivalent to bottlenecks. Most of the time the processor waits (issuing NOPs) for new data to be fetched into its high-speed registers. The outer data, requested by the CPU, is present in slower memory systems with respect to CPU SRAM cache, like DRAM, which may in turn need to load data from other slower memory devices like hard drives. An evident limitation to the overall system performances.

2.1.2 Future memory trends: the "dream memory"

None of the current memory technology can be classified as the best. Cache memories are small and fast, but rather expensive and volatile. The main memory is denser and less expensive, but slower (requires refresh) and volatile. Finally, secondary storage is denser, non-volatile and even less expensive, but is far slower and not long-term reliable (for example, floating gate degradation issues in Flash technology). Researcher are constantly working on improving memory technologies, and the desire of a "**dream memory**" could soon become completely true: a universal technology overcoming the limitations of all the current memory types and combining all of them together. A unique, universal, fast memory technology applicable to all memory hierarchy levels, combining only the advantages of current memories at a small extra cost and reduced additional design efforts:

- **High speed of operations**: time of access a given memory location reduced to a few nanoseconds, in both reading and writing modes. We are basically requesting cache-level performances;
- **Low power**: the memory does not consume a lot of dynamic power during its operations (read, write) and especially the static (standby) power is reduced ideally to zero. This is a feature typically presents in the lower tiers of current memory hierarchy;
- **High density of integration**: being able to pack a rather large memory capacity into as little area as possible. A mandatory requirement for current cache and RAM;
- **Reliability, high endurance**: a prerequisite for any memory, employing memory error detection and/or correction with minimum hardware overhead;
- **Non-volatility**: ability to store information even when the power is completely shut down and in reduced power states. Nowadays, a typical feature of secondary storage only;
- **Absence of refresh**: memory cells are able to retain their content through the entire cycle of operation at power on without leakage current causing the need of refreshing the information, the principal cause of current RAM main memory slow-down.
- **Low cost**: both in design and in manufacturing phases. A requirement strictly related to the design complexity and the materials used.
- **Integration on top of silicon technology**: avoid to use a completely different/new technology to physically implement a new memory but rather exploit current silicon technologies, which are quite inexpensive.

ITRS is a committee defining standards for an **International Technology Roadmap for Semiconductors**. Inside its most recent report, **Edition 2013** [1], the main predictions and trends reflect the considerations highlighted so far. For the *near term, 2013-2020*, it is expected that designers and researchers work on CMOS efficient scaling, endurance, noise margin reduction and solve memory latency gap into systems. Moreover, for the *long-term trend, 2021-2028*, ITRS forecasted that new memory structures will need to be identified and implemented, replacing both DRAM and SRAM with more efficient solutions, indicating **non-volatile memories** as possible candidates. At the same time, the typical desired combination of requirements of “low-cost, high-density, low-power and fast-latency memory” [1] are required especially for large systems. A comparison of speed versus capacity (density) of memory technologies is qualitatively shown in Figure 1. **Non-volatile memories** look like a promising solution to address ITRS concerns and roadmap guidances as well as a valid candidate to become a universal memory. Possibly, the dream memory we are looking for. Figure 1, in particular, shows the two generations of MRAM: the first generation, slower and less dense (Field MRAM) and the second and last generation, STT-MRAM, overcoming most of the limitations of the Field approach especially in writing speed and power consumption, as well as density.

2.1.3 MRAM as non-volatile universal memory candidate

Among the possible types of non-volatile memories, magnetic memory (MRAM) looks like a very viable solution. Based on a particular stack of materials called magnetic tunneling junction (MTJ), it brings a unique combination of advantages establishing a new class of digital

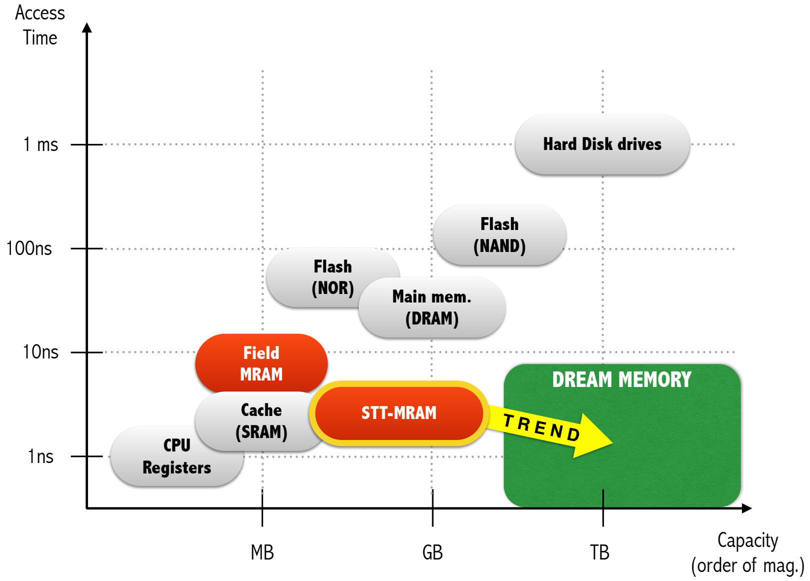


Figure 1. Main digital memories: Capacity vs Access Time. In red, magnetic memories

memory. A summary of these advantages is well documented in the book *Emerging Memories Technologies and Trends* [2]. In a few words, MRAM allows to achieve:

- An **unlimited number of read and write** operations;
- Both **read and write access times orders of magnitude faster** than conventional Flash and EEPROM, reaching, in the most advanced implementations, even SRAM speeds of the order of few nanoseconds;
- **High memory cells density**: area of occupation is reduced to a minimum thanks to a vertical stacking technique with respect to the access transistor of the memory cell itself.

MTJ fits on top of a single transistor, thus the memory cell size is comparable (and even better) to DRAM or Flash technologies;

- **Low power, low operation voltages**; absence of refresh operation, plus write operations do not require an erase to be done in advance;
- **Non-volatile storage**, enabling implementations as *instant-on* main memory;
- **Immunity to radiation**, a strict requirement for military and space applications;

Moreover, the book [2] lists the main players on the MRAM market: IBM, Sanyo, Infineon and Motorola are worth being mentioned. In particular, IBM can be considered the pioneer of the field with its first viable solution realized in 1997 [3].

Currently, researchers are working on overcoming the **main limitations of MRAM**: during write operations, both power consumptions and latency are still higher than SRAM. This factor, combined with nanoelectronics-level considerations, also limits the density of integration and the GigaByte barrier still needs to be broken.

2.2 MTJ as Magnetic Memory using Magnetoresistance: MRAM

2.2.1 Using electromagnetism to store digital information: main idea

Every digital memory technology relies on a specific microscopic working principle. For example, DRAM is based on capacitive effects, Flash memories on charges trapped on a floating gate within a thin oxide. Magnetic memory (MRAM) is based on **electromagnetism**, and in particular playing with the magnetic field effects on ferromagnetic materials, to store data in a binary form. MRAM is physically realized as a stacks of ferromagnetic materials. Each of these ferromagnet is characterized by a well defined magnetization M, a macroscopic vectorial quantity characterized by a direction and an intensity. The key point for MRAM is about controlling and changing in a stable manner the magnetization direction of a ferromagnetic layer ("switching").

2.2.2 Magnetoresistance effect and Spin-Torque transfer

Magnetization needs to be translated to an higher-level concept in order to be exploited in digital applications. The **magnetoresistance effect** is the link between nanoscale and digital level, and expresses the observed resistance across a particular stack of materials including ferromagnetic ones, while playing with their magnetization direction.

MRAM are based on the discovery, in 1975, of the Tunneling Magnetoresistance, or TMR [4]. Let us consider a stack made, from the top through the bottom, of:

- A first ferromagnetic (FM) layer, called **free layer** or storage layer, whose magnetization vector M_1 is free to switch under the influence of an external quantity (a magnetic field);
- A thin insulating layer, **oxide**, through which electrons tunneling happens;

- A second ferromagnetic (FM) layer, named **pinned layer** or fixed layer, whose magnetization vector M_2 is always kept fixed in any operating condition.

This structure is called Magnetic Tunneling Junction, or MTJ. Both FM layers are characterized by a proper magnetization vector M aligned along a precise direction called easy axis. An external applied magnetic field causes the free layer to switch its direction by 180 degrees, letting it be either parallel or antiparallel to the magnetization vector of the pinned layer. Two possible states are then defined:

- Parallel configuration (P): M_1 and M_2 are in phase, resulting in a parallel state associated resistance R_p rather low;
- Antiparallel configuration (AP): M_1 and M_2 are out of phase by 180 degrees, resulting in an antiparallel state associated resistance R_{ap} rather high.

These states of high and low resistance can be used to store a digital binary value, 0 or 1. For convention, R_p corresponds to the low state (0) and R_{ap} to the high state (1). Figure 2 sums up the MTJ structure and its interesting property to behave as a **variable (binary) resistor**.

The **TMR ratio** is the expression of the distance between the two resistive states, defined in Equation 2.1. Nowadays the TMR can go up to more than 100% at room temperature. Using thin films causes an abrupt P/AP difference, thus an high TMR. To make a magnetic memory reliable, the TMR must be as high as possible in order to be able to discriminate correctly the two digital levels leaving very little room to sensing errors.

$$TMR_{\%} = \frac{\Delta R}{R_p} \cdot 100 = \frac{R_{ap} - R_p}{R_p} \cdot 100 \tag{2.1}$$

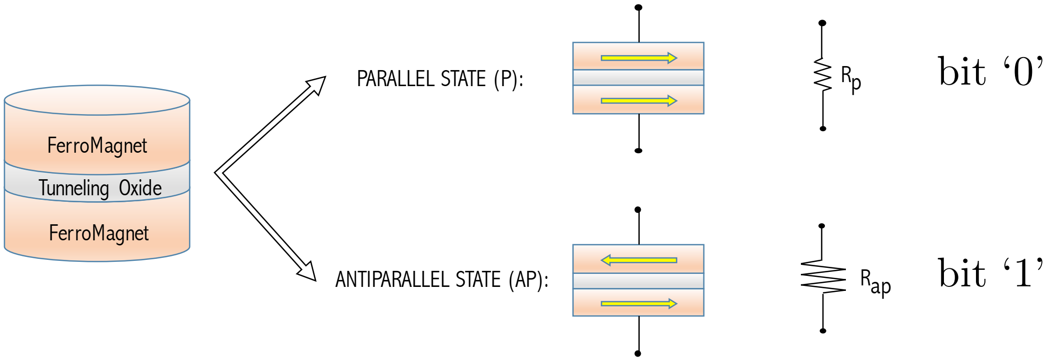


Figure 2. Magnetic tunneling junction: basic structure, P and AP states from the analog (resistor) and digital (bit value) perspective.

2.2.3 STT-MRAM: spintronic meets "second generation" magnetic memory

The TMR effect has been used for years in the head of hard disk drives to read and write data in the spinning plates. First applications to MRAM began with the so called **Field MRAM**, which used a magnetic field to toggle the magnetization of the free layer. Unfortunately, this approach had several tradeoffs:

- The need of additional lines in a memory array to discriminate read operations from writes [1];
- High write power, since a large current in a wire placed nearby the free layer of the MTJ needed to be used to trigger a magnetic field whose intensity is able to realize the switching;

[1] Switching is actually the write operation in a magnetic memory

- Disturbance problems: the magnetic field used to write a cell caused potential switching of nearby cells.

In the last few years, the idea of implementing TMR effect into MRAM has thus shifted towards a novel approach: **spintronics**. **Spin-torque transfer MRAM**, **STT-MRAM**, is based on letting a current flow through the MTJ device itself to realize the switching of the magnetization of the free layer. This means that a current flowing through the device can both sense the resistance (for reading operation) and program (write operation) the memory.

Spin-torque transfer was discovered only in 1996. The current is seen as a set of spin-up and spin-down electrons. While flowing through a non-magnetic material, nothing happens to both populations of electrons. On the contrary, when the current crosses a ferromagnet, they are both subject to interesting alterations: the electrons whose spin is aligned with the magnetization direction grows in terms of population size, and the growth is compensated by an equal loss of electrons from the opposite spin population. While the magnetization vector of a ferromagnet can change the populations of spin-up and spin-down electrons, it has been observed that also the opposite principle can arise: a polarized current can switch the magnetization itself. This behavior is exactly the one used to drive the switching in a STT-MRAM.

With respect to Field MRAM, STT-MRAM does not use any magnetic field: the write selectivity is very precise (since the spin-torque effect happens within a cell without influencing surrounding ones), thus magnetic interferences are not present anymore. Write current is also much lower than the one required by Field MRAM as well as the architecture is simpler (no longer multiple wires to discriminate between read and write operations). Moreover, when device scaling occurs, transistors length decreases as well, leading to a reduced maximum drain-to-source current: fortunately, the required current to perform MRAM cell switching scales down very nicely with MTJ size shrinking.

2.2.4 STT-MRAM digital memory cell

The typical structure of a memory cell constituted of STT-MRAM is depicted in Figure 3. Each cell, in its basic configuration, is called 1T1MTJ being constituted by a single access transistor and and a single MTJ stack. Figure 3 also shows qualitatively the dependance of the binary MTJ resistance on the applied bias to the cell, and the hysteresis realized by the switching behavior. When the applied voltage rises above a well known quantity (typically a few hundreds mV) the switching between the two states (P to AP or viceversa) happens, and it is driven by the polarity of the voltage. In order to program the cell, either a voltage or a current source can be used. The red arrows in Figure 3 indicate the required direction of the current flowing across the device to trigger the spin-torque transfer effect.

Basic memory operations are realized as follows:

- Memory cell selection happens through the access transistor. Its gate is driven by the WL (word line) signal;
- To write in the memory, a current is applied. The direction of the current determines how the cell is programmed.
- Reading operation happens sensing either the current or the voltage across the MTJ, which is simply treated as a variable binary resistor.

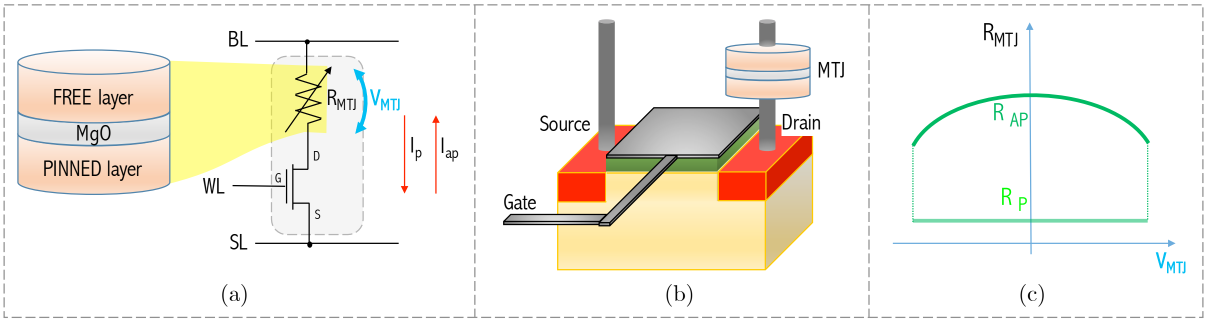


Figure 3. STT-MRAM basic digital memory cell, constituted of an MTJ stack on top of an access MOS transistor. In (a) the electrical scheme, in (b) the simplified physical layout and in (c) the dependance of the parallel and antiparallel resistance on the voltage drop insisting on the MTJ.

It is extremely important to keep the current flowing across the MTJ stack during reading phase much smaller compared to the one required for writing operation, in order to avoid the MRAM cell changing its value while a read operation takes place.

A qualitative idea on how a MRAM memory array looks like is illustrated in Figure 4.

A further advantage of MRAM is the density of integration: with respect to current memory technology competitors, in fact, the MTJ stack is grown immediately above the drain contact of its access transistor, translating into a 3D stacked structure that starts from the bit line all the way through the MTJ and the MOSFET. Building MRAMs simply means adding a few thin extra layers of materials over a MOS-based integrated circuit, leading to a density of integration extremely high.

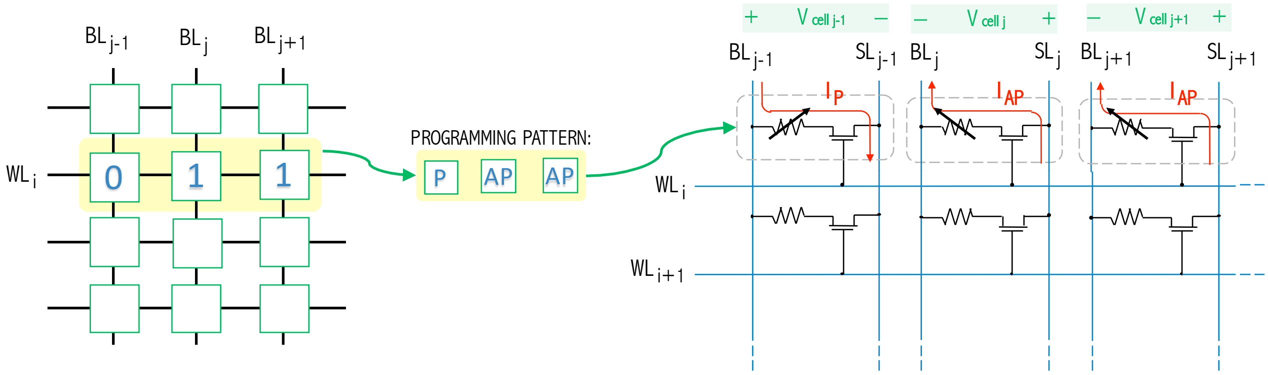


Figure 4. STT-MRAM memory array: WL is the word line employed to activate a given row, then read and write operations happen exploiting the BL (bit line) and SL (source line), which provide access to the MTJ. In this particular example a '0 1 1' binary string has to be written.

2.2.5 Main applications: consolidated work and novel trends

In the last few years, different research groups and memory Companies spent remarkable efforts to design, build and test a multitude of MRAM ideas and applications. In this subsection, the most populars ones are briefly listed, together with some additional information on MRAM. The first implementations regarded the first generation, Field MRAM, then researchers started to transition towards the adoption of STT-MRAM, given that more recently this second generation magnetic memory has become more mature and stable.

Nanoscale ideas

A magnetic tunnel transistor has been proposed [5] based on a double tunneling junction.

A sort of BJT based on hot-electrons transport to generate the collector current, making use of two MTJs within the emitter/base/collector junction and exploiting the principles behind spintronic. More recently, a new pseudo-spin-MOSFET has been fabricated and characterized [6]: a circuit using an ordinary MOSFET plus an MTJ to reproduce functions of spin transistor, offering a transconductance which is function of the status of the MTJ stack (parallel or antiparallel). To enhance spintronic applications improving the TMR ratio, MTJ tunneling oxide based on aluminium has been replaced with magnesium, in a stack including Fe/MgO/Fe: Yuasa et al reported earlier in 2004 [7] a giant TMR ratio up to 180% at room temperature, using MTJs based on single-crystal stacks using MgO as oxide. Nowadays, every MTJ structure is based on complex sets of layers, but the oxide is still MgO.

A universal memory

The obvious application of STT-MRAM is memory. In 2006, IBM was able to realize a 16Mb Field MRAM prototype chip [8], after experimenting the performance and playing with the characteristics of magnetic tunnel junctions development. The latest advancement in using STT-MRAM as the new universal memory has been achieved by Toshiba with the p-STT-MRAM (perpendicular) [9]. This technology is currently the closest to the idea of universal memory, and it is based on having the magnetization vector in the two ferromagnetic layers perpendicular to their surface rather then parallel, becoming finally a promising solution for high density, high speed, non-volatile random access memory: density similar to DRAM, non-volatility and speeds comparable with SRAM, as well as a theoretically infinite write endurance and low power, yet high speed switching.

Unbalanced MTJ flip-flop and FPGA

A new class of circuits, built by adding an MTJ the the source side of the nMOS of each of the two cross-coupled inverters constituting an SRAM memory element. MTJs are always programmed in a complementary state and until metastability is triggered the flip-flop keeps latching a defined binary value. Triggering the metastability causes the configuration stored in the MTJ couple to be loaded in the flip-flop replacing its previous content. The device can be seen as an hybrid of volatile and non-volatile memory element, where the volatile part is exactly the same of an SRAM while the two MTJs add the capability to store a configuration that can resist to long power-off periods and can be loaded at any time to replace the current SRAM cell (flip-flop) stable state. Furthermore, while the SRAM is holding its current configuration to perform a given memory operation, the MTJs can be simultaneously programmed to a different state (shadow programming) and the new configuration is immediately ready to be loaded after triggering the next metastability. This huge advantage found immediate applications into experimental FPGA, when the reconfiguration of memory blocks defining interconnections and logic blocks behaviors needs to happen as fast as possible. Remarkable results are illustrated in [10] using traditional Field MRAM, while [11] also adopts a more recent version of Thermally Assisted Switching which ensure faster operations over traditional Field MRAM.

MTJ-based nonvolatile Logic in Memory

Logic-in-memory architecture is a rather old concept: including storage elements embedded and distributed over a CMOS plane getting rid of large delays and power required every time logic needs to access memory modules (typically over an external bus). MTJ is a good memory fit for this goal. More or less complex digital devices (from basic NOR or NAND to more complex LUT implementations) can be built and, together with the normal inputs and outputs, they come with some configurations inputs which instruct the logical block to its function. First, configuration inputs program some data stored into MTJ devices, which allows to execute an arbitrary logic operation taking the data inputs and providing the related output. Typical applications include not only Nonvolatile FPGA but also CAM (content-addressable memories). A remarkable example of Logic-in-Memory architecture is offered by the work of Matsunaga et al [12], describing how a non-volatile full adder can be implemented using this approach.

Associative Computing

Logic-in-Memory enabled new CAM to be built. The work of Guo et al in [13] describes how CMOS and MTJ can work together to realize a TCAM (ternary content-address memory) key-data, using STT-MRAM and integrable with DDR3 protocol (the device can be plugged into a normal DDR3 socket). The goal of TCAM is processing in memory, reducing access times to an external slower memory, and offering not only CAM functionality (rapid search of data) but also processing, within the same memory chip, the output (a reduced instruction set is offered to the user) before being transferred eventually to the CPU, which is now simply in charge of handling the result rather then processing computation on raw data.

Normally-Off Processor with p-STT-MRAM

The use of the advanced version p-STT-MRAM proposed by Toshiba in a more recent sub-30nm flavor enabled an effective reduction in power of cache memory within a CPU. Previous generation of conventional STT-MRAM, when used as cache memory, caused an extremely high active energy in the CPU (due to rather high write energy) although ensuring an almost-zero standby power. The breakthrough by Toshiba's advanced p-STT-MRAM in 2012 [14] and 2014 [15] enabled a decrease in power for short CPU standby state while a combination with power gating techniques ensures the same advantage also for longer CPU standby states, testing the results on HP-mobile processors. Overall, stand-by power is reduced by roughly 50%, processor performance are comparable to that of SRAM and using nonvolatile p-STT-MRAM L2 cache also active power is drastically reduced (about 30% while CPU active state is dominant, 65% in moderate usage of CPU resources and 90% in almost totally-idling state). Future ideas include avoiding a net difference between volatile (SRAM) and non-volatile (MRAM) parts in a CPU cache devising a solution which is an hybrid of the two: non-volatility feature will run all the way down from L2, L3 caches down to CPU core, L1 cache and registers files. In this optic, Big.LITTLE architectures sound according to Toshiba a viable next implementation.

Resistive computation: a non-volatile CPU microarchitecture

Similar to the achievements by Toshiba, Guo et al introduced in a remarkable paper [16] a proposed set of guidelines to implement the concept of resistive computation. The main idea is based on solving the power wall problem while scaling down beyond 45nm migrating most of the architecture of a modern CPU from traditional CMOS to STT-MRAM. Every main CPU inner block is analyzed in details carefully re-engineering its peculiarity embedding a non-volatile approach, then an 8 core Sun Niagara-like CMT processor is modeled at 32nm technology node, showing huge reduction in terms of power dissipation and leakage power yet offering performance comparable to standard CMOS.

Hardware Security and PUF

Last but not least, preliminary work has been conducted in [17] and [18] to show how STT-MRAM can be employed to generate a unique Signature (or ID) for a digital MRAM memory chip, exploiting the process variation of the parallel and antiparallel configurations as Physical Unclonable Function. No more words will be spent in this paragraph since this is the main topic of the thesis work and will be widely treated in the next chapters.

2.3 STT-MRAM electrical models and characterizations

A wide variety of models and equations has been proposed both on literature and papers to represent the behavior of an STT-MRAM memory cell under different conditions (associated equivalent resistance, writing current pulse, switching behavior,. . .). At the moment there are no accepted standards on which model is better to be used for a specific purpose. In order to use STT-MRAM as PUF, a goal of this thesis work, the memory needs to be modeled in the simulations that have been conducted. After researching among different proposed equations, only one model seems representative enough for our purposes, since it takes into account a precious link: the effect of microscopic quantities (such as, but not limited to, the oxide thickness of the MTJ) over the digital STT-MRAM itself.

In order to characterize a PUF based on STT-MRAM, we obviously first need to characterize the STT-MRAM memory chip itself. What we need is a model that expresses the resistance of the parallel and antiparallel states ($\mathrm{R_p}$ and $\mathrm{R_{ap}}$) of each memory cell based on the MTJ stack properties. First of all, then, focus has been put into researching among the equations regarding a DC static modeling of the equivalent resistance of the MTJs rather then dynamic models which focus more on the switching behavior of the device (writing).

The present thesis work is an extension of the work by Zhang et al, illustrated in two subsequent papers [17] and [18]. Both papers used a particular way to model $\mathrm{R_p}$ and $\mathrm{R_{ap}}$. Particularly in [18], more focused on STT-MRAM rather than non-volatile memories in general, the model which has been used is somehow based on the fact that the MTJ resistance is related to the area of the MTJ stack and to the oxide thickness, and although no direct equation is showed, the paper actually only links these two physical parameters to an intermediate quantity (rather than the resistance of P/AP states): the **product RA**, an intrinsic quantity defined for each MTJ stack. *RA* is the resistance-area product. The mathematical formulation of this peculiar dependance is illustrated in Equation 2.2.

$$RA \propto \left(e^{a_0 \cdot t_{ox} + b_0} + \sum_{m=1}^{c} (-1)^{m-1} \cdot V_{MTJ}^{2m} \cdot e^{a_m \cdot t_{ox} + b_m} \right)^{-d} \tag{2.2}$$

Equation 2.2, proposed by [19] and used in [18], links oxide thickness and feature size of the MTJ (dimension of each layer composing its stack) to the physical value RA. Keeping constant the voltage applied to the MTJ structure, V_{MTJ}, different oxide thicknesses t_{ox} lead to a different RA value. For example, for t_{ox}*=0.85nm*, it is usually assumed *RA=10Ωμm*^{2}. Except for t_{ox} and V_{MTJ}, the remaining terms of the right end side of Equation 2.2 are simply fitting parameters, technology dependent.

This equation does not really provide a complete model to the effective resistances $\mathrm{R_p}$ and $\mathrm{R_{ap}}$. A well recognized model in this sense is the one provided in [20] and more extensively improved in [21], and is here mentioned and explained in the following equations.

Equation 2.3 expresses the **conductance physical model of the MTJ**, showing how the conductance at zero bias is a decreasing function of the oxide thickness t_{ox}, and depends on the potential barrier height φ(for MgO MTJs φ=0.4).

$$G(0) = 3.16 \cdot 10^{10} \cdot \varphi^{1/2} \cdot \frac{e^{-1.025 \cdot t_{ox} \cdot \varphi^{1/2}}}{t_{ox}} \tag{2.3}$$

However, in the electrical macro-models of MTJs, the resistance performance of the device can be expresses through the **simplified resistance equation** shown in Equation 2.4. It seems important to highlight that we refer here to $\mathrm{R_p}$.

$$R(0) = \frac{t_{ox}}{F \cdot \varphi^{1/2} \cdot Area} \cdot e^{1.025 \cdot t_{ox} \cdot \varphi^{1/2}} \tag{2.4}$$

The **parameter F** is a factor computed starting from the resistance-area product according to Equation 2.5. For example, *RA=10Ωμm²* gives then *F=332.253*.

$$F = \frac{3322.53}{RA} \tag{2.5}$$

The variable **Area** is instead the cross-section area of the MTJ, and it is computed in different ways according to the physical shape of the device. Equation 2.6 lists down the typical possibilities.

$$Area = \begin{cases} \pi \cdot A^2, & \text{if circular shape} \\ \frac{\pi}{4} \cdot A \cdot B, & \text{if elliptical shape} \\ A \cdot B, & \text{if rectangular shape} \end{cases} \tag{2.6}$$

The resistance of the MTJ is not really constant. $\mathrm{R_p}$, modeled in Equation 2.4, gives the value of this resistance at zero-voltage bias. In order to determine how its value changes according to the bias applied to the MTJ, V_{MTJ}, Equation 2.7 is used. In particular, it is worth remarking that $\mathrm{R_p}$ exhibits a very weak dependance on the applied bias and it can practically be considered constant over the entire range of variability of V_{MTJ}.

$$R(V) = \frac{R(0)}{1 + \dfrac{t_{ox}^2 \cdot e^2 \cdot m}{4 \cdot \hbar^2 \cdot \varphi} \cdot V_{MTJ}^2} \tag{2.7}$$

The important contribution of V_{MTJ} appears instead in the **TMR** (used to evaluate $\mathrm{R_{ap}}$ starting from $\mathrm{R_p}$, as previously mentioned into Equation 2.1). For simplicity, now we refer to $\mathrm{TMR_0}$ assuming it is no longer a percentage value but rather a number (then, *$TMR_{\%}$=120%* becomes now *TMR_0=1.2*). The real TMR is actually very dependent on the applied voltage to the MTJ, as described in Equation 2.8 - here V_h is the voltage at which TMR becomes half of *TMR_0*.

$$TMR_{real} = \frac{TMR_0}{1 + \dfrac{V_{MTJ}^2}{V_h^2}} \tag{2.8}$$

Finally, given then R_p and TMR_{real}, it is possible to evaluate the antiparallel resistance using Equation 2.9, which is a simple rephrasing of Equation 2.1.

$$R_{AP} = R_P \cdot (1 + TMR_{real}) \tag{2.9}$$

Table I recalls the principal parameters defining any MTJ structure, particularly when it comes about STT-MRAM memory applications, including the variables and constants encountered so far.

TABLE I

MTJ AS STT-MRAM: PARAMETERS AND VARIABLES

Parameter	**Description**	**Typ.Value/Example**
Microscopic quantities		
t_{ox}	Oxide thickness	typ. around 1nm
φ	potential barrier across tunneling oxide	MgO: 0.4eV; AlO: 2eV
RA	Resistance-Area product, intrinsic parameter of a MTJ stack of given materials	$\propto 10 \div 10^2 \Omega \mu m^2$
Feature size	Manufacturing size of the MTJ and CMOS, not necessarily coincident	typ. 45÷65nm
A, B	Physical dimension of the cross-section of the MTJ, related to feature size	typ. 65nm
Area	MTJ cross-section area (from A, B, MTJ shape)	$\propto nm^2$
e	Electron elementary charge	$1.60 \cdot 10^{-19} C$
m	Electron mass	$9.1 \cdot 10^{-31} kg$
$\hbar$	Reduced Planck's constant	$1.0545 \cdot 10^{-34} J \cdot s$
Macroscopic quantities		
R (R_p, R_{ap})	Effective resistance across the MTJ stack. "R" typically refers to the parallel state	ex. $(2k\Omega, 4k\Omega)$
TMR_0	Tunneling magnetoresistance ratio at 0V bias	ex. 100%
TMR_{real}	Tunneling magnetoresistance ratio at given bias	ex. 95%
V_{MTJ} *or* V_{bias}	Voltage across the MTJ structure; in this work assumed to be used to read a cell	typ. 100mV
V_h	Voltage at which TMR_{real} becomes half TMR_0	typ. 0.5V

Other kind of models have been developed to shape different behaviors of MTJ STT-MRAM structures. However, the one illustrated in this section is the only one which appears the most complete as well the closest to size the behavior of a wide range of MTJ structures in terms of physical size and voltages, granting its use in different papers related to the STT-MRAM topic.

Other models use typically too simplified equation to describe the behavior of the MTJ resistance, for example based on a few fitting parameters related to a specific set of physical feature sizes [22], appearing as a mere modeling as function of the applied bias only. Another lack in the field is a model which simultaneously takes into account the equations shown so far and the dependance over temperature. Currently, temperature-dependent models regarding STT-MRAM are not available at the macroscopical level, and they typically relate the temperature dependance to microscopic quantities like the tunneling electron spin polarizations of the two ferromagnets [23].

2.4 Hardware Security and the role of PUF

Every idea should be protected against theft and plagiarism. Electronics, and in general every product of engineering in general, is the result of clever insights, improvements and breakthrough achievements on which a team responsible for its design invests manpower, thus brains and costs.

Although market competition is a good source of continuous positive challenge and improvement, design theft, on the other side, is something that needs to be fought, and a source of substantial income losses. Hardware Security is a set of design practices that contribute in the intellectual property protection, fighting electronic chips piracy and design thefts.

Any electronic component can only hit the target market after several steps. Engineering design and manufacturing are two among them. In other words, **hardware design** and **manufacturing** are two key (and obviously unavoidable) steps. They require many steps to be accomplished, as well as the involvement of several Companies typically spread worldwide.

The globalization of the market has involved both aspects, and while chip design is usually kept in-house, in a given Company A, responsible for the design and the selling of a given chip, this Company will typically outsource the manufacturing phase to (at least) another Company B dislocated worldwide (China and Brazil are a few examples). We call Company A fab-less, while Company B is the responsible of the physical implementation of the device.

This scenario involves **Company A to share chips designs and specifications with Company B**, and during this phase, one or more third parties may access, steal and re-use them to produce illegally other lots of that particular technology. The worst case scenario includes the manufacturer itself, Company B, since it is the one in possess of both the intellectual property (the design) and the physical resources (its silicon foundry) to produce, out of control of company A, some chips and selling them to a black market - or even to the consumer market.

A basic illustration in Figure 5 shows the design path from Company A to B and the desire of protecting somehow the design during this phase, until the chips hits the market.

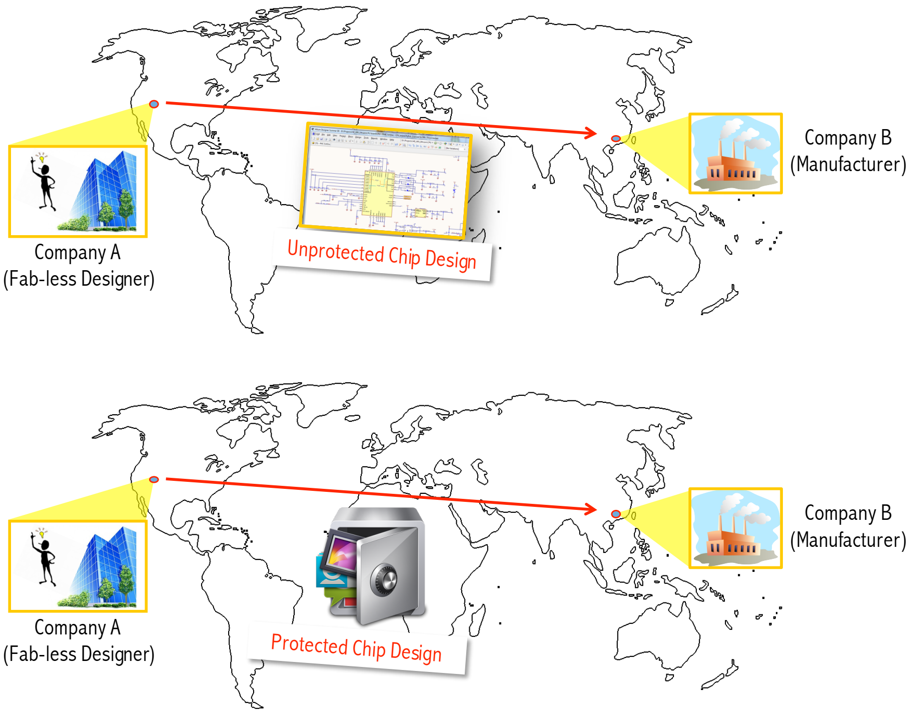


Figure 5. During chip manufacturing, unprotected (top) chip design may lead to design thefts. Design protection is then becoming a must (bottom).

Hardware designers have then started, especially during the last years, to implement Hardware Security features in a given chip that needs to hit the production. By **Hardware Security** we mean a set of rules, protocols and implementation techniques, usually embedded at design stage of the original chip, to fight and hopefully avoid design thefts, adding some sort of layer of security.

This layer can be implemented in different ways. For example, hardware is re-engineered in such a way to hide parts of the schematics, or similar techniques that require huge computational power to reverse-engineer the protection scheme and access back the original design subject of the possible theft. More recently, Hardware Security is relying more and more on another idea: identifying chips through a Physical Unclonable Function or PUF.

2.4.1 The need of identifying chips

A viable solution to fight design (and thus physical chips) thefts is the possibility of forging a **unique identification number (ID) on each chip** of a given produced lot. When the chip, after production, hits the market, eventually implanted on a wider system, each device that has been sold needs to be activated to work. The activation process requires the device to send its ID to the Company which, after checking in its database of recognized chips, will grant the OK and unlock its functionalities.

The classical approach would be embed a small non-volatile memory in each produced chip, a ROM, to permanently hold the ID. The main problem to this classical approach is that the chip producer, in charge of physically realizing the chip, together with its ID, would be capable of knowing in advance each ID itself. At the same time, if the ID is written from the chip design Company, Company A, after the lot is produced and handed back with an empty programmable read-only memory (PROM), once the chip hits the market anybody can access anyway the ID and realize similar chips with the same ID. Or, even worse, be able to detect the general algorithm behind the ID generation and replicate it to produce IDs that would still give a PASS when queried for the activation. Plus, embedding a different kind of memory (like a ROM) for the only purpose of identification implies the use of additional masks on silicon and an increased cost of the produced chips.

This idea needs then to rely on a completely different approach: a key needs to be present and associated to each chip, but simultaneously it should not be embedded permanently at all inside a given produced chip. This is the main idea behind a Physical Unclonable Function, subject of next subchapters.

2.4.2 PUF: Physically Unclonable Functions

PUF stands for **Physical(ly) Unclonable Function**, and identifies a well known physical characteristic which is associated univocally to a specific structure like an integrated circuit at the physical level. The word *physical* is very important, since the idea behind PUF is to ***rely on specific parameters at the materials level constituting a given device to generate some useful information for identification purposes***.

In particular, what is exploited are some features that are, on the contrary, rather fought at the digital level: we refer to **manufacturing process inaccuracies**, that in the worst possible scenario would lead to a low yield of a chip lot.

Independently on how accurate the manufacturing process is, in fact, there will always be an uncertainty accompanying the physical quantities related to it: feature size, doping concentration are just a very few examples. Furthermore, with microelectronic device scaling, the process variation, cause of a discrepancy from an ideal behavior, starts to play more a dominant role.

A PUF is then an one-way function that, evaluated for each chip of a production lot, takes into account process variation to **generate a unique signature for that chip and that chip only**, solely based on the process variation itself. In particular:

- Process variation is then the random (thus unpredictable) input x to the PUF;
- The PUF is a function that is applied to the input: *f(x)*;
- The generated output, *y=f(x)* is the unique signature of that given chip and hopefully unique.

This response is then used for security- or identification-related goals. No more than one chip can come with the same signature. The advantage of PUF is that it is a **one-way function**: this means that, starting from the input domain, it is possible to get a univocal result y=f(x), however, the image (result) to which y belongs, is very difficult to invert. Given y, it is very hard to find x. Going back into the electronic domain, the random input x is obviously a physical electronic quantity which is subject to some manufacturing process variation. This variable, ideally a constant, because of process variations during manufacturing can actually assume a mean value and a well known amount of tolerance. A spread around the ideal value can be observed and it is usually characterized, apart a few exceptions, by a **Gaussian distribution**: the ideal value is the *mean value* and around it a *standard deviation* is the expression of the process variation itself.

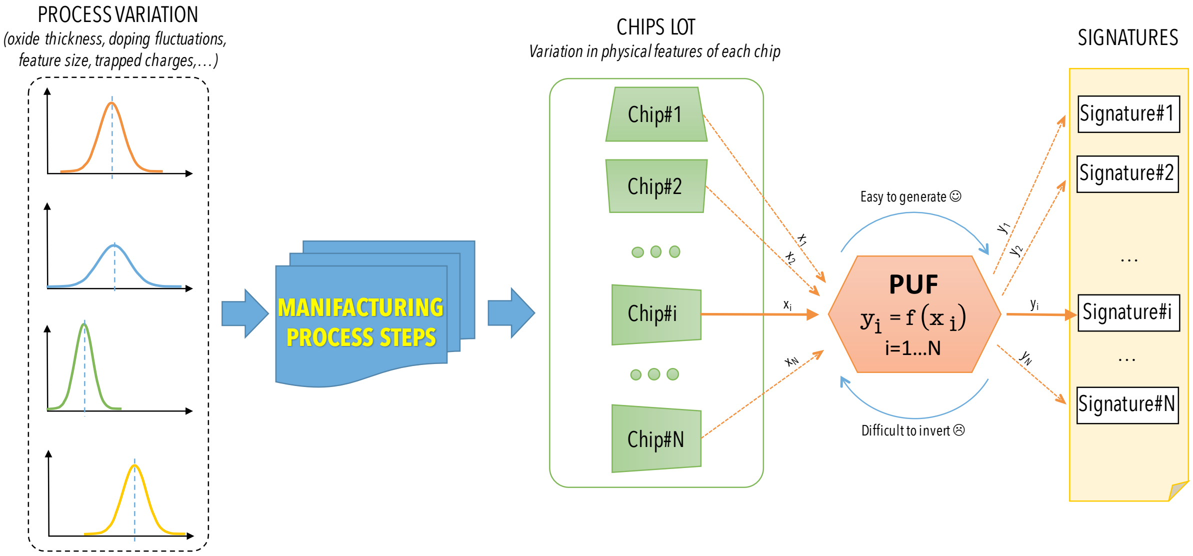


Figure 6. From chips manufacturing to unique signature: PUF generation

Figure 6 show qualitatively the general steps of manufacturing process and when PUF generation takes place, leading to secure signatures generation for a reliable chips authentication: a necessary measure to detect original chips (which come with a signature) from fraud ones.

The suitability of a particular device, either analog or digital, as PUF, relies then mostly on its physical characteristics and the amount of variability of them. There are different reasons to select a particular technology as PUF for Hardware Security purposes rather than another. Obviously, we need to define what suitability as PUF means. Although this aspect is of crucial importance since it determines how reliable a specific design is when used as PUF, it also needs to keep pace with another criteria, which is of higher-level: compatibility with the chip technology that the designer is using to implement a certain more or less complex VLSI logic function. For example, it sounds obvious that an analog PUF can hardly be hosted within a completely digital integrated circuit scenario, as well as the opposite. Also, volatile memories are getting popular as PUF applications, but one would hopefully choose the same kind of memory as PUF as the one that is currently employing into his chip.

In the next two sub-chapters, a few classical PUF examples are briefly introduced.

2.4.3 An analog example of PUF

A remarkable paper which built the fundamentals on chip authentication and Physical One-Way Functions is the work of Pappu et al. [24]. Given a population of 3D tokens, they are scanned by an incident laser light; exploiting the **interference phenomena**, a 2D image is obtained which is in turn transformed into a 1D binary key with a particular technique called Gabor hash. Varying the angle of the incident laser beam on the 3D token, different binary keys are obtained for each particular chip. This procedure is called **enrollment**.

Each chip is then identified by a given number of couples (angle; 1D key) and the entire databases is stored in a secure server. During the **authentication** phase, the chip (3D token) verification takes place at a possibly insecure location. The server sends out a challenge (an angle) used to scan the 3D token at the unsecure terminal. The response is then sent back to the server. If the response is coincident with the one stored on the server side, for that particular angle of that token, then that 3D token being verified is authentic.

This example represents an introduction on how **physical process is used to generate a secure key with PUF**: laser light scattering is a phenomena which depends on the physical properties on the medium, on the location of the atoms and subatomic particles interactions.

This aspect, varying with 3D tokens manufacturing, is totally uncontrollable and completely random.

2.4.4 Typical digital PUFs: Ring Oscillator and SRAM

Moving to the digital domain, PUF needs to be implemented using the technology already present and embedded in the chip. A first example worth citing is the use of **Ring Oscillators (RO) as digital PUF** [25]. These devices, built as a loop chain of inverters, are used to generate a digital clock at a given frequency. However, due to process variation, the exact frequency of operation among different RO is never exactly the same. Using a wide population of more than 100 FPGAs, each of which uses 512 RO already embedded in each FPGA. These ROs are compared in group of two and, for each of these couples, if the first RO is faster than the second one an output bit '1' is generated, otherwise the output of the comparison returns a '0'. Repeating this procedure for all the 512 ROs a 511-bit wide binary signature is obtained, identifying each FPGA.

A second example of digital PUF, the work of Holcomb et al in [26], relies on **exploiting the power-up phase of SRAM memory cells**. Each SRAM cell contains a value and its complement either in the form of (0;1) or (1;0). These are the only two possible stable states. However, when the memory cells are not powered, the output of the SRAM cells is obviously (0;0) which represents anyway an unstable state. Rising the supply voltage of the SRAM from 0 to the nominal V_{DD} leads to the transition to one of the two stable states. It is not known a priori which state will be reached since it depends on physical features of each SRAM memory cell such as transistors mismatch and noise. Associating a '0' or a '1' according to the stable state reached by each memory cell allows to map the memory matrix into a binary matrix constituting the signature of the chip.

2.4.5 Introduction on evaluating the quality of a PUF

This second example [26] shows very well **what "challenge" and "response" mean in a PUF**. For the SRAM case, the challenge is the memory address, which allows to consider and select each memory cell requesting a response. The "response" is the digital bit which is function of the transitioning between the power-off metastable state and one of the two possible stable states. In the analog PUF example in [24], the "challenge" is the incident laser beam while the "response" is the 2D diffraction pattern of the light (further processed to get a digital signature).

Independently from the particular technology and its analog or digital domain, PUF is a technique that allows to associate a digital signature to each physical device. **The signature is never stored in the device itself**, can be generated multiple times when required and it is stored securely in the server of the Company owning the intellectual property over that particular design.

The aspect "*generated multiple times*" is of crucial importance in a PUF: the signature generated from the same chip must be the same when generated again. This aspect is called **reliability** or stability. On the other side, the signature of a given chip must differ from the signature of another chip: we refer here to **uniqueness**. Both reliability and uniqueness are the two quantitative parameters that allow to evaluate the **quality of a PUF**. In the real scenarios, no PUF is perfectly reliable and unique. However, a given amount of tolerance is accepted and enables to discriminate between "*good PUFs*" and "*bad PUFs*".

Both papers [25] and [26] contain useful information on how to evaluate the quality of the PUF. For these reason, their insight will be used in the next chapter to assess the stability and performance of the **PUF subject of this thesis work: STT-MRAM**.

2.5 STT-MRAM as PUF: introduction and current work

The introduction on PUF done so far, together with a couple of digital-electronics examples, has hopefully provided an initial flavor on the main idea behind PUF. Once again, it is worth remarking that every time that, in electronics, we observe a not so negligible parameter variation in a given microelectronic device, this feature can be used to try to implement a PUF. We refer then to quantities that are typically assumed steady or constant in the digital domain, after exiting a metastable state, an initial transient, etc. PUF exploits instead the transitioning to stability.

2.5.1 MRAM resistance dispersion as PUF

Using a digital memory as PUF is an idea worth trying. It allows to exploit an hardware resource already present in a digital system for a new purpose: Hardware Security. STT-MRAM is a good candidate to become the universal memory in the near future, at least until the main memory level.

There are different ideas behind STT-MRAM as PUF – at a digital level, they are based on exploiting the parallel and antiparallel states' associated resistances, R_p and R_{ap}. It has been observed, in fact, that given a population of MTJ stacks, when put in parallel and antiparallel state, they assume a possible range of values: in other words, manufacturing process variation over physical parameters of the MTJ stack leads to a dispersion of both R_p and R_{ap}.

What happens in reality is that both the parallel and antiparallel state of a population of MTJ are distributed according to a Gaussian probability distribution function (or pdf). Figure 7 illustrates this behavior from a qualitative standpoint, highlighting the main peculiarities of both R_p and R_{ap}.

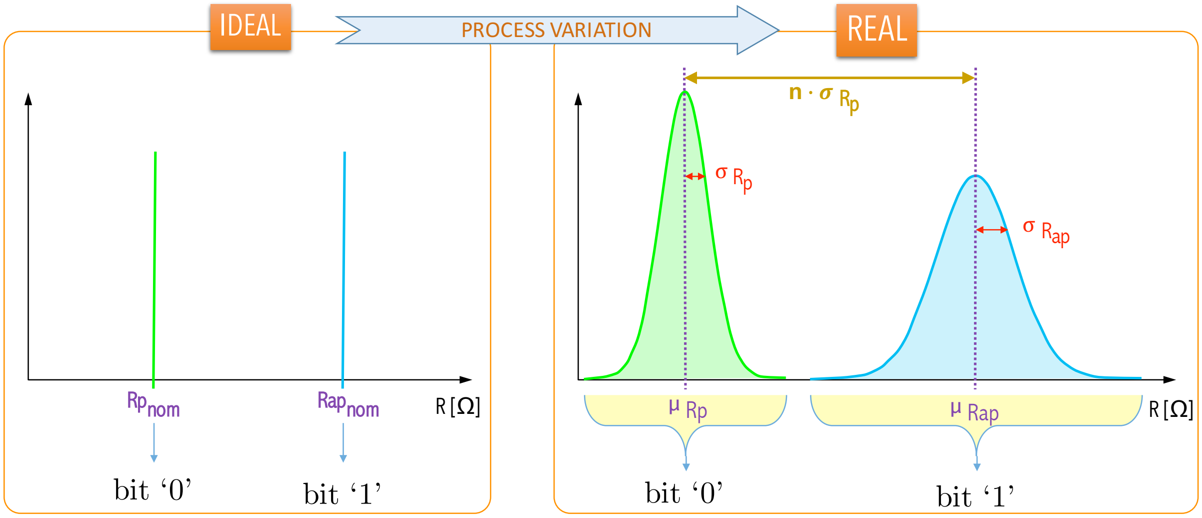


Figure 7. P and AP states have a low- and high-state resistance. In reality process variation leads to a spread of them both.

The general expression for a Gaussian pdf is recalled in Equation 2.10, where μ is the mean value and σ is the standard deviation.

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.10}$$

Combining the peculiarity of the population of MTJ devices highlighted in Figure 7 and more rigorously defined in Table II, it is possible to describe the population of MTJ (STT-MRAM) memory cells, both in parallel and in antiparallel configuration, through Equation 2.11 and Equation 2.12.

$$f(R, \mu_{Rp}, \sigma_{Rp}) = \frac{1}{\sqrt{2\pi}\sigma_{Rp}} \cdot e^{-\frac{(R-\mu_{Rp})^2}{2\sigma_{Rp}^2}} \tag{2.11}$$

$$f(R, \mu_{Rap}, \sigma_{Rap}) = \frac{1}{\sqrt{2\pi}\sigma_{Rap}} \cdot e^{-\frac{(R-\mu_{Rap})^2}{2\sigma_{Rap}^2}} \tag{2.12}$$

TABLE II

STT-MRAM: MTJ STACK MAIN PARAMETERS OF RESISTANCE STATISTICAL DISTRIBUTION IN PARALLEL AND ANTIPARALLEL STATES.

Parameter	**Description** *(P=parellel; AP=antiparallel)*
μ_{Rp}	Mean value of P state resistance distribution *Rp*
σ_{Rp}	Standard deviation of P state resistance distribution *Rp*
$\frac{\sigma_{Rp}}{\mu_{Rp}} \cdot 100$	Measure of variability of P state resistance wrt mean value
$max\{Rp\}$	Max value of P resistance among samples in the population
σ_{Rap}	Mean value of AP state resistance distribution *Rap*
σ_{Rp}	Standard deviation of AP state resistance distribution *Rap*
$\frac{\sigma_{Rap}}{\mu_{Rap}} \cdot 100$	Measure of variability of AP state resistance wrt mean value
$min\{Rap\}$	Min value of AP resistance among samples in the population
$n \cdot \sigma_{Rp}$	Distance between μ_{Rp} and μ_{Rap} expressed as multiple of σ_{Rp}

STT-MRAM used as PUF is based on exploiting this process variation, **engineering the resistance dispersion in such a way that this random physical phenomena can become source of a robust signature generation** for chip identification purposes.

2.5.2 Current work proposed schemes: comparison and tradeoffs

The aim of the present thesis work is to **investigate on the use of STT-MRAM as Memory PUF** (also called *MemPUF* for short). The starting point is the work of Zhang et al. in [17] and [18], which has posed the basis on the analysis of STT-MRAM as viable generator of Physical Unclonable Functions.

- In the first of the two papers, "*Feasibility Study of Emerging Non-Volatile Memory Based Physical Unclonable Functions*" [17], different non-volatile memory technologies are analyzed (STT-MRAM, Resistive RAM and Phase Change RAM). Here **STT-MRAM, especially in antiparallel configuration, appear to be among the best non-volatile memory candidates as PUF due to an high quality of the generated signature**. Generally, all three types of non-volatile memories are validated as viable PUF solution.
- In the second of the two papers, "*Highly Reliable Memory-based Physical Unclonable Function Using Spin-Transfer Torque MRAM*" [18], STT-MRAM only is chosen and further developed. The idea behind the paper is **not only to generate a PUF** as in [17], **but also to augment the digital architecture in order to write back (to save) in the STT-MRAM memory chip itself the generated signature**: after all, the memory is used to save data. This might sound not safe from the Hardware Security

standpoint, but a PUF serves different goals - for example, providing a simple signature which is in turn further processed to become a *true random number generator*.

The main principle behind PUF generation through STT-MRAM is described as follows, and it represents the starting point for most of this entire thesis work.

First of all, the present work relies on using an STT-MRAM memory chip seen as an usual 2D memory matrix, of M words of L bits each, thus of size $M \times L$. Each row, that is each word, is processed and from it a response string, in a binary form, is generated, applying a physical one-way function which exploits the resistance variation of each MTJ constituting the single bit, either in parallel or in antiparallel state. After processing, the memory can be associated to a 2D matrix of equal size containing M strings representing the responses. Figure 8 summarizes this architecture from a conceptual standpoint.

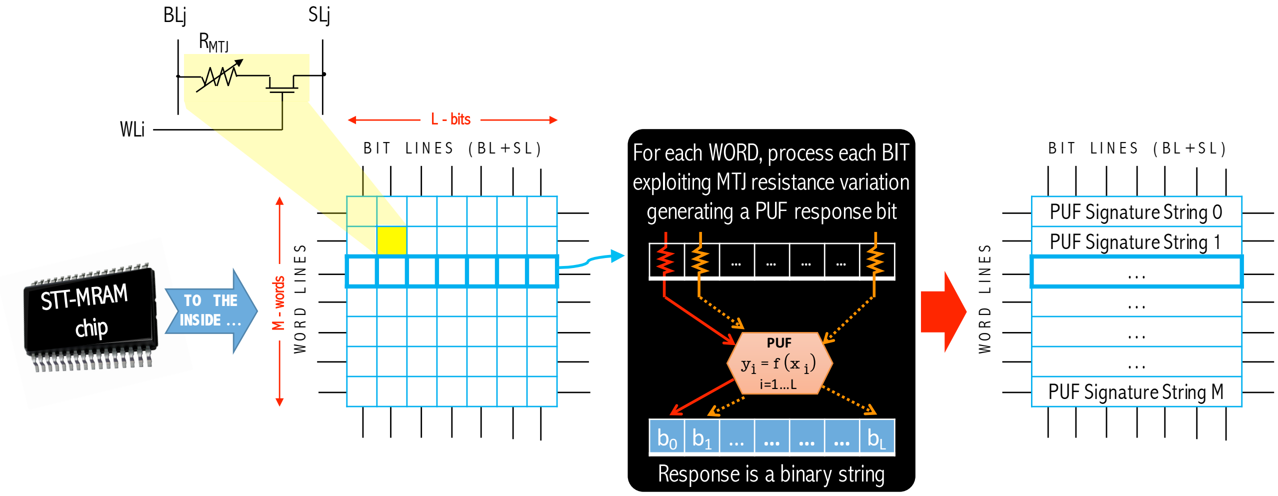


Figure 8. Generating a PUF signature from a STT-MRAM memory chip

In particular, two possible solutions are used to implement a physical unclonable function over this structure. In order to introduce a nomenclature which will become useful and further extended in the next chapter, illustrating this thesis work, we call these two solutions **CASE 1** and **CASE 2**.

First of all, both solutions are qualitatively explained and introduced in Figure 9.

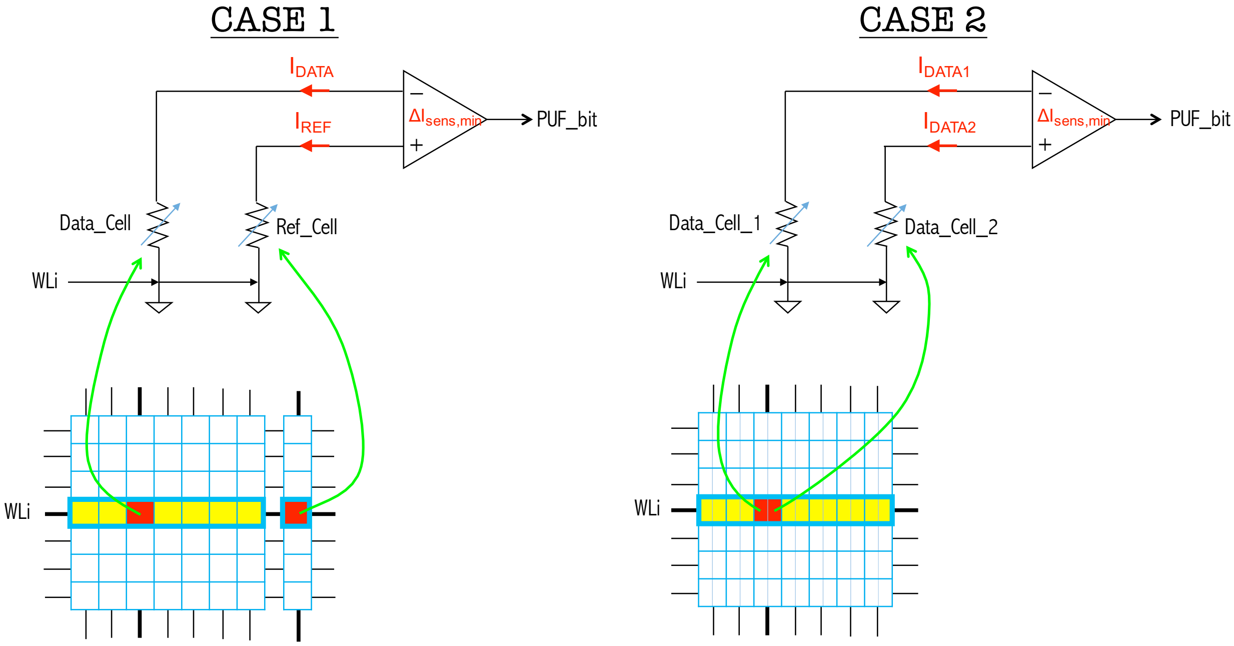


Figure 9. A representation of the methodology used in the two papers constituting the current work in order to generate a signature using the PUF method on a chip of STT-MRAM.

In particular:

- **CASE 1**: the memory uses an additional column, for each word line, of **reference cells**: this is a possible approach used in MRAM memories when it comes to read data for normal memory read operations. In the PUF context, both memory data cells and reference cell, for each row, are of the type **1T1MTJ memory cell** (one transistor plus one MTJ) and they are always programmed in the same state (either parallel or antiparallel) and then the world line is activated: by letting a current flow in both cells, I_{DATA} and I_{REF}, a sense amplifier is able to sense the current difference and produce a digital output, 0 or 1, according to the currents mismatch. Ideally, the two currents are the same (since the nominal value of R_P or R_{AP} is constant) but, due to process variation, as already described before in 2.5.1, the resistance dispersion of the parallel (or antiparallel) state leads to two different ohmic values and then, for the same read voltage applied to the MTJ cells, to two different currents. The sense amplifier is then able to sense the difference and produce the output bit – each of the bits constituting the PUF signature for that world line. *The produced bit is then function of the intrinsic randomness that lies in the resistance dispersion of the MTJ state due to the unpredictable process variation: the PUF idea itself*;

- **<u>CASE 2</u>**: no reference cell is present, and for each word line each memory cell is constituted by two MTJs rather than one. We refer in this case to **self-referencing**, or **2T-2MTJ memory cell** (since constituted by two access transistors and two MTJ stacks). This second possibility, an alternative to reference-base reading for normal memory read operations, can be again exploited for PUF purposes. The idea is very similar to CASE 1, and the difference resides in the fact that the currents to be evaluated by the sense amplifier now both come from the same memory cell, I_{DATA1} and I_{DATA2}. Again, as in

CASE 1, it is possible to use the memory matrix programmed entirely either in parallel or antiparallel configuration.

Table III summarizes the peculiarities and working principles of these two approaches.

TABLE III
PUF GENERATION: COMPARISON OF CASE 1 AND CASE 2.

	Memory Cells Type	**Sensing Scheme**	**Cells Programming**
CASE 1	1T1MTJ: an access transistor driving a single MTJ stack	Reference-cell method: 1T1MTJ data cell resistance compared to 1T1MTJ reference cell resistance	Two options: all as Rp; all as Rap
CASE 2	2T2MTJ: two access transistors each of which driving one of the two MTJs	Self-referencing method: a single 2T2MTJ cell, resistance of the two MTJs compared against each other. No reference cell is present.	Two options: all as Rp; all as Rap

Iterating the procedure, for both cases, for all the memory words across the entire memory chip, and putting together the PUF response bits, it is possible to **end up with a binary string that can be seen as signature associated to every word line**. This result has already been remarked and highlighted previously in Figure 8.

The results shown by the papers and are very promising. In particular, using STT-MRAM as PUF leads to the following achievements:

- Given a certain amount of noise effect in the system, the bit-error rate in the generated signature, an indicator of reliability of the PUF signature, is the lowest in STT-MRAM when compared to other non-volatile memory technologies [17];

- The uniqueness of the signatures (how much they differ from each other) is better in differential mode (CASE 2) for all the memory technologies - in particular, for STT-MRAM, the presence of 2T2MTJ cells means doubling the number of MTJs thus increasing the number of available binary resistors (for the chosen state, P or AP) enabling to a wider difference among the generated signatures [17];

- Using non-volatile memories as PUF, with respect to standard CMOS based memories, allows to get far higher density of integration [17];

- Thanks to the unlimited endurance of MTJ-based memories the reliability of the generated PUF is rather high [18];

- The intrinsic working principle of STT-MRAM, based on spin-torque transfer, allows very high resistance to external disturbances as well as disturbances caused during the writing phase [18];

- To increase the reliability of the STT-MRAM PUF system, it is possible to write back in the memory itself the generated PUF. In other words, after the signature string, for each word line, has been generated, an automatic write-back mechanism stores the binary string itself in the memory cells [18].

2.5.2.1 STT-MRAM as PUF from the nano electronic standpoint

Apart from the digital level standpoint, additional work has been done on STT-MRAM as PUF from the nanoelectronic point of view. The related papers, although more than worth being mentioned, are not strictly related to the digital level system design and belong to the nano electronic field thus are not strictly related to the focus area of this thesis (more digital-oriented). In particular, in [27], PUF is evaluated exploiting the fact that variations in the MTJ device geometry lead to an unpredictable slight energy barrier variation related to the magnetization vector of the ferromagnets. In particular, the magnetization vector is excited and then moved from the easy axis (its preferred alignment in the free-layer ferromagnet) to a perpendicular direction (the hard axis) and then released back to its resting condition (the easy axis), which can happen in two different and unpredictable ways. This work has been further extended in [28], brought closer to the digital level with a possible authentication protocol and a system level architecture.

Another approach used in [29] is instead based on the study around the switching threshold of an MTJ between parallel and antiparallel states, working on bias voltages which ensure an almost 50% probability of switching.

Finally, Spintronic is used in [30] applied to nanowires used as Domain Wall Memory exploiting surface roughnesses of the nanowire itself - not really an MTJ but a very similar device.

CHAPTER 3

MY WORK

The aim of this Chapter 3 is to report in detail the work carried out with the present thesis. After highlighting, in the previous Chapter 2, the theoretical background behind both STT-MRAM and Hardware Security principles, including the main work realized so far on the use of magnetic memories as PUF, as reported in the previous Section 2.5.2, the latter represents a bridge of connection between the work done so far and the new contribution provided on the field by the present thesis work.

In the following paragraphs, several of the theoretical concepts and models highlighted in Chapter 2 will be used and cited since they represent the starting point for this main Chapter.

Chapter 3 is then about explaining in details the goals, the accomplishments and the overall contribution of the present thesis work, already introduced in Chapter 1.

3.1 Main motivations

New non-volatile memories, especially in the STT-MRAM flavor, are the most promising candidate in the near and far future to replace current conventional SRAM and DRAM technologies in digital computer systems. In the last couple of decades, evolutions in both spintronics and microelectromagnetic structures has led to the development of magnetic memories which aim to become a new standard for digital memories.

A second trend is the use of Hardware Security techniques at a design level which is becoming more popular during chips design and prototyping: design theft is a phenomena that

Companies owning an intellectual property are constantly trying to fight and in the last few years conferences and standards on the matter of Hardware Security are starting to bring everybody on the same page. Physical Unclonable Function, PUF, has been shown to be a promising solution against chips piracy.

Bonding the trend of the emerging use of magnetic memories in STT-MRAM flavor and Hardware Security, the present thesis works aims at investigating in details techniques and methodologies, as extension of the current work presented in Section 2.5.2, to effectively protect manufactured chips against unwanted uses using PUF principles and exploiting process variation in the associated resistance of STT-MRAM cells.

All simulations conducted in this Chapter have been realized implementing models and testing methodologies in **Matlab**.

3.2 Initial analysis on the static electrical model of STT-MRAM

In order to realize a simulation framework to test and validate PUF designs applied to STT-MRAM, the magnetic memory itself needs to be characterized. Typically, studies on PUF conducted so far do not focus a lot on a rigorous statistical analysis of the random physical feature exploited for PUF purposes but they rather start by applying the PUF methodology immediately.

Before proposing then new schemes on the use of STT-MRAM as PUF device, then, a **population of STT-MRAM memory cells is simulated and studied from the statistical standpoint**. In particular, we **focus on the dispersion of the resistance of the associated parallel and antiparallel states of the MTJ stacks**, studying how the variation of physical parameters and electrical quantities affects them, understanding how much it is possible to push the resistance dispersion of MTJs playing with process variation in order to get a good base for a PUF implementation and, simultaneously, still obtaining a memory cell which can be reliably used for its traditional digital applications: storing data.

In particular, Equation 2.4 has been used as starting point of a model that has been built to analyze the performance of a MTJ population representative enough for statistical purposes.

3.2.1 Statistical study of a population of STT-MRAM MTJ memory cells

A population of $N = 5000$ STT-MRAM memory cells has been generated through Monte Carlo simulations and studied. In particular, the simulation framework has been built up and based on the following parameters, assumed to be provided and well defined in a given manufacturing process (and using Table I as reference to characterize the technology node used):

- Oxide thickness of the MTJs: nominal value $\mu_{tox} = 0.85nm$ and standard deviation $\sigma_{tox} = 1\%$;
- A resistance-area product chosen to be $RA = 10\Omega\mu m^2$ and assumed to be constant, a reasonable value for oxide thickness around $1nm$;
- An elliptic shape for the MTJs, with a feature size of 65nm ($\mu_A = \mu_B = 65nm$) and a standard deviation $\sigma_A = \sigma_B = 3\%$;
- A tunneling magnetoresistance ratio, at zero bias, equal to $TMR_0 = 120\%$;
- An applied bias to read each cell equal to 100mV: nominal value $\mu_{V_{bias}} = 100mV$ and standard deviation $\sigma_{V_{bias}} = 1\%$

The remaining parameters coincide with the typical values already quoted in Table I. By letting the variable parameters (oxide thickness, feature size and applied reading voltage) vary as a Gaussian distribution with a Monte Carlo approach, thus described through Equation 2.10, and applying overall the model proposed in Equation 2.4, taking into account the voltage variation effects on the parallel and antiparallel resistance of the populations of MTJs (described by Equation 2.7, Equation 2.8 and Equation 2.9), a population of $N = 5000$ STT-MRAM memory cells has been generated and studied.

For each memory cell, the simulator implemented in Matlab provides three values: the real parallel and antiparallel resistances and the real applied voltage. Overall, then, the output population can be seen as a stream of three elements, called: Rp_pop, Rap_pop and $Vbias\_pop$.

3.2.2 MTJ population performance evaluation

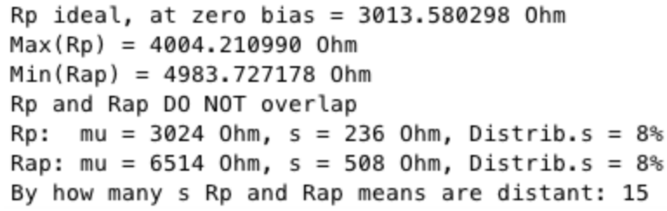
The population generated, constituted by 5000 units, is then evaluated to establish its performance as PUF. Simply speaking, we require here a good dispersion of the resistance for both states, P and AP, in order to rely on a robust foundation to build up a Physically Unclonable Function out of it.

In the past work, the distribution of R_p and R_{ap} of magnetic memory cells has typically been determined by physically producing a sample chip of STT-MRAM and measuring each cell. On the other side, to have an estimation of the nominal value of R_p and R_{ap} for fixed physical parameters, some equations have been proposed, which in turn might be either the result of fitting curves to experimental data (it is the case of Equation 2.2) or the outcome of a purely theoretical study (Equation 2.3 then simplified with Equation 2.4).

A first contribution of this thesis work is to apply Equation 2.4 (that indicates an ideal exact value of R_p and R_{ap}) to a software-simulated population of magnetic memory cells, verifying that, if the physical parameters of the MTJs are affected by a simulated amount of process variation, then the generated population is normally distributed as reported in the literature [31] and in different papers and presentations as [32]. The chosen starting point is Equation 2.4 because it is, at the moment, the only one that includes an high number of physical parameters and applied voltage as variables to determine the resistance of an MTJ stack.

The following six Figures represent the output of the simulator and show the result of the framework built do generate the population of MTJs in both parallel and antiparallel states.

First of all, Figure 10 is a brief initial output of the simulator providing some relevant basic parameters over the generated population: for the chosen parameters, we roughly get a $\mu_{Rp} = 3k\Omega$ and a $\mu_{Rap} = 6.5k\Omega$. Comparing Figure 10 with the specifications provided by Table II, we get an 8% coverage of the standard deviation with respect to the mean value both for R_p and R_{ap}. The two states' mean values are distant from each other by 15 σ_{Rp}.



```
Rp ideal, at zero bias = 3013.580298 Ohm
Max(Rp) = 4004.210990 Ohm
Min(Rap) = 4983.727178 Ohm
Rp and Rap DO NOT overlap
Rp:  mu = 3024 Ohm, s = 236 Ohm, Distrib.s = 8%
Rap: mu = 6514 Ohm, s = 508 Ohm, Distrib.s = 8%
By how many s Rp and Rap means are distant: 15
```

Figure 10. STT-MRAM generated population: first statistical parameters over the MTJ stacks' resistance dispersion

Figure 11 shows the dependance of the resistance of the parallel and antiparallel states as function of the applied bias V_{MTJ} to the memory cell for read operation. While keeping any other parameter constant (feature size and oxide thickness) the applied reading voltage is made varying by the simulator in a reasonable range and the variation over the nominal (zero-bias) value of R_p and R_{ap} is detected. While R_p has a weak dependance on the applied reading voltage, R_{ap}, on the contrary, exhibits a strong variation which is roughly parabolic. The result reflects Equation 2.8.

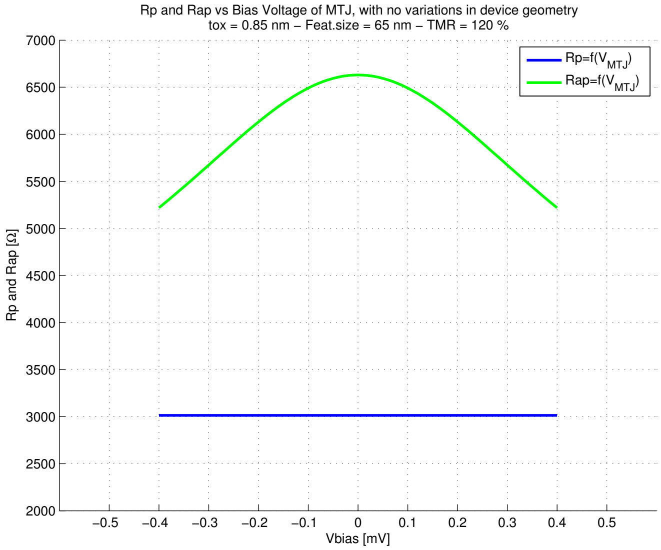


Figure 11. Dependence of R_p and R_{ap} on V_{MTJ} applied to a memory cell

Finally, taking into account in the simulator process variation of MTJ area and oxide thickness, and letting the applied reading voltage to each cell vary as well as a Gaussian distribution, the sample population of $N = 5000$ elements is plotted on an histogram chart in Figure 12. Overimposed to it, the associated Gaussian fit, confirming that if both the process variation and the applied voltage are normally distributed, this will lead to a normally distributed population of parallel and antiparallel states associated resistances of the MTJs constituting the population itself. The parameters of these two populations have already been discussed in Figure 10.

It seems worth of consideration to highlight the fact that the antiparallel distribution shows a much wider standard deviation: the reason principally lies on the fact that it is a lot more dependent on perturbations over the applied voltage with respect to the parallel state associated resistance. Given that for a PUF we require a lot of variability, at least for the uniqueness of the

PUF signature we could already expect the antiparallel state to perform better (a fact already confirmed and validated in the first approach used by [17] and [18] employing STT-MRAM as PUF).

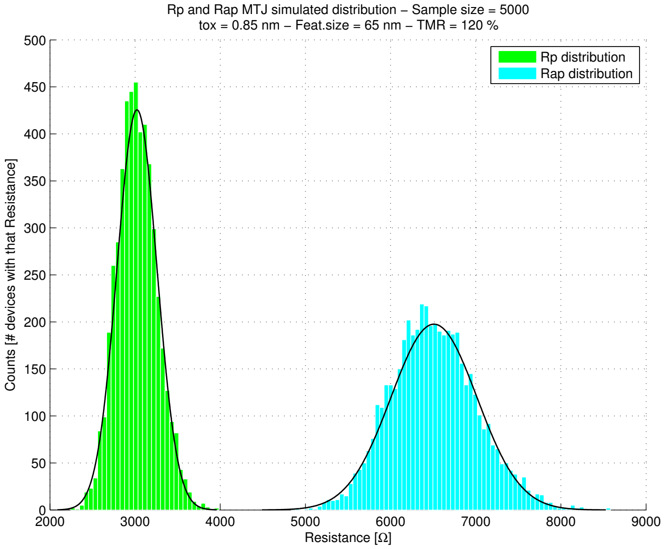


Figure 12. Distribution of the simulated sample set of MTJ elements: R_p and R_{ap} distribute normally given normally distributed physical parameters and applied voltage

The next three Figures are a further study of the populations R_p and R_{ap} from a particular standpoint, investigating the influence of particular aspects to obtain further information on the resistances, necessary to exploit the STT-MRAM as PUF.

Figure 13 shows the possible pairs of $(R_{ap} - R_p)$ taken as resistance differences: the mean value of this new population is equal to the distance between the mean values of the single populations R_p and R_{ap}. So far, the work on STT-MRAM as PUF considers cells either programmed simultaneously as R_p or R_{ap}. We want to try new layouts and flavors verifying whether, combining both states at the same time for different cells, it is possible to realize a signature whose quality as PUF is still very high.

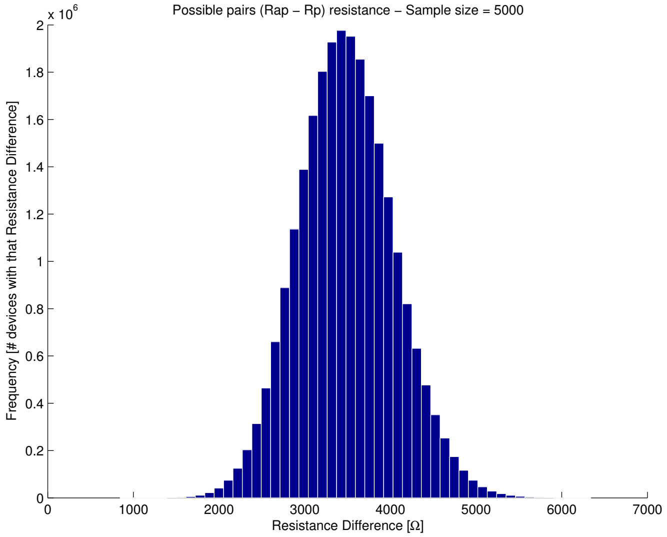


Figure 13. Possible pairs of $(R_{ap} - R_p)$ taken as resistance differences

A similar result is shown in Figure 14, which compares now the possible pairs of $(R_{ap_i} - R_{ap_j})$ and $(R_{p_i} - R_{p_j})$. The histograms referred to these two groups shows that taking the differences between resistances of antiparallel states leads to a much wider distribution, that is, characterized by an higher dispersion.

The reason why we bother about resistance difference lies again on the fact that this attribute will be exploited later when proposing new PUF schemes for magnetic memory: given a certain amount of applied voltage to a set of memory cells, in fact, resistance difference between two cells can be translated immediately into current difference. And the difference between the two currents can be sensed by an amplifier leading to a PUF bit.

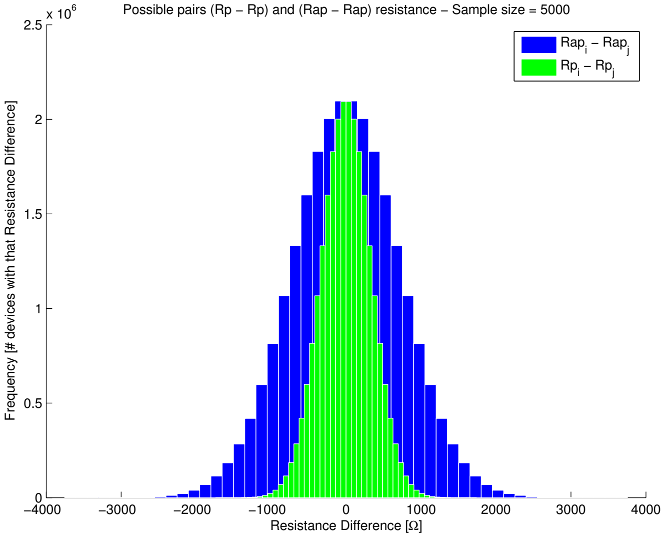


Figure 14. Possible pairs of $(R_{ap_i} - R_{ap_j})$ and $(R_{p_i} - R_{p_j})$

Direct consequence of Figure 14 is in fact Figure 15, highlighting the difference between the two currents flowing across each of the possible couples of MTJs constituting the population, both of them programmed either in parallel or antiparallel state. The R_{ap} population performs better in terms of number of different values (higher dispersion, that is higher standard deviation). However, being the resistance higher (more than double with respect to R_p with the chosen TMR_0), this leads to a lower current for the same V_{MTJ} across the cell, if compared with the parallel state. This implies the need of a much more reliable sensing amplifier to sense correctly differential currents and produce a reliable PUF bit. It seems than that while PUF uniqueness seems to fit better the R_{ap} population, on the other side reliability is more robust for the the R_p population. The concepts of Reliability and Uniqueness, already briefly introduced qualitatively in 2.4.5, will be defined, studied and evaluated later in this Chapter.

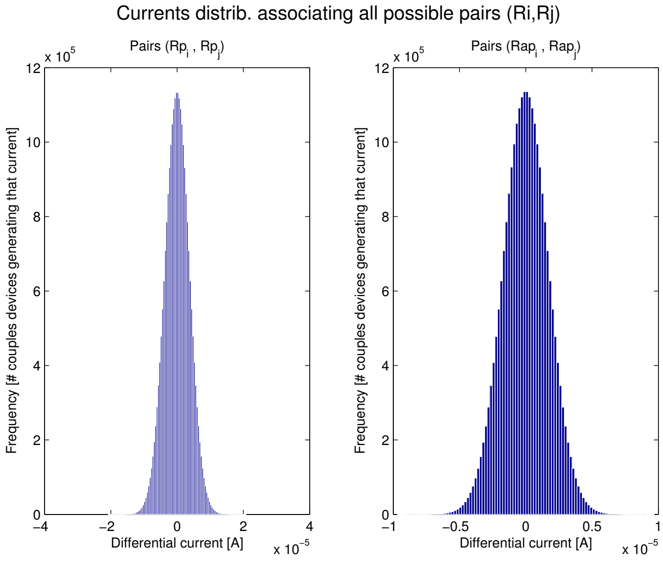


Figure 15. Differential currents when $V_{MTJ} = 100mV$, normally distributed, is applied to the memory cells. The parallel configuration ensures a population of differential currents with an intensity slightly more than doubled with respect to R_{ap}

3.2.3 Effect of the variation of the MTJ oxide thickness over the population

The oxide thickness parameter, t_{ox}, is a fundamental variable that plays an important role in defining the associated resistance of the MTJs constituting an STT-MRAM memory chip. The reason lies in the microelectronic world: wherever there is a quantum tunneling barrier, the probability of an electron (synonym of current) to cross it is inversely proportional to the barrier height (the energy in electron-volts, eV) and thickness (a physical distance in nanometers). The longer the path through an insulation layer, the lower the probability of an electron to go through the barrier and reach the second ferromagnet. This impedance to cross the barrier can be seen as conductance (Equation 2.3) and then easily transformed in an equivalent resistance (Equation 2.4).

The aim of this Subsection 3.2.3 is then to investigate, with a simulation framework, how the oxide thickness variation influences the performance of the populations R_p and R_{ap}.

First of all, assuming V_{MTJ} and the feature size (A, B) constant (without any process variation), the dispersion of the MTJ resistances is then analyzed solely as function of the variability of the oxide thickness. In particular, at three different thicknesses ($t_{ox} = 0.85nm; 1nm; 1.15nm$), the oxide thickness standard deviation is let vary in a range $0\% \leq \sigma_{tox} \leq 5\%$. The effect over the generated population of MTJ memory cells (as usual, $N = 5000$) is then measured detecting as relevant parameter the amount of standard deviation of both R_p and R_{ap} over their respective mean values, and the result is depicted in Figure 16.

As the result shows, for no variability over the oxide thickness we obtain an ideal population of parallel and antiparallel state, that is, only one possible value for both of them (the nominal one). This is due to the fact that no process variation occurs at all.

On the other hand, increasing the amount of process variation over the oxide thickness (always present in any manufacturing process), the amount standard deviation of the parallel and antiparallel resistances increases linearly. In is important to note that both parallel and antiparallel configurations behave exactly the same; after all, the ratio between the standard deviation and the mean value is the same both for parallel and antiparallel configuration, an intrinsic property of MTJs.

Furthermore, for the same amount, in percentage, of process variation affecting the oxide thickness, at higher oxide thickness corresponds an higher amount of variability of the generated populations of MTJs.

In other words, the populations R_p and R_{ap} both exhibit an higher variation when the oxide thickness increases and its intrinsic variability, unavoidable and due to the manufacturing process, increases as well.

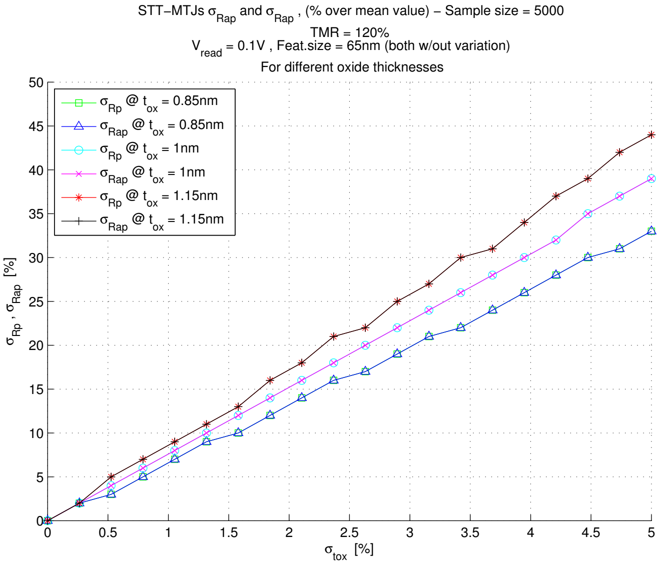


Figure 16. σ_{Rp} and σ_{Rap} variation as function of oxide thickness t_{ox} and its amount of standard deviation. V_{MTJ} and feature size assumed to be constant and not varying at all.

Then, also V_{MTJ} and the feature size are varied according to a Gaussian probability density function, with a standard deviation of 5%. The effect of introducing this additional pair of process variation is to make the population of parallel and antiparallel resistance of the magnetic memory cells vary more (higher standard deviation), but the effect is visible and present only when the oxide thickness' standard deviation is roughly less than 2% of the nominal value: above it, it does not matter. Practically any MTJ stack used nowadays in STT-MRAM is below this threshold (typically around or below $1nm$), thus not only oxide thickness but also reading voltage and feature size variations play an important role in assessing the performance of the normal distribution characterizing R_p and R_{ap}. The result is shown in Figure 17.

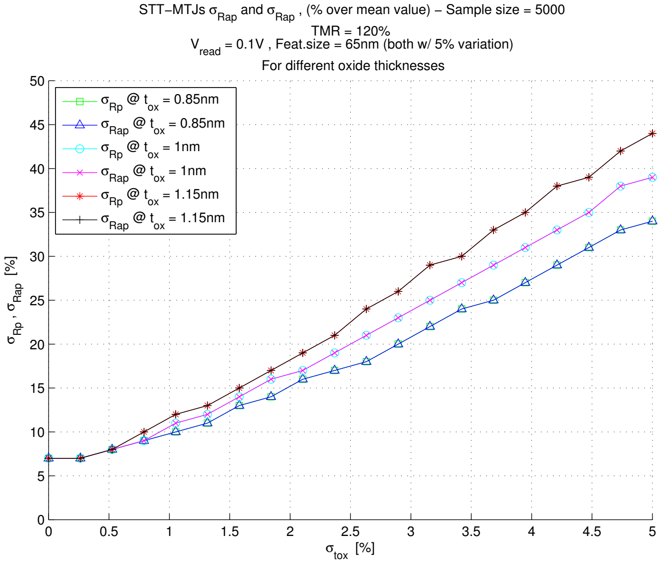


Figure 17. σ_{Rp} and σ_{Rap} variation as function of oxide thickness t_{ox} and its amount of standard deviation. V_{MTJ} and feature size are now normally distributed as well, thus both affected by a standard deviation (kept to 5%).

The suitability of an STT-MRAM chip as digital memory imposes total absence of overlapping between the two resistances populations R_p and R_{ap}: a binary zero (expressed by a cell characterized by R_p) must be always recognized and distinguished from a binary one (a cell programmed in antiparallel state thus with an associated resistance R_{ap}). It is then important that, in other words, $max\{R_p\} < min\{R_{ap}\}$. The effect of the process variation of oxide thickness, feature size and reading voltage can then be interpreted again as function of the distance between the two populations of resistances characterizing the parallel and antiparallel state. Asking absence of overlapping means, in the limit case, that: $min\{R_{ap}\} - max\{R_p\} \geq 0$; the complementary condition is not acceptable when STT-MRAM is used as digital memory.

It is interesting to note, as shown in Figure 18, that for small increases in the nominal value of the oxide thickness, as soon as its associated standard deviation increases, the performance worsens: in other words, the two populations start to become closer to each other and to overlap at a much faster rate. As long as the model adopted in the simulations framework is concerned, this dangerous trend is instead almost influent for lower oxide thicknesses, below $1nm$.

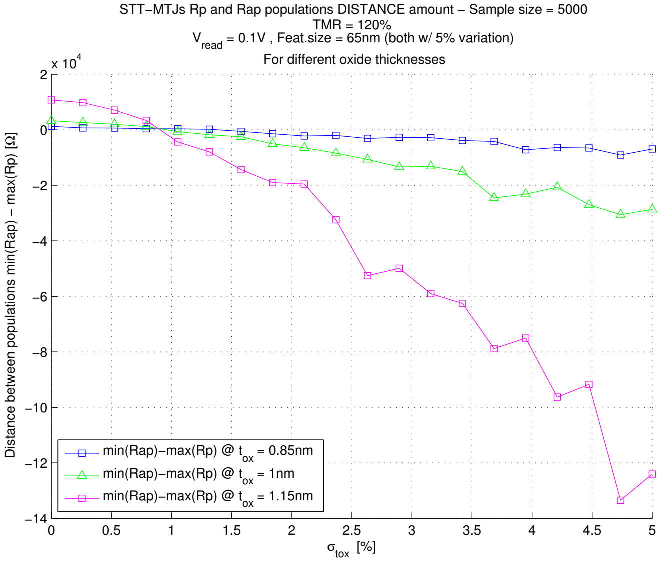


Figure 18. Distance between populations of parallel and antiparallel states function of the standard deviation of oxide thickness, for three different values of t_{ox}; $\sigma_{V_{MTJ}}$ and $\sigma_{A,B} = 5\%$.

Finally, Figure 19 shows the same behavior highlighted in Figure 18 but expressing now the distance between the two means of the populations as multiple of the standard deviation of the parallel state associated resistance. Typically, in fact, when it comes to resistance analysis of MRAM, it is convention to say “by how many sigma” the two populations are distant, and the ideal value usually is above 10.

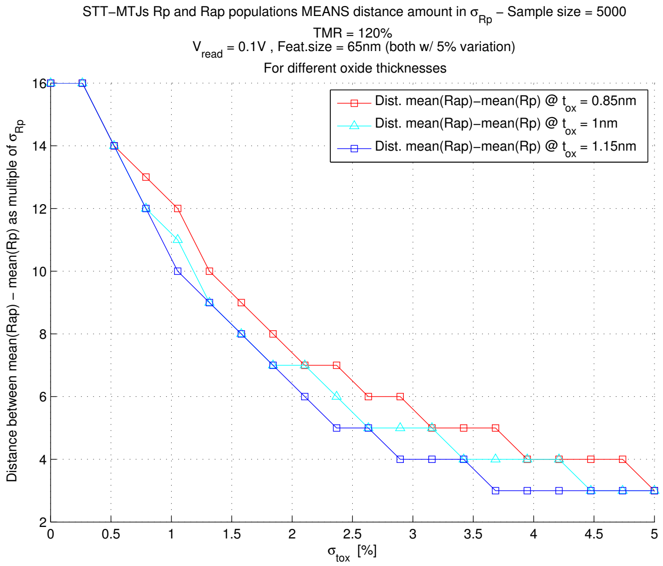


Figure 19. Distance between the mean values of the populations of parallel and antiparallel states expressed as multiple of σ_{Rp}, as function of the oxide thickness.

From now on, an oxide thickness of $0.85nm$ with a 1% standard deviation is chosen – together with the other parameters already mentioned at the beginning of Subsection 3.2.3. These are the specifications that we simulate to be provided by a silicon foundry for a given manufacturing process of STT-MRAM, that will be used in the next sections to build up schemes and methodologies on the use of STT-MRAM chips as Memory PUF.

3.3 Improving the usage of STT-MRAM as PUF

After studying the model built up to generate a population of MTJ memory cells, the simulation script has been used in the second part of the thesis work: applying new methodologies and circuits layouts to generate a PUF signature out of an STT-MRAM memory chip.

In particular, it is worth highlighting that in the previous work [17] a single STT-MRAM memory chip is used to generate a PUF signature: each word of the memory matrix represents a different PUF signature and the sets of these response strings is evaluated in terms of stability and randomness. The principle has already been depicted in Figure 8.

However, this approach is rather experimental since it focuses on validating the intrinsic behavior of an STT-MRAM chip as capable of generating Physical Unclonable Function, where the challenge is a selection of a particular word line, and the response is the result given by currents mismatch flowing into couples of magnetic tunneling junctions (Figure 9).

To expand this approach, the first remark that needs to be done is the assumption that STT-MRAM will be present, in the near future, in a particular set of chips. Let us then consider a production lot of N printed circuit boards, each of which containing integrated chips interconnected among them and that part of the memory technology embedded in the PCB is based on STT-MRAM. For example, we might assume that the L2 cache of the on-board microprocessor, or a rather secondary storage memory chip is implemented as magnetic memory for the need of non-volatility, speed, high density or any combination among the advantages that MRAM brings to digital systems. The scenario is highlighted in Figure 20, where on the left we find the design of a chip including an STT-MRAM module that enters production: in a manufacturing facility, then, several chips composing a production lot are fabricated and each of them includes a module of STT-MRAM used for a particular memory-related purpose.

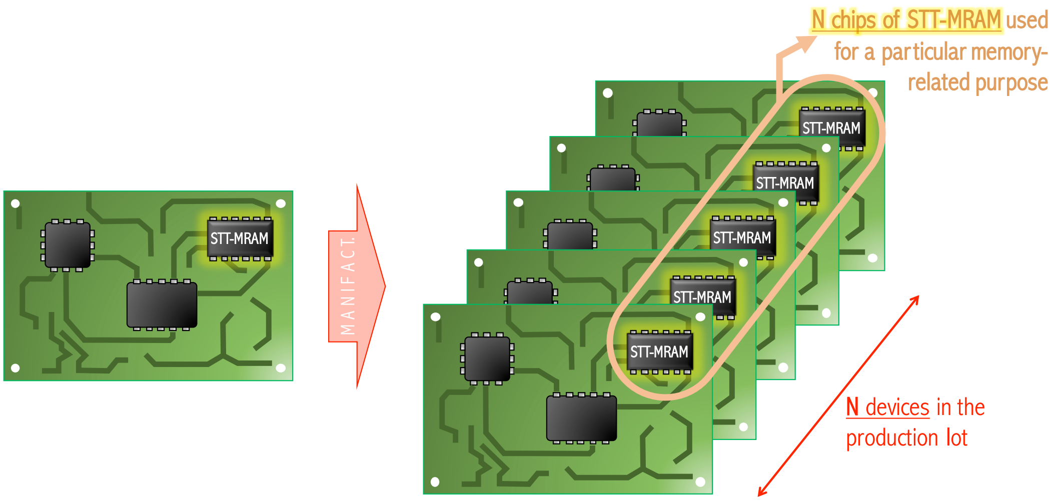

STT-MRAM
MANIFACT.
N chips of STT-MRAM used for a particular memory-related purpose
N devices in the production lot

Figure 20. A set of N devices produced at a manufacturing facility each of which containing a STT-MRAM memory chip.

Keeping in mind the goal of Hardware Security, achievable for example through PUF, each of these STT-MRAM chips, within its respective belonging PCB, can be used to generate a PUF signature. Each device of the production lot will then be able to be identified through a unique signature generated exploiting the non-volatile magnetic memory embedded inside it and primarily used for for digital memory purposes.

The first big difference with the previous work is now evident: **rather than using a single chip of STT-MRAM as trial to generate PUF Signatures, one for each word line**, now a population of different chips, counting N elements, is analyzed, and now **for each**

element (STT-MRAM chip) the entire memory chip is used to generate a single PUF Signature rather than single memory words. This scenario is depicted in Figure 21.

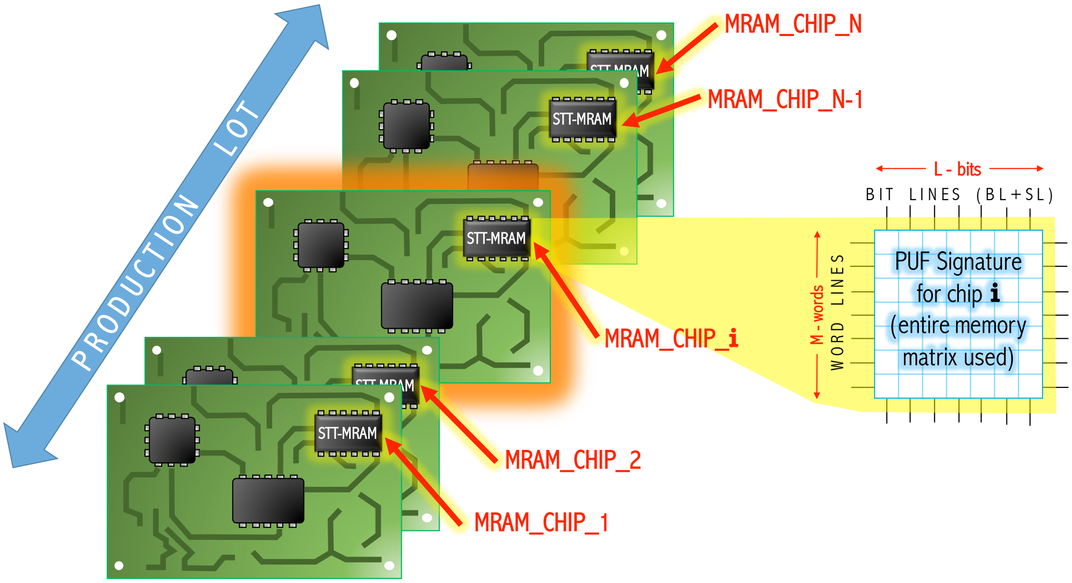


Figure 21. The PUF signature is produced using, for each chip, the entire STT-MRAM memory matrix for a single signature. Each PCB can then be identified exploiting the resistance variation of the MTJs constituting the cells of the magnetic memory.

Clearly, there are challenges coming with this approach: since now the memory space used to generate a single PUF Signature is an entire memory matrix rather than simply a word, we basically require the use of far more MTJ memory cells as target for the generation of the signature. However, at the same time, the Gaussian distribution of the parallel and antiparallel resistances is obviously the same. This means that there is a far higher probability that two or more memory cells have a much alike resistance value, when increasing the overall number of elements constituting the population of MTJs itself. Direct consequence of this observation is the fact that the quality of the generated PUF, both in terms of uniqueness (randomness) and reliability might worsen. To verify this new methodology:

1. The previous PUF generation schemes proposed in [17], being called Case 1 and Case 2 in Figure 9, are now applied to this new architecture variation: now for each STT-MRAM chip a unique PUF signature is generated;
2. Two new schemes, that we call **Case 3** and **Case 4**, have been proposed in the next Section 3.4 inspired by the same of Case1 and Case2, but exploiting the fact that using now an entire STT-MRAM memory chip for a single PUF Signature means having far more cells to play with which can then **grouped** in particular sets to generate and sense differential currents with new methodologies.

3.4 Design schemes variations

While the implemented Case 1 and Case 2 are the same used in the previous work in [17], two new layouts have been proposed within the new scenario introduced in Section 3.3. In particular, Case 1 and Case 2 have already been extensively described in Subsection 2.5.2, while the two new cases, extension of the previous methodology, are now introduced as follows:

- **CASE 3**: extension of CASE 1, the memory uses still a reference column but now both data cells and reference cells are of the type 2T2MTJ. A single data cell is compared with a single reference cell and, in order to take advantage of the slight mismatches leading to the feasibility of generating PUF random bits, a new sensing scheme for differential currents is adopted. In particular, now for both cells: one MTJ is programmed as antiparallel (AP) and the second one as parallel (P). Using the same reading voltage, two currents, I_{DATA1} and I_{DATA2} are generated, and their difference ΔI_1 is considered. The same approach is used for the reference cell as well (I_{REF1} and I_{REF2} whose difference leads to a ΔI_2), and finally the two currents ΔI_1 and ΔI_2 are sent to a sense amplifier which determines the PUF output bit. Ideally $\Delta I_1 = \Delta I_2$ but thanks to the dispersion of the resistances R_P and R_{AP} the value of the output bit provided by the sense amplifier is totally random and unpredictable, suitable for PUF purposes;

- **CASE 4**: extension of CASE 2 - no reference cell is present, and the entire memory is of type 2T2MTJ. The idea of comparison and PUF generation is the same as in CASE 3 but now the absence of a reference cell imposes the use of an higher number of memory cells, of the type 2T2MTJ. Each couple of cells guarantees a single PUF bit then, in order to get a PUF Signature as wide as CASE 3, a memory of double size must be used. Hopefully, as in CASE 1 vs CASE 2, CASE 4 would provide a better uniqueness in the generated signatures with respect to CASE 3, since the systematic similarity introduced by the use of the same reference cell for more than on PUF bit is avoided.

It is possible to note that these two new sensing schemes are based on 2T2MTJ memory cells. This kind of layout is becoming more and more dominant within STT-MRAM chips for different reasons: not only an higher density of integration but especially for sensing purposes during the use of magnetic memory as digital memory. In fact, with device scaling parasitic components start to play a more dominant role thus sensing a single MTJ by means of current and voltage may lead to an high bit error rate (BER) due to parasitic terms over imposing and dominating in the reading/writing phases. From here the need of differential sensing schemes that make use of a couple of MTJ always programmed in dual form. For example, to store a binary '0' the sequence is $AP - P$, while an '1' is identified by $P - AP$ or viceversa. Particular electronic circuits, surrounding the read/write circuitry, are then used to either store or read out a value memorized in this complemented form.

Figure 22 shows a direct comparison among CASE 1, CASE 2, CASE 3 and CASE 4.

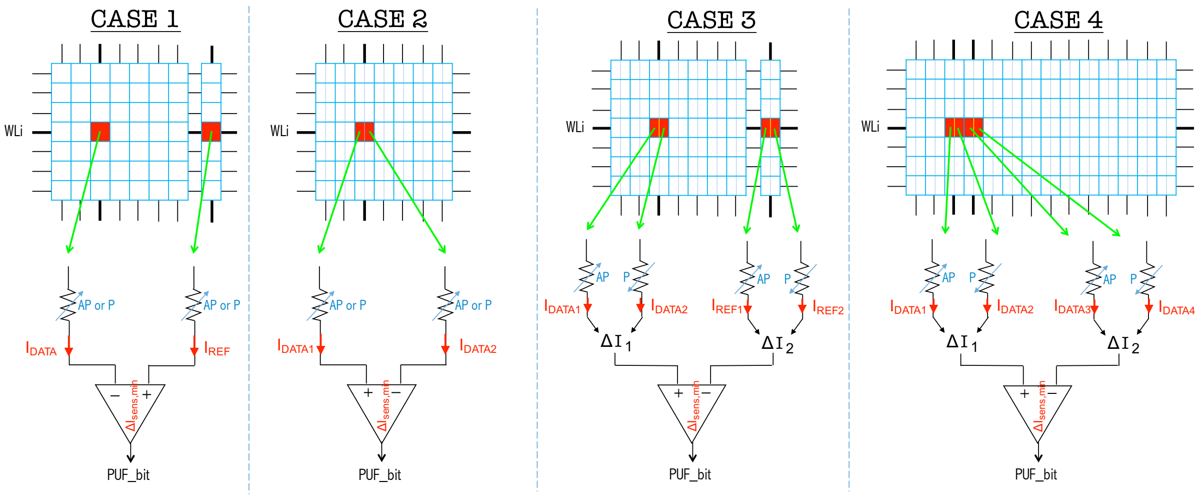


Figure 22. A comparison among CASE 1, CASE 2, CASE 3 and CASE 4 - the sensing schemes used in the simulation framework to evaluate the quality of the PUF coming from a set of memory matrices.

To be more precise, for CASE 1, 2, 3 and 4 a different number of MTJ-MOS couples and/or memory cells is required in order to end up having a PUF signature, in a matrix form, of the same length. Taking in mind Figure 22, Table IV shows the number of devices and the size of the memory needed for an example of a PUF Signature composed of a binary matrix of $M \times L$ bits in size (be aware that now M and L are used to identify the dimension of the PUF Signature, rather different from the physical size of the memory according to the Case implemented).

For all the simulations, we chose a **PUF Signature size of** 32×32 **bits** for each of the N chips of the production lot. Also, **the production lot is constituted of** $N = 100$ **different chips** (or, better, PCBs, each of which embeds an STT-MRAM chip).

TABLE IV
A COMPARISON OF THE EFFECTIVE MEMORY SIZE NEEDED TO REALIZE A PUF SIGNATURE OF A GIVEN SIZE, CASE BY CASE.

Given a PUF Signature matrix of size $L \times M$***:***

Case	**Words length**	**N of words**	**Cells Type**	**N of MTJs**
1	$L_{real,1} = L + 1$	M	1T1MTJ cell + 1T1MTJ Ref.	$(L + 1) \times M$
2	$L_{real,2} = 2 \times L$	M	2T2MTJ cell, no Ref.	$(2 \times L) \times M$
3	$L_{real,3} = 2 \times L + 2$	M	2T2MTJ cell + 2T2MTJ Ref.	$(2 \times L + 2) \times M$
4	$L_{real,4} = 4 \times L$	M	two 2T2MTJ cells, no Ref.	$(4 \times L) \times M$

An important final remark worth mentioning is the need of keeping consistency among the results of the simulation for all the four cases. The generated population of MTJs is a 3D matrix of sizes $M \times (4 \times L) \times N$ – this ensures that the biggest memory is generated (Case 4 needs more memory space with respect to all the other cases) and then more or less memory is effectively used and grouped (1T1MTJ vs 2TMTJ, the presence of a Reference column, etc.) according to the specific Case to be studied. By doing so, the population of MTJ memory cells is generated once and is used as baseline for all the test Cases. This is of fundamental importance when it comes to comparing the quality of the generated PUF among Cases, since starting from different populations generated at different times (due to different sizes) leads for sure to differences among the populations generated each time, in opposition to the principle of a common baseline to evaluate uniformly and consistently which would be lost.

3.5 Simulations details and metrics: evaluating the quality of a PUF

The second part of the simulations has been built focusing on CASES 1, 2, 3 and 4 implementation and evaluation. The main steps carried out by the simulator are here summarized:

1. **Generate the population of STT-MRAM memory chips**, each of which constituted by a memory matrix of size as big as $M \times (4 \times L)$ – having a population of N element, then, the entire data set has been implemented as a three-dimensional matrix of sizes $M \times (4 \times L) \times N$. Each slice of the matrix (elements indexed by i=1...N) is a 2D matrix corresponding to the memory matrix of a single STT-MRAM chip of the production lot. Each element within each matrix represents and emulates the memory cell, and is implemented as a structured data (struct) with fields R_p, R_{ap}, V_{MTJ} and these values are generated using the algorithm already presented in Section 3.2 to generate a population of MTJs. It is worth recalling that for all these simulations, a **PUF Signature size of 32×32 bits** has been selected and **the production lot is constituted of $N = 100$ chips**.

2. **Generate the PUF response string** by applying methods of CASE 1, CASE 2, CASE 3 and CASE 4. Every time the PUF signature is generated, this binary string is then saved into each 2D matrix itself – one more field of the structured data representing each memory cell is then the *Response_bit*. It is important to remark that from the PUF standpoint, the size of the response string is a 2D matrix of size $M \times L$. For each case, then, we have more memory cells available than response bits (as already highlighted in Table IV). The matrix depicted on the left side of Figure 23 provides an additional and clearer explanation on this matter;

3. **Evaluate the PUF Quality, in terms of uniqueness and reliability / bit–error rate (BER)**. The aim of the present Section 3.5 is to provide an analytical introduction on the equations and methodologies used to characterize quantitatively the quality of the generated PUF Signatures for all four Cases.

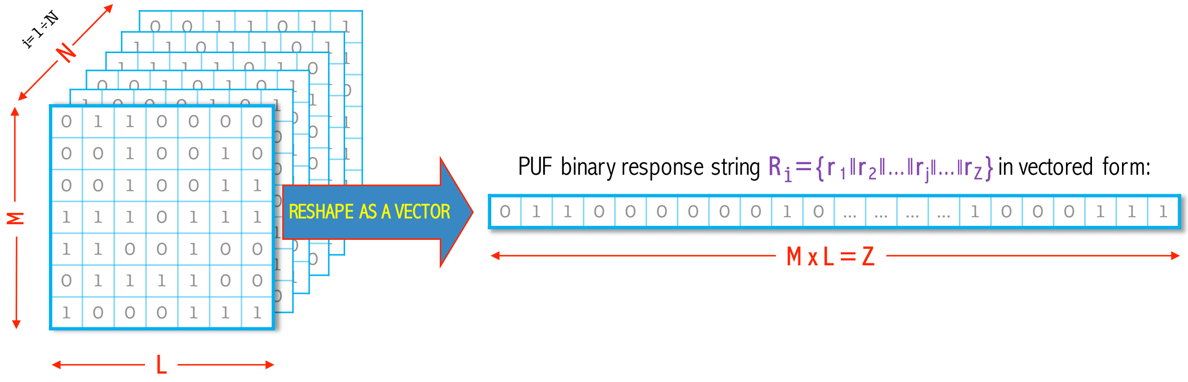


Figure 23. Transforming the binary response string of the PUF from matrix to vectored form.

In order to evaluate the PUF Quality, the initial requirement is to structure the PUF binary response string in a vectored form. Since the initial data structure is a matrix, for each response string, it is needed to reshape it as a vector. The length of the vector will then be $M \times L := Z$.

The matrix–to–vector reshaping is highlighted in Figure 23. Each response string, corresponding to a given chip $i = 1 \dots N$, is indicated in the form $R_i = \{r_1 \parallel r_2 \parallel \dots \parallel r_j \parallel \dots \parallel r_z\}$. The generic PUF response bit is then identified as r_j.

The methodology used to evaluate PUF quality follows an approach that is the combination of the benchmark metrics presented in [25], [26] and [24], and it represents one of the most extensive set of equations used to test and check the quality of a generated PUF Signature from the uniqueness standpoint. Subsections 3.5.1 and 3.5.2 are dedicated to explain these PUF quality measurement metrics.

3.5.1 Uniqueness evaluation metrics

Let us assume that the generated set of PUF Signatures is not affected by errors, that is absence of uncertainties in the response strings due to systematic or random errors that may lead in the generation of some of the response bits in an unpredictable and unstable manner. The fundamental parameter to evaluate the quality of a PUF is called uniqueness, and it is a measure of how unique (different) the PUF Signatures are among each other. In order to get a good PUF out of a certain digital system, we require an architecture that leads to a signature that is easy to generate and as random as possible. In other words, process variation must vary a lot in such a way that the possible number of non-identical signatures, among different chips, is as high as possible. Uniqueness refers then to the analysis of the entire population of chips, thus we call it an *inter-chip* (or equivalently inter-die) quantity. In order to carry out from a mathematical standpoint a quantitative analysis of the uniqueness parameter, the concept of *Hamming Distance* is used. For this reason, uniqueness is typically a synonym of **inter-chip Hamming Distance**.

The **Hamming Distance** expresses how much alike (or different) two binary strings (thus streams of data constituted by 1's and 0's) are. The formula used to evaluate the uniqueness is stated in Equation 3.1 and it gives the average inter-die Hamming Distance. It considers all the possible couples (using subscripts u, v) of binary strings R_i and provides an average over the entire population of Hamming Distances. All the other constants and variables in Equation 3.1 have exactly the same meaning discussed in Section 3.5.

The ideal value os 0.5, or 50%. It is important to remark that together with the average value, considering all the possible single couples, we get a set of Hamming Distances that can be represented in terms of mean value (given by Equation 3.1 itself) and standard deviation (computing the dispersion over the population of possible Hamming Distances).

$$HD_{avg} = \frac{2}{N \times (N-1)} \cdot \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \frac{HD(R_u, R_v)}{Z} \cdot 100\% \tag{3.1}$$

Uniqueness evaluation comes with two sub-parameters that contribute to assess the quality of the PUF from the uniqueness standpoint itself. The first one is the **Hamming Weight** (here HW for short), which expresses whether the response string is uniformly distributed – in other words, for each of the chips i constituting the population of N elements, it provides information on whether an equal number of 0s and 1s is used. Again, the ideal value is 0.5, or 50%. Equation 3.2 states the formal definition of Hamming Weight.

$$HW = \frac{\sum_{t=1}^{Z} r_{i,t}}{Z} \cdot 100\% \tag{3.2}$$

The second parameter is the **Bit Aliasing** (BA for short), a concept similar to the Hamming Weight – now the bits in the same position within different strings are simultaneously considered and we check whether they have the same value. This quality metric checks whether common patterns are present in the same regions among different signatures. The extreme negative case is $BA = 0$ meaning that the inter-die Hamming Distance would be zero too (worst case). Again, the ideal value is 0.5, or 50%. The formula for Bit Aliasing is quoted in Equation 3.3.

$$BA = \frac{\sum_{t=1}^{N} r_{i,t}}{N} \cdot 100\% \tag{3.3}$$

3.5.2 Reliability – BER evaluation metrics

Let us assume that the generated set of PUF Signatures is now affected by errors. **Reliability** is a concept of wider meanings and interpretations when it comes about PUF, since it is strongly dependent on the specific technology used. Typically, it identifies by how much the output string of the PUF changes when some particular operating conditions, typically temperature T and voltage V, fluctuate. Since the analysis focuses on analyzing different sets of data within the same chip, we refer to **intra-chip Hamming Distance**.

The generic formulation is introduced in Equation 3.4, where for a given chip i the Hamming Distance is computed on the same associated response string for different operating conditions (considering x different operating conditions, each of which identified by the variable $y = 1 \ldots x$ leading to a set of responses produced by the same chip and thus hopefully and ideally coincident). When it comes to Reliability, the lower the better (ideally, zero).

$$Reliability = \frac{1}{x} \cdot \sum_{y=1}^{x} \frac{HD(R_i, R_{i,y}^{'})}{Z} \cdot 100\% \tag{3.4}$$

The result of Equation 3.4, once computed for all the N chips, is then again a population that can be characterized, as the HW, by an average value and a standard deviation.

The sensibility to voltage and temperature fluctuations is very important to assess the performance of a PUF. In our specific case, when STT-MRAM needs to be analyzed as PUF system, this approach is not so useful – the model used, in fact, even though is currently the best to describe a population of MTJs in a STT-MRAM memory array, does not contemplate temperature dependance while voltage dependance, although present, will not lead to appreciable results which are relevant and representative for Reliability evaluation purposes. We will observe this more rigorously in the next Sections when the simulation outputs about Reliability are presented and discussed.

Coming to STT-MRAM, an equivalent concept of Reliability is the **Bit-Error Rate**, or **BER** for short. We assume that the Reliability of our system is dependent only on the quality of the sense amplifier, which is characterized by a minimum differential current $\Delta I_{sens,min}$ that it is able to sense, below which the output is considered an error (and called *error bit*) since unpredictable and not correctly sensed.

We refer here to an **extrinsic reliability**, due to the sensing amplifier not any longer ideal and easily computable through Equation 3.5. For higher $\Delta I_{sens,min}$ we expect a poorer BER.

$$BER_{\%} = \frac{N^{\circ}\_error\_bits}{Z} \cdot 100\% \tag{3.5}$$

A final important remark on BER: in the simulation framework, **not only $\Delta I_{sens,min}$ but also V_{MTJ} is considered to evaluate the Bit-Error Rate**. In fact, when the nominal applied reading bias to the memory cells varies, so does the current flowing through them and then the current difference needed to be estimated in all Cases 1, 2, 3 and 4 varies. This last quantity, then, can become closer to $\Delta I_{sens,min}$ when the applied reading bias decreases, thus the influence of V_{MTJ} needs to be actively taken into account as well.

3.6 PUF Quality simulations results and critical analysis

The simulation framework has been built both to implement the four different design schemes (Case 1, 2, 3 and 4) and to evaluate the Quality of the generated PUFs case by case, applying the concept of Hamming Distance and Bit-Error Rate. The next two Subsections 3.6.1 and 3.6.2 report the results of these simulations and provide a critical insight comparing the results of each implementation case.

3.6.1 PUF Uniqueness study for all design schemes

First of all, Uniqueness has been evaluated for Case 1, Case 2, Case 3 and Case 4. In the present Subsection, results are first discussed individually for each case, then they are compared all together in a table to understand similar trends or differences.

3.6.1.1 Case 1 study

CASE 1, P configuration: the first memory layout (Case 1), using all the MTJs initially programmed in the parallel state, provides a good inter-chip Hamming Distance, extremely close to the ideal value (50%). Also, the standard deviation is reasonably small. Both Hamming Weight and Bit Aliasing are not very noisy, confirming that there are not common pattern among the PUF Signatures of different chips as well as 0s and 1s are used in a quite balanced way in each signature. Results are depicted in Figure 24.

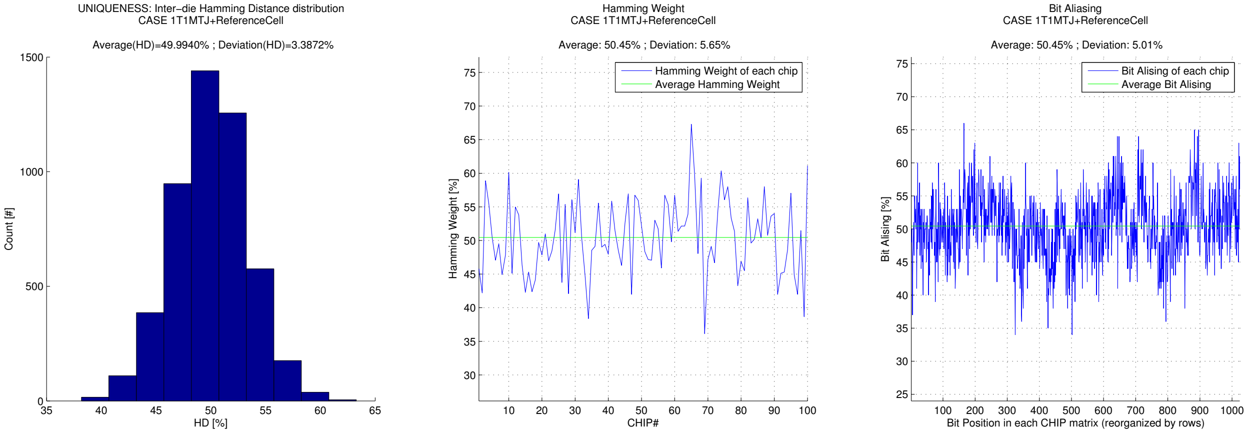


Figure 24. PUF Uniqueness evaluation for CASE 1, MTJs all pre-programmed in parallel (P) configuration.

CASE 1, AP configuration: it is interesting to note how the Uniqueness results for the antiparallel (AP) configuration follows exactly the same pattern of the parallel (P) case – surprisingly, programming the MTJs as antiparallel does not lead here to any significant improvement in terms of Signatures uniqueness. Considering the signatures all together, they appear to be all as random as in the case of parallel cells programming. Results are depicted in Figure 25.

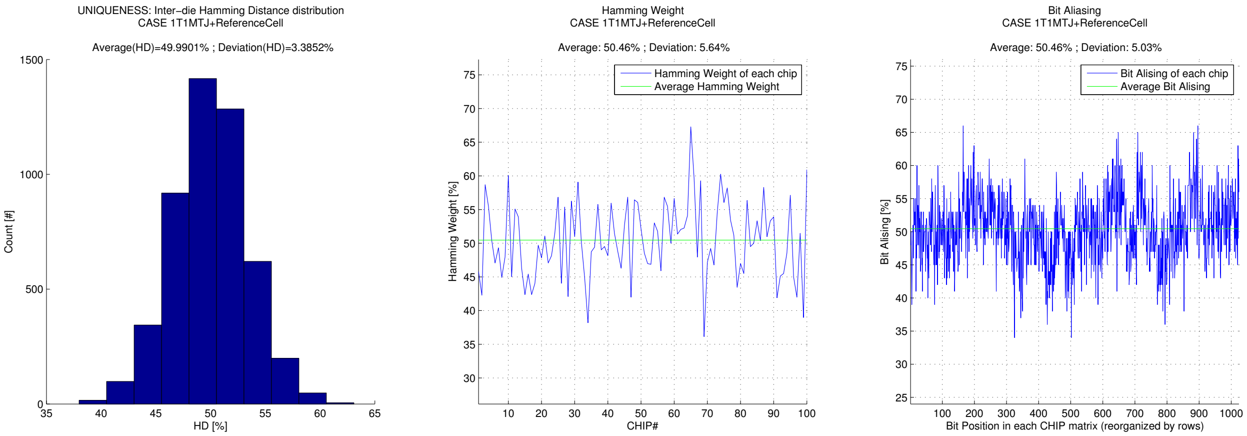


Figure 25. PUF Uniqueness evaluation for CASE 1, MTJs all pre-programmed in antiparallel (AP) configuration.

3.6.1.2 Case 2 study

CASE 2, P configuration: removing the presence of a reference cell, thus using 2T2MTJ memory cells, an improvement in terms of Uniqueness is immediately visible. Not only the inter-chip Hamming Distance standard deviation is much reduced, but the Hamming Weight is significantly improved as well. The presence of an higher number of MTJs (doubled with respect to CASE 1) leads to a more balanced usage of 0's and 1's within the same signature (less standard deviation). No significant improvement is observed instead in the Bit Aliasing standard deviation. Furthermore, as in CASE 1, both inter-chip Hamming Distance and Hamming Weight and Bit Aliasing are characterized by a nominal value that is always close to the ideal one (50%). Here the improvement with respect to Case 1 is not really about the nominal value but rather about the standard deviation of the metrics used to evaluate the PUF Quality itself – the PUF Signatures experience quality standards that change much less when moving in the analysis among the entire populations of chip, as confirmed by a reduction in terms of standard deviation of both Hamming Distance and Hamming Weight. Results are depicted in Figure 26.

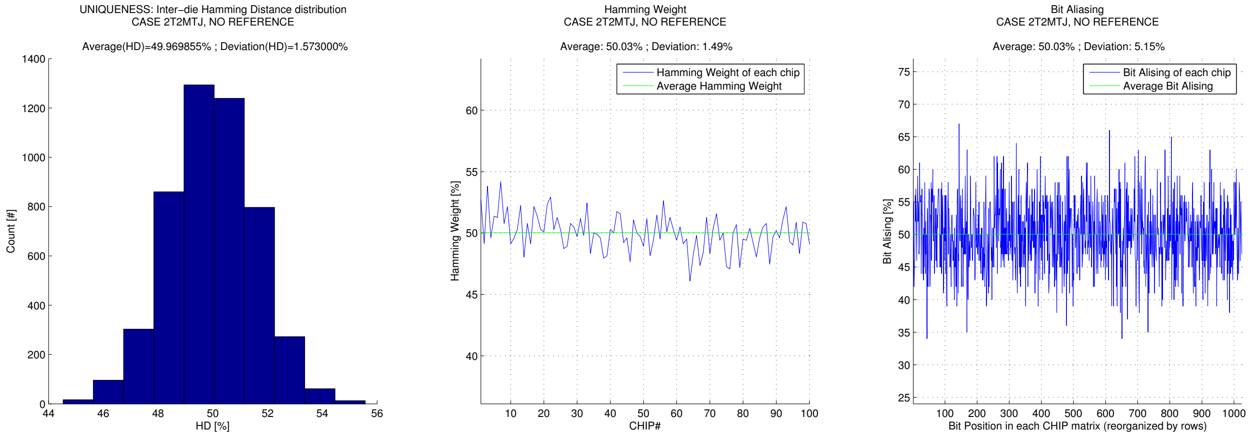


Figure 26. PUF Uniqueness evaluation for CASE 2, MTJs all pre-programmed in parallel (P) configuration.

CASE 2, AP configuration: the situation is similar to what happens for Case 1, in the analysis P vs AP – no reasonable change in terms of PUF Quality is observed with respect to Case 2's P configuration. Yet, both P and AP in Case 2 offer higher PUF Quality in terms of Uniqueness with respect to P and AP of Case 1. Results are depicted in Figure 27.

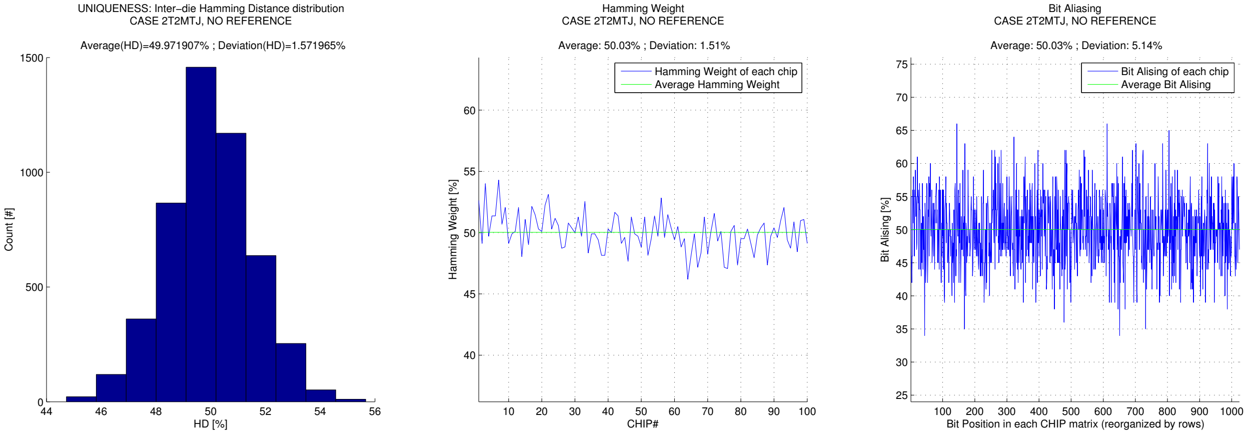


Figure 27. PUF Uniqueness evaluation for CASE 2, MTJs all pre-programmed in antiparallel (AP) configuration.

3.6.1.3 Case 3 study

CASE 3, reversed-combo (P+AP) configuration: using a rather higher number of MTJs with respect to Case 1 and Case 2, the PUF Quality is not worsened. The hypothetical trouble is due to the fact that increasing (roughly doubling) the number of MTJ elements in Case 3 (as well as in Case 4), keeping obviously the same the standard deviation of the parallel and antiparallel MTJ resistive states, having an higher number of different MTJs means that their resistive values are less varying from each other thus potentially leading to a worse PUF Quality.

Fortunately this is not the case, because introducing as additional element of randomness the difference of currents ΔI_1 and ΔI_2 counterbalances and compensates it. Then, when digital system design needs to bring the memory chip to operate in a reversed-combo configuration, for example to improve the reading sensitivity as memory and the signal-to-noise ratio, the PUF Quality is not affected by this choice at all. Case 3 performance as PUF are comparable to Case 1. The reference cell is dominating the PUF Quality in both of them and in a comparable manner. This is not all – not only the PUF Quality is still very high, but with respect to Case 1 the standard deviation of both Inter-Die Hamming Distance and Hamming Weight is slightly improved, even though Case 3 still uses Reference cell. Bit Aliasing is again the one whose standard deviation keeps playing around 5%. Results are depicted in Figure 28.

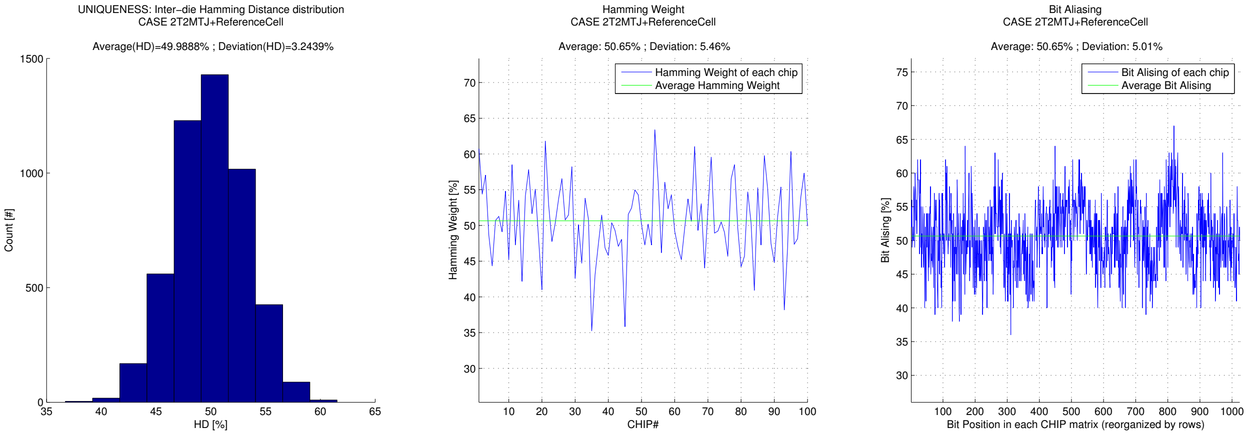


Figure 28. PUF Uniqueness evaluation for CASE 3, MTJs pre-programmed in reversed-combo (P+AP) configuration.

3.6.1.4 Case 4 study

CASE 4, reversed-combo (P+AP) configuration: avoiding the use of a Reference cell, thus doubling the size of the MTJ memory matrix, does not bring additional benefits in terms of PUF Quality with respect to Case 3. In other words, while between Case 1 (with reference) and Case 2 (no reference) the standard deviation of PUF Quality parameters is drastically improved, for Case 4 vs Case 3 this does not happen. After all, Case 3, although theoretically limited by the presence of the systematic reference cell resistance for each row of the PUF, offers already a much higher number of memory cells with respect to Case 1 and Case 2, contributing into maintaining the set still composed by signatures that vary a lot among them. The only real benefit here is constituted by a drastic reduction of the standard deviation of the Hamming Weight. Results are depicted in Figure 29.

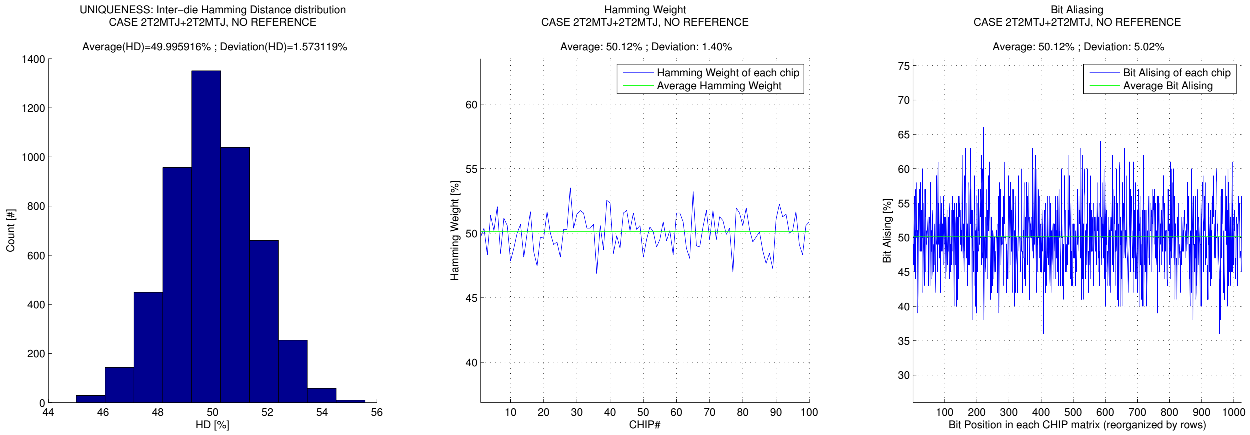


Figure 29. PUF Uniqueness evaluation for CASE 4, MTJs pre-programmed in reversed-combo (P+AP) configuration.

3.6.1.5 Comparative study among Cases

Finally, PUF Quality in terms of Uniqueness is reported in Table V which compares Case 1, Case 2, Case 3 and Case 4 simultaneously. As it is possible to note, All cases offer a good nominal value for Inter-chip Hamming Distance, Hamming Weight and Bit Aliasing, proving that STT-MRAM can be effectively used as PUF generator. However, Case 1 is the worst in terms of standard deviation of its parameters.

From Case 2 on, they all behave more constantly close to their respective nominal value and increasing (doubling for Case 3, and making roughly four times bigger for Case 4) the size of each STT-MRAM memory chip does not impact in a negative manner on the PUF Uniqueness at all.

TABLE V
PUF UNIQUENESS EVALUATION: A COMPARATIVE STUDY AMONG CASE 1, 2, 3 AND 4.

CASE	***MTJ Config.***	***Inter-Die H.D.***		***H.W.***		***B.A.***	
		HD_{avg}	HD_{dev}	HW_{avg}	HW_{dev}	BA_{avg}	BA_{dev}
CASE 1	P	49.994%	3.387%	50.45%	5.65%	50.45%	5.01%
	AP	49.440%	3.385%	50.46%	5.64%	50.46%	5.03%
CASE 2	P	49.970%	1.573%	50.03%	1.49%	50.03%	5.15%
	AP	49.972%	1.572%	50.03%	1.51%	50.03%	5.14%
CASE 3	Rev.Combo (P+AP)	49.989%	3.244%	50.65%	5.46%	50.65%	5.01%
CASE 4	Rev.Combo (P+AP)	49.996%	1.573%	50.12%	1.40%	50.12%	5.02%

3.6.2 PUF Reliability (BER) study for all design schemes (Cases 1, 2, 3, 4)

Uniqueness study has shown good results for all four cases, especially for Case 2, 3 and 4. It is then necessary to investigate whether, while combined with Reliability, one or more Cases performs better than others, in order to assess whether there is a unique winner, or at least a few of them ex aequo.

It is important to note that for Uniqueness evaluation no errors were introduced in the sensing schemes: in other words, to evaluate Uniqueness, Bit-Error Rate was kept to zero. A way to see this principle is the superposition effect. Since Uniqueness and BER are two different parameters used to evaluate the PUF Quality, they cannot interfere with each other, thus while evaluating the former the latter needs to be zeroed and viceversa.

The simulation framework built up to evaluate Uniqueness has been re-adapted to evaluate Reliability. Both uniqueness and reliability, in fact, share the same main idea to generate the PUF signature (just slightly different sets of input and different input values are used).

Now, then, errors are introduced back in the system, which behaves less close to the ideal case. For each of the test cases, in fact, the simulator allows to select a specific one (thus, a specific memory layout). Then, letting the error introduced by the sense amplifier's minimum detectable differential current vary in a range of $(0 \dots 5)\mu A$, the PUF Signature is generated and the associated Bit-Error Rate (BER), for each $\Delta I_{sens,min}$, is calculated. Also, a zoom-in of the sub-range of $(0 \dots 50)nA$ is provided. Additionally, for Cases 1 and 2, of course, the analysis is again repeated twice: for both parallel (P) and anti-parallel (AP) configurations (but contrarily to Uniqueness, P and AP configurations are now compared within the same chart).

3.6.2.1 Case 1 study

CASE 1, P vs AP configuration: for Case 1 we observe that while the sense amplifier's minimum detectable current difference increases, the BER does not increase at a a very high rate: the dependance starts to be linear and then flattens to almost a constant. The antiparallel configuration offers a worse quality in terms of BER: since the AP resistance is higher than the P one, for the same reading voltage lower currents are generated and their difference is much lower in terms of intensity when compared to the P case. Then, the sense amplifier starts to encounter troubles into detecting in a reliable manner the PUF bit and the BER increases faster. The parallel case allows for a much low BER; to achieve a good Reliability, the parallel configuration is the best choice when it comes to pre-programming all the MTJs prior to the PUF Signature computation. The results are depicted in Figure 30. In the zoom-in of the chart on the right, we can see that for the same BER value the parallel states allow to relax a lot the specification of the sense amplifier. For instance, setting to BER to 0.5% requires roughly a $\Delta I_{sens,min} = 12.5nA$ for the AP case which becomes $\Delta I_{sens,min} = 25nA$ circa for the P case. In other words, moving from AP to P doubles the performance in terms of BER for the same quality of the sense amplifier.

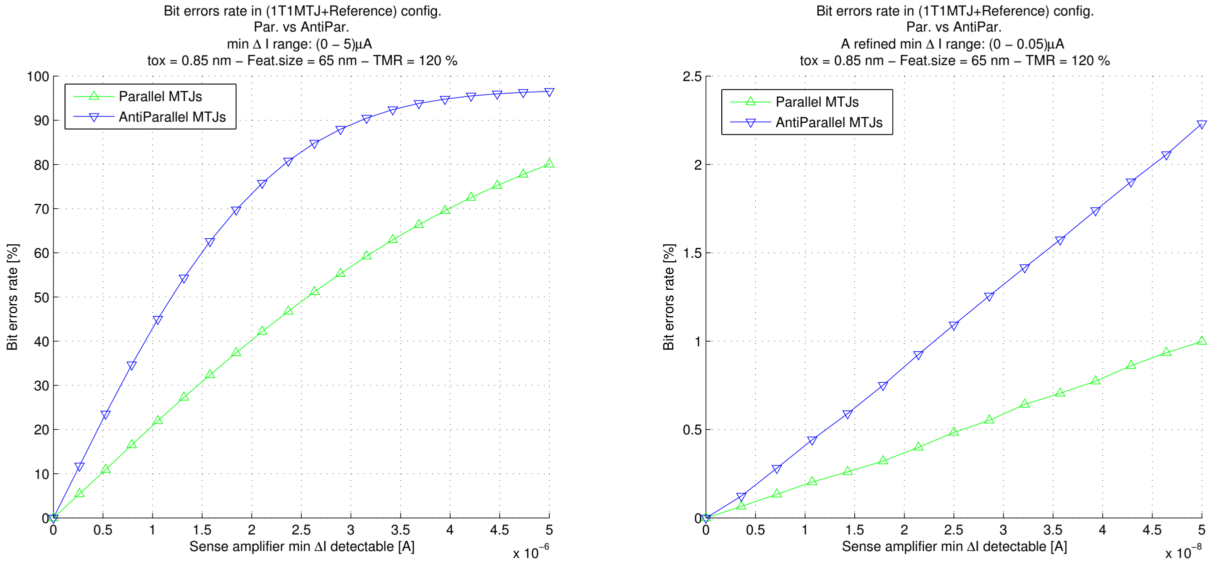


Figure 30. Bit-Error Rate for Case 1, comparing parallel (P) vs antiparallel (AP) case. The former is the winner here.

3.6.2.2 Case 2 study

CASE 2, P vs AP configuration: with respect to Case 1, the trend is practically identical – the only difference lies when focusing on the AP configuration: in Case 2, it saturates to 100% (completely unreliable signature) slightly faster when compared to Case 1. For the P configuration, instead, the Reliability does not change much. Figure 31 shows the BER analysis for Case 2.

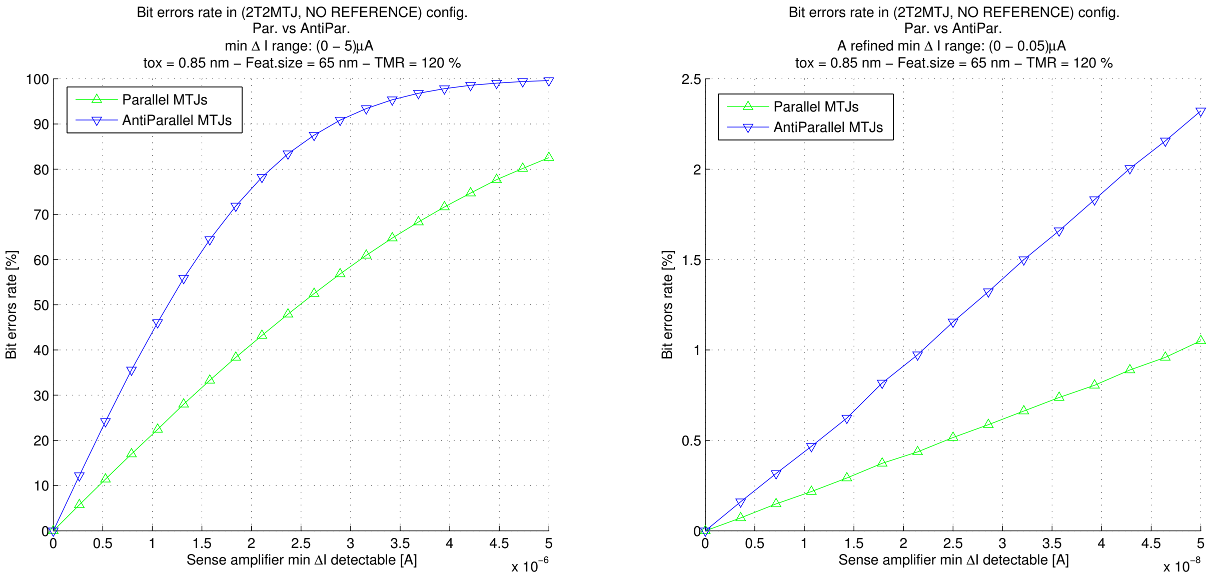


Figure 31. Bit-Error Rate for Case 2, comparing parallel (P) vs antiparallel (AP) case. The former is the winner here.

3.6.2.3 Case 3 study

CASE 3: since Case 3 (as well as Case 4) includes a reversed-combo approach, one BER curve is computed against the sense amplifier minimum detectable differential current. The dependance looks linear over all the range, but, with respect to Case 1 and 2, Case 3 guarantees a smaller Bit-Error Rate when compared to the AP configuration of both Case 1 and Case 2. Overall, the BER levels are kept equal to the ones of Case 1 and Case 2 in P configuration, even if the reversed-combo approach used in Case 3, as well as in Case 4, includes the half of the MTJs programmed in an AP way. Results of BER for Case 3 are depicted in Figure 32.

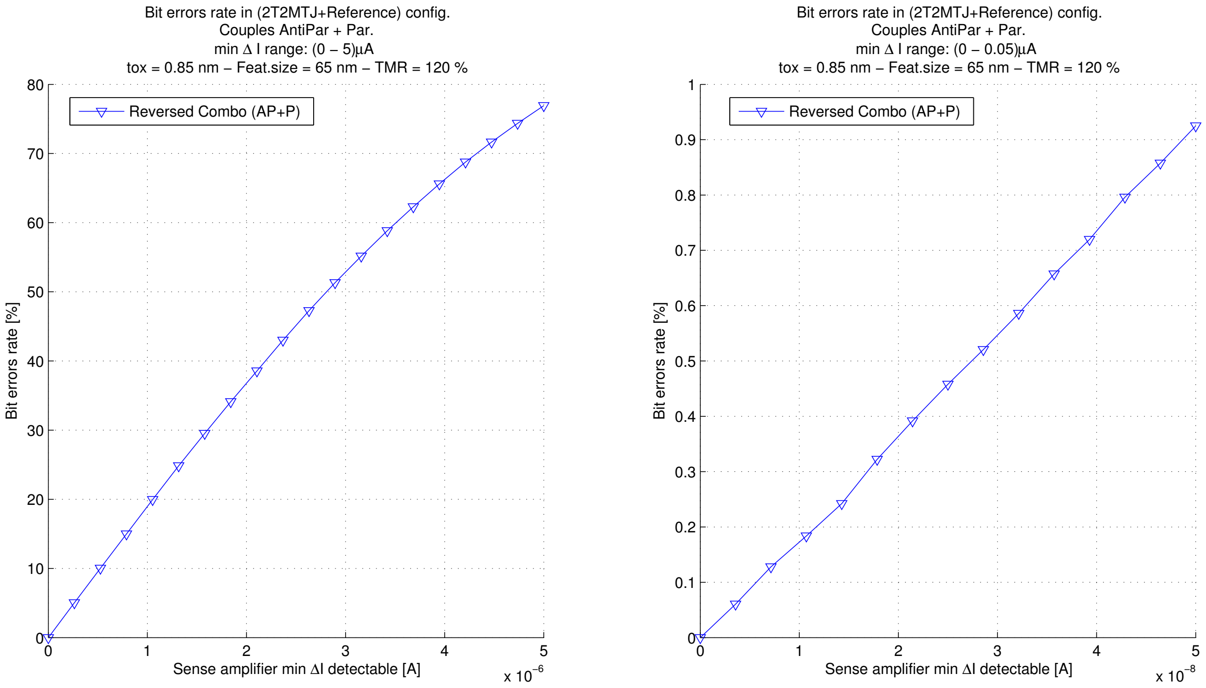


Figure 32. Bit-Error Rate for Case 3, which includes a reversed-combo (P+AP) programming and the use of reference cells.

3.6.2.4 Case 4 study

CASE 4: finally, Case 4 offers BER levels comparable to Case 3. In fact, no big differences are observable with respect to the previous case, only a slight improvement is present. Results are observable in Figure 33.

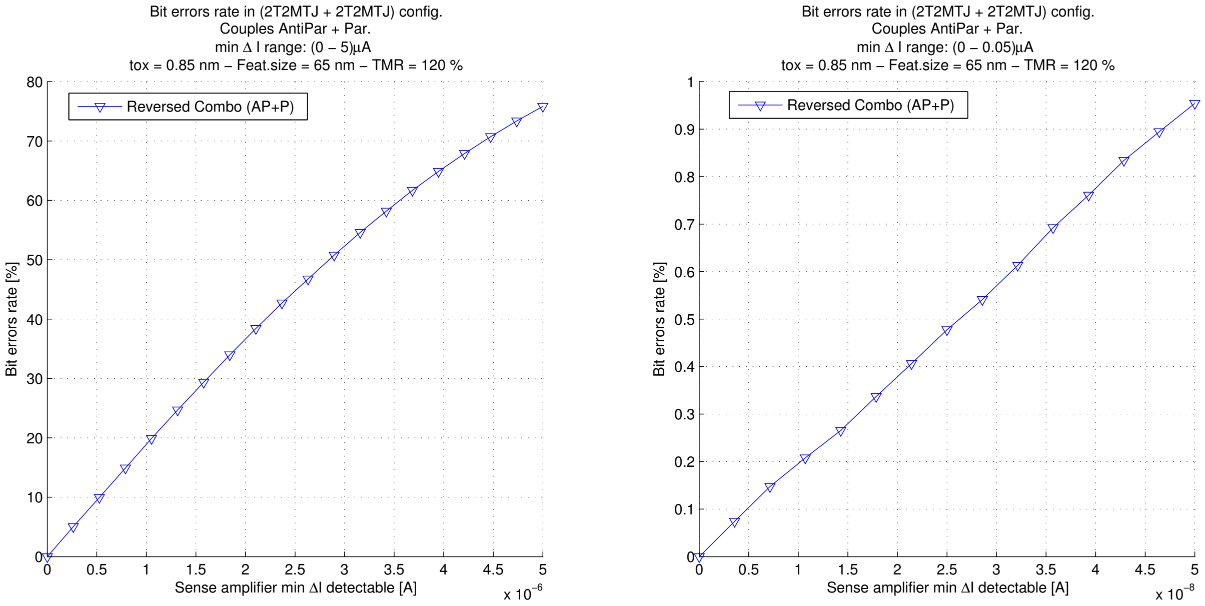


Figure 33. Bit-Error Rate for Case 4, which includes a reversed-combo (P+AP) programming and no reference cells.

3.6.2.5 Comparative study among Cases

In order to depict an overview on BER, Cases 1, 2, 3 and 4 need to be looked at simultaneously. Figure 34 offers then a comparison among cases. The right side of the Figure includes a bar chart for a sense amplifier error set to $50nA$. This error is still very high, since it causes the BER to be in the order of $1 \dots 2\%$. Good memory systems need to offer BER far below the 1% threshold, thus when it comes to carefully comparing the four different Cases a $BER = 10nA$ is also used in order to appreciate better BER differences among Cases. In this scenario, depicted in the bar chart on the left in Figure 34, the benefits of Case 3 and Case 4 become more evident: not only even though using half of the cells in AP state the BER is far less than the one of Case

1 and 2 in AP state, but at the same time the value of BER is overall kept lower even when compared to Case 1 and Case 2 in P state as well. Moving from Case 3 to Case 4, furthermore, the BER slightly improves.

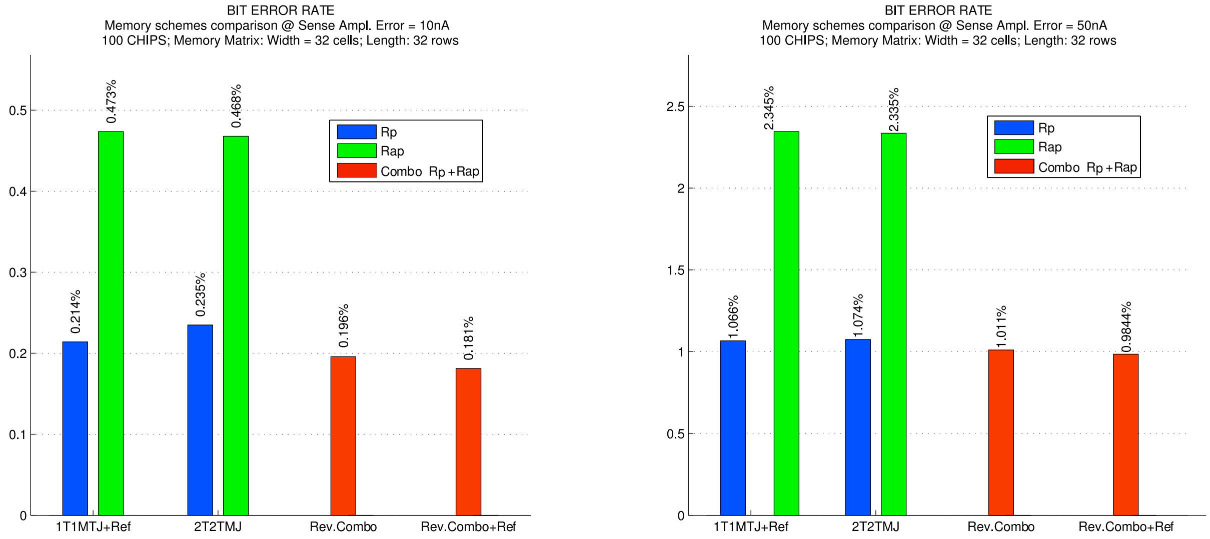


Figure 34. BER comparison at sense amplifier error of $10nA$ and $50nA$. Smaller detectable differential currents lead to more expensive and more precise sense amplifiers which reduce the BER. Also, at smaller currents, the BER differences among Cases is more appreciable.

3.6.2.6 Influence of variation of the reading voltage over BER

As already stated in Subsection 3.5.2, in order to complete the analysis over the BER, the parameter V_{MTJ} (the reading voltage of the cells) needs to be taken into account too. For each of the four Cases, then, at three different MTJ voltages ($V_{MTJ} = \{90mV; 100mV; 110mV\}$, the BER is again computed, keeping now the $\Delta I_{sens,min} = 10nA$ of the differential amplifiers. Table VI shows the obtained results. When the reading voltage increases, the R_{AP} values decrease as well as the R_P ones (recalling that the dependance of the MTJ's associated resistances as function of the reading voltages is inversely proportional). Then, the current flowing through the MTJ devices, given by the ratio between the voltage and the resistance, rises. With higher currents, finally, we expect a smaller BER, which is confirmed by the simulation results.

We can then conclude than not only the BER reduces while moving from Case 1 towards 4, but simultaneously becomes better when the applied voltage is increased. The upper boundary of this benefit is given by the fact that we cannot increase a lot the applied voltage, otherwise it will become too high to trigger a current across the MTJ devices that will cause the switching (unwanted changes of states of the MTJ, thus an undesired writing operation over the memory).

TABLE VI
BER COMPARISON WHEN THE APPLIED READING VOLTAGE VARIES.

BER with $\Delta I_{sens,min} = 10nA$				
CASE	***MTJ Config.***	$\mathbf{V_{MTJ,1} = 90mV}$	$\mathbf{V_{MTJ,2} = 100mV}$	$\mathbf{V_{MTJ,3} = 110mV}$
CASE 1	P	0.2339%	0.2140%	0.1866%
	AP	0.5038%	0.4735%	0.3750%
CASE 2	P	0.2441%	0.2349%	0.2051%
	AP	0.5088%	0.4678%	0.4326%
CASE 3	Rev.Combo (P+AP)	0.2158%	0.1958%	0.1816%
CASE 4	Rev.Combo (P+AP)	0.2031%	0.1812%	0.1753%

3.6.2.7 Influence of reading voltage over the MTJs' resistance

The results shown in Subsection 3.6.2.6 suggest further analysis of the effect of the reading voltage V_{MTJ} over the population of MTJs. In order to highlight how R_P but especially R_{AP} is (inversely) proportional to it, we can magnify voltage variations and plot the distribution of R_P and R_{AP} at three different reading voltages, chosen to be: $V_{MTJ} = \{50; 100; 150\} mV$. As depicted in Figure 35, the antiparallel state population is highly dependent on the reading voltage variation. In particular, while the voltage rises, the antiparallel population starts to move towards the origin of the axis.

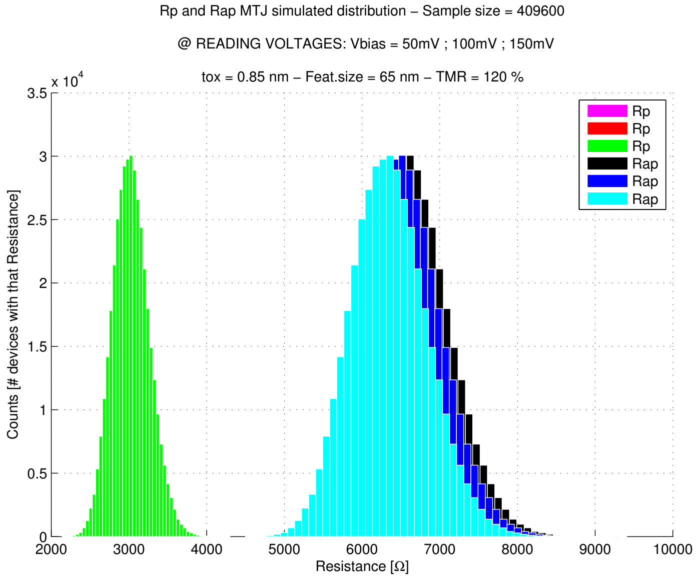


Figure 35. Influence of the nominal reading voltage over the shifting of the population of resistances of the MTJs.

We have already seen in Subsection 3.6.2.6 that an increase over V_{MTJ} benefits the BER which decreases. However, we can not increase V_{MTJ} a lot, since the following disadvantages would start to be present and dominate:

- The higher the reading voltage, the higher the power dissipation, since an higher current will flow through the MTJs. Although generating the PUF Signature is a procedure which is done only once, it is yet worth remarking that especially in read and write operations, when eventually combined with the PUF generation, using the STT-MRAM as a digital memory, this drawback is worth of consideration;
- Increasing too much V_{MTJ} would also lead, sooner or later, the antiparallel distribution to start getting in touch (overlapping with) the parallel one. From a digital standpoint, we would not be able to recognize parallel (binary 0) from antiparallel (binary 1), with an increase in the BER not only when it comes to PUF but, again, also in the memory applications;
- Finally, an high V_{MTJ} could cause the memory cell MTJ to switch its magnetization, implying an event of an unwanted write operation which alters the original memory content (a 0 becoming a 1 or viceversa).

3.7 Improving PUF Signature unclonability against software simulations

The signature generated so far, by means of the Physically Unclonable Function technique, can be seen as a **unique ID** associated to a certain chip. Not only when referring to STT-MRAM, but also talking about PUF in general, this consideration means that this binary identification number can be seen as a **Barcode**. For several applications, like associating a unique ID to a chip, this is more than enough.

However, in order to realize a good PUF, we need stricter requirements dictated from the field of Hardware Security. In particular, when we refer to PUF, we emphasize its unclonability feature. So far, we can conclude that the PUF investigated in the present thesis work is physically unclonable: in other words, we can be sure that no manufacturer can realize an exact hardware replica of a specific device, since process variation introduces a natural component of randomness that cannot be physically reproduced.

Unfortunately, for Hardware Security purposes, we need more. A PUF needs to be not only physically unclonable but also **unclonable under the software simulations standpoint**. A subject identifiable as design thief cannot realize a physical replica of a chip Signature but can still take a "snapshot" of it, realize a copy and simulate it in software.

It is thus required to add an **extra layer of randomness**, this time in control of the chip designer, which will bond with the already present process variation randomness. By doing so, with some additional hardware overhead, the chip designer can be able to realize a PUF Signature domain (the set of possible signatures generated by a chip) which is not anymore well defined (unique Signature) but can be augmented in a very high set of different possible signatures, and the one that will be picked up and chosen to become the final signature for each of the chips of the production lot is a decision controlled by the chip designer itself.

Looking back at the previous Section 3.6, Case 3 and 4 are the ground towards new possibilities within this approach. In fact, although they are originally designed to provide an extension over Case 1 and Case 2, they simultaneously represent, after all, the concept of **grouping memory cells into macro groups** before sensing currents and generating a PUF bit. This observation can thus be brought to an higher level, to be precise to a system design level, and further augmented, to provide a more robust framework to Hardware Security people.

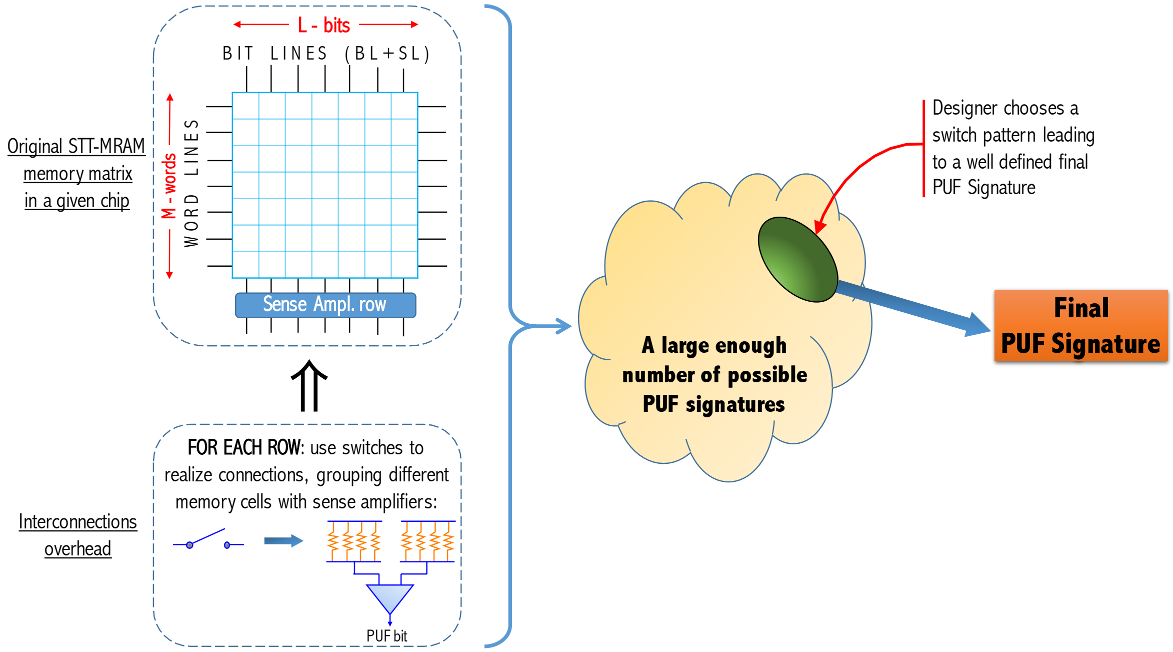

L - bits
BIT LINES (BL+SL)
M - words
WORD LINES
Original STT-MRAM memory matrix in a given chip
Sense Ampl. row
FOR EACH ROW: use switches to realize connections, grouping different memory cells with sense amplifiers:
Interconnections overhead
PUF bit
A large enough number of possible PUF signatures
Designer chooses a switch pattern leading to a well defined final PUF Signature
Final PUF Signature

Figure 36. Beyond Cases 1. . . 4: Increasing the space of Signatures of the PUF generated by means of STT-MRAM

The idea here is to realize even bigger groups of memory cells before sending their resulting current to an input of a sense amplifier. The organization of the grouping that we propose here can be done using **external switches**. Figure 36 provides a qualitative outlook at this idea. For each row, then, there exists a set of interconnections (MOS switches) that decides how to group cells together for each of the two inputs of each of the available sense amplifiers. Now **all the MTJs must be kept programmed exactly in the same way** (either parallel or antiparallel) and once again connecting different resistors, ideally equal in terms of resistance value, in parallel and sending these two groups of parallel elements to each sense amplifier, a PUF bit can be again generated. The initial essence of PUF generation by means of current mismatching is then still used but augmented by means of a generalized interconnection approach brought by the use of additional switches.

According to the position of each switch, then, for each row the designer can realize different possible interconnections, leading in turn to a huge space of PUF signatures (a large signatures domain). Once **the designer chooses a specific switches pattern**, this will lead to a well defined final PUF response for that chip, which will be provided as ID for identification / Hardware Security purposes.

With this additional approach, the PUF implemented exploiting the resistance of MTJs composing the STT-MRAM memory array is not only physically unclonable, but the technique is also harder to reproduce through simulations. The reason lies to the fact that now we do not get a trivial PUF Signature out of each chip applying one of the four Cases analyzed so far, but we add an extra layer of randomness that is chosen by the chip designer and effectively slows down (exponentially, as we will prove later) the time needed by an attacher to find out the right sequence of grouping of MTJ memory cells for each row used to generate, at the end, a specific PUF response for a given chip.

The goal of this section is then to provide an introductory study on how difficult we can make the life of the attacker with this extra level of security, deriving some basic equations which describe the size of the PUF space and proving that it is large enough. In order to make this idea viable for Hardware Security purposes, the **growth of the attacker complexity should be exponential** with respect to the original memory matrix size.

3.7.1 Strong vs Weak PUF: from Chip Signature to challenge-response pairs

The work of Herder et al. [33] classifies Physical Unclonable Functions in two different sets, basing the study on the relation between the challenges applied to the physical medium exploited as PUF and its related responses:

- **Weak PUF**: they offer a few challenge - response pairs. When a set of challenges is applied, the number of responses that the PUF element is able to provide is rather limited. For example, both the previous two papers [17] and [18] on which this thesis work is based, including the results and new methodologies proposed so far (with Case 1, Case 2, Case 3 and Case 4) belong to this category. Also some of the classical digital PUFs mentioned before, such as the work in [26], behave as a Weak PUF;
- **Strong PUF**: they allow to retrieve a rather huge set of possible challenges and responses. In other words, the number of possible challenges that are able to generate a different and associated response is very high. Analog PUFs typically have this behavior, and the

work in [24], briefly introduced in the Subsection 2.4.3, provides a complete example. The number of responses is very wide since the challenge set is an analog orientation angle (offering thus theoretically an infinite number of challenges).

An important point highlighted in [24] is the introduction of the concept of **(challenge-response) pairs** for the same chip. Using Strong PUFs, in other words, we do not refer anymore to a PUF Signature (unique and well defined for each chip) but rather to a set of signatures, called challenge-response pairs, which are associated to a given chip.

Rather than Signature application (Weak PUF), then, challenge-response pairs, proper of Strong PUFs, allow to obtain an high number of different possible signatures for a given chip, an aspect used in other applications such as **authentication**. As stated in [24] and again highlighted in Subsection 2.4.3, each chip is identified by a set of challenge-response pairs which are stored in a database of a secured server of the chip designer. When the designer wants to authenticate a chip as valid, a transaction begins: the server sends a particular challenge and awaits the answer of the remote chip consisting of a response. Whether this response is very close (ideally coincident) with the associated response present on the server then the chip is marked as authentic, otherwise it is rejected. A possible application would be, for example, enabling only some devices (belonging to a certain production date or production lot) to download and install a new revision of a firmware.

While a Weak PUF finds then typical application as a tool to store an authentication key (and used thus as key generation mechanism), Strong PUFs are employed to realize chip authentication at a far lower cost with respect to current specialized hardware required to implement authentication of electronic chips.

In a Weak PUF, the challenge-response pair (that in the worst case is only one) must be kept secret since, if revealed, it would let an attacker replicate (emulate) that specific device passing authentication procedures. On the other end, a Strong PUF, supporting an high number of challenge-response pairs, does not requires this design restriction and no additional authentication hardware is required (Weak PUFs still need this additional overhead if they are meant to be used for authentication purposes).

3.7.2 Augmenting STT-MRAM PUF as Strong PUF

It is possible to augment the properties and the potential of a Weak PUF into a Strong PUF. The idea proposed in this Section 3.7 is exactly an **extension of PUF Signature into a (challenge-response) pair**. According to Figure 36, in fact, the chip designer can change as many times as he likes the pattern of switches generating then different number of PUF Signatures, now more properly called **challenge-response pairs**.

A more rigorous representation of this scenario is then a slight variation of Figure 36, offered by Figure 37.

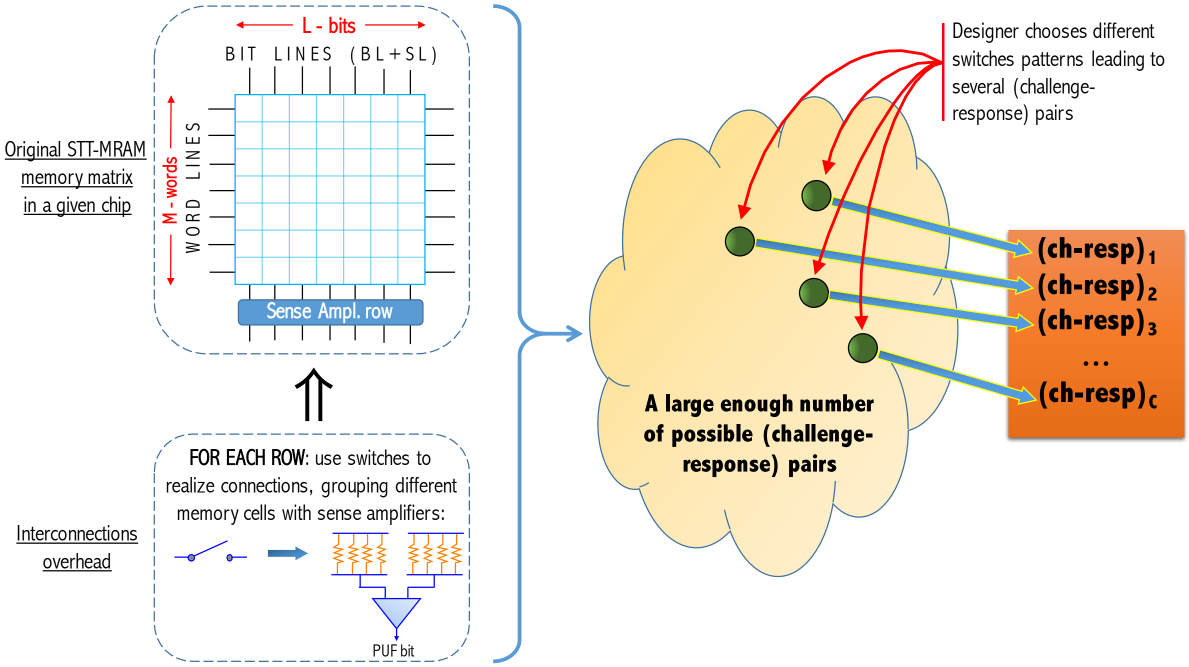

L - bits
BIT LINES (BL+SL)
M - words
WORD LINES
Original STT-MRAM memory matrix in a given chip
Sense Ampl. row
FOR EACH ROW: use switches to realize connections, grouping different memory cells with sense amplifiers:
PUF bit
Interconnections overhead
Designer chooses different switches patterns leading to several (challenge-response) pairs
A large enough number of possible (challenge-response) pairs
(ch-resp)$_1$
(ch-resp)$_2$
(ch-resp)$_3$
...
(ch-resp)$_C$

Figure 37. STT-MRAM as Strong PUF: obtaining a set of different (challenge-response) pairs for a given STT-MRAM chip.

3.7.3 Memory layout assumptions

The analysis that follows is an introductory mathematical verification on the increase of the PUF Signatures space, that will be called, from now on, the set of (challenge-response) pairs. From now on, let us shorten this term with **CRP**.

We would like to validate the assumption that an increase of the size of the memory words (L-bits wide) will lead to an exponentially increased size of the challenge-response pairs domain.

Let us assume that the elementary memory cell is of the simplest type 1T-1MTJ, composed then by a single access transistor driven by the world line WL in series with a MTJ. The memory matrix is composed then by these elementary cells and is $(M \times L)$ in size. Also, the memory chip comes with a row of sense amplifiers whose number is half the size of the memory words L – the assumption is related to the fact that each sense amplifier comes with two inputs, I_+ and I_-, and each memory cells offers an output current during read phases that can be connected to a single input of a sense amplifier. With L bits per word we normally have then ${}^{L}/_{2}$ available sense amplifiers. The scenario is depicted in Figure 38.

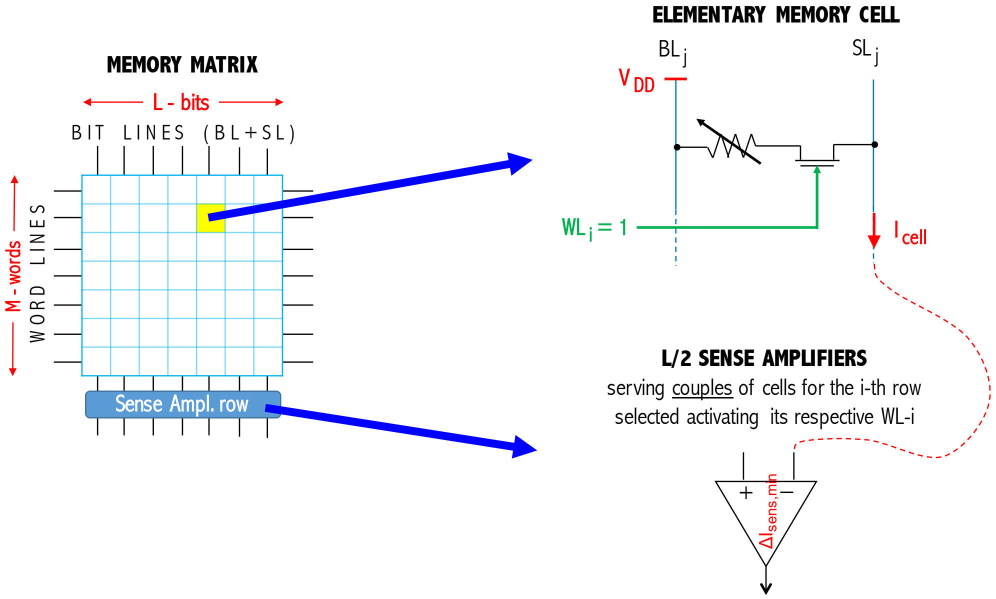


Figure 38. Memory layout assumptions

3.7.4 Mathematical formulation of the challenges space

We first would like to determine how many different combinations of grouping the memory cells can be achieved using the switches and connecting them to the available sense amplifiers. We define then the **space of the challenges** as the number of total different challenges that it is possible to apply to our PUF STT-MRAM, and we call this space Ω. We will find an equation for Ω step by step, starting from a simplified memory configuration and adding more constrains one step at a time.

SINGLE ROW, SINGLE AMPLIFIER

- Assume to call A and B the two groups of elements (MTJs in parallel) connected to each of the inputs of the sense amplifier. Assume not to consider that the number of elements in A must be equal to the number of elements in B. Having a memory word long L, we have then 2^L possible combinations of elements in A as well as in B and, being the two groups independent, we get a number of total combinations equal to $2^L \times 2^L = 2^{2L}$. A simplified interpretation of this scenario is offered by Figure 39.

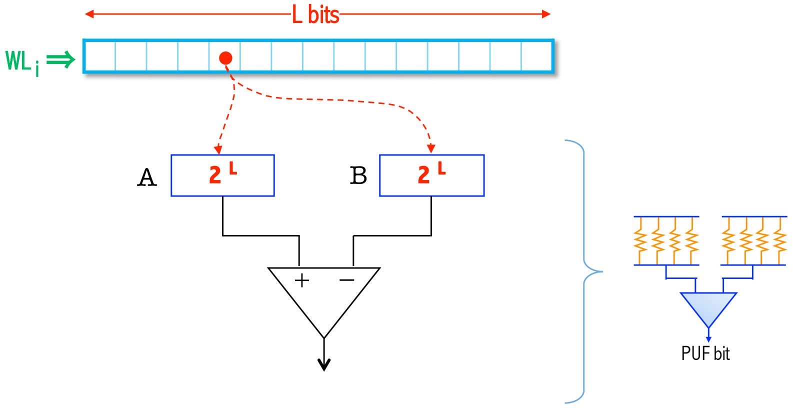


Figure 39. Preliminary thoughts on possible combinations of grouping memory cells (seen as MTJs' associated resistance) in parallel

- Now add the following constraints: A and B must have the same number of elements K (we want a balanced structure every time we realize a connection between a set of memory cells and an input of the sense amplifier) and the very same group cannot be connected any longer to two groups simultaneously. For example, for $K = 3$: $(265); (267)$ is OK, but $(265); (265)$ is NOT OK. For different values of $K = \{1 \ldots L\}$ we have then to consider binomial coefficients as derived in Table VII.

TABLE VII
REASONING ON POSSIBLE GROUPINGS OF THE AVAILABLE MEMORY CELLS

K	**No. Possible Groups in A**	**No. Possible Groups in B**
1	$L \equiv \binom{L}{1}$	$L - 1 \equiv \binom{L}{1} - 1$
2	$\binom{L}{2}$	$\binom{L}{2} - 1$
3	$\binom{L}{3}$	$\binom{L}{3} - 1$
4	$\binom{L}{4}$	$\binom{L}{4} - 1$
...	...	...
L	$\binom{L}{L} \equiv 1$	$\binom{L}{L} - 1 \equiv 0$

Then, the total number of different possible challenges Ω is given by multiplying, for each K, the number of possible groups in A and B, then adding all of them (for $K = \{1 \ldots L\}$):

$$\Omega = L \cdot (L-1) + \binom{L}{2} \cdot \left[\binom{L}{2} - 1\right] + \binom{L}{3} \cdot \left[\binom{L}{3} - 1\right] + \ldots + \binom{L}{L} \cdot \left[\binom{L}{L} - 1\right] \quad (3.6)$$

We can now get rid of the (-1) terms recalling the property in Equation 3.7 (here n represents a generic variable):

$$\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{n} = 2^n \tag{3.7}$$

Leading to a more compact result with respect to Equation 3.6, as shown in Equation 3.8:

$$\Omega = \binom{L}{1} \cdot \binom{L}{1} + \binom{L}{2} \cdot \binom{L}{2} + \binom{L}{3} \cdot \binom{L}{3} + \ldots + \binom{L}{L} \cdot \binom{n}{n} - 2^L \tag{3.8}$$

Equation 3.8 could be further simplified using the properties of binomial coefficients, however, since this is just the beginning of our analysis, we can proceed further on (as shown later, the final equation to consider and evaluate is not the one proposed in Equation 3.8).

- Since the sense amplifier is a symmetrical structure, let us consider now $|I_{diff}| \equiv I_{+} - I_{-} \equiv I_{-} - I_{+}$. In other words, we want to treat cases like $(267);(289)$ and $(289);(267)$ as the same case. The new space of the PUF can be obtained dividing by 2 Equation 3.8, as shown in Equation 3.9.

$$\Omega = \frac{1}{2} \cdot \left[\binom{L}{1} \cdot \binom{L}{1} + \binom{L}{2} \cdot \binom{L}{2} + \binom{L}{3} \cdot \binom{L}{3} + \ldots + \binom{L}{L} \cdot \binom{L}{L} - 2^L \right] \tag{3.9}$$

- We can now add the restriction that a cell cannot be connected simultaneously to both inputs of the sense amplifier, otherwise a current partitioning would arise. This translates into a modification with respect to Table VII, shown in Table VIII.

TABLE VIII
REASONING ON POSSIBLE GROUPINGS OF THE AVAILABLE MEMORY CELLS WITH MORE RESTRICTIONS ON THE GROUPING RULES

K	**No. Possible Groups in A**	**No. Possible Groups in B**
1	$\binom{L-1}{1}$	$\binom{L-1}{1}$
2	$\binom{L-2}{2}$	$\binom{L-2}{2}$
3	$\binom{L-3}{3}$	$\binom{L-3}{3}$
...	...	...

The space of the challenges applicable to the PUF, Ω, is now again derived in Equation 3.10

$$\Omega = \frac{1}{2} \cdot \left[\binom{L-1}{1} \cdot \binom{L-1}{1} + \binom{L-2}{2} \cdot \binom{L-2}{2} + \binom{L-3}{3} \cdot \binom{L-3}{3} + \ldots \right] \quad (3.10)$$

The analysis conducted so far with one sense amplifier is just the starting point. To have a PUF signature big enough, in fact, we need to consider, for each memory word, more than one sense amplifier (since the number of PUF bits corresponds to the number of available sense amplifiers).

In order to have a response binary string big enough, we could keep using one sense amplifier, and changing at different time frames (in a sequential fashion) the switches connections, generating one response bit at a time, then iterating the procedure for a number of times equal to the desired size of PUF response of each row (word line). This approach allows to have few switches (one sense amplifier only is available, and we need two switches per cell, thus we only need $2 \times L$ switches, or MOSFETs, in total). However, it leads to a slow generation of the signature and needs to be implemented with a sort of sequential logic.

The second option, instead, consists on using more sense amplifiers with more switches. It is faster but leads to an higher number of switches. After all, according to our initial hypothesis, our memory matrix already comes with an array of ${}^{L}/_{2}$ sense amplifiers which can be used to generate, for each row, a binary response string of the same length.

SINGLE ROW, MULTIPLE AMPLIFIERS

Considering more amplifiers simply means, for each K, expanding Table VIII horizontally with more columns (thus more groups: $A, B, C, D, \ldots$). The binomial coefficients of the type $\binom{a}{b}$ will have a numerator a which decreases by an amount of K moving from a group to the next one.

The result is shown in Table IX, where for reasons related to the available space the column titles “No. Possible Groups in X” are simply replaced by “X”, where X is the name of each group.

TABLE IX

CONSIDERING MORE SENSE AMPLIFIERS FOR EACH WORD LINE

	Sense Amplifier 1		**Sense Amplifier 2**		$\dots$
K	**A**	**B**	**C**	**D**	$\dots$
1	$\binom{L}{1}$	$\binom{L-1}{1}$	$\binom{L-2}{1}$	$\binom{L-3}{1}$	$\dots$
2	$\binom{L}{2}$	$\binom{L-2}{2}$	$\binom{L-4}{2}$	$\binom{L-6}{2}$	$\dots$
3	$\binom{L}{3}$	$\binom{L-3}{3}$	$\binom{L-6}{3}$	$\binom{L-9}{3}$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

It is now finally time to find a compact expression for Ω, applying the same sum-of-products principle carried out so far. In particular, let us define the following quantities:

– **L**: number of elementary memory cells for each row;

– **K**: ranging from 1 to L, identifies the number of elements per group, and each group is a set of parallel resistors seen as a connection of multiple memory cells' MTJs programmed simultaneously either in parallel or in antiparallel;

– **S**: the number of available sense amplifier (typically we let $S \equiv^{L} /_{2}$).

The number of possible combinations of interconnections, leading to a space Ω of challenges, is expressed in a compact form by Equation 3.11, based on the preliminary derivations carried out in Table IX.

$$\Omega = \frac{1}{2} \cdot \sum_{K=1}^{L} \left[\prod_{i=0}^{2 \cdot S-1} \binom{L - K \cdot i}{K} \right] \tag{3.11}$$

Equation 3.11 has a problem that needs to be addressed: for some values of K, the numerator of the binomial coefficient might become negative. This is due to the fact that we are requesting groups too wide (K big) having only a relatively small number L of available cells per memory word. The equation is then valid until the condition in Equation 3.12 is met.

$$(N - K \cdot i) \geq K \Rightarrow K \leq \frac{N}{1+i} \tag{3.12}$$

Equation 3.12 imposes an important boundary on K, which should not exceed a certain amount function of L, which makes sense also from the logical and physical standpoints.

MULTIPLE ROWS (WHOLE MEMORY MATRIX), MULTIPLE AMPLIFIERS

Considering then simultaneously Equation 3.11 and Equation 3.12, we can multiply the result in Equation 3.11 by M to finally find Ω for the entire STT-MRAM matrix. The final result is stated in Equation 3.13.

$$\Omega = \frac{M}{2} \cdot \sum_{K=1}^{L} \left[\prod_{i=0}^{2 \cdot S-1} \binom{L - K \cdot i}{K} \right], K \leq \frac{N}{1+i} \tag{3.13}$$

Let us now consider Equation 3.13 and plot Ω as function of L to verify how the space of the challenges grows in relation to the width of the STT-MRAM memory chip. We expect the designer to be able to play among a huge number of possible challenges with a rather small increase of memory size, and this assumption is correctly validated in Figure 40. Here we have fixed the memory length $M = 32$ words. The plot has been realized in semilog scale (that is, taking the logarithm of every value on the y axis) in order to better appreciate the behavior of this dependance. In a linear plot, the increase is in fact way too steep.

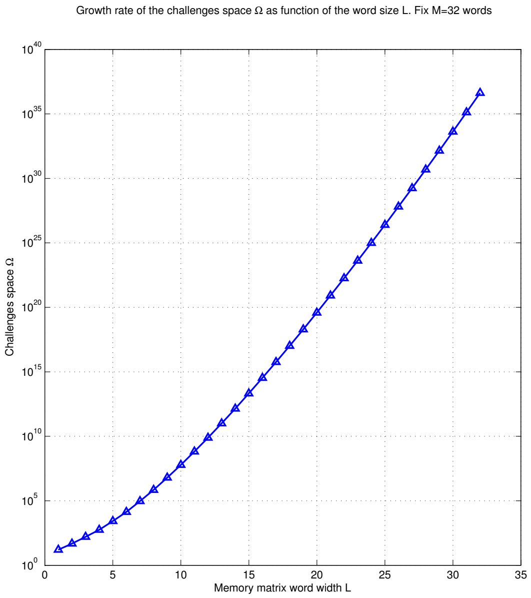


Figure 40. PUF challenges space Ω dependance on memory width L (semilog plot).

3.7.5 Mathematical formulation of the response space

The response of the PUF, also called in the previous Sections of this thesis binary response string R, is the product of two terms. The first one is equal to the number of sense amplifiers (since every output of a sense amplifier corresponds to one PUF bit). The second term is instead the number of rows M for the memory matrix. Also, we previously set the number of sense amplifiers to be equal to half the width of the memory matrix, that is: $L/2$. The **size of the PUF response**, called from now on Λ, then, can be expresses through Equation 3.14.

$$\Lambda = 2^{L/2} \times M \tag{3.14}$$

Although Equation 3.14 is rather simple to evaluate, we offer anyway in Figure 41 a plot of Λ against the growth of the memory width L. Once again, we set $M = 32$ to keep consistency with the previous analysis in Subsection 3.7.5. The dependance is parabolic.

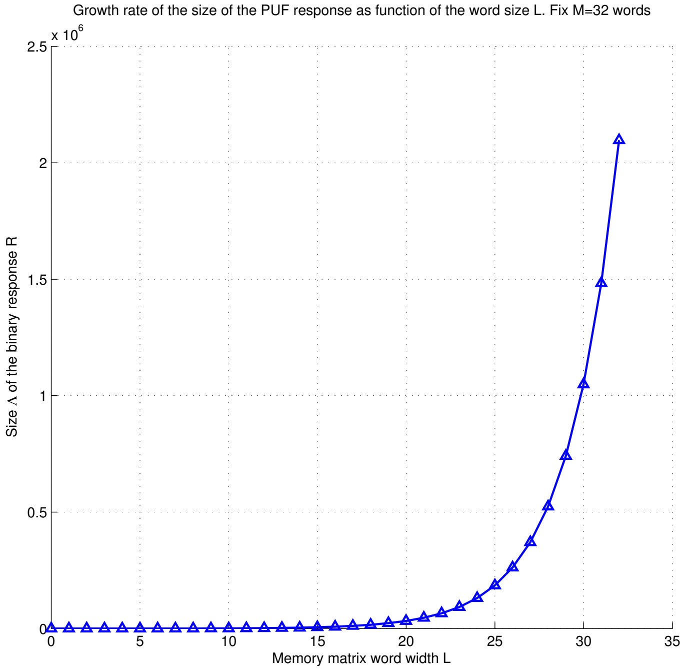


Figure 41. PUF response string size Λ dependance on memory width L.

3.7.6 Complexity for the attacker to defeat the Strong PUF

For Hardware Security applications, a good PUF should have the property of growing exponentially, as function of a certain variable N, in terms of **complexity from the clonability through software simulations** carried out by an attacher. In our case, N represents the words size of the memory matrix, indicated as L in the previous Sections.

The **complexity** Φ for the attacker is calculated keeping in mind that it is not dependent on the number of sense amplifiers but only on the memory matrix word size L. Looking back at Table IX, then, it is given by the product of the terms in A and B only (Since we need to consider only $SenseAmplifier1$), then summing them up for each $K = 1 \dots L$, as expressed by Equation 3.15.

$$\Phi = \sum_{K=1}^{L} \binom{L}{K} \cdot \binom{L-K}{K} \tag{3.15}$$

The term $M/2$ is not considered in front of the equation because, when it comes to complexity, we are more interested in a proportionality relation rather than an exact value. Furthermore, the quantity Φ must increase at an exponential rate. In order to verify this aspect, let us find a lower bound MIN and an upper bound MAX into which Φ is comprised:

- **MIN**: the lower bound consists, for each K, of $\binom{L}{K}$ possible elements connected to one input of the amplifier and 1 possible element connected to the other. We have then:

$$MIN = \binom{L}{1} \cdot 1 + \binom{L}{2} \cdot 1 + \binom{L}{3} \cdot 1 + \ldots + \binom{L}{L} \cdot 1 = 2^L \tag{3.16}$$

– **MAX**: the upper bound is exactly what already derived and represented in Figure 39:

$$MAX = 2^{2L} \tag{3.17}$$

We can then immediately see that:

$$MIN < \Phi < MAX \Rightarrow 2^L < \sum_{K=1}^{L} \binom{L}{K} \cdot \binom{L-K}{K} < 2^{2L} \tag{3.18}$$

Equation 3.18 shows that Φ is bounded by two exponentials, thus it follows an exponential behavior too (actually, the lower bound is enough to prove that Φ grows *at least* exponentially). Figure 42 shows the behavior of Φ as function of L, with the lower and the upper bounds MIN and MAX in a semilog scale (where linear proportionality expresses an exponential tendency).

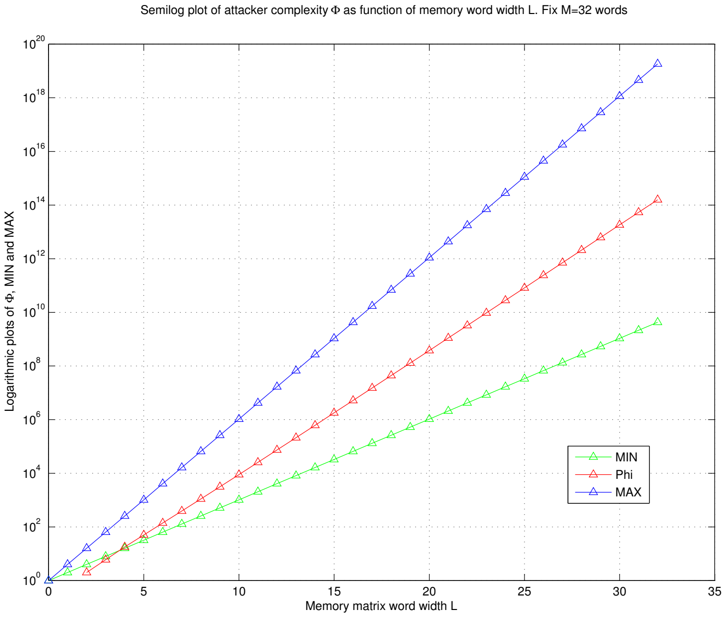


Figure 42. Attacker complexity Φ dependance on memory width L (semilog plot).

Using a semilog scale for the y axis we see a linear dependence of Φ with respect to L indicating and confirming then the assumption of an exponential dependance.

A final but extremely important remark: the complexity Φ for the attacker is different from the concept of **guessing a signature**: while, in fact, Φ is independent on the number of the sense amplifiers S, the probability of guessing correctly an hidden PUF signature is instead dependent on the number of the sense amplifiers. In fact, given that each output of the sense amplifier can be either the binary 0 or the binary 1, the attacker has a $50\% = 0.5$ probability of guessing the bit. Increasing the number of amplifiers to two, the probability reduces to: $0.5 \times 0.5 = 0.25$. For three, $0.5 \times 0.5 \times 0.5 = 0.125$, and so on.

3.7.7 A basic high-level digital approach to implement the switches

In order to realize a versatile connection of every possible memory cell to every possible input of the available set of sense amplifiers, an hardware overhead is necessary. Here a possible preliminary idea is proposed which includes the use of a Decoder for each memory cell. The designer is now able to select the challenge by choosing a word line to process plus feeding the decoder, for each STT-MRAM word bit, with a binary sequence that will realize the connection of each MTJ of each memory cell only to one among the available inputs of the sense amplifiers. The structure of this implementation idea is shown in Figure 43.

The **hardware cost** is dominated by the decoders, since we need one for each memory cell, together with L switches (MOSFETs). Being each memory word composed by L bits, this cost needs to be multiplied by L: we end up with L decoders and $L \times L = L^2$ switches.

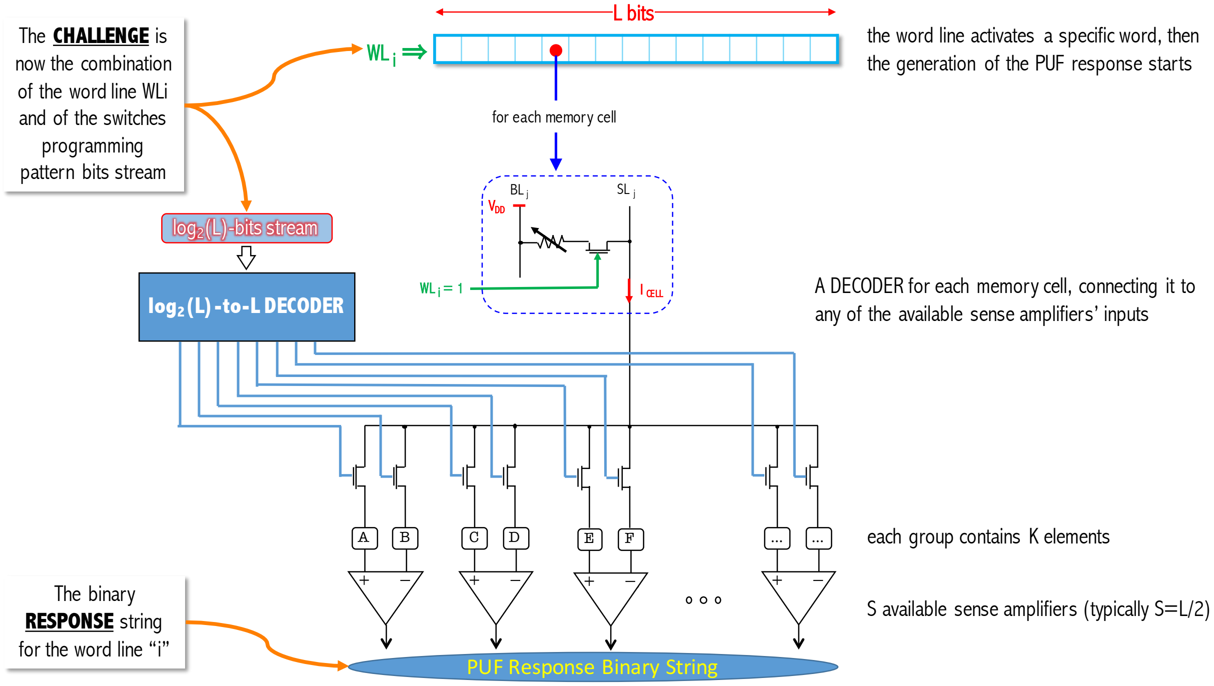

The CHALLENGE is now the combination of the word line WLi and of the switches programming pattern bits stream
L bits
WL i
the word line activates a specific word, then the generation of the PUF response starts
for each memory cell
BL j
SL j
VDD
WL j = 1
I CELL
log2(L)-bits stream
log2(L)-to-L DECODER
A DECODER for each memory cell, connecting it to any of the available sense amplifiers' inputs
A
B
C
D
E
F
...
...
+
−
each group contains K elements
S available sense amplifiers (typically S=L/2)
o o o
The binary RESPONSE string for the word line "i"
PUF Response Binary String

Figure 43. A possible high-level digital implementation of STT-MRAM as Strong PUF. A Decoder is needed for each memory cell.

Finally, it is important to remark that the need for the designer to feed now each row with a bit stream to program the decoders imposes the necessity to store, each time a row is processed, a sequence of bits long $L \times \log_2 L$. Since the PUF response is generated processing one word at a time (activating the respective WL_i) it might be possible to store this additional input pattern of the decoders in the unused rows (that is, any row except the one processed at that time to generate part of the PUF response).

This possible solution shows how dynamic and flexible our STT-MRAM can be, proving that it can be exploited at the same time both as memory and as Physically Unclonable Function, letting the two flavors coexist and support each other for Hardware Security applications, that include the storage of additional information (which would otherwise cause an hardware overhead since an additional memory would be necessary).

Clearly, the solution proposed in Figure 43 is very preliminary, and according to the specific STT-MRAM chip used and the particular technology adopted, as well as the type of sensing scheme for the memory cells when it comes to using the MRAM as a proper digital memory, more or less hardware overhead can be necessary. Once STT-MRAM technology will become more unified, further efforts could lead to universally accepted hardware implementations to generate a PUF out of an STT-MRAM memory chip.

CHAPTER 4

CONCLUSIONS

4.1 Final conclusions and remarks

After introducing the main ideas and principles behind magnetic memory (especially STT-MRAM) and Physical Unclonable Function in Hardware Security, the two topics, apparently far away from each other, are brought together to present a set of ideas and simulations validating the use of magnetic memory to generate a unique signature for a chip within a production lot as well as challenge-response pairs for more proper Hardware Security applications such as chip authentication.

A population of STT-MRAM memory cells, based on MTJs, has been first of all studied from the statistical standpoint showing how the influence of process variation, simulated through software, over physical quantities of the MTJs can effectively lead to a dispersion over the parallel and antiparallel distribution of the two binary states of a magnetic memory cell in terms of resistance.

Then, selecting some specific manufacturing values, the magnetic memory has been used to implement a Physical Unclonable Function showing four possible design schemes variations. In all cases, the quality of the generated PUF Signature is very high, showing how an entire STT-MRAM chip can be used to generate a signature which is based on its microscopic properties affected by process variation. A deep analysis over Uniqueness and Bit-Error Rate has been carried out on the proposed schemes taking into account fluctuations of the supply voltage provided to the memory cells during read operation. Comparisons about cases show common trends and trade-offs, highlighting at the same time that the scheme called Case 4, with a reading voltage as high as possible (yet complain to power dissipation and unwanted writes requirements), leads to the best results even though bringing as requirement a memory size which is doubled.

A specific trend has been observed while playing with Cases 1, 2, 3 and 4: the possibility of grouping memory cells together to evaluate the PUF. This observation has been further developed and used to transform the STT-MRAM from a Weak PUF to a Strong PUF, for more proper Hardware Security related applications. A digital scheme based on switches and digital Decoders has been proposed showing how this hardware overhead is effectively able to increase the space of PUF challenges as well as increase exponentially the difficulty for the attacker to defeat through software simulations the security of physical unclonable functions provided by an STT-MRAM chip.

A secondary (but certainly not a minor) contribution of the present thesis work is the effort of offering a comprehensive theoretical baseline grouping together some of the most relevant background and ideas proposed by different papers concerning MRAM technology. When it comes about magnetic memory, in fact, both Field MRAM and STT-MRAM, at the moment a unified literature is still missing and there are still plenty of proposed digital implementations but very little common ground. In the effort of bringing MRAM integrated into digital systems effectively in the near future, this challenge should be addressed proposing standards and general rules on digital design for magnetic memories.

4.2 Future Work

The research project, a bridge between a study on STT-MRAM and Physically Unclonable Functions, opens to different possible future advancements and improvements on both topics. In particular, the following is a brief list according to each of the main areas touched by this thesis.

On MRAM as digital memory, joint efforts should be done by MRAM researchers to propose standard and more unified schemes to convince current electronic designers to adopt MRAM as innovative memory in their digital systems, offering protocols and more uniform outcomes that enable, for example, to translate early prototypes (like cache implementations proposed by Toshiba, or processing-in-memory) into fully functional and easy-to-adopt innovative digital blocks. The potential of MRAM and its advantages over several current digital memory technologies has been widely demonstrated and translating the potential into real engineered products would enormously contribute to overcome current limitations due to integrated circuit scale down, bringing more speed and non-volatile memory with less power consumption.

When MRAM is studied as variable binary resistor, more accurate and comprehensive models and equations should be devised, taking into account, at a macroscopic scale, not only the dependance of spin-torque transfer MTJs resistance on physical dimensions and applied voltages but also the dependance over temperature. From this standpoint, a wider model would allow to carry out more realistic stability analysis over the memory cells, crucial for applications like Physical Unclonable Function.

The field of the use of MRAM as digital PUF is still very new and experimental – there is still a lot to be done and this thesis work presents possible schemes and methodologies that can be further expanded. Different kind of grouping could be devised starting from the ones proposed by this thesis work, studying tradeoffs between security level of the PUF, hardware overhead and speed into generating in real-time a challenge-response pair.

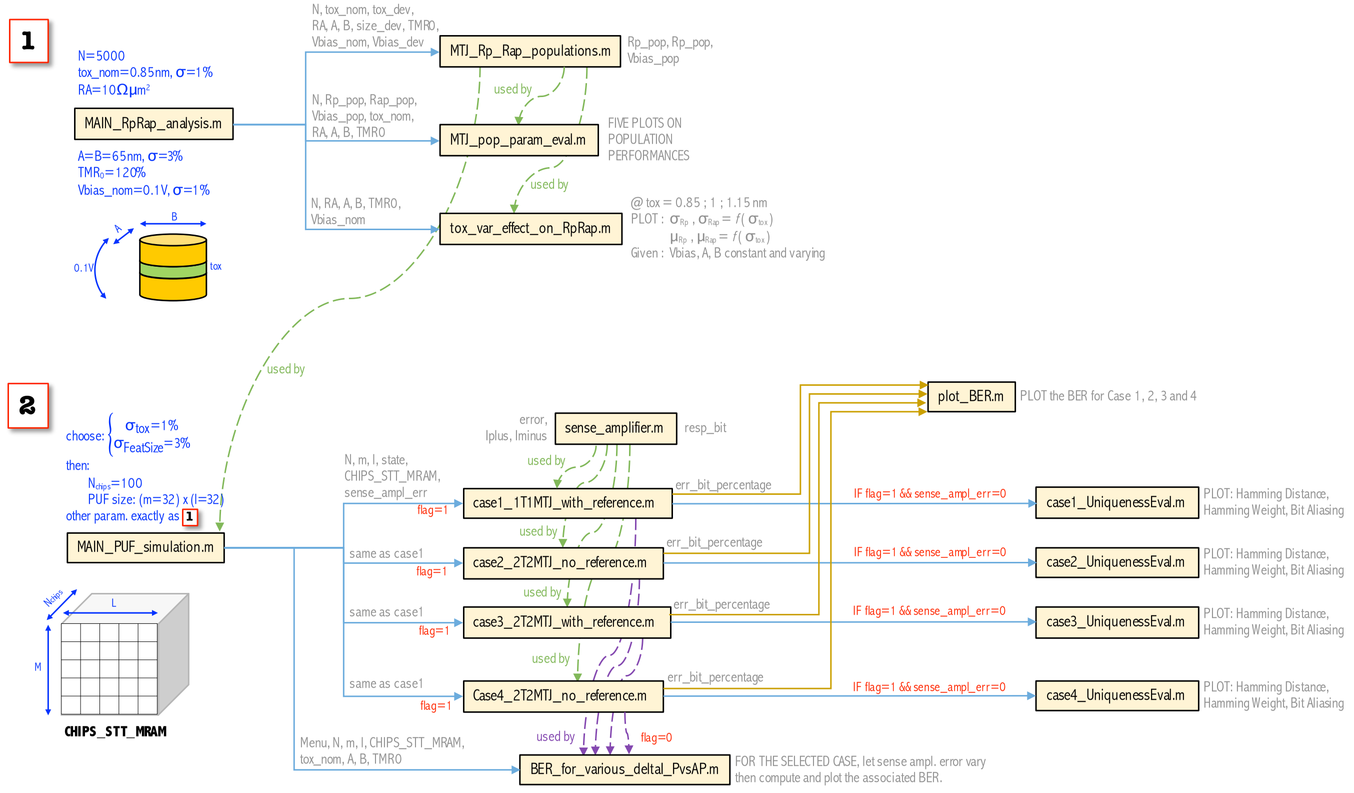
Once some common ground on MRAM as PUF is established, this implementation could be tested on current hardware security related applications such as authentication, verifying how a PUF based on MRAM could simplify the architecture of these schemes drastically reducing the use of specialized extra hardware to implement a security layer over a digital chip.

Finally, to overcome the challenge of reduction of the Bit-Error Rate and realizing a PUF which can be considered very stable with respect to fluctuations of physical quantities, a final stage, called Fuzzy Extractor, can be interleaved between the raw PUF response and the final output. Practically any paper and research about PUFs proposes this additional block, whose main function is to help reducing the intra-chip variation due to noise (temperature, voltage, etc.). Given a noisy input (the raw PUF response), the Fuzzy Extractor is able to produce a noise-tolerant final output string, together with some helper data. Thanks to the Fuzzy Extractor, if its input (the PUF raw response) changes slightly over time due to noise, the device is still able to detect and tolerate, within a certain boundary, the noisy component and generate a stable (the same) output. The helper data is used to assist into reproducing the same stable response when a noisy (thus slightly different) input comes to the Fuzzy Extractor. An extensive and remarkable theoretical background on Fuzzy Extractors is offered by [34]. ■

APPENDICES

Appendix A

SOFTWARE SIMULATIONS MAIN FRAMEWORK



Appendix B

SOME RELEVANT CODE OF THE SIMULATION FRAMEWORK

Four among the most representative functions and files constituting the simulation framework built to implement a PUF out of STT-MRAM chips are quoted in this Appendix B.

FIRST SCRIPT: how the population of MTJs is generated, starting from the relevant equations describing the resistance of the parallel and antiparallel state and augmenting them with a Monte Carlo approach to simulate the process variation of the physical parameters and applied voltage to the MTJ stacks.

Listing B.1. Generating thenpopulations R_p and R_{ap}

```
function [ Rp_pop, Rap_pop, Vbias_pop ] = MTJ_Rp_Rap_populations(N,tox_nom,...
                             tox_dev,RA,A,B,size_dev,TMR0,Vbias_nom,Vbias_dev)
    %
    % MgO Barrier Tunnel Resistance Model
    % ALL SIZES MUST BE FED IN METERS at function call!!!
    %
    % ------ INPUTS DESCRIPTION: ----------------|------- Typical Value ---|
    % N: population size to be created           |                         |
    % tox_nom: Nominal oxide thickness           |           0.85nm        |
    % tox_dev: Std dev of tox, % nom value       |             2%          |
    % RA: R*A product                            |         10ohm*um^2      |
    % A: MTJ lenght                              |            65nm         |
    % B: MTJ width                               |            65nm         |
    % size_dev: A and B dev                      |             5%          |
    % TMR0: Nominal TMR at 0V bias               |            120%         |
    % Vbias_nom: Nominal MTJ read bias voltage   |            0.1V         |
    % Vbias_dev: Std dev of MTJ reading bias     |             5%          |
    % -------------------------------------------|-------------------------|


phi = 0.4;
Vh  = 0.5;
e=1.60*(10^-19);
m=9.1*(10^-31);
h=1.0545*(10^-34);


% --------------------------- FUNCTION BODY ---------------------------
F = 3322.53/RA;
% Create populations of Oxide thicknesses
tox_pop  = tox_nom  + randn(N,1)*tox_dev;

% Create populations of lenght A
A_pop = A + randn(N,1)*size_dev;

% Create populations of width B
B_pop = B + randn(N,1)*size_dev;

% Create the population of MTJs areas
Area_pop  = A_pop.*B_pop*(pi/4);

%Area_dev = (A*B)*sqrt((size_dev/A)^2+(size_dev/B)^2);
%Area_pop = A*B + randn(N,1)*Area_dev;

% Generates Rp distribution at 0 bias
Rp_pop = (tox_pop*1.0e10./(F*(phi^0.5)*Area_pop*1.0e12)).* exp(1.025*...
          tox_pop*1.0e10*(phi^0.5));

% Generates Vbias distribution
Vbias_pop  = Vbias_nom + randn(N,1) * Vbias_dev;

% Generates Rp population dependent now on bias voltage:
Num=(tox_pop.^2)*(e^2)*m;
Den=4*(h^2)*phi;
Rp_pop=Rp_pop./(1+((Num/Den).*(Vbias_pop.^2)));

% Generates TMR distribution
TMR_real_pop = (TMR0/100)./(1 + (Vbias_pop.^2) / (Vh^2));

```

```
    % Generates Rap distribution
    Rap_pop = Rp_pop .* (1 + TMR_real_pop);
    % ---------------------------------------------------------------------
end
```

SECOND SCRIPT: the main among the scripts written to assess the statistical distributions of the populations of MTJs, verifying how the parallel and antiparallel resistances are spread. A second script, not mentioned here, but equally important, brings some of the lines of codes of Listing B.2 to conduct the analysis of the influence of the variation of the oxide thickness (and its standard deviation) over the population of MTJs.

Listing B.2. Statistical evaluation of the populations R_p and R_{ap}

```
function [ ] = MTJ_pop_param_eval(N,Rp_pop,Rap_pop,Vbias_pop,tox_nom,RA,A,B,TMR0)

    phi = 0.4;
    Vh  = 0.5;
    e   = 1.60*(10^-19);
    m   = 9.1*(10^-31);
    h   = 1.0545*(10^-34);

    F = 3322.53/RA;

    % ----- PLOT [Rp,Rap]=f(Vbias) (keep tox and feature size = const) ----
    % 1 - Calculate Rp at zero bias:      %ELLIPSE!!
    Rp0 = (tox_nom*1.0e10/(F*(phi^0.5)*A*B*(pi/4)*1.0e12)).* exp(1.025*...
           tox_nom*1.0e10*(phi^0.5));
    % 2 - Generates a vector of possible Vbiases:
    Vbias=-0.4:0.01:0.4;
    % 3 - Evaluate Rp at real bias voltages
    Rp=Rp0./(1+((((tox_nom^2)*(e^2)*m)/(4*(h^2)*phi)).*(Vbias.^2)));
    % 4 - Calculate the real TMR according to the real Vbias:
    TMR_real = (TMR0/100)./(1 + (Vbias.^2) / (Vh^2));
    % 5 - From Rp and TMR_real computes finally Rap
    Rap = Rp.*(1 + TMR_real);
    % finally plots Rp and Rap against Vbias
```

```
fig1=figure(1);
title(sprintf('Rp and Rap vs Bias Voltage of MTJ, with no variations...
      in device geometry\ntox = %.2f nm - Feat.size = %.0f nm - TMR ...
      = %i %%',tox_nom*10^9,A*10^9,TMR0));
xlabel('Vbias [mV]');
ylabel('Rp and Rap [\Omega]');
hold on;
plot(Vbias, Rp, 'b-', 'LineWidth',2);
plot(Vbias, Rap, 'g-', 'LineWidth',2);
legend('Rp=f(V_{MTJ})','Rap=f(V_{MTJ})');
grid on;
axis([-0.6,0.6,2000,7000]);
hold off;
saveas(fig1,'1_RpRap_function_Vbias.eps','epsc');
% ---------------------------------------------------------------------



% ---- COMPUTE DISTRIBUTION PARAMETERS OVER Rp AND Rap POPULATIONS ----
fprintf('     Rp ideal, at zero bias = %f Ohm\n',Rp0);
fprintf('     Max(Rp) = %f Ohm\n     Min(Rap) = %f Ohm\n',max(Rp_pop),...
        min(Rap_pop));
if max(Rp_pop) > min(Rap_pop)
    fprintf('     Rp and Rap overlap of %f Ohm!\n',max(Rp_pop)-min(Rap_pop));
else
    fprintf('     Rp and Rap DO NOT overlap\n');
end

fprintf('     Rp:  mu = %i Ohm, s = %i Ohm, Distrib.s = %i%%\n',...
        uint32(mean(Rp_pop)),uint32(std(Rp_pop,1)),uint32((std(Rp_pop,1)/...
        mean(Rp_pop))*100));
fprintf('     Rap: mu = %i Ohm, s = %i Ohm, Distrib.s = %i%%\n',...
        uint32(mean(Rap_pop)),uint32(std(Rap_pop,1)),uint32((std(Rap_pop,...
        1)/mean(Rap_pop))*100));
fprintf('     By how many s Rp and Rap means are distant: %i\n',...
        uint32((mean(Rap_pop)-mean(Rp_pop))/std(Rp_pop,1)));
% ---------------------------------------------------------------------



```

Appendix B (Continued)

```
% ------ PLOTS HISTOGRAM and GAUSSIAN APPROX. OF THE POPULATIONS ------
fig2=figure(2);
hold on;
title(sprintf('Rp and Rap MTJ simulated distribution - Sample size = %i\n...
      tox = %.2f nm - Feat.size = %.0f nm - TMR = %i %%',N,tox_nom*10^9,...
      A*10^9,TMR0));
xlabel('Resistance [\Omega]');
ylabel('Counts [# devices with that Resistance]');
%hist(Rp_pop,50);
hist(Rp_pop,uint32(std(Rp_pop,1))/7);
hold on
%hist(Rap_pop,50);
hist(Rap_pop,uint32(std(Rap_pop,1))/7);
x = findobj(gca,'Type','patch');
set(x(1),'Facecolor','c','EdgeColor','w'); %Rap
set(x(2),'Facecolor','g','EdgeColor','w'); %Rp
legend('Rp distribution','Rap distribution');


X1 = linspace(mean(Rp_pop)-4*std(Rp_pop,1),mean(Rp_pop)+4*std(Rp_pop,1),1000);
X2 = linspace(mean(Rap_pop)-4*std(Rap_pop,1),mean(Rap_pop)+4*std(Rap_pop,...
     1),1000);
Y1 = normpdf(X1,mean(Rp_pop),std(Rp_pop)).*(N*50);
Y2 = normpdf(X2,mean(Rap_pop),std(Rap_pop,1)).*(N*50);
plot(X1,Y1,'k','LineWidth',1);
plot(X2,Y2,'k','LineWidth',1);
grid on;
hold off;
saveas(fig2,'2_RpRap_pop.eps','epsc');
% ---------------------------------------------------------------------



% ---------- COMPUTE THE POSSIBLE DISTANCES Rap - Rp COUPLES ----------
delta_Rap_Rp=zeros(1,N*N);
k=1;
for i=1:N
    for j=1:N
       delta_Rap_Rp(k) = Rap_pop(i) - Rp_pop(j);
       k=k+1;
```

```
        end
    end

    fig3=figure(3);
    hold on;
    title(sprintf('Possible pairs (Rap - Rp) resistance - Sample size = %i', N));
    xlabel('Resistance Difference [\Omega]');
    ylabel('Frequency [# devices with that Resistance Difference]');
    hist(delta_Rap_Rp,50);
    h = findobj(gca,'Type','patch');
    %display(h)
    set(h(1),'EdgeColor','white');
    hold off;
    saveas(fig3,'3_Distances_RapRp_couples.eps','epsc');
    % ---------------------------------------------------------------------



    % ------ COMPUTE ALL POSSIBLE DISTANCES Rp-Rp AND Rap-Rap COUPLES -----
    delta_Rp_Rp=zeros(1,N*(N-1));
    k=1;
    for i=1:N
        for j=1:N
            if j~=i
                delta_Rp_Rp(k) = Rp_pop(i) - Rp_pop(j);
                % Computes also the associated current parallel-parallel
                delta_I_pp(k) = Vbias_pop(i)/Rp_pop(i)-Vbias_pop(j)/Rp_pop(j);
                k=k+1;
            end
        end
    end

    delta_Rap_Rap=zeros(1,N*(N-1));
    k=1;
    for i=1:N
        for j=1:N
            if j~=i
                delta_Rap_Rap(k) = Rap_pop(i) - Rap_pop(j);
                % Computes also the associated current antiparallel-antiparallel
                delta_I_apap(k) = Vbias_pop(i)/Rap_pop(i)-Vbias_pop(j)/Rap_pop(j);
                k=k+1;
            end
        end
    end

    fig4=figure(4);
    hold on;
    title(sprintf('Possible pairs (Rp - Rp) and (Rap - Rap) resistance - ...
          Sample size = %i', N));
    xlabel('Resistance Difference [\Omega]');
    ylabel('Frequency [# devices with that Resistance Difference]');
    hist(delta_Rap_Rap,50);
    hist(delta_Rp_Rp,50);
    legend('Rap_{i} - Rap_{j}','Rp_{i} - Rp_{j}');
    h = findobj(gca,'Type','patch');
    %display(h)
    set(h(1),'FaceColor','g','EdgeColor','white');
    set(h(2),'FaceColor','b','EdgeColor','white');
    hold off;
    saveas(fig4,'4_Distances_RpRp_RapRap_couples.eps','epsc');

    fig5=figure(5);
    hold on;
    subplot(1,2,1)
    hist(delta_I_pp,100);
    title(sprintf('Pairs (Rp_{i} , Rp_{j})'));
    xlabel('Differential current [A]');
    ylabel('Frequency [# couples devices generating that current]');
    h = findobj(gca,'Type','patch');
    set(h(1),'EdgeColor','white');
    subplot(1,2,2)
    hist(delta_I_apap,100);
    title(sprintf('Pairs (Rap_{i} , Rap_{j})'));
    xlabel('Differential current [A]');
    ylabel('Frequency [# couples devices generating that current]');
    h = findobj(gca,'Type','patch');
    set(h(1),'EdgeColor','white');
    suptitle(sprintf('Currents distrib. associating all possible pairs (Ri,Rj)'));
    hold off;
    saveas(fig5,'5_Currents_Distr_RpRp_RapRap.eps','epsc');
```

```

    % ---------------------------------------------------------------------

end
```

THIRD SCRIPT: an example of how the PUF response is evaluated, using the electrical scheme for Case 4. The output includes the Bit-Error Rate that accompanies the generation of the response for the entire set of memory matrices.

Listing B.3. Evaluation of the PUF Signature for Case 4

```
function [ err_bit_percentage ] = case4_2T2MTJ_no_reference(N,m,l,...
                                  CHIPS_STT_MRAM,Sense_ampl_err,call_from_main)
        % 2T2MTJ memory cell with another 2T2MTJ memory cell
        for i=1:N
            % for each chip...
            for j=1:m
                % for each row...
                for k=1:4:l-3
                    % save the left 2T2MTJ cell current parameter
                    DeltaILeft=(CHIPS_STT_MRAM(j,k,i).Vbias/...
                                CHIPS_STT_MRAM(j,k,i).Rap)-(CHIPS_STT_MRAM...
                                (j,k+1,i).Vbias/CHIPS_STT_MRAM(j,k+1,i).Rp);
                    % save the right 2T2MTJ cell current parameter
                    DeltaIRight=(CHIPS_STT_MRAM(j,k+2,i).Vbias/CHIPS_STT_MRAM...
                                (j,k+2,i).Rap)-(CHIPS_STT_MRAM(j,k+3,i).Vbias...
                                /CHIPS_STT_MRAM(j,k+3,i).Rp);
                    % compute the response bit:
                    resp_bit=sense_amplifier(Sense_ampl_err,DeltaIRight,...
                                             DeltaILeft);
                    % Then save the response bit in all four cells
                    CHIPS_STT_MRAM(j,k,i).bit_val=resp_bit;
                    CHIPS_STT_MRAM(j,k+1,i).bit_val=resp_bit;
                    CHIPS_STT_MRAM(j,k+2,i).bit_val=resp_bit;
                    CHIPS_STT_MRAM(j,k+3,i).bit_val=resp_bit;
                end
            end
        end

        % compute the percentage of -1 (errors):
        err_bits=0;
        for i=1:N
            for j=1:m
                for k=1:4:l-4
                    if CHIPS_STT_MRAM(j,k,i).bit_val == -1
                        err_bits=err_bits+1;
                    end
                end
            end

        end
        err_bit_percentage=(err_bits/(N*m*(l/4)))*100;

        % Compute UNIQUENESS: inter-die Hamming distance (among chips)
        % But only if this function is called from MAIN! flag call_from_main=1
        % and, only if the SENSE AMP ERROR = 0!
        if call_from_main==1 && Sense_ampl_err==0
            [avg_inter_HD,disp_inter_HD]=case4_UniquenessEval(CHIPS_STT_MRAM,...
                                                              N,m,l);
        end

end
```

<u>FOURTH SCRIPT</u>: script used, for Case 4, to evaluate the PUF Uniqueness in terms of inter-die Hamming Distance, Hamming Weight and Bit Aliasing.

Listing B.4. Evaluation of the PUF Uniqueness for Case 4

```
function [avg_inter_HD,var_inter_HD]=case4_UniquenessEval(CHIPS_STT_MRAM,N,m,l)
    % Computes the average inter-die Hamming Distance among chips, and
    % provieds it back as output in the variable avg_inter_HD. Also, it
    % computes the following:
    %   1. The population of inter-die Hamming Distances, plotting the hist
    %   2. The Hamming Weight, plotting the graph
    %   3. The Bit Aliasing, plotting the graph
    %  ------------------------------------------------------------------
    % |                                                                  |
% | [!] TO PERFORM THIS EVALUATION, MUST ASSUME NO ERRORS IN THE PUF! |
% | That is, BER=0 => Bring to 0 any source of error (in our case, the|
% | sensitivity of the Current Differential Amplifier!).              |
% |-------------------------------------------------------------------|
%
% FIRST: reshape the 3D matrix so that it becomes a single matrix: each
% row corresponds to each mxl matrix: mxlxN --> Nx[m*(l/4)]
% for case1, careful to consider: [l-1] (last column is the reference!)
CHIPS_STT_MRAM_reshaped=zeros(N,m*(l/4));
for i=1:N
    %for each original matrix (chip) CHIPS_STT_MRAM, transform it into
    % a vector and save it into each ROW of CHIPS_STT_MRAM_reshaped
    z=1;
    for j=1:m
        for k=1:4:l-3
            CHIPS_STT_MRAM_reshaped(i,z)=CHIPS_STT_MRAM(j,k,i).bit_val;
            z=z+1;
        end
    end
end
% now CHIPS_STT_MRAM_reshaped contains 3D --> 2D matrix. Rows are chips

% SECOND: compute the Hamming distances between chips (=> rows), in %:
HD_inter = pdist(CHIPS_STT_MRAM_reshaped,'hamming');
% and transform that fractional distance into a %...
HD_inter = HD_inter*100;

avg_inter_HD=mean(HD_inter);
var_inter_HD=std(HD_inter,1);

% THIRD: Compute the Hamming Weight
X_HW=1:1:N;
HW=zeros(1,N);
for i=1:N
    HW(i)=(sum(CHIPS_STT_MRAM_reshaped(i,:))/(m*(l/4)))*100;
end
% FOURTH: Compute the Bit Aliasing
X_BA=1:1:m*(l/4);
BA=zeros(1,m*(l/4));
for i=1:m*(l/4)
```

```
        BA(i)=(sum(CHIPS_STT_MRAM_reshaped(:,i))/N)*100;
    end

    figure;
    subplot(1,3,1);
    hold on;
    title(sprintf('UNIQUENESS: Inter-die Hamming Distance distribution\n...
          CASE 2T2MTJ+2T2MTJ, NO REFERENCE\n\n Average(HD)=%f%% ; ...
          Deviation(HD)=%f%%',avg_inter_HD,var_inter_HD));
    hist(HD_inter,10);
    xlabel('HD [%]');
    ylabel('Count [#]');
    subplot(1,3,2);
    hold on;
    title(sprintf('Hamming Weight\nCASE 2T2MTJ+2T2MTJ, NO REFERENCE\n\n...
          Average: %.2f%% ; Deviation: %.2f%%',mean(HW),std(HW,1)));
    plot(X_HW,HW);
    plot(X_HW,ones(size(HW))*mean(HW),'g-');
    axis([-inf,+inf,min(HW)-10,max(HW)+10]);
    legend('Hamming Weight of each chip','Average Hamming Weight');
    xlabel('CHIP#');
    ylabel('Hamming Weight [%]');
    grid on;
    subplot(1,3,3);
    hold on;
    title(sprintf('Bit Aliasing\nCASE 2T2MTJ+2T2MTJ, NO REFERENCE\n\nAverage:...
          %.2f%% ; Deviation: %.2f%%',mean(BA),std(BA,1)));
    plot(X_BA,BA);
    plot(X_BA,ones(size(BA))*mean(BA),'g-');
    axis([-inf,+inf,min(BA)-10,max(BA)+10]);
    legend('Bit Alising of each chip','Average Bit Alising');
    xlabel('Bit Position in each CHIP matrix (reorganized by rows)');
    ylabel('Bit Alising [%]');
    grid on;
    hold off;

end
```

CITED LITERATURE

1. International technology roadmap for semiconductors (ITRS), emerging research devices, 2013 edition. Retrieved December 23, 2014, from http://www.itrs.com.

2. Prince, B.: Emerging memories: technologies and trends. Springer Science and Business Media, 2002.

3. Gallagher, W., Parkin, S. S., Lu, Y., Bian, X., Marley, A., Roche, K., Altman, R., Rishton, S., Jahnes, C., Shaw, T., et al.: Microstructured magnetic tunnel junctions. Journal of Applied Physics, 81(8):3741–3746, 1997.

4. Julliere, M.: Tunneling between ferromagnetic films. Physics letters A, 54(3):225–226, 1975.

5. Rodary, G., Hehn, M., Dimopoulos, T., Lacour, D., Bangert, J., Jaffrès, H., Montaigne, F., van Dau, F. N., Petroff, F., Schuhl, A., et al.: Development of a magnetic tunnel transistor based on a double tunnel junction. Journal of magnetism and magnetic materials, 290:1097–1099, 2005.

6. Shuto, Y., Nakane, R., Wang, W., Sukegawa, H., Yamamoto, S., Tanaka, M., Inomata, K., and Sugahara, S.: A new spin-functional metal–oxide–semiconductor field-effect transistor based on magnetic tunnel junction technology: Pseudo-spin-mosfet. Applied physics express, 3(1):013003, 2010.

7. Yuasa, S., Nagahama, T., Fukushima, A., Suzuki, Y., and Ando, K.: Giant room-temperature magnetoresistance in single-crystal fe/mgo/fe magnetic tunnel junctions. Nature materials, 3(12):868–871, 2004.

8. Gallagher, W. J. and Parkin, S. S.: Development of the magnetic tunnel junction mram at ibm: From first junctions to a 16-mb mram demonstrator chip. IBM Journal of Research and Development, 50(1):5–23, 2006.

9. Kishi, T., Yoda, H., Kai, T., Nagase, T., Kitagawa, E., Yoshikawa, M., Nishiyama, K., Daibou, T., Nagamine, M., Amano, M., et al.: Lower-current and fast switching of a perpendicular tmr for high speed and high density spin-transfer-torque mram. In Electron Devices Meeting, 2008. IEDM 2008. IEEE International, pages 1–4. IEEE, 2008.

10. Bruchon, N., Torres, L., Sassatelli, G., and Cambon, G.: New nonvolatile fpga concept using magnetic tunneling junction. In Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on, pages 6–pp. IEEE, 2006.

11. Guillemenet, Y., Torres, L., Sassatelli, G., and Bruchon, N.: On the use of magnetic rams in field-programmable gate arrays. International Journal of Reconfigurable Computing, 2008:1, 2008.

12. Matsunaga, S., Hayakawa, J., Ikeda, S., Miura, K., Endoh, T., Ohno, H., and Hanyu, T.: Mtj-based nonvolatile logic-in-memory circuit, future prospects and issues. In Proceedings of the Conference on Design, Automation and Test in Europe, pages 433–435. European Design and Automation Association, 2009.

13. Guo, Q., Guo, X., Patel, R., Ipek, E., and Friedman, E. G.: Ac-dimm: associative computing with stt-mram. In ACM SIGARCH Computer Architecture News, volume 41, pages 189–200. ACM, 2013.

14. Nomura, K., Abe, K., Yoda, H., and Fujita, S.: Ultra low power processor using perpendicular-stt-mram/sram based hybrid cache toward next generation normally-off computers. Journal of Applied Physics, 111(7):07E330, 2012.

15. Ando, K., Fujita, S., Ito, J., Yuasa, S., Suzuki, Y., Nakatani, Y., Miyazaki, T., and Yoda, H.: Spin-transfer torque magnetoresistive random-access memory technologies for normally off computing. Journal of Applied Physics, 115(17):172607, 2014.

16. Guo, X., Ipek, E., and Soyata, T.: Resistive computation: avoiding the power wall with low-leakage, stt-mram based computing. In ACM SIGARCH Computer Architecture News, volume 38, pages 371–382. ACM, 2010.

17. Zhang, L., Fong, X., Chang, C.-H., Kong, Z. H., and Roy, K.: Feasibility study of emerging non-volatilememory based physical unclonable functions. In Memory Workshop (IMW), 2014 IEEE 6th International, pages 1–4, May 2014.

18. Zhang, L., Fong, X., Chang, C.-H., Kong, Z. H., and Roy, K.: Highly reliable memory-based physical unclonable function using spin-transfer torque mram. In Circuits and Systems (ISCAS), 2014 IEEE International Symposium on, pages 2169–2172, June 2014.

19. Fong, X., Choday, S. H., and Roy, K.: Bit-cell level optimization for non-volatile memories using magnetic tunnel junctions and spin-transfer torque switching. Nanotechnology, IEEE Transactions on, 11(1):172–181, 2012.

CITED LITERATURE (Continued)

20. Zhao, W., Belhaire, E., Mistral, Q., Chappert, C., Javerliac, V., Dieny, B., and Nicolle, E.: Macro-model of spin-transfer torque based magnetic tunnel junction device for hybrid magnetic-cmos design. In Behavioral Modeling and Simulation Workshop, Proceedings of the 2006 IEEE International, pages 40–43. IEEE, 2006.

21. Zhang, Y., Zhao, W., Lakys, Y., Klein, J.-O., Kim, J.-V., Ravelosona, D., and Chappert, C.: Compact modeling of perpendicular-anisotropy cofeb/mgo magnetic tunnel junctions. Electron Devices, IEEE Transactions on, 59(3):819–826, 2012.

22. Garg, R., Kumar, D., Jindal, N., Negi, N., and Ahuja, C.: Behavioural model of spin torque transfer magnetic tunnel junction, using verilog-a. International Journal of Advancements in Research &amp; Technology, 1(6):36–42, 2012.

23. Mejdoubi, A., Prenat, G., and Dieny, B.: A compact model of precessional spin-transfer switching for mtj with a perpendicular polarizer. In Microelectronics (MIEL), 2012 28th International Conference on, pages 225–228. IEEE, 2012.

24. Pappu, R., Recht, B., Taylor, J., and Gershenfeld, N.: Physical one-way functions. Science, 297(5589):2026–2030, 2002.

25. Maiti, A., Casarona, J., McHale, L., and Schaumont, P.: A large scale characterization of ro-puf. In Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on, pages 94–99. IEEE, 2010.

26. Holcomb, D. E., Burleson, W. P., and Fu, K.: Power-up sram state as an identifying fingerprint and source of true random numbers. Computers, IEEE Transactions on, 58(9):1198–1210, 2009.

27. Das, J., Scott, K., Burgett, D., Rajaram, S., and Bhanja, S.: A novel geometry based mram puf. In Nanotechnology (IEEE-NANO), 2014 IEEE 14th International Conference on, pages 859–863. IEEE, 2014.

28. Das, J., Scott, K., Rajaram, S., Burgett, D., and Bhanja, S.: Mram puf: A novel geometry based magnetic puf with integrated cmos. 2015.

29. Marukame, T., Tanamoto, T., and Mitani, Y.: Extracting physically unclonable function from spin transfer switching characteristics in magnetic tunnel junctions. Magnetics, IEEE Transactions on, 50(11):1–4, 2014.

30. Iyengar, A., Ramclam, K., and Ghosh, S.: Dwm-puf: A low-overhead, memory-based security primitive. In Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on, pages 154–159. IEEE, 2014.

31. Iniewski, K.: Nano-semiconductors: Devices and Technology. CRC Press, 2011.

32. Driskill-Smith, A.: Latest advances and future prospects of stt-ram. In Non-Volatile Memories Workshop, 2010.

33. Herder, C., Yu, M.-D., Koushanfar, F., and Devadas, S.: Physical unclonable functions and applications: A tutorial. 2014.

34. Dodis, Y., Ostrovsky, R., Reyzin, L., and Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM journal on computing, 38(1):97–139, 2008.

VITA

NAME	Paolo Vinella
EDUCATION	B.S., Information Technology and Electronic Engineering, Politecnico di Torino, Politecnico di Milano and Tongji University of Shanghai, Italy and China, Double-Degree, 2012 M.S., Electronic Engineering, Politecnico di Torino, Italy, 2015 M.S., Electrical and Computer Engineering, UIC, U.S., 2015
HONORS	POLITONG mobility scholarship for Special Program, 2010-2011 TOP-UIC mobility scholarship for Special program, 2014 Scholarship for Thesis abroad in U.S., Fall 2014 Teaching and Research Assistantship with full tuition waiver at UIC for Spring 2015