Exploration of non-volatile magnetic memory for processor architecture

Sophiane Senni

Collège Doctoral du Languedoc-Roussillon

THÈSE

Pour obtenir le grade de

Docteur

Délivré par l'**Université de Montpellier**

Préparée au sein de l'école doctorale
I2S - Information, Structures, Systèmes

Et de l'unité de recherche
LIRMM - Laboratoire d'informatique, Robotique et Microélectronic de Montpellier

Spécialité: **Systèmes Automatiques et Microélectroniques**

Présentée par **Sophiane Senni**

Exploration of non-volatile magnetic memory for processor architecture

Soutenue le lundi 14 décembre 2015 devant le jury composé de

Jacques-Olivier KLEIN	Professeur, Université Paris Sud	Rapporteur
Jean-Michel PORTAL	Professeur, Université d'Aix Marseille	Rapporteur
Gregory DI PENDINA	Ingénieur de Recherche, CNRS/SPINTEC	Examinateur
Bruno MUSSARD	Ingénieur, Crocus Technology	Examinateur
Ian O'CONNOR	Professeur, Ecole Centrale de Lyon	Examinateur
Abdoulaye GAMATIE	Directeur de Recherche, CNRS/LIRMM	Examinateur
Francky CATTHOOR	Directeur Scientifique, IMEC	Examinateur
Lionel TORRES	Professeur, Université de Montpellier	Directeur de Thèse

UNIVERSITÉ DE MONTPELLIER

Dedicated to my family

Abstract

With the downscaling of the Complementary Metal-Oxyde Semiconductor (CMOS) technology, designing dense and energy-efficient system-on-chip is becoming a real challenge. Reducing the CMOS transistor size faces up to manufacturing constraints leading to many issues. Regarding the energy, a significant increase of the power density and dissipation obstructs further improvement in performance. The increase of the leakage current leads to a significant growth of the static energy consumption in current integrated systems. Embedded volatile memories, such as Static Random Access Memory (SRAM) and Dynamic Random Access Memory (DRAM), occupy a predominant part of the total silicon area in those systems-on-chip. As a result, a significant proportion of total power is spent into memory systems. In the past two decades, alternative memory technologies have emerged with attractive characteristics to mitigate the aforementioned issues. Among these technologies, Magnetic Random Access Memory (MRAM) is a promising candidate as it combines simultaneously high density and very low static power consumption while its performance is competitive compared to SRAM and DRAM. Moreover, MRAM is non-volatile. This capability, if present in embedded memories, has the potential to add new features to enhance energy efficiency and reliability. In this thesis, an area, performance and energy exploration of embedding the MRAM technology in the memory hierarchy of a processor architecture is investigated. A first fine-grained exploration was made at cache level for multi-core architectures. A second study evaluated the possibility to design a non-volatile processor integrating MRAM at register level. Finally, within the context of low power applications for the internet of things, some new features brought by these emerging memory technologies were studied.

Keywords: Magnetic Random Access Memory (MRAM), non-volatility, processor, memory hierarchy

Résumé

De par la réduction continuelle des dimensions du transistor CMOS, concevoir des systèmes sur puce à la fois très denses et énergétiquement efficients devient un réel défi. Réduire la dimension du transistor CMOS est sujet à de fortes contraintes de fabrication, entraînant de nombreuses problématiques. Pour l'aspect énergétique, une augmentation importante de la puissance dissipée par unité de surface freine l'évolution en performance. Ainsi l'augmentation des courants de fuite entraîne une augmentation de l'énergie statique des systèmes intégrés considérés. Notons que les mémoires embarquées volatiles telles que la SRAM et la DRAM occupent une part prédominante de la surface silicium de ces systèmes sur puce. C'est la raison pour laquelle une partie significative de la puissance totale consommée dans les circuits actuels provient des composants mémoires. Ces deux dernières décennies, de nouvelles mémoires non volatiles sont apparues possédant des caractéristiques pouvant aider à résoudre ces problèmes. Parmi ces nouvelles technologies mémoires, la MRAM (mémoire magnétique) est une candidate avec un fort potentiel, elle permet d'allier une forte densité d'intégration et une consommation d'énergie statique quasi nulle, tout en montrant des performances comparables à la SRAM et à la DRAM. De plus, la MRAM a la capacité d'être non volatile. Ceci est particulièrement intéressant pour l'ajout de nouvelles fonctionnalités afin d'améliorer l'efficacité énergétique ainsi que la fiabilité. Ce travail de thèse a permis de mener une exploration en surface, performance et consommation énergétique de l'intégration de la MRAM au sein de la hiérarchie mémoire d'une architecture de processeur. Une première exploration fine a été réalisée au niveau mémoire cache pour des architectures multicœurs. Une seconde étude a permis d'évaluer la possibilité d'intégrer la MRAM au niveau registre pour la conception d'un processeur non volatile. Enfin, dans le cadre d'applications électroniques embarquées faible consommation pour les objets connectés, de nouvelles fonctionnalités que peuvent apporter ces technologies ont été étudiées.

Mots clés : MRAM, non volatilité, processeur, hiérarchie mémoire

Acknowledgements

First of all, I would like to thank Pr. Lionel Torres, my advisor, for his guidance, support and patience over the last three years. Thanks for giving me the opportunity to work on the fascinating field of MRAM. Thanks for the fruitful discussions we had together to complete successfully this thesis. Thanks for the invaluable advice you gave me for my future career.

Then, I would like to thank Crocus Technology which funded this thesis and thus have made this research possible. A special thank to the team I have joined at Crocus Technology, including Bruno Mussard, Alain Faburel, Christophe Gineste and Ali Alaoui for the time I spent with them during this thesis.

At LIRMM, I would also like to thank Dr. Gilles Sassatelli for his insights about the evaluation of processor architecture, Abdoulaye Gamatié for his help and advice on writing papers, all the PhD students for the fruitful discussions I had with them about new research ideas, and all the people that somehow helped making the achievement of this thesis possible.

Many thanks to Pr. Ian O'Connor, Pr. Jacques-Olivier Klein, Pr. Jean-Michel Portal, Dr. Gregory Di Pendina and Pr. Francky Catthoor, who accepted to be part of my thesis committee, and for the fruitful discussions during my PhD defense.

Finally, I would like to thank a lot my family for their help and patience during the long hours I spent working on this thesis.

Contents

List of Figures

List of Tables

Acronyms

AMR Anisotropic Magnetoresistance

CAM Content-Addressable Memory

CMOS Complementary Metal-Oxyde Semiconductor

DRAM Dynamic Random Access Memory

FeRAM Ferroelectric Random Access Memory

FPGA Field-Programmable Gate Array

GMR Giant Magnetoresistance

IC Integrated Circuit

IoT Internet of Things

ISA Instruction Set Architecture

LLC Last-Level Cache

MCU Microcontroller

MeRAM Magnetoelectric Random Access Memory

MLU Magnetic Logic Unit

MR Magnetoresistance

MRAM Magnetic Random Access Memory

MTJ Magnetic Tunnel Junction

NVM Non-Volatile Memory

OxRAM Oxyde-based Resistive Random Access Memory

PCRAM Phase-Change Random Access Memory

RAM Random Access Memory

ReRAM Resistive Random Access Memory

SoC System-On-Chip

SRAM Static Random Access Memory

SOT Spin Orbit Torque

SOT-MRAM Spin Orbit Torque MRAM

STT Spin Transfer Torque

STT-MRAM Spin Transfer Torque MRAM

TAS Thermally Assisted Switching

TAS-MRAM Thermally Assisted Switching MRAM

TMR Tunneling Magnetoresistance

1

INTRODUCTION

1.1 Context

Intensive investigations are underway to resolve the most critical problem of current nano-electronic systems: energy efficiency. Major issues encountered in today's Integrated Circuits (ICs) include high leakage current, performance saturation, increased device variability and process complexity. For battery-powered applications, energy consumption is unquestionably the most critical metric. In dynamic mode, fast switching at low power is targeted. In static mode, low leakage power is desired. Current systems embed volatile devices such as flip-flops, Static Random Access Memory (SRAM) and Dynamic Random Access Memory (DRAM), which lose information when powered off. Circuit design techniques, such as clock and power gating, are currently used to reduce the power consumed during standby mode. Although these techniques can reduce the consumption of static energy, it is not so easy to manage the total power consumption. First, System-On-Chips (SoCs) are becoming more and more complex with the increasing number of transistors per die. Regarding on-chip memories, as they are mostly volatile, several power modes are required such as active, standby, retention, deep sleep and power down for various application demands [2]. The possibility of integrating Non-Volatile Memory (NVM) would greatly facilitate the power-saving techniques implementation.

One possible way to overcome the energy efficiency issue is non-volatile SoC using non-volatile devices. In this case, a complete power down is possible with no loss of data or logic states. A promising candidate for non-volatile SoCs is magnetic memory (MRAM) based on Magnetic Tunnel Junction (MTJ) component. Both academia and industry regard MRAM as a suitable technology to become a universal memory as it combines low leakage, high density and has low access time compared to other existing and emerging NVMs such as flash, Phase-Change Random Access Memory (PCRAM) or Resistive Random Access Memory (ReRAM). However, despite the many attractive fea-

tures of MRAM, two challenges are still under intensive investigation. First, MTJ switching requires a significant amount of current. Second, even if MTJ is orders of magnitude faster than conventional NVM, e.g. flash or embedded flash, it is slower than typical 6-transistor-based SRAM, especially for write operations. However, Toshiba recently published very encouraging results [3] on a perpendicular MTJ technology with an access time of $3ns$ and read/write bit energy that is almost equivalent to SRAM. MRAM has attracted many researchers, and many studies have been conducted to evaluate integration of MRAM in the memory hierarchy of processor architecture.

1.2 Thesis objectives and contributions

1.2.1 Objectives

Within the context introduced above, the global objective of this thesis is to explore how MRAM can improve the overall performance of a SoC, according to the three metrics speed/energy/area, by integrating it at different level in the memory hierarchy (Figure 1.1). This thesis focus on processor architecture.

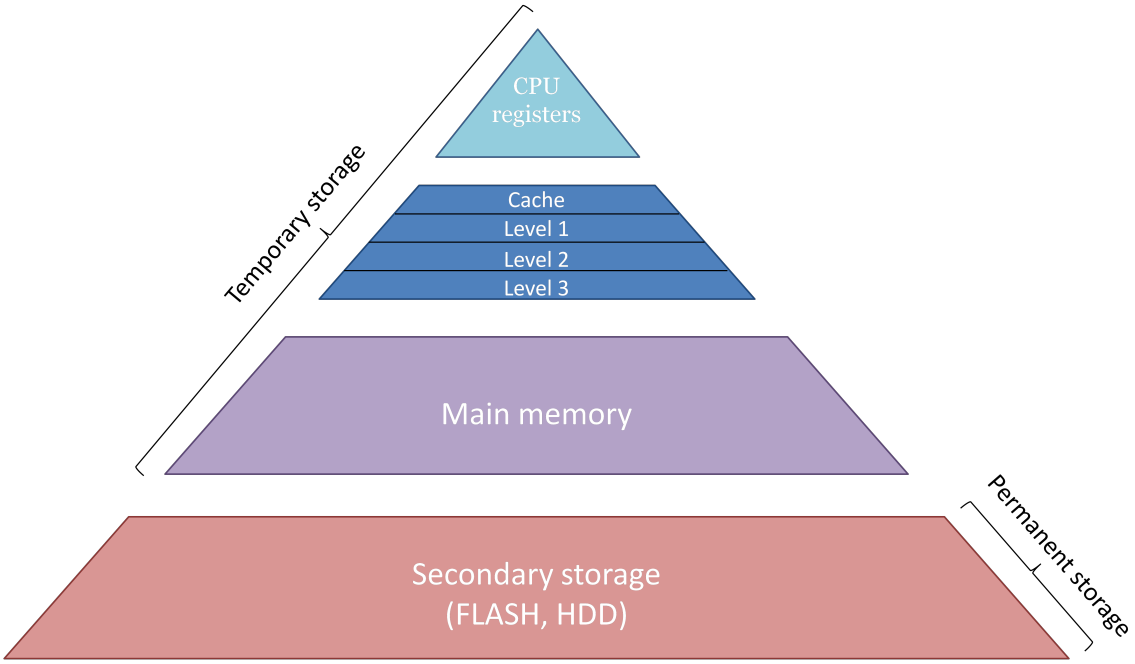


Figure 1.1: Memory hierarchy

Although it combines attractive features such as low leakage power and non-volatility, MRAM suffers from both high write latency and energy, which need to be mitigated if integration at deep level in a SoC is desired, e.g. at register level. This thesis does not

propose any techniques or optimizations at device, circuit or architecture level to address the drawbacks of MRAM. Instead, this work aims at setting up a fine-grain exploration flow to evaluate MRAM considering its current characteristics. In addition to the speed/energy/area/exploration, investigation on the potential new features that MRAM can bring into SoCs thanks to the non-volatility is also part of the objectives.

1.2.2 Contributions

As part of the contributions, this work explored use of MRAM into the memory hierarchy of processor architecture at register, cache and main memory levels. The impact in terms of performance, energy and area has been analyzed and new computing paradigms enabled by the non-volatility of MRAM has been studied. The main contributions of this thesis are:

1. Development of a fine-grain exploration flow to evaluate NVM-based cache thanks to:
    - a modified version of the gem5 simulator to model asymmetric read/write latencies
    - the extraction of important information on the memory hierarchy activity: read/write ratio, dynamic/static energy ratio, miss rate, bandwidth
2. Performance/energy/area evaluation of both L1 and L2 caches based on STT-MRAM
3. Performance/energy/area evaluation of L2 cache based on TAS-MRAM
4. Validation on a full 32-bit RISC embedded processor of:
    - the possibility to save/restore the complete state of a processor (*instant-on/off*)
    - the possibility to restore a previous valid state of the processor (*rollback*), for instance in the case of an execution error.
5. Analysis of a non-volatile processor with *instant-on/off* and *rollback* capabilities:
    - Analysis of the architectural changes
    - Analysis of the impact in terms of performance and energy

1.3 Thesis organization

To have a clear overview of the thesis organization, a diagram is given in Figure 1.2.

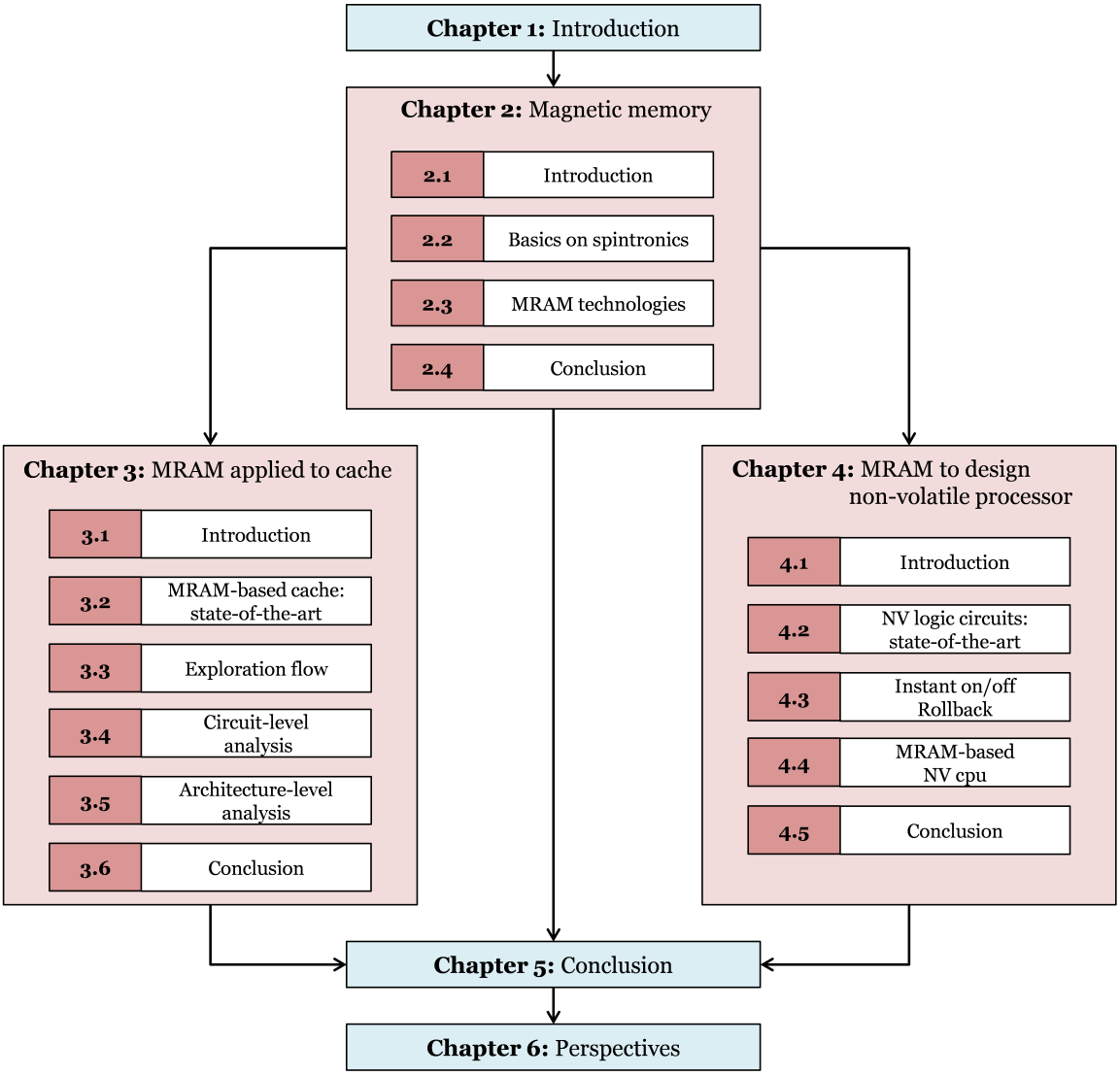


Figure 1.2: Thesis organization

Following this introduction, Chapter 2 gives basics on spintronics and describes the Physics of MRAM technology. Then, existing MRAM technologies are reviewed.

Chapter 4 examines integration of MRAM at cache level into the memory hierarchy of multi-core architecture. First of all, previous studies on MRAM-based cache are reviewed. Afterwards, an exploration flow to evaluate MRAM-based cache memory in terms of speed, energy, and area is described. Finally, a full exploration of MRAM-based cache at both circuit and architecture levels is reported.

Chapter 4 studies the possibility to have a non-volatile processor thanks to MRAM-based registers, within the context of low-power applications related to the internet of

things. Firstly, an overview of the previous works on non-volatile logic circuits is given. Secondly, this chapter introduces two features when using non-volatile registers into a processor: *instant-on/off* and *rollback*. Thirdly, performance and energy consumption of such a non-volatile processor are estimated.

Chapters 5 and 6 respectively conclude this work and give an insight on perspectives.

1.4 Company supporting this thesis

The work carried out during this thesis was funded by Crocus Technology [4], a company which develops and supplies magnetic sensors and embedded memory solutions designed with Magnetic Logic Unit (MLU) technology based on Thermally Assisted Switching MRAM (TAS-MRAM). Due to its proprietary MLU technology, Crocus' magnetic sensors bring significant advantages to industrial, consumer electronics and automotive applications requiring high sensitivity, high temperature, low-noise and low-cost. MLU's distinguishing properties for enabling speed and endurance afford new levels of robustness to Crocus' embedded memory solutions aimed at the Internet of Things (IoT) and security applications. Crocus is headquartered in Santa Clara, California, and has offices in Grenoble and Rousset, France. It co-owns Crocus Nano Electronics, a Russian-based advanced magnetic semiconductor manufacturing facility.

2

MAGNETIC MEMORY

2.1 Introduction

For information processing, electronics use a fundamental property of the electron: its electric charge. In the 1980s, discoveries on the spin-dependent electron transport phenomena gave birth to what is known today as spintronics [5]. Spintronics is a new paradigm for information storage and logic operation using another fundamental property of the electron: the spin. In this new technology, this is the electron spin that carries information instead of the electron charge. Use of spin as information is currently based on the orientation of a spin ("up" or "down") relative to a reference (e.g. magnetic orientation of a ferromagnetic film). The detection of the relative orientation of the spin is performed using the spin-dependent electron transport properties of semiconductor-ferromagnet interfaces. Compared to semiconductor devices, spintronics-based devices are expected to be faster, more energy efficient and denser, with the capability of non-volatile data storage.

This chapter gives the basics of spin-based electronics by presenting the main discoveries related to spin interactions with the magnetic properties of a material. In addition, an insight into the spintronic applications is given with a specific focus on memory devices.

The rest of the chapter is organized as follows: Section 2.2 presents the major phenomena related to spin interactions in solid-state devices. Then, the basic element to build magnetic memories is introduced. Section 2.3 reviews existing MRAM technologies. Section 4.6 concludes this chapter.

2.2 Basics

2.2.1 Magnetoresistance effect

In a conducting solid, the mean velocity of the electron is proportional to the electric field. This can be described as $v = \mu \cdot E$, where v is the mean electron velocity, μ is the electron mobility (dependent on the material), and E is the electric field. The presence of magnetic field affects the electron transport through the Lorentz force, which can be described as $F = ev \cdot B$, where e is the electron charge, and B the magnetic induction. As a result, the resistance of the material varies applying a magnetic field: this is the Magnetoresistance (MR) effect. This effect (referred as anisotropic magnetoresistance) was first discovered by William Thomson in 1856. Few years later, researchers would discover two more pronounced MR effects known as the giant magnetoresistance and the tunnel magnetoresistance. The rest of this section will describe these three MR effects.

Anisotropic magnetoresistance

William Thomson discovered the Anisotropic Magnetoresistance (AMR) through experiments on iron and nickel. He observed a variation of the electrical resistance when an external magnetic field is applied. The electron scattering (i.e. electrons are deviated from their original trajectory) rate is affected depending on the direction of the field. When the magnetization is perpendicular to the current direction, the electron scattering is smaller, whereas when the magnetization is parallel to the current direction, the electron scattering is larger. Figure 2.1 illustrates the AMR effect resulted by the interaction between the magnetization (green arrows in the figure) and the electron spin. Electrons whose spin is in the opposite direction of the magnetization (white spheres in the figure) are scattered more than electrons whose spin is parallel with the magnetization (red spheres in the figure). In ferromagnetic materials, AMR affects the resistance in the order of a few percent. However, in the late 1970s, it was sufficient to successfully develop AMR sensors to replace inductive sensors as the read head in hard-disks.

Giant magnetoresistance

In the late of 1980s, two groups of researchers, led by Albert Fert and Peter Grünberg, independently discovered the Giant Magnetoresistance (GMR) effect in structures alternating ferromagnetic (FM) and non-magnetic (NM) layers. The group of A. Fert investigated the MR of thirty to sixty stacked Fe/Cr structures and observed almost a factor of 2 between the resistivities at zero field and in the saturated state, respectively [7] (Figure 2.2). On the other hand, the group of P. Grünberg studied the MR of a simple Fe/Cr/Fe

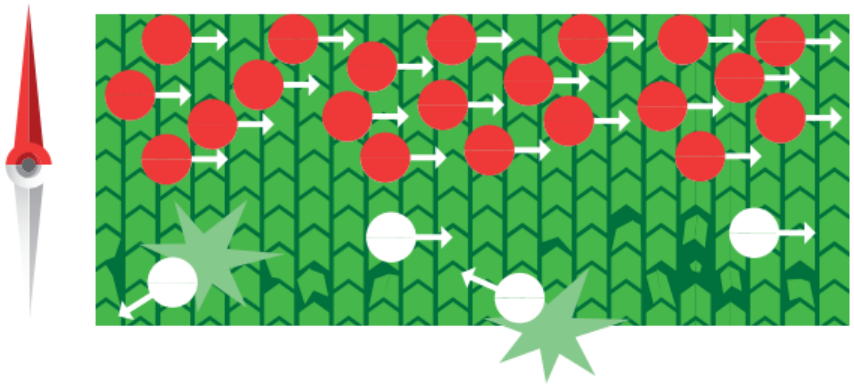


Figure 2.1: Magnetoresistance effect [6]

structure and noticed that an anti-parallel alignment of the magnetization of the Fe layers increases the electrical resistivity, much more than AMR effect.

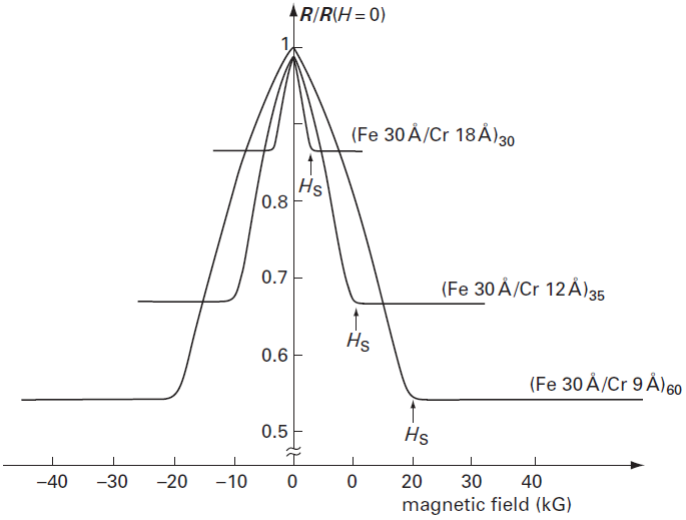


Figure 2.2: Giant magnetoresistance: experiment results on N stacked Fe/Cr structures with N = 30, 35 and 60 at 4.2K [7]

Grünberg and his group explained this effect by spin-flip scattering, as shown in Figure 2.3. When the spin of the electrons is parallel to the direction of magnetization of the FM layer, the electrons are weakly scattered and the FM layer shows a small resistance. On the other hand, when the spin of the electrons is anti-parallel to the direction of magnetization of the FM layer, the electrons have a strong scattering and the FM layer shows a large resistance.

With no external magnetic field, the FM/NM/FM structure shows an anti-ferromagnetic behavior (i.e. the magnetization of the FM layers are in opposite direction). In such struc-

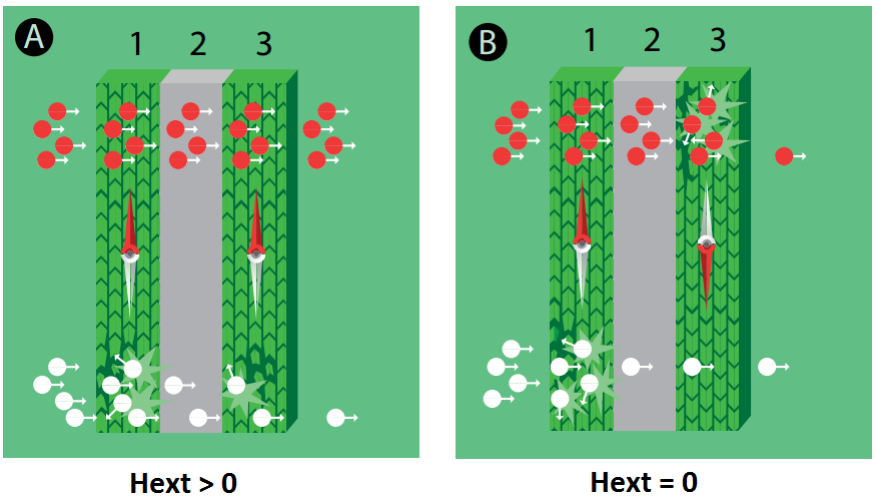


Figure 2.3: Giant magnetoresistance effect [6]

ture, both the spin-up electrons and the spin-down electrons will have a strong scattering since they will necessary cross a layer with a magnetization anti-parallel to the spin. When an external magnetic field is applied, both FM layers are aligned in the same direction. As a result, the electrons whose the spin is parallel to magnetization of the FM layers will be weakly scattered. In this case, the electrical resistivity is lowered compared to the anti-parallel configuration of the structure.

The material has a resistance consisting of two resistances in parallel. One resistance for the spin-up electrons and on resistance for the spin-down electrons. If we consider $R_{\downarrow}$ as the resistance of a FM layer with weak scattering, and $R_{\uparrow}$ as the resistance of a FM layer with strong scattering, then the total resistance of the parallel (anti-parallel) configuration of the structure can be given by the equation 2.1 (equation 2.2):

$$\begin{aligned} R_p &= (2 \cdot R_{\downarrow}) \parallel (2 \cdot R_{\uparrow}) \\ &= \frac{2 \cdot R_{\uparrow} \cdot R_{\downarrow}}{R_{\uparrow} + R_{\downarrow}} \end{aligned} \tag{2.1}$$

$$\begin{aligned} R_{ap} &= (R_{\uparrow} + R_{\downarrow}) \parallel (R_{\downarrow} + R_{\uparrow}) \\ &= \frac{R_{\uparrow} + R_{\downarrow}}{2} \end{aligned} \tag{2.2}$$

Therefore, the difference in resistance between the two configurations is given by the equation 2.3:

$$\begin{aligned} \Delta R &= R_p - R_{ap} \\ &= -\frac{1}{2}\frac{(R_\uparrow - R_\downarrow)^2}{R_\uparrow + R_\downarrow} \end{aligned} \tag{2.3}$$

Tunneling magnetoresistance

Tunneling Magnetoresistance (TMR) is a phenomenon first discovered by Tedrow and Meservey in 1970 [8]. They observed that the tunneling electrons through junctions between very thin superconducting aluminum layers and ferromagnetic nickel layers is spin dependent. In 1975, Michel Jullière studied the conductance of two ferromagnetic layers separated by a thin insulator [9]. He measured the tunneling conductance dependence on voltage at 4.2K. He observed on a Fe/Ge-O/Co structure a relative resistance change of 14% at zero bias, and 2% when the voltage bias is increased at 6mV.

Unlike GMR which is related to electron scattering, TMR relies on the spin polarizations of the conduction electrons. In a parallel configuration of the material, electrons whose spin is parallel to the direction of the magnetization of the layers will tunnel through the barrier, whereas electrons whose is anti-parallel will be filtered. In an anti-parallel configuration of the structure, both spin-up and spin-down electron flows are reduced, resulting in a large resistance.

Jullière measured the conductance ratio, which is known today as the TMR ratio given by the equation 2.4, where P_1 and P_2 are the spin polarizations of the two ferromagnetic layers:

$$\begin{aligned} TMR &= \frac{\Delta R}{R_p} \\ &= \frac{R_{ap} - R_p}{R_p} \\ &= \frac{2 \cdot P_1 \cdot P_2}{1 - P_1 \cdot P_2} \end{aligned} \tag{2.4}$$

In 1995, Terunobu Miyazaki and Nobuki Tezuka reported a TMR ratio of 2.7% at room temperature in a NiFe/Al_2O_3/Co structure [10]. In the same year, Moodera et al. observed the first giant TMR ratio of 11.8% in a Al_2O_3-based junction at room temperature [11]. Since 2004, junctions based on a Al_2O_3 barrier have reached a TMR ratio of 70% [12]. However, the breakthrough regarding the TMR ratio will come with barriers based on the MgO. In 2001, Butler et al. and Mathon and Umersk theoretically predicted that a giant TMR ratio higher than 1000% could be obtained in fully epitaxial

Fe(001)/MgO(001)/Fe(001) structure [13, 14]. In 2008, experimental TMR reached 600% at room temperature in a CoFeB/MgO/CoFeB junction [15].

The TMR effect have raised a great interest to design spintronic devices. Figure 2.4 summarizes the evolution of the magnetoresistance ratio and the device applications. The following section will describe the magnetic tunnel junction as a data storage unit.

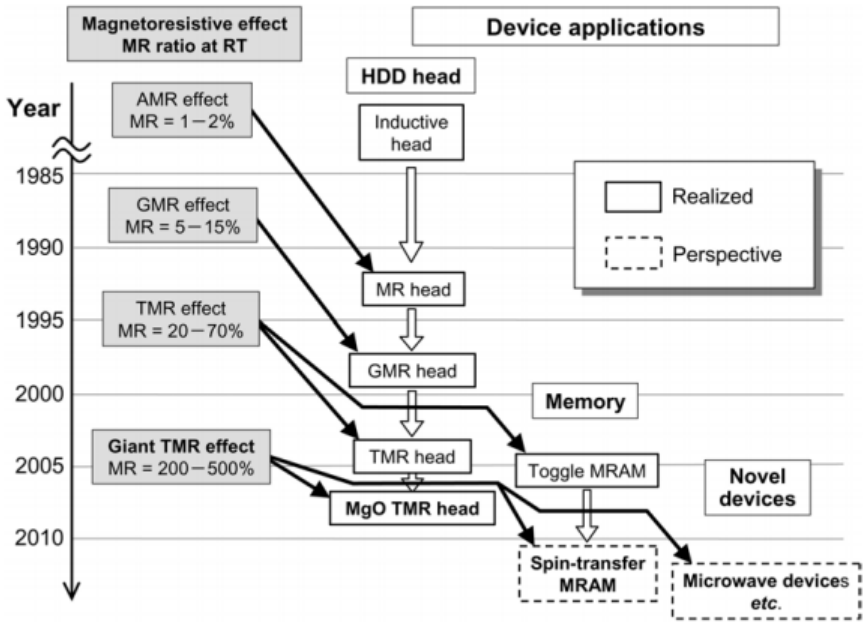


Figure 2.4: Magnetoresistance ratio evolution [16]

2.2.2 Magnetic tunnel junction

A MRAM bit is a MTJ consisting of two ferromagnetic layers separated by a thin insulating barrier. The information is stored as the magnetic orientation of one of the two layers, called the free layer (FL) or storage layer. The other layer, called the reference layer or fixed layer (RF), provides the fixed reference magnetic orientation required for reading and writing. The TMR effect causes MTJ resistance to depend significantly on the relative orientation of the two magnetic layers: the antiparallel state provides much larger resistance than the parallel state. It enables the magnetic state of the FL to be sensed thanks to a current flowing through the MTJ. Hence, stored information can be read. Five methods have been proposed to switch the orientation of the FL: toggle [17], Thermally Assisted Switching (TAS) [18], Spin Transfer Torque (STT) [19], voltage-induced switching and the most recent method is called Spin Orbit Torque (SOT) [20].

2.3 Magnetic Random Access Memory Technologies

2.3.1 Conventional

A conventional MRAM, shown in Figure 2.5, uses a simple way to program the MTJ where sufficient magnetic field is generated thanks to a combination of two current flows applied simultaneously through a row and a column of an MTJ array. Two problems arose with this method. First, large current is needed to generate sufficient magnetic field to reverse the magnetization of the FL. Second, this approach suffers from selectivity problem: some of the bits sharing the same row or column of the cell being programmed might be exposed to sufficient magnetic field and be switched unintentionally. This effect is one consequence of process variability. The magnetic field necessary to reverse the magnetization is not exactly the same for all the bits [21].

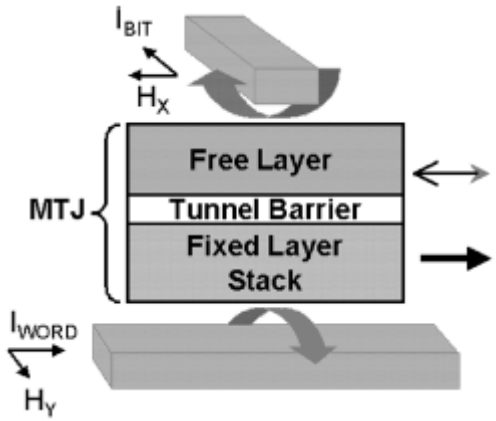


Figure 2.5: Conventional MRAM [22]

2.3.2 Toggle

This technology is currently commercialized by Everspin [23]. Figure 2.6 illustrates a standard toggle MRAM MTJ. A Complementary Metal-Oxyde Semiconductor (CMOS) access transistor provides a current through the MTJ needed for the read operation. Each MTJ is located at the intersection of two conductive lines, I_{WORD} and I_{BIT} in Figure 2.6. Toggle MRAM was proposed to deal with the selectivity problem observed in the standard MRAM. The toggle MRAM adds a second FL and an anti-ferromagnetic coupling layer above the first FL, as shown in Figure 2.6. In addition, a specific timing sequence of the write-current pulses, shown in Figure 2.7, is used to switch only the MTJ at the intersection of the conductive lines. These changes improve the stability of the magnetic orientation of the bit cell and avoid the selectivity issue. Unlike other MRAM technologies,

the toggle MRAM scheme does not drive the bit cell to a predetermined state, instead it always reverses the current magnetization of the FL. As a result, a read of the current state of the MTJ is required before a write if the opposite state is desired.

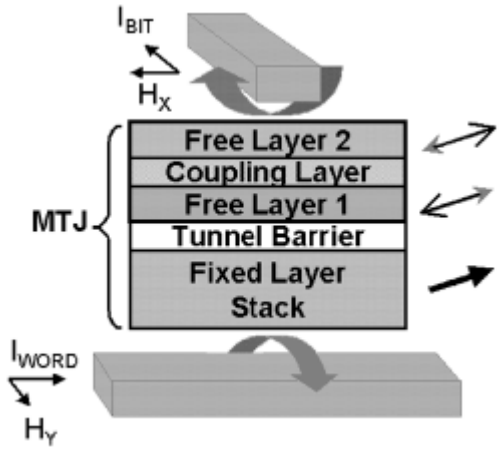


Figure 2.6: Toggle MRAM [22]

The toggle MRAM has certain limitations. Although it resolves the selectivity issue of the conventional MRAM, it still needs a significant amount of current to switch the bit cell, thereby limiting the upper bound of the write speed. Moreover, the amount of current needed for writing remains almost the same even when the size of the bit cell is reduced. Consequently, the selectivity issue can appear again when scaling the MTJ. In addition, as the switching current does not shrink scaling the technology node, the area of the peripheral circuits of the MTJ array also remains the same, thus limiting the density [21]. Toggle MRAM is not predicted to be suitable at nodes less than 90 nm.

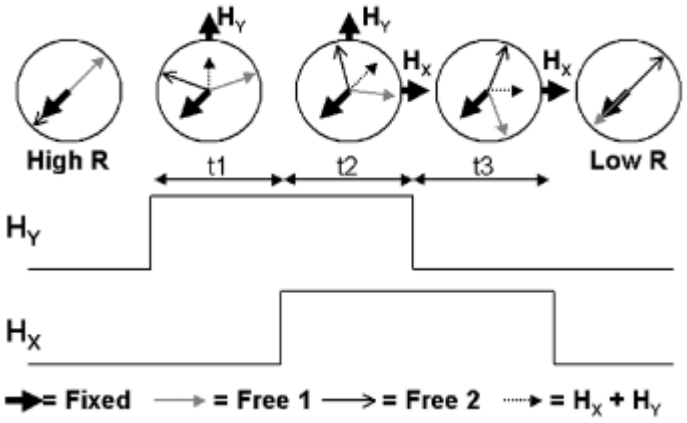


Figure 2.7: Toggle write sequence [22]

2.3.3 Thermally assisted switching

The aim of the TAS concept was to improve the downsize scalability of MRAM. The concept was developed by the SPINTEC laboratory and TAS-MRAM is currenly commercialized Crocus Technology. As shown in Figure 2.8, TAS-based MTJ uses an anti-ferromagnetic layer (low T_B in Figure 2.8) to block the magnetic orientation of the FL under a threshold temperature. To switch the bit cell, a select transistor provides a flow of current to heat the MTJ above the blocking temperature thereby enabling storage of new information thanks to application of a magnetic field. Heating the FL allows TAS-MRAM to use a smaller magnetic field and hence less current than toggle MRAM to write the bit cell, since a single conductive line is sufficient to generate the required magnetic field. Blocking the FL's state using a coupling anti-ferromagnetic layer also significantly improves data stability, even scaling the technology node. As a result, TAS-MRAM makes it possible to reduce the switching energy while ensuring excellent data retention. This new method also solves the selectivity issue, since the MTJ has to be heated before writing.

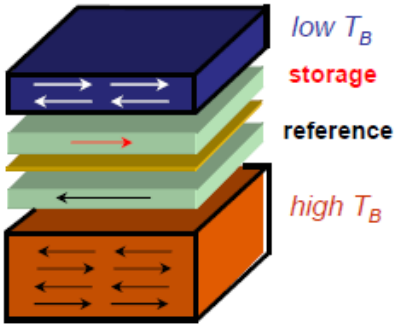


Figure 2.8: MTJ structure of TAS-MRAM [18]

Figure 2.9 shows a complete TAS write operation. Assuming the MTJ stores a "0" state (parallel state), the first step in the TAS method is to heat the FL by flowing a current through the MTJ to reach the blocking temperature (heating step in Figure 2.9). The second step is to generate an external magnetic field to switch the FL while heating the MTJ (switching step in Figure 2.9). Once the FL switches to the "1" state, the CMOS transistor responsible for the heating process is switched off whereas the MTJ remains under the external magnetic field (cooling step in Figure 2.9).

Crocus technology designed another implementation of TAS-based MTJs, called MLU [25], in which the reference layer (RL) is replaced by a self-reference layer (SRL). As a result, the SRL can easily be switched by applying an external magnetic field because the magnetic orientation of the SRL is not fixed like that of the RL, as shown in Figure 2.10.b. While the write scheme remains the same, the read operation is quite different. In this

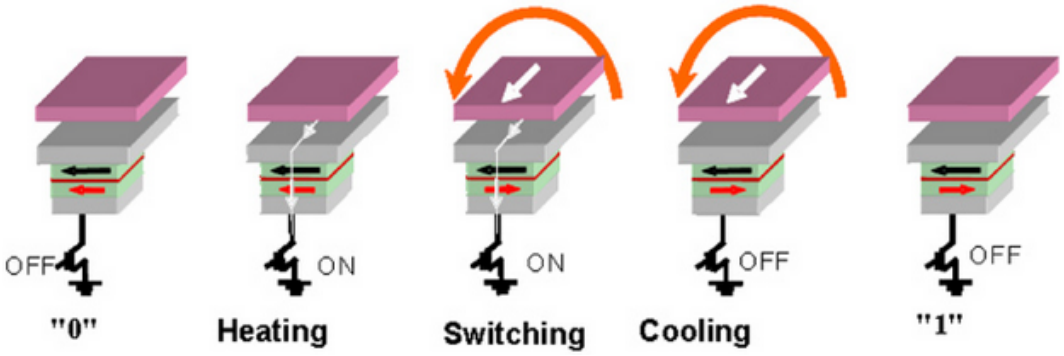


Figure 2.9: Thermally assisted switching MRAM [24]

case, reading consists of two steps: the MTJ resistance is measured while the magnetization of the SRL is set in one direction. The MTJ resistance is then measured again when the magnetization of the SRL is reversed to the opposite direction. The resistance variation between the two measurements provides information about the magnetization of the FL. The new approach increases the read time, but tolerance to process variation is clearly improved since each bit cell is self-referenced. Moreover, this approach significantly reduces reading errors. Usually, the two resistance states need to be well separated, which can lead to manufacturing problems when scaling the technology node. For MLU, using the difference in the two states for reading is not sensitive to this manufacturing problem.

The innovative MLU concept also leads to another interesting feature: the device can also act as an exclusive-OR logic gate (XOR). Assuming that the two magnetic layers are the inputs and the MTJ's resistance is the output, the truth table of a XOR logic gate can be built (Figure 2.10.c). This makes MLU a particularly useful component of security applications. An example of application introduced by Crocus technology is the *match-in-place* [25], shown in Figure 2.11. If the current direction applied on the field line during a read operation is considered as an information input, the device can use its XOR logic capability to compare in situ the data stored in memory with the input and control if there is a match or a mismatch comparing the output resistance with a reference. Other possible applications of MLU are: content-addressable memory, NOR-MRAM, NAND-MRAM [24].

Although the structure of TAS-MRAM means it has better scalability than toggle MRAM, TAS-MRAM needs a non-negligible time to complete its write operation due to the heating/cooling processes. Moreover, since an external magnetic field is used to switch the MTJ, the amount of current is still high, even if it is lower than for the toggle. One possible way to reduce the switching energy is to combine the TAS method with the STT effect. TAS-MRAM is expected to be scalable down to 45 nm [24]. However, the aim of combining TAS with STT is to further improve scalability. The strongest advantage

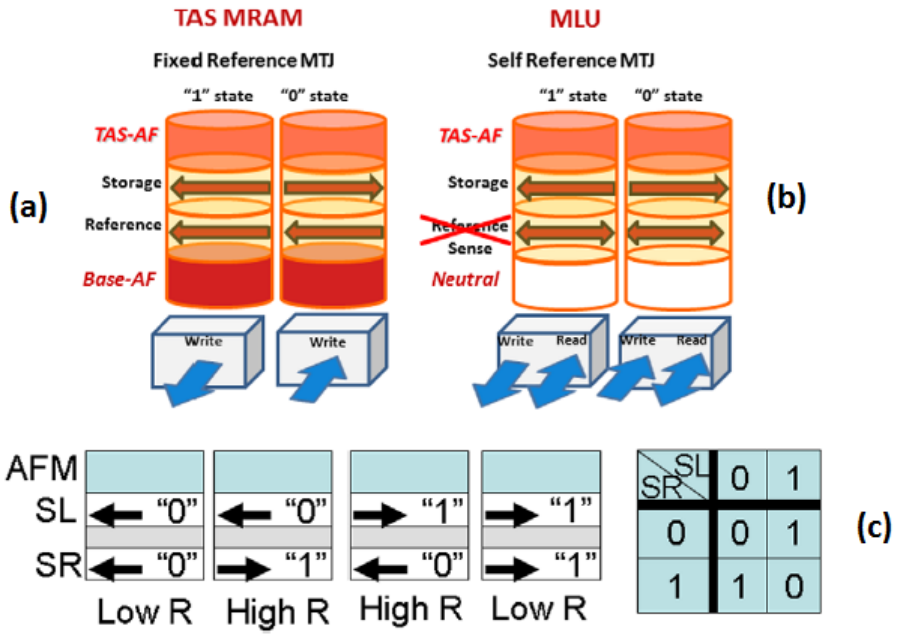


Figure 2.10: Magnetic Logic Unit: (a) Magnetic stack of TAS-MRAM (b) Magnetic stack of MLU (c) Virtual XOR logic gate of MLU [24]

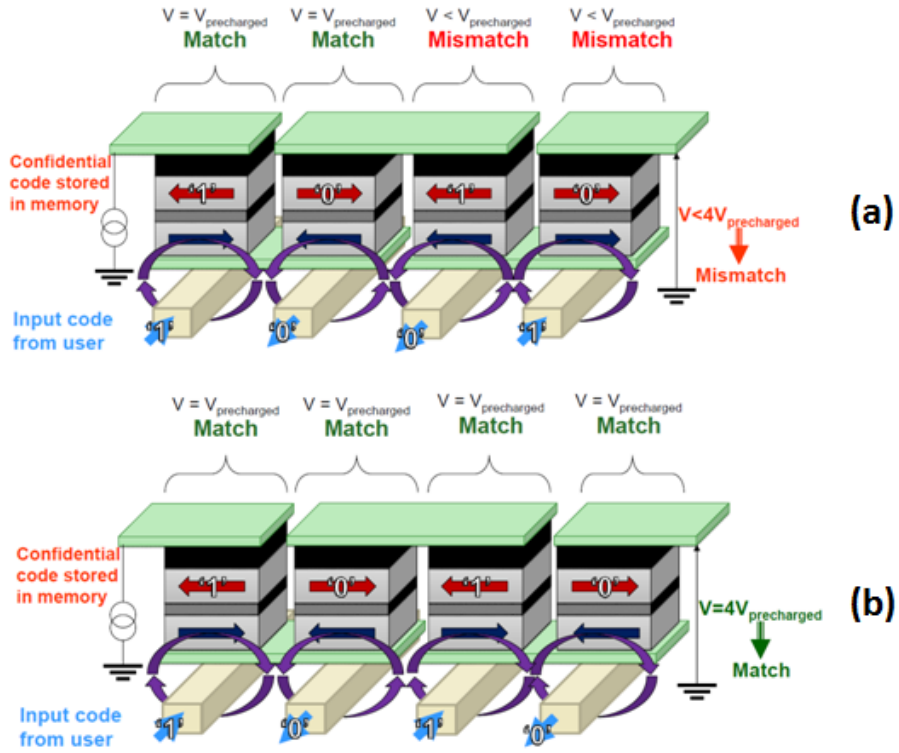


Figure 2.11: Match-in-place principle: (a) mismatch, (b) match

of TAS-MRAM is its high thermal stability thanks to its MTJ structure. The latter allows very good data retention, and good reliability against magnetic field disturbance [24].

2.3.4 Spin transfer torque

Spin Transfer Torque MRAM (STT-MRAM) appeared with the need to reduce the switching energy consumption of MRAM. Unlike the previous MRAM technologies, which use an external magnetic field to program a bit cell, STT-MRAM write operations are based on another physical phenomenon to switch the magnetic orientation of the FL called STT. The idea is that the FL can be switched by direct transfer of the spin angular momentum from spin-polarized electrons. In this way, a highly spin-polarized current flowing through the MTJ causes a "torque" applied by the injected electron spins on the magnetization of the FL. Applying sufficient current will cause sufficient torque to switch the bit cell, thereby enabling information to be written. Figure 2.12 depicts the STT effect.

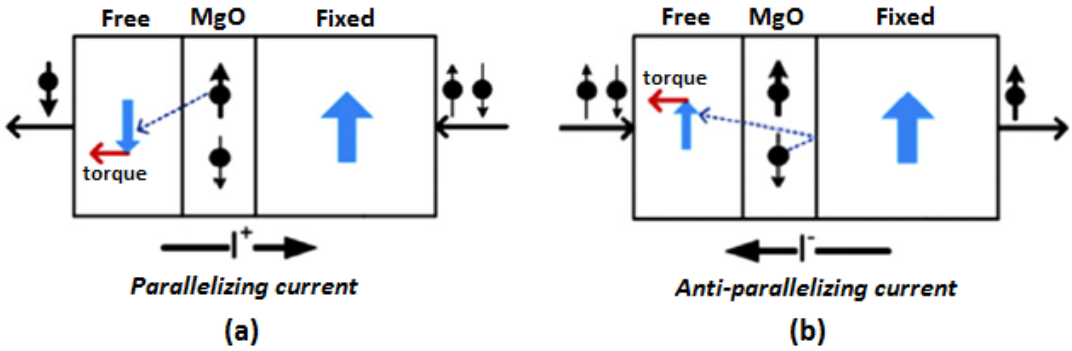


Figure 2.12: Spin transfer torque effect: (a) illustration of the transition from an antiparallel to a parallel state, and (b) the transition from a parallel to an antiparallel state [19]

Figure 2.12.a shows the transition from an antiparallel to a parallel state. In this case, electrons go through the fixed layer first, and the fixed layer acts as a polarizer. Thus, electrons are spin-polarized in the magnetic orientation of the fixed layer. Once the insulating barrier (MgO) is crossed, the spin-polarized electrons exert torque on the magnetization of the FL until a magnetic orientation reversal occurs. A similar effect is depicted in Figure 2.12.b for the transition from a parallel to an antiparallel state. In this case, electrons go through the FL first. While the majority of the electrons will be spin-polarized in the magnetic orientation of the FL, a minority of electrons will still be spin-polarized in the opposite direction of the FL. These minority electrons will be reflected at the barrier interface and will exert torque on the magnetization of the FL.

Two kinds of magnetization of the magnetic layers can be found in STT-MRAM: in-

plane and perpendicular. In-plane magnetization is also used in toggle MRAM and TAS-MRAM, in which the magnetic orientation is parallel to the plan of the MTJ, whereas in perpendicular magnetization, the magnetic orientation is perpendicular to the plan of the MTJ. Perpendicular STT-MRAM was introduced to further reduce the switching current of the MTJ and to improve scalability.

State-of-the-art showed that STT-MRAM read access time is similar and sometimes better than its SRAM equivalent [26, 27, 28]. Concerning write operations, despite the fact that STT-MRAM considerably reduces switching energy compared to the previous MRAM technologies, some limitations were observed. Some of them were mitigated or eliminated while others still remain.

First, read and write operations use the same path, which can lead to unexpected writes when reading is underway, particularly with advanced technology nodes. To mitigate this issue, a solution was proposed at device level designing a three-terminal dual-pillar MTJ structure with two spatially and electrically independent ports for writes and reads [29].

Second, the current needed to switch the MTJ from the parallel to the antiparallel state (and vice-versa) is not symmetrical [30]. Switching from a parallel to an antiparallel state requires more current than the reverse. This is because switching from an antiparallel to a parallel state is performed by spin-polarized electrons going through the MTJ (majority of the electrons), whereas switching from a parallel to an antiparallel state is performed by reflected spin-polarized electrons (a minority of the electrons). A solution was also proposed to eliminate this problem by adding a complementary polarizer [31, 32]. In this proposed device, the MTJ has two pinned layers instead of one, with opposite magnetic orientations. Depending on the information to write, the switching current will flows through the corresponding pinned layer.

Third, STT-MRAM is confronted to scalability issue. When a STT-MRAM cell is scaled, the thermal stability factor scales down linearly with the area, and can cause unreliability due to retention failure [33]. Moreover, although its switching energy remains low compared to Toggle and TAS-MRAM, STT-MRAM needs access transistor sizes larger than the minimum size at advanced node (32 nm and below) [34], limiting thus the memory density . This is an issue also for high performance applications which require high write speed, since the switching current of STT-MRAM increases when the write pulse width decreases.

2.3.5 Voltage Induced Switching

In order to improve the scalability and reduce the switching energy observed with STT-MRAM, a voltage-controlled MTJ were proposed [35, 36, 37, 38, 39], also known as Magnetoelectric

Random Access Memory (MeRAM). As shown in Figure 2.13, this approach uses voltage rather than current to reverse the magnetization of the free layer thanks to the recently demonstrated voltage-controlled magnetic anisotropy effect (VCMA) [40]. The free layer has a magnetic anisotropy that can be changed by voltage. Hence, voltage-induced switching of the magnetization can be performed modifying the magnetic anisotropy of the MTJ. The voltage-controlled MTJ (VMTJ) structure uses materials commonly used by previous MRAM technologies, thus maintaining manufacturability [35]. VMTJ has an unipolar voltage-controlled behavior, i.e. switching is performed by set/reset voltages of different amplitudes but same polarity, whereas STT-MRAM uses opposite current polarities to switch the bit cell.

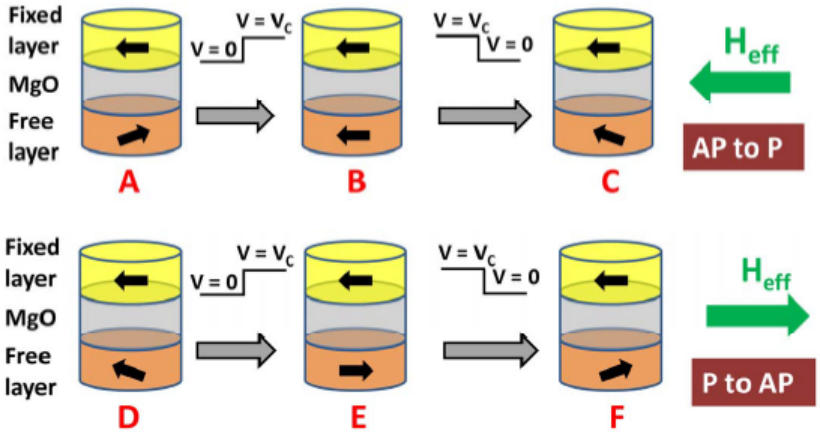


Figure 2.13: Voltage induced switching [35]

Since switching for VMTJ is performed via voltage, the barrier thickness can be increased to reduce the parasitic conductance and hence the effect of current-induced torques (i.e. STT effect). Moreover, a high TMR (greater than 100%) is possible allowing the read-out of the magnetization of the free layer [41]. Although it is still at experimental level and needs further improvements in the design, MeRAM is expected to improve the scalability by eliminating the need for large currents, which is currently a real issue with STT-MRAM for advanced technology nodes.

2.3.6 Spin orbit torque

Spin Orbit Torque MRAM (SOT-MRAM) is the most recent technology for MRAM. It was developed to mitigate the issues observed in STT-MRAM. Contrary to STT-MRAM, this new technique uses a three-terminal structure to separate the read and write paths, as shown in Figure 2.14.b. The physical effect responsible for the reversal of magnetization of the FL is not yet fully understood. According to some authors, the Rashba effect [42]

or the spin Hall effect [43] could explain the switch in magnetization of the storage layer.

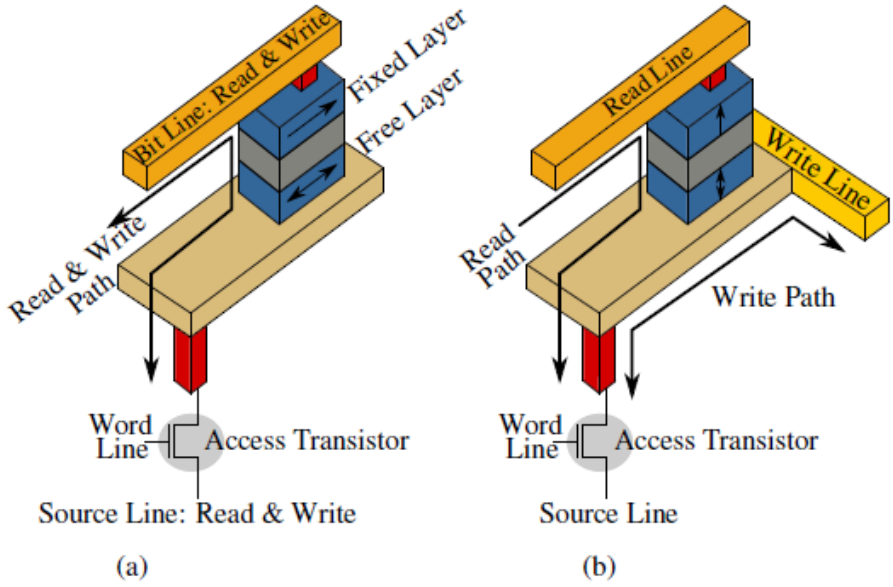


Figure 2.14: SOT-MRAM: (a) Conventional STT-MRAM (b) SOT-MRAM [44]

Unlike STT-MRAM, SOT-MRAM intrinsically separates the read and write paths and allows symmetrical switching current between the two states of the MTJ. Hence, read stability is improved, strongly reducing the possibility of a bit flip (the bit changes its state) during a read operation. Also, designers can optimize the read and write separately. On the other hand, SOT-MRAM has a bigger cell size than STT-MRAM because of its three-terminal structure. As SOT-MRAM is a young technology compared to other MRAM technologies, further research is needed to optimize the SOT-based MTJs. Like MeRAM, a great potential is expected from this technology to reach same performance as SRAM.

2.4 Conclusion

This chapter presented the main phenomena related to the MR effect, which lead to the development of MRAM technology. As described above, a material consisting of alternate FM and NM layers shows lowest electrical resistance when the magnetic moments of the FM layers are aligned, and highest electrical resistance when they are anti-aligned. Thus, it was observed that the transport of the electrons is spin-dependent due to interactions between the electron spin and the magnetic properties of the material. This discovery marked the beginning of spintronics. Advances on the MR effect allowed to reach a ratio of more than 100% between the lowest and the highest resistance of a mate-

rial. As a result, MRAM technologies emerged and intensive investigations are currently underway to improve their performances. To have an overview of the differences between them, MRAM technologies reported above are summarized in Table 2.1. Due to its voltage-controlled switching scheme, MeRAM needs a very low write current compared to other MRAM technologies. Hence, very high scalability is expected. STT-MRAM and SOT-MRAM show almost the same overall performance and are very good candidates to be part of the memory hierarchy of SoCs. Unlike SOT-MRAM, first test chips have already been developed for STT-MRAM. Compared to other MRAM technologies, TAS-MRAM is the most reliable thanks to its MTJ structure which allows excellent thermal stability, and then very good data retention.

The following chapters analyze integration of some MRAM technologies into the memory hierarchy of processor architecture. Chapter 4 focus on multicore architecture and evaluate MRAM in cache memory. Chapter 4 explore the benefits of having a non-volatile processor including MRAM at register level.

Compared to other MRAM technologies, Toggle MRAM has a very high switching energy and and its scalability is limited. Hence, it is not considered for the remaining of this report. Although MeRAM and SOT-MRAM show very promising performance, they are always at experimental level and need further development. On the contrary, TAS-MRAM and STT-MRAM are quite mature since test chips already exist [45, 46, 47, 4]. Therefore, only these two technologies are considered for the next chapters.

Technology	**Cell size** (F^2)	**Access time read/write**	**Write current**	**Endurance**	**Maturity**	**Advantages/Drawbacks**
Toggle MRAM [17, 22, 23]	50	35 ns / 35 ns	>30 mA	10^{15}	Commercialized	(+) Maturity (-) High power
TAS-MRAM [4, 18, 24]	<50	30 ns / 30 ns	A few mA	10^{15}	Test chip [4]	(+) Reliability (-) Access time
STT-MRAM [19, 30]	<50	2-20 ns / 2-20 ns	50 uA	$> 10^{16}$	Test chip [45, 46, 47]	(+) Low power (-) Reliability
MeRAM [41, 48]	<10	<10 ns	very low	$> 10^{16}$	Prototype	(+) Low power (-) Maturity
SOT-MRAM [26, 44, 49]	<50	A few ns	<100 uA	$> 10^{16}$	Prototype	(+) Low power (-) Maturity

Table 2.1: MRAM technologies

3

MRAM APPLIED TO CACHE MEMORY

3.1 Introduction

Since the advent of ICs, the number of transistor per die never stops increasing to reach today several billion of transistors [50]. The decreasing size of the CMOS transistor has made possible the fabrication of small devices able to run at high speed. On the other hand, the power consumption of SoC has significantly increased due to the high density of integrated components. As a result, current nanoelectronic systems are confronted with heat issue because of the high power dissipation, which is a real obstacle to the increase of the frequency. With the limit of the frequency scaling, a shift to parallel computing has been observed to form the era of multi-core processors.

Regarding the three metrics speed/energy/area, memory is a key element for future SoCs. Richard Sites, one of the fathers of computer architecture, said in his article entitled "It's the memory, Stupid!" [51]:

> *Across the industry, today's chips are largely able to execute code faster than we can feed them with instructions and data... The real design action is in memory subsystems—caches, buses, bandwidth, and latency.*

Since processing elements have to be fed with instructions and data from memories, the latter plays an important role on the overall performance of the system. Furthermore, an increasing trend of embedding more volatile memory in SoCs is observed. As shown in Figure 3.1, memory systems occupy more than half of the die area. As a consequence, a significant proportion of total power is spent on memory systems (Figure 3.2). The predominant technology is SRAM, currently used for both cache memory and registers because of its fast access time compared to other technologies. However, it consumes more and more static energy due to the increase of the leakage current when decreasing the technology node. This is a major issue to energy efficiency.

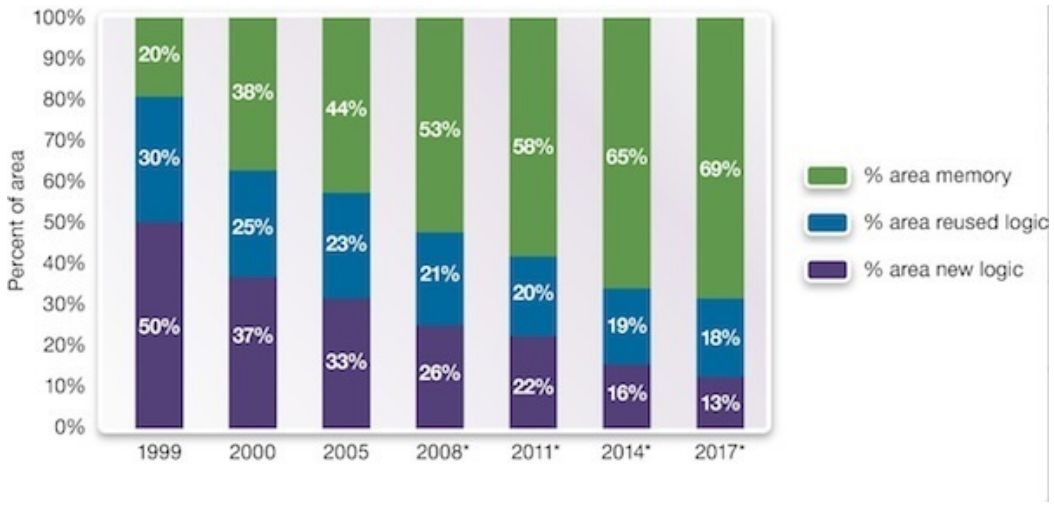


Figure 3.1: SoC area repartition between logic and memory (from Semico Research Corporation [52])

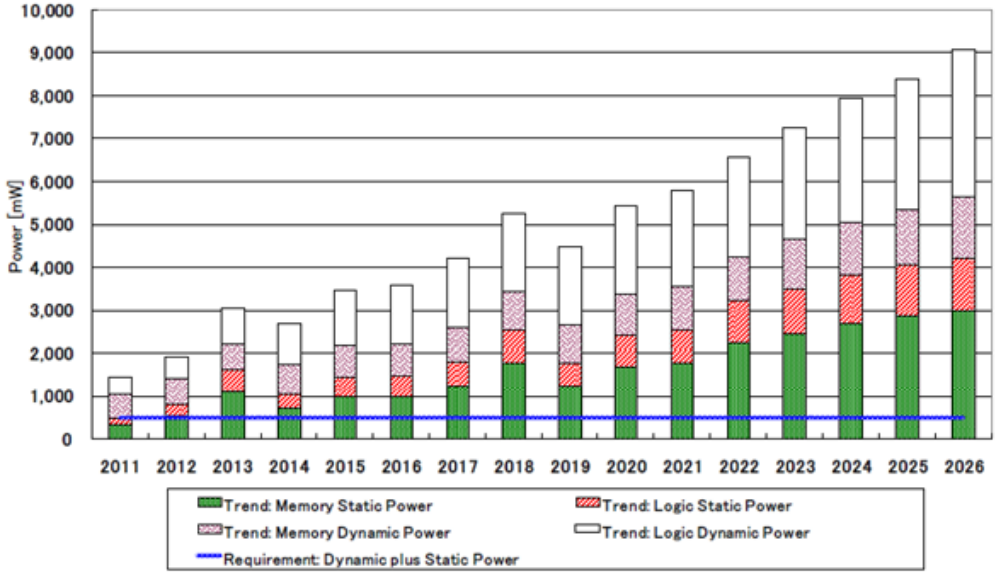


Figure 3.2: SoC energy repartition between logic and memory (from ITRS [53])

In the past two decades, alternative memory technologies have emerged with attractive characteristics to mitigate the aforementioned issues. Among these technologies, MRAM is a promising candidate as it combines simultaneously high density and very low static power consumption while its performance is competitive compared to SRAM and DRAM.

This chapter presents a fine-grain exploration to evaluate the performance and energy impacts of including MRAM in the memory hierarchy of processor architecture. The ex-

ploration is discussed based on L1 and L2 cache memories. In the present study, useful information about the memory traffic are extracted, such as the cache miss rate and the cache bandwidth. This information is monitored over time to better understand the behavior of the workloads in terms of memory access. Hence, a clear vision of the influence of these parameters on performance and energy is possible. In addition, a set of data including the read/write ratio, the static/dynamic energy ratio and L1/L2 access ratio are also extracted to better analyze the impact of the different read/write latencies, of the high dynamic energy and of the low leakage of MRAM, thus enabling fine-grain analysis of the performance and the total energy consumption of MRAM-based cache.

The rest of the chapter is organized as follows: Section 4.2 reviews the state-of-the-art on MRAM-based cache memory. Section 3.3 describes the NVM exploration flow used in this study to allow a fine-grain evaluation of including MRAM in the memory hierarchy of processor architecture. Section 3.4 analyzes and compares both MRAM and SRAM caches at circuit level. Section 3.5 explores MRAM-based cache at architecture level for both L1 and L2 caches (L2 as last-level-cache). Section 4.6 concludes this chapter.

3.2 State-of-the-art review

Many studies have been conducted on integration of MRAM into the memory hierarchy of single-core and multi-core architectures. All these studies explored a hybrid cache hierarchy using SRAM and STT-MRAM technologies, but a few studies also explored use of DRAM and PCRAM. Most of the studies evaluated the use of STT-MRAM for last-level cache (Last-Level Cache (LLC)). A few authors explored MRAM for upper levels of cache such as L1.

3.2.1 3D-stacking MRAM

Some authors studied the benefit of the 3D-stacking ability of MRAM combined with its high density to evaluate 3D-processor architecture. They analyzed the performance and energy impacts of having a MRAM-based LLC on top of a 2D-processor architecture. [54] and [55] evaluated a 3D-stacked STT-MRAM-based L2 on top of a 2D-processor architecture. Considering the same area constraint, use of MRAM in L2 results on a 89% and 73% total power reduction compared to a conventional L2 based on SRAM. [56] explored three different memory hierarchy configurations with hybrid L2/L3 cache architectures using MRAM, PCRAM and embedded DRAM in an 8-core processor considering a 3D chip integration. Under the same area constraint and considering a 3-level cache hierarchy in which L1 is based on SRAM, results showed 18% IPC[1] improvement and up to

[1] Instruction Per Cycle

70% of total power reduction over a 3-level cache hierarchy based on SRAM only.

3.2.2 MRAM-based non-uniform cache architecture

Other authors explored non-uniform cache architectures (NUCA) using both SRAM and MRAM in one cache level. [57] and [56] proposed a hybrid cache consisting of a large but slow MRAM-based region and small but fast SRAM-based region. Using data migration policies, the objective is to write mostly in the SRAM region (because it is faster) and to read data from the MRAM region. In this way, performance degradation due to the high write latency of MRAM can be mitigated. In addition, larger cache capacity is possible thanks to the high density of MRAM. Simulation results of such a hybrid cache in L2 showed 55% of the total power reduction on average and 5% IPC improvement over a SRAM L2 baseline. [58] proposed a hybrid SRAM/STT-MRAM cache architecture for chip-multiprocessors. In addition, micro-architectural mechanisms were introduced to reduce the number of writes in STT-MRAM regions. Considering an 8-core architecture, use of this hybrid SRAM/STT-MRAM architecture in a shared L2 cache showed that the overall power consumption is reduced by 37.1% and performance is improved by 23.6% on average compared with SRAM based static NUCA L2 cache under the same area configuration.

3.2.3 Novel management policies for MRAM-based cache

Several cache management techniques have been proposed to mitigate the two main drawbacks of MRAM (high write latency and high write energy). [59] proposed a novel technique called early write termination (EWT). EWT aims at removing unnecessary writes (i.e. writing same value) to reduce the write energy consumption in the cache. Because MTJ does not switch gradually but abruptly at the end of a STT-based write, this technique proposed to read the stored value during a write operation and to stop the write if it is redundant. Considering a 4-core architecture, evaluation of the EWT technique on a shared L2 cache based on STT-MRAM shows that up to 80% of write energy reduction can be achieved through EWT, resulting on 33% less total energy consumption, and 34% reduction in energy-delay product compared to a STT-MRAM-based L2 cache without EWT. [55] introduces the read-preemptive write buffer technique in which write buffers are used to mitigate the long write latency of MRAM. In addition, when there is a conflict between a read (from the upper level cache) and a write (from the write buffer), a read-preemptive policy gives priority to the read in order to prevent write operations from blocking read operations due to the long write latency of writes. By using the read-preemptive write buffer technique on a STT-MRAM-based L2, results showed up to 9% of performance improvement and up to 67% of power reduction compared to

SRAM-based L2. [28] proposed a similar technique called the obstruction-aware policy (OAP) for cache management of a single-port STT-MRAM-based L3 (LLC). In addition to preventing the delay of several read operations caused by a long write operation, OAP can mitigate the performance degradation of the MRAM-based LLC while significantly reducing total energy consumption thanks to the ultra-low leakage of MRAM. After adopting OAP and considering 4-core system with an 8MB STT-RAM L3 cache, results showed 14% performance improvement on average, and a reduction of 64% on the total L3 energy consumption. [60] proposed a new STT-MRAM cache architecture called asymmetric write architecture with redundant blocks (AWARE) to reduce the average cache write latency. This is done by taking advantage of the asymmetric write characteristics of STT-MRAM. Use of this technique on a STT-MRAM-based L2 cache showed a reduction of the average cache write latency by 30% over conventional STT-MRAM cache design, at the cost of an increase of the cache write energy by 7%.

3.2.4 Other studies on MRAM-based cache

Among studies on MRAM-based cache, [61] evaluated the performance/energy impacts of a STT-MRAM-based L2 when the retention time of the MTJ is reduced. Reducing the retention time of the MTJ can reduce both switching energy and switching latency. 10+ years, $1s$ and $10ms$ retention times for STT-MRAM were explored. In addition, a cache revive policy was proposed for the $10ms$-retention-time-based STT-MRAM to refresh data if necessary. Using this scheme on a L2 cache based on STT-MRAM showed an average $10 - 12\%$ improvement in performance compared to the traditional SRAM-based L2 cache design, while reducing the energy consumption by 60%. [62] proposed fine-grain power gating on STT-MRAM peripheral circuits to further reduce the total energy consumption of STT-MRAM-based LLC. The simlation results showed a reduction up to 80% of leakage power in state-of-the-art STT-MRAM LLC. [26, 44] made a first study of using the new SOT-MRAM technology in caches. Both SOT-MRAM-based L1 and L2 were explored and compared with SRAM-based and STT-MRAM-based caches. The best configuration simulated was a hybrid combination of SRAM for the L1-Data-cache, SOT-MRAM for the L1-Instruction-cache and L2-cache, which can reduce the energy consumption by 60% while the performance increases by 1% compared to an SRAM-only configuration. Although this work is close to the study presented in this chapter, some important aspects on the architecture and the memory behavior were not taken into account for relevant analysis. First, the exploration were only made for a single-core architecture. It is also important to analyze the impact of having a multi-core architecture because critical information such as the memory bandwidth can significantly influence the results (especially for a cache shared between the cores). Second, device and circuit-level

information were mostly used to analyze performance and energy results at architecture level. As proposed in this chapter, it is necessary to take into account the memory activity of the cache, such as the miss rate and bandwidth since our results have demonstrated the high influence of these parameters on the overall performance and energy consumption. [63] also explored MRAM-based L1 cache (Data-Cache) using an advanced perpendicular STT-MRAM (ap-STT-MRAM) proposed in [64]. This work highlighted that the read latency of STT-MRAM is the new bottleneck when it is used in upper level cache (i.e. L1). After demonstrating the major performance penalty of replacing SRAM by STT-MRAM in L1-Data cache, micro-architectural modifications by means of an intermediate buffer placed between the processor and the L1-Data cache have been proposed to overcome the read limitations of the STT-MRAM. In addition, appropriate data allocations schemes coupled with code transformations and optimizations are performed to reduce the performance penalty introduced by the STT-MRAM to extremely tolerable levels (8%) compared to a SRAM L1-Data cache design.

3.2.5 Summary

Many works on MRAM-based caches point to the real interest of this memory technology for future IC. The common trend in these studies was to take advantage of the non-volatility, high density, low leakage, and 3D-stacking capability of MRAM while mitigating its drawbacks, which are high write energy and latency. Results showed that for large cache capacity (e.g. LLC), systems can significantly benefit from the high density and the ultra-low leakage of MRAM. For write intensive workloads, cache management needs to be optimized to mitigate the high write energy and latency of MRAM. The energy and area gains of MRAM-based cache can be potentially important not only at circuit level but also at system level. [65] showed that for the big.LITTLE system [66], the L2 cache area is about 40% and 30% of the total area of the cortex-A7 cluster (4-core) and of the cortex-A15 cluster (4-core), respectively. An energy evaluation showed that cache energy consumption (including L1 and L2) represents around 50% and 25% of the total energy consumption of the cortex-A7 cluster and the cortex-A15 cluster, respectively, when only one core is active.

Table 3.1 gives a summary of previous studies on MRAM-based cache. Although they evaluated many architectures when including MRAM into cache, these works did not analyzed the influence of the memory traffic on the overall performance and energy consumption of MRAM-based cache. In this chapter, the repercussion of many parameters (read/write ratio, static/dynamic energy ratio, ratio of the number of accesses between different levels of cache, cache miss rate, cache bandwidth) on MRAM-based cache is investigated.

Reference	Number of cores	Cache hierarchy	Architectures & techniques	Main results
[54]	1	SRAM 16kB L1, **STT-MRAM 16MB L2** **baseline**: SRAM 4MB L2	3D-stacking	89% total L2 power reduction
[55]	8	SRAM 16kB L1, **shared STT-MRAM 8MB L2** **baseline**: shared SRAM 2MB L2	3D-stacking + read-preemptive write buffer	9% performance improvement 67% total L2 power reduction
[56]	8	SRAM 32kB L1, SRAM 256kB L2, **STT-MRAM 4MB L3** **baseline**: SRAM 1MB L3	Direct replacement	5% performance improvement 65% total L3 power reduction
[57]	1	SRAM 32kB L1, **hybrid SRAM/STT 4MB L2** **baseline**: SRAM 4MB L2	Hybrid cache + data migration policies	5% IPC improvement 55% total L2 power reduction
[58]	8	SRAM 16kB L1, **shared hybrid SRAM-512kB/STT-12MB L2** **baseline**: shared SRAM 2MB L2	Hybrid cache + micro-architectural mechanisms	23.6% performance improvement 37.1% total L2 power reduction
[59]	4	SRAM 32kB L1, **shared STT-MRAM 16MB L2** **baseline**: shared STT-MRAM 16MB L2 without EWT techniques	Early Write Termination (EWT techniques)	33% total L2 power reduction 34% energy-delay reduction
[28]	4	SRAM 32kB L1 & 256kB L2, **shared STT-MRAM 8MB L3** **baseline**: shared SRAM 8MB L3	Obstruction Aware Policy (OAP techniques)	14% performance improvement 64% total L3 power reduction
[60]	1	SRAM 32kB L1, **STT-MRAM 4MB L2** **baseline**: STT-MRAM 4MB L2 without AWARE techniques	AWARE techniques	30% average write latency reduction
[61]	4	SRAM 32kB L1, **shared STT-MRAM 4MB L2** **baseline**: shared SRAM 1MB L2	Retention time write latency/energy trade-off	12% performance improvement 60% total L2 power reduction
[62]	2	SRAM 32kB L1, **shared STT-MRAM 1MB L2** **baseline**: shared SRAM 512kB L2	Power-gating	80% leakage power reduction
[26, 44]	1	SRAM 32kB L1-D, **SOT-MRAM 32kB L1-I**, **SOT-MRAM 512kB L2** **baseline**: SRAM 32kB L1-I & L1-D, SRAM 512kB L2	Direct replacement	1% performance improvement 60% energy reduction
[63]	1	SRAM 32kB L1-I, **STT-MRAM 64kB L1-D**, SRAM 2MB L2 **baseline**: SRAM 64kB L1-D	Very Wide Buffer + optimizations	8% performance penalty

Table 3.1: MRAM-based cache: state-of-the-art review

3.3 Non-volatile memory exploration flow

3.3.1 Overview

As already mentioned, MRAM has attractive features such as low leakage, high density, and non-volatility. However, MRAM still suffers from high write latency and high write energy. To evaluate the impact of including MRAM in the memory hierarchy of processor architecture, an exploration flow based on both circuit-level and architecture-level tools is needed. A circuit-level tool needs to provide characteristics of a complete memory circuit (i.e. including data array and peripheral circuits). An architecture-level tool simulates a complete processor-based system with its memory hierarchy. For area, performance, and energy evaluations, the minimum information required is:

- Circuit-level requirements: access latency, access energy, static power, area.
- Architecture-level requirements: execution time of the simulated applications, amount of memory transactions for each level of the memory hierarchy.

Another important point is that the flow needs flexibility (i.e. extension or modifications should be possible) to make it possible to model any kind of architecture.

In this section, we propose an exploration flow based on gem5 [67], a processor architecture simulator widely used by the research community. gem5 is able to simulate a complete processor-based system with devices and operating system in full system mode (i.e. nothing is emulated). The use of gem5 makes it possible to define the total processor system architecture, including memory hierarchy specifications: cache size, cache and main memory latencies, etc. Execution time and memory transactions can be extracted for a given application, i.e. cache read/write accesses including cache hits and misses. In addition, the cache miss rate, the cache miss latency, and the memory bandwidth can be monitored over time to better understand the activity of the memory. Hence, a fine-grain analysis of performance and energy results for each simulated workload is possible.

Using gem5 is a judicious choice for processor architecture researchers for three main reasons. First, it is open source. Second, it is a community-supported tool, i.e. extension of this tool is done by gem5 users from both industry and academia, making gem5 a sustainable solution. Third, the flexibility of gem5 allows users to easily model new architectures, new cache management policies, or any new optimization techniques at architecture level. In addition, gem5 is potentially able to allow exploration of manycore architecture including more than one hundred cores applying a trace-driven approach proposed in [68].

For the rest of this section, detailed information on the gem5 simulator is given first. Then, a circuit-level model for NVMs, which is used in this thesis to explore MRAM

into cache memory, is described. Finally, the complete NVM exploration flow set up to evaluate MRAM-based cache is detailed.

3.3.2 The gem5 simulator

The gem5 simulator is the merger of the M5 [69] and GEMS [70] simulators. The objective was to provide a flexible tool focused on architectural modeling, including multiple CPU models, memory systems, and devices models. This sub-section aims at describing the simulation capabilities of gem5.

Instruction set architecture

gem5 currently supports most commercial Instruction Set Architectures (ISAs) including ARM, ALPHA, MIPS, Power, SPARC and x86.

CPU models

Four CPU models are provided by the gem5 simulator: AtomicSimple, TimingSimple, In-Order, and Out-Of-Order (O3). AtomicSimple and TimingSimple model a minimal one IPC[2] CPU. AtomicSimple is a purely functional model commonly used for fast simulation purposes and cases that do not require a detailed CPU model (e.g. cache warm-up periods, testing the functionality of a program). Unlike AtomicSimple, TimingSimple also models the timing of memory accesses.

In-order and O3 model a detailed pipelined CPU. In addition to the timing of memory accesses, they also simulates the timing of each pipeline stage. Unlike the In-Order CPU, the O3 models a out-of-order pipeline which simulates dependencies between instructions, functional units, memory accesses, and pipeline stages. Parameterizable pipeline resources such as the load/store queue and reorder buffer allow O3 to simulate superscalar architectures and CPUs with multiple hardware threads.

Memory system

gem5 features a detailed, event-driven memory system including caches, crossbars, snoop filters, and a fast and accurate DRAM controller model [71], for capturing the impact of current and emerging memories, e.g. LPDDR3/4, DDR3/4, HBM, WideIO1/2. The components can be arranged flexibly, e.g. to model complex multi-level non-uniform cache hierarchies with heterogeneous memories.

[2] Instruction per cycle

Execution modes

The gem5 simulator can operate in two modes: System-call Emulation (SE) and Full-System (FS). SE mode is used to simulate individual applications without the need to model devices or an operating system (OS). System calls are emulated by calling the host OS.

On the other hand, FS mode executes both user-level and kernel-level instructions and models a complete system including the OS and devices. This includes support for interrupts, exceptions, privilege levels, and I/O devices.

3.3.3 NVSim: a circuit-level model for NVM

NVSim [1] is a circuit-level model for NVM performance, energy, and area estimations, which supports different NVM technologies including STT-MRAM, ReRAM, and PCRAM. NVSim uses the same modeling principles as the well-known CACTI [72, 73], but starting from a new framework and adding specific features for NVM technologies. Like CACTI, NVSim also has the capability of modeling SRAM. NVSim is validated against several industry prototype chips within the error range of 30%. The main objectives of this tool is to facilitate the architecture-level NVM research by:

- Estimating the access time, access energy, and silicon area of NVM chips with a given organization and specific design options before the effort of actual fabrications.
- Exploring the NVM chip design space to find the optimized chip organization and design options that achieve best performance, energy, or area.
- Finding the optimal NVM chip organization and design options that are optimized for one design.

The rest of this sub-section gives an insight of the NVSim features.

Device model

NVSim uses data from ITRS for their device models. This tool covers the process nodes from $180nm$, $120nm$, $90nm$, $65nm$, $45nm$, $32nm$ to $22nm$ and supports three transistors types: high performance, low operating power, and low stand-by power.

Array organization

Figure 3.3 illustrates the memory array organization in NVSim. Three hierarchy levels are observed: Bank, Mat, and subarray.

Bank is the top-level structure modeled in NVSim. One non-volatile memory chip can have multiple banks. The bank is a fully-functional memory unit, and it can be operated independently. In each bank, multiple mats are connected together in either H-tree or bus-like manner.

Mat is the building block of bank. Multiple mats in a bank operate simultaneously to fulfill a memory operation. Each mat consists of multiple subarrays and one predecoder block.

Subarray is the elementary structure modeled in NVSim. Every subarray contains peripheral circuitry including row decoders, column multiplexers, and output drivers.

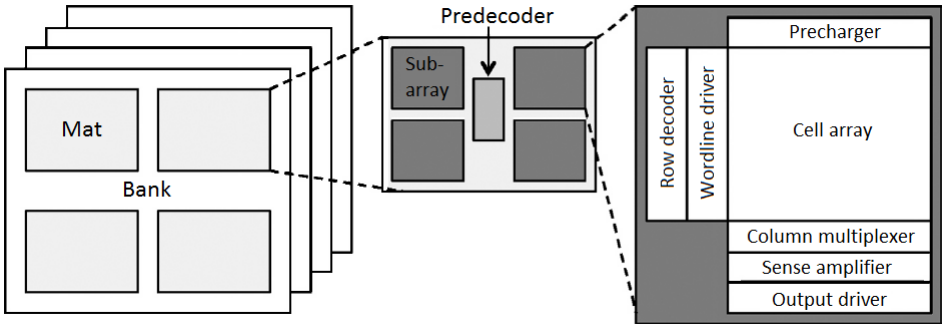


Figure 3.3: Memory array organization in NVSim

Memory bank type

NVSim models three types of memory designs: Random Access Memory (RAM), set-assiocative cache, and Content-Addressable Memory (CAM). RAM takes the address of data as input and returns the content of data. Set-associative cache contains two separate RAMs (data array and tag array), and can return the data if there is a cache hit by the given set address and tag. CAM, usually implemented in fully-associative cache, output the data address at the I/O interface given the data content as input.

For the set-associative cache, three different access manners are models:

- **Normal access**: start to access the cache data array and tag array at the same time; the data content is temporarily buffered in each mat; if there is a hit, the cache hit signal generated from the tag array is routed to the proper mats and the content of the desired cache line is output to the I/O interface.
- **Sequential access**: access the cache tag array first; if there is a hit, then access the cache data array with the set address and the tag hit information, and finally output the desired cache line to the I/O interface.

- **Fast access**: access the cache data array and tag array simultaneously; read the entire set content from the mats to the I/O interface; selectively output the desired cache line if there is a cache hit signal generated from the tag array.

3.3.4 Exploration flow

In this sub-section, we provide details on the NVM exploration flow illustrated in Figure 3.4. Evaluating performance/energy/area of NVM-based memory hierarchy can be divided into 5 steps:

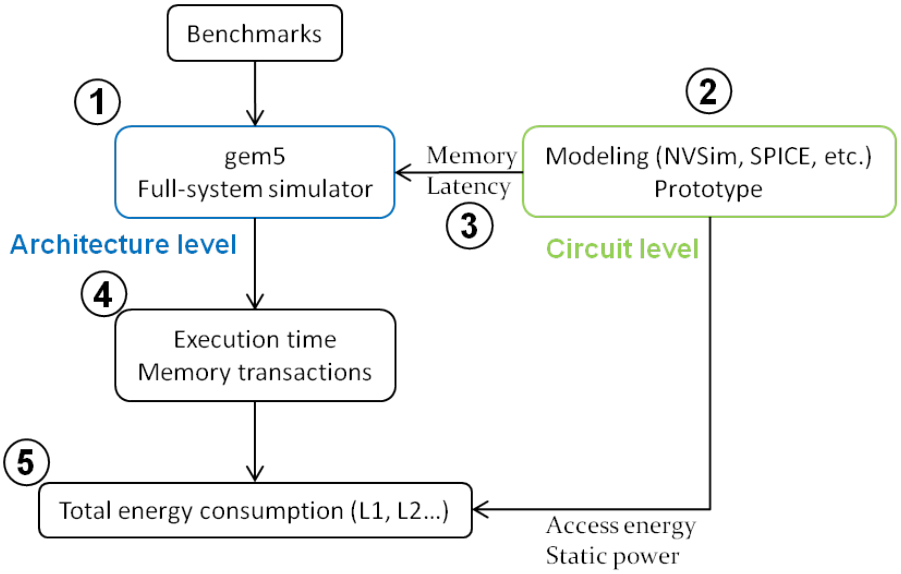


Figure 3.4: NVM exploration flow

1. Defining the architecture: Single or multi-core, ISA choice (e.g. ARM, x86), defining the memory hierarchy (e.g. level of cache, memory size).
2. Obtaining memory characteristics at circuit level (Area analysis possible at this step): access latency, access energy, static power, area.
3. Calibrating each level of the memory hierarchy with the access latencies obtained in step 2
4. Extracting the outputs of the gem5 simulation (Performance analysis possible at this step)
5. Calculating the energy consumption of each level of the memory hierarchy using the memory characteristics obtained in step 2 and the amount of transactions obtained in step 4. (Energy analysis possible at this step)

To calibrate the memory hierarchy in terms of access latency (step 3), data from a circuit-level models such as NVSim or outcomes from a real prototype can be used. NVSim should be used for a rapid estimation of electrical features of a complete memory chip including read/write access time, read/write access energy, and static power. For more precise evaluations, the results from SPICE simulation of a full design or the electrical features of a real prototype are more appropriate.

In step 5, the total cache energy consumption is calculated as shown in the equation 3.1. P_{static} is the leakage power of the cache, N_r (N_w) is the number of reads (writes), and E_r (E_w) is the energy per access for a read (write) operation.

$$\begin{aligned} E_{total} &= E_{static} + E_{dynamic} \\ &= P_{static}.Runtime + N_r.E_r + N_w.E_w \end{aligned} \tag{3.1}$$

3.4 MRAM-based cache: circuit-level analysis

In this section, we provide a comparative analysis of the performance/energy/area of MRAM-based and SRAM-based caches at circuit level (i.e. a direct comparison of the memory characteristics). STT-MRAM-based, TAS-MRAM-based and SRAM-based caches are analyzed. Table 3.2 shows the cache parameters (latency, energy per access, static power, area) of a $512kB$ L2 for the three memory technologies concerned. Table 3.3 shows the same parameters for a $32kB$ L1. The sizes of the L1 and L2 caches are also used for the architecture-level analysis in Section 3.5. Note that results of both SRAM and STT-MRAM come from NVSim, while for TAS-MRAM, outcomes from a real prototype we used thanks to support provided by Crocus Technology. Because TAS-MRAM is not suitable for use in L1 (because of its slow access time), we only evaluated this technology for the L2 cache. To take into account the state-of-the-art of MRAM technology and to evaluate performance and energy fairly, we compared $45nm$ STT-MRAM-based cache results with a baseline $45nm$ SRAM-based cache, and $130nm$ TAS-MRAM-based cache results with a baseline $120nm$ SRAM-based cache.

3.4.1 Analysis of $512kB$ L2 cache

As expected, both MRAM technologies have higher write latency than SRAM. Regarding read latency, STT-MRAM is faster than SRAM (45 nm). This is due to a smaller load capacitance for STT-MRAM as it is denser than SRAM. Hence, when the cache capacity increases, SRAM read latency increases much more than that of STT-MRAM [44]. Concerning the difference in cache area between the two technologies, a SRAM bit cell consists of six CMOS transistors whereas a STT-MRAM bit cell consists of one CMOS

transistor and a MTJ. As a result, for the same capacity, the area of the cell array for STT-MRAM is smaller than for SRAM resulting in smaller total L2 cache area (Table 3.2). These two advantages of STT-MRAM in terms of read latency and cache area are only noticeable in the case of large cache capacity, when the area of the cell array occupies a large proportion of the total cache area compared to the area occupied by the peripheral circuitry.

	Latency		**Energy**			**Cache area**	
Technology	**Read** (ns)	**Write** (ns)	**Read** (nJ)	**Write** (nJ)	**Leakage** (mW)	**Total** (mm^2)	**Cell** (F^2)
$45nm$ SRAM	4.28	2.87	0.27	0.02	320	1.36	146
$45nm$ STT	2.61	6.25	0.28	0.05	23	0.82	57
$120nm$ SRAM	5.95	4.14	1.05	0.08	82	9.7	146
$130nm$ TAS	35	35	1.96	4.62	10	11.7	35

Table 3.2: $512kB$ L2 cache features

TAS-MRAM write and read latencies are respectively 8.5 and 6 times higher than those of SRAM ($120nm$). As explained in Section 2.3.3, the long write latency of TAS-MRAM in mainly due the heating/cooling stages required to switch the MTJ. The total cache area of TAS-MRAM is slightly larger than that of SRAM ($120nm$), for which there are two explanations. First, contrary to STT-MRAM, the TAS-MRAM bit cell is written by a magnetic field generated by a current flowing through a conductive line. Hence, conductive lines have to be added to TAS-MRAM-based memory. Second, as explained in Section 2.3.3, TAS-MRAM requires a high write current (higher than STT-MRAM). As a result, the write drivers have to be large enough to generate sufficient current for a write operation.

Regarding L2 energy consumption, using MRAM instead of SRAM results in higher write energy for both STT-MRAM and TAS-MRAM. STT-MRAM read energy is very similar to that of SRAM, whereas a TAS-MRAM read consumes around $2\times$ more energy than SRAM. However, in terms of leakage power, MRAM has a considerable advantage over SRAM: a 45 nm STT-MRAM-based L2 consumes over one order of magnitude less power than $45nm$ SRAM-based L2, while a TAS-MRAM-based L2 consumes around $8\times$ less power than a $120nm$ SRAM-based L2. This is because most of the static power of large capacity memories comes from cell arrays. Since MRAM cell has zero standby power and the CMOS access transistor does not require a power supply (to retain data), all the static power in MRAM-based memory is due to peripheral circuitry such as address decoding, drivers, and sense amplifiers.

3.4.2 Analysis of $32kB$ L1 cache

As shown in Table 3.3, STT-MRAM and SRAM read latencies are similar. But a higher latency is still observed for STT-MRAM writes. STT-MRAM consumes around $4\times$ and $7\times$ more energy than SRAM for read and write operations, respectively. A significant gain in static power is obtained by replacing SRAM with STT-MRAM due to the zero leakage of the MTJ. The total cache area of a STT-MRAM-based L1 is slightly larger than that of a SRAM-based L1 cache. This is because the capacity of the cache is small ($32kB$), and so, the area occupied by the peripheral circuits is not negligible compared to the area of cell array. Since MTJ writes need a large amount of current, the transistors of the write circuitry for STT-MRAM have to be large enough to generate sufficient write current. As a result, the peripheral circuits for STT-MRAM-based cache occupy more area than their SRAM equivalents.

	Latency		**Energy**			**Cache area**	
Technology	**Read**	**Write**	**Read**	**Write**	**Leakage**	**Total**	**Cell**
	(ns)	(ns)	(nJ)	(nJ)	(mW)	(mm^2)	(F^2)
$45nm$ SRAM	1.25	1.05	0.024	0.006	22	0.091	146
$45nm$ STT	1.94	5.94	0.095	0.04	3.3	0.117	57

Table 3.3: $32kB$ L1 cache features

3.4.3 Summary

The comparison of MRAM and SRAM at the circuit level showed that the high density of MRAM may be advantageous in terms of read latency for large cache capacity. In addition, considerable static power can be saved using MRAM instead of SRAM. On the other hand, our results clearly reveal the disadvantage of using MRAM in terms of write latency and write energy compared to its SRAM equivalents. The difference between the latency parameter in SRAM and MRAM will of course depend on the frequency used by the processor. In this study, the frequency used for the processor was $1GHz$. Table 3.4 shows the access latencies in terms of CPU cycle for L1 and L2. Since TAS-MRAM was evaluated only for L2 cache, L1 latencies for TAS-MRAM and the baseline $120nm$ SRAM are not shown.

3.5 MRAM-based cache: architecture-level analysis

3.5.1 Experimental setup

A few workloads of SPLASH-2 [74] and PARSEC [75] benchmark suites were used to explore STT-MRAM and TAS-MRAM based caches for a quad-core processor ARM architecture. While SPLASH-2 workloads are mostly focused on high performance computing (HPC), PARSEC includes emerging workloads in many different areas such as computer vision, financial analytics, data mining, animation physics, image processing and video encoding. According to [76], PARSEC includes workloads that handle a huge amount of data compared to SPLASH-2. Figure 3.5 and Table 3.4 show the architecture layout and configuration we used for simulation. Table 3.5 provides details on the simulated workloads.

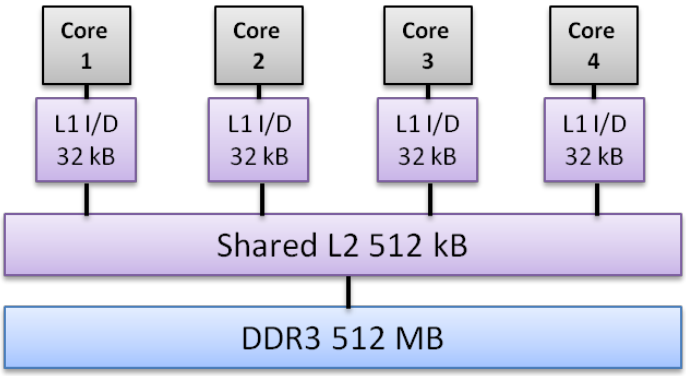


Figure 3.5: Quad-core architecture layout

Hierarchy level	Configuration
Processor	4-core, $1GHz$, 32-bit RISC ARMv7 (Linux OS)
L1 I/D cache	Private, $32kB$, 4-way associative, $64B$ cache line
	$45nm$ SRAM (read: 2, write: 2)
	$45nm$ STT (read: 2, write: 6)
L2 cache	Shared, $512kB$, 8-way associative, $64B$ cache line
	$45nm$ SRAM (read: 5, write: 3)
	$45nm$ STT (read: 3, write: 7)
	$120nm$ SRAM (read: 6, write: 5)
	$130nm$ TAS (read: 35, write: 35)
Main Memory	DRAM, 512MB, DDR3, 100-cycle latency

Table 3.4: Architecture configuration

Splash-2 benchmark	**Input set**
barnes	16K Particles, Timestep = 0.25, Tolerance 1.0
fmm	16K Particles, Timestep = 5
ft	2^{20} total complex data points
lu1	Contiguous blocks, 512x512 Matrix, Block = 16
lu2	Non-contiguous blocks, 512x512 Matrix, Block = 16
ocean1	Contiguous partitions, 514x514 Grid
ocean2	Non-contiguous partitions, 258x258 Grid
radix	4M Keys, Radix = 4K
Parsec benchmark	**Input set**
blacksholes	4,096 options
bodytrack	4 cameras, 1,000 particles, 5 layers, 1 frame
ferret	3,544 images, 16 queries
fluidanimate	35,000 particles, 5 frames
streamcluster	4,096 points per block, 32 dimensions, 1 block
x264	640 x 360 pixels, 8 frames

Table 3.5: Benchmarks

gem5 modifications

By default, gem5 assumes the same latency for reading and writing. Therefore, the original version of the simulator is not adapted to model MRAM-based memories which have asymmetric read/write latencies. The NVM exploration flow presented in Section 3.3 actually use a modified version of gem5 which is able to simulate cache memory with different read/write latencies.

Originally, the configuration files of the cache in gem5 only provide one parameter for the hit latency which is applied for both read and write accesses. First, we added an additional parameter for the hit latency. Thus, the modified version of gem5 provides two cache parameters for the hit latency, one for read, and one for write.

Then the source code that models the cache behavior has been adapted to consider the actual hit latency depending on if a read or a write is performed. We essentially modified two functions. The first function models the access of the cache from upper levels in the memory hierarchy (i.e. from the CPU side), the other function models the access of the cache from lower levels (i.e. from the memory side).

Different read/write latency feature has been validated by checking the time between a read/write request sent by the CPU and the response sent by the cache memory.

3.5.2 Analysis of the cache memory activity

Overview

Here we provide prior study on the workload behavior in terms of memory activity. Information including the read/write ratio, static/dynamic energy ratio, L1/L2 access ratio, the cache miss rate and the cache bandwidth are analyzed. The results are shown for the baseline architecture (SRAM-based cache). The aim of this preliminary study is to obtain useful information for a more comprehensive analysis of subsequent results concerning the impacts of incorporating MRAM into cache memory on performance and energy.

Read/Write ratio

Since access time and energy requirements for reading and writing differ considerably in MRAM, it is important to analyze the read/write ratio of the workloads. As explained in Sections 1 and 2, a MRAM write operation consumes more energy and is slower than its SRAM equivalent. Figures 3.6 and 3.7 depict the read/write ratio for L2 and L1 caches, respectively.

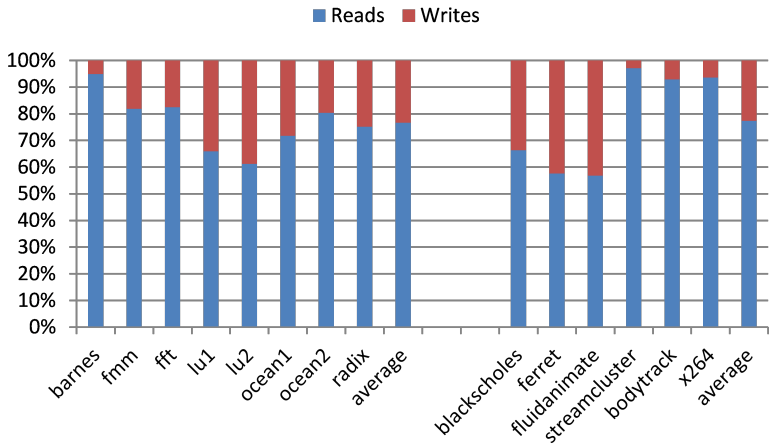


Figure 3.6: L2 read/write ratio

Our results showed that majority of the accesses are reads for both L1 and L2 caches. In L2, three workloads (lu1, lu2, blackscholes) have more than 30% writes, and two workloads (ferret, fluidanimate) have more than 40% writes. The number of L1 writes is small compared to the number of reads. Note that I-Cache is read only, I-Cache writes are consequently equal to 0.

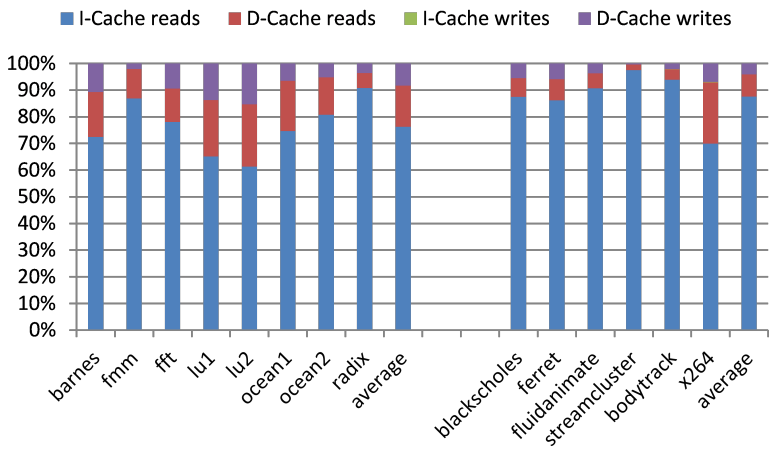


Figure 3.7: L1 read/write ratio

Static/Dynamic energy ratio

The static/dynamic energy ratio in memory is helpful when studying different memory technologies. As mentioned above, the main challenge for existing and future ICs is energy efficiency. Due to the high leakage current of SRAM, a memory device like MRAM with ultra-low leakage is very attractive. Our results showed that more than 80% of the total energy consumption in the L1 cache is static, and more than 90% in the L2 cache. In Sections 3.5.3 and 3.5.4, we analyze the results concerning the MRAM-based cache to see whether the notable gain in static power consumption in MRAM compensates for the high dynamic energy loss of MRAM in both L1 and L2.

L1/L2 access ratio

The L1/L2 access ratio provides a rough idea of the impact of the L2 (and L1) cache on performance (i.e. execution time). For instance, if there is a big difference between the number of L1 accesses and the number of L2 accesses, L2 will have little impact on the execution time. Table 3.6 lists the number of accesses in L1 and L2 caches. Note that each value is the average of all the workloads.

Benchmark	Number of accesses	
	L1	L2
Splash-2	2 billion (0.5 billion/CPU	26 million
Parsec	12 billion (3 billion/CPU)	16 million

Table 3.6: L1/L2 access ratio

For both benchmarks (SPLASH-2 & PARSEC), L1 is much more accessed than L2. The L1/L2 ratio for PARSEC is clearly higher than for SPLASH-2. As a result, if MRAM is used in L2 when executing PARSEC workloads, the drop in performance due to the long write latency of MRAM will be less visible.

Cache miss rate & cache bandwidth

Other interesting results concerning memory activity are related to the cache miss rate and the cache bandwidth. The cache miss rate reveals the number of times data is not found in the cache, thus requiring access to a lower level of the memory hierarchy to update (write) the corresponding cache block. A high cache miss rate can have a negative effect on performance due to the long write latency of MRAM. This is the case if the cache bandwidth is also high. The cache bandwidth is the number of bytes accessed per second. Therefore, if both cache miss rate and cache bandwidth are high, using MRAM will likely have a negative effect on performance. The L2 cache miss rate and bandwidth for each workload of SPLASH-2 are listed in Figures 3.8 and 3.9 respectively.

Regarding the L1 cache, STT-MRAM and SRAM have the same read access time in terms of CPU cycles (Table 3.4). On the other hand, STT-MRAM has three times slower write access time than SRAM. In this case, the reduction in performance of STT-MRAM-based L1 will obviously come from write access. To better analyze the following results concerning the STT-MRAM-based L1, the L1-Data cache write bandwidth is shown in Figure 3.10, for each SPLASH-2 workload (L1-Instruction cache is read only). The write bandwidth informs us about the number of bytes written per second. Consequently, the biggest reduction in performance using STT-MRAM instead of SRAM would be expected for workloads with the highest write bandwidth. Although the same results as in Figures

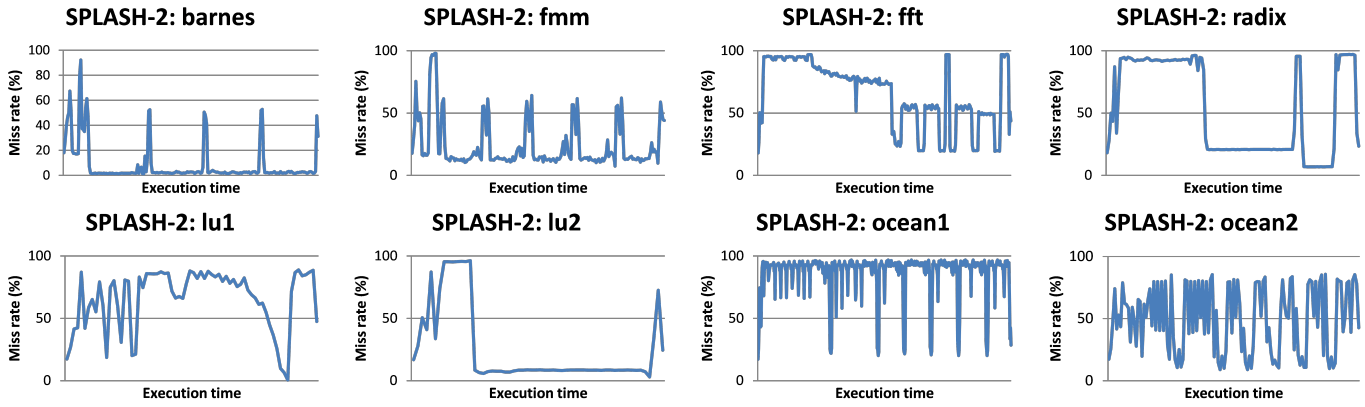


Figure 3.8: L2 cache miss rate

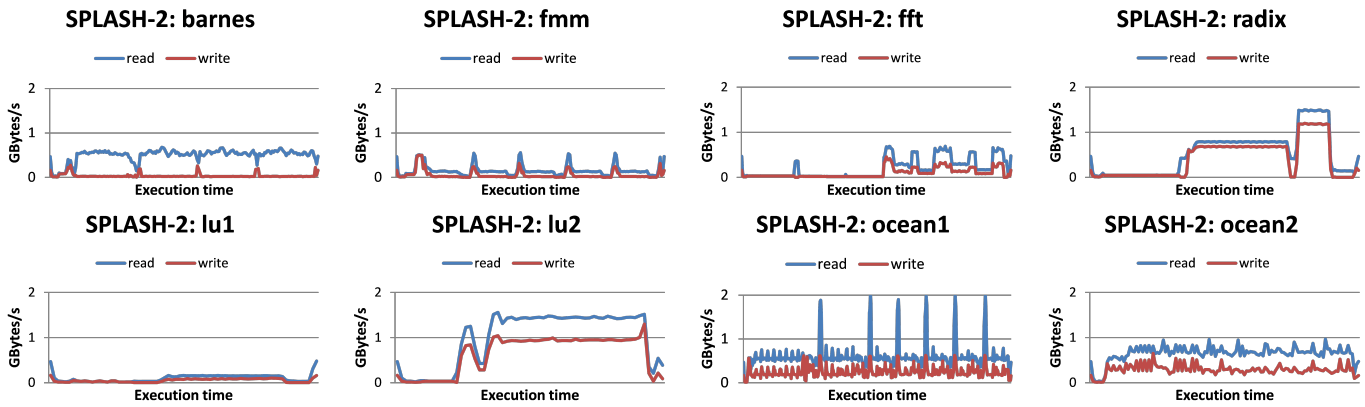


Figure 3.9: L2 cache bandwidth

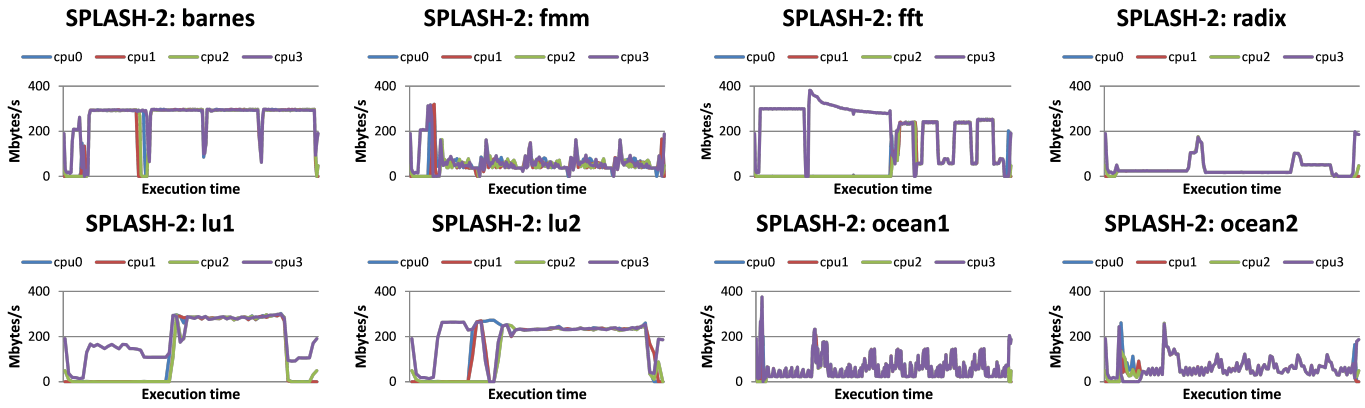


Figure 3.10: L1 data cache write bandwidth

3.8, 3.9 and 3.10 are available for the PARSEC workloads, we do not show them here for the sake of brevity.

3.5.3 Exploration of the L2 cache

Performance evaluation

Figure 3.11 shows the execution time of SPLASH-2 and PARSEC workloads for both STT-MRAM and TAS-MRAM based L2 caches. Figure 3.11 shows that the performance of STT-MRAM- based L2 scenario is similar and sometimes better (ocean1, ocean2) than the baseline. This is because STT-MRAM has a smaller read latency than its SRAM equivalent and also to the fact that, as mentioned in Section 3.5.2, L2 is more accessed in read.

TAS-MRAM-based L2 performance penalties of 14% and 2% on average were observed for SPLASH-2 and PARSEC workloads respectively. In the case of ocean2, 38%

of performance degradation was observed using TAS-MRAM. The cache memory traffic analysis (Section 3.5.2) showed that the difference in the number of L1 and L2 accesses was larger for PARSEC than for SPLASH-2. Hence, L2 has less impact on the execution time for PARSEC than for SPLASH-2 explaining why the decline in performance is very small for PARSEC workloads, even using TAS-MRAM.

The highest penalties in execution time using TAS-MRAM were observed for ocean1, ocean2 and radix workloads. This is understandable when the cache miss rate and cache bandwidth in Figures 3.8 and 3.9 are analyzed: all three workloads have both a high cache miss rate and a high cache bandwidth compared with other SPLASH-2 workloads. As explained in Section 3.5.2 above, for memory activity, this kind of behavior does not favor MRAM due to its long write latency.

Energy evaluation

Figure 3.12 shows total L2 energy consumption (including dynamic and static energy). Simulation results showed using STT-MRAM is 92% more energy efficient on average than SRAM for both SPLASH-2 and PARSEC workloads. Using TAS-MRAM, 63% and 84% energy gains on average were observed for SPLASH-2 and PARSEC respectively. This notable difference in energy consumption between the two technologies is explained by the low leakage power of MRAM compared to SRAM. As we noticed during the analysis of the cache memory activity (Section 3.5.2), using SRAM, more than 90% of the total L2 energy consumption is static when using SRAM. As a result, replacing SRAM with MRAM can dramatically reduce the total energy consumption in L2. This makes MRAM-based cache memory an attractive alternative for energy efficient systems because despite

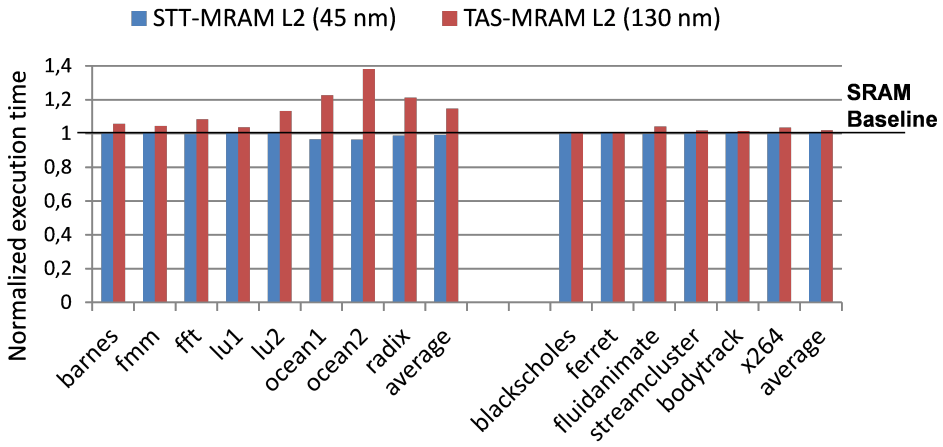


Figure 3.11: Execution time with MRAM-based L2 cache

the reduction in energy consumption, the performance remains reasonable.

Figure 3.12 shows that lu2, ocean1, ocean2 and radix are the workloads with the lowest energy gain for TAS-MRAM-based L2. This makes sense given the previous results on the cache bandwidth in Figure 3.9, showing that L2 in these four workloads, L2 is more frequently accessed than the others. As seen in Table 3.2, TAS-MRAM consumes more energy than SRAM, not only for writes, but also for reads in the L2 cache. As a result, less energy is gained for these four workloads because of the loss in dynamic energy when SRAM is replaced by TAS-MRAM.

3.5.4 Exploration of the L1 cache

Performance evaluation

It will be recalled that we only evaluated STT-MRAM in L1 because it is currently the most competitive option with SRAM. Figure 3.13 shows the execution time with different cache configurations: a scenario in which both the Instruction- Cache (I-Cache) and Data-Cache (D-Cache) are based on STT-MRAM, a scenario with STT-MRAM-based I-Cache only, and a scenario with STT-MRAM-based D-Cache only.

As already observed in Table 3.4, the read latency is the same (in terms of CPU cycles) with both technologies. Therefore, STT-MRAM is slower than SRAM only in write operations.Since I-Cache is read only, replacing SRAM by STT-MRAM only in the I-Cache does not affect performance. Penalties are observed only in the other scenarios in which the D-Cache is based on STT-MRAM. For some workloads (barnes, fft, lu1, lu2), using STT-MRAM in a D-cache reduces overall performance by around 20% due to its high write latency. For others, such as streamcluster, the execution time penalty is very

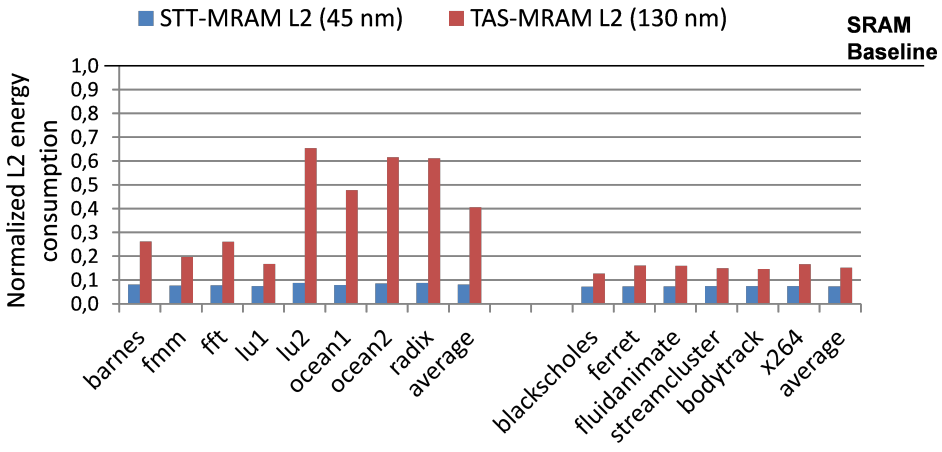


Figure 3.12: MRAM-based L2 energy consumption

small, even with a STT-MRAM-based D-Cache. This can be explained by analyzing the cache write bandwidth of DCache, previously shown in Figure 3.10, which clearly shows that the D-Cache is more frequently accessed in write for barnes, fft, lu1 and lu2 workloads than for other SPLASH-2 workloads. Hence, the impact of the long write latency of STT-MRAM on the execution time is more visible for these four workloads. Overall, for all the workloads, the simulation results show that the execution time penalty of STT-MRAM-based D-Cache does not exceed 21%, since the L1 cache is much more accessed in read for the simulated workloads (Figure 3.7).

Energy evaluation

Figure 3.14 shows total L1 energy consumption (including for the L1 cache of each core). Replacing SRAM with STT-MRAM in L1 does not gain as much energy as with the L2 cache. The reason is the L1 cache is much more accessed than the L2 cache, as already shown in Table 3.6. As a result, the dynamic energy impact of STT-MRAM is more visible. It will be recalled that previous circuit-level analysis showed that STT-MRAM consumes respectively around 4 times and 7 times more energy than SRAM for read and write operations in L1 (Table 3.3). Concerning SPLASH-2 workloads, Figure 3.14 shows average energy gains of 38% for STT-MRAM in both I-Cache and D-Cache, of 9% for STT-MRAM in I-Cache only, and of 24% for STT-MRAM in D-Cache only. Concerning PARSEC workloads, an average energy loss of 10% is observed for STT-MRAM in I-Cache only. For the other cache configurations, average energy gains of 26% are observed for STT-MRAM in D-Cache only, and of 19% for STT-MRAM in both I-Cache and D-Cache.

The cache memory traffic analysis showed that the I-Cache is the most accessed mem-

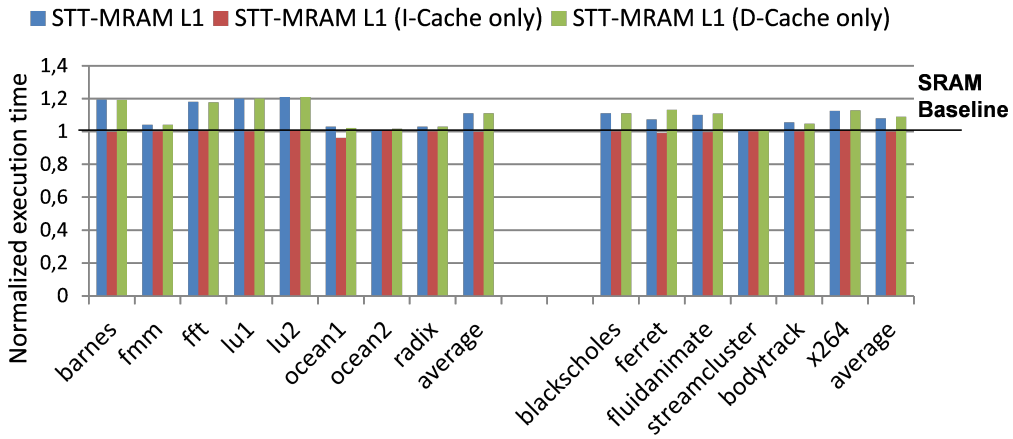


Figure 3.13: Execution time with MRAM-based L1 cache

ory in L1 (Figure 3.7). I-Cache accesses represent 76% and 87% of total L1 accesses for SPLASH-2 and PARSEC workloads, respectively. When STT-MRAM is only used in I-Cache, an increase in total L1 energy consumption is observed for two workloads of SPLASH-2 (barnes, fmm) and for five workloads of PARSEC (blacksholes, ferret, fluidanimate, streamcluster bodytrack). This is because for PARSEC workloads, the total number of accesses in L1 is higher than for SPLASH-2, as seen previously in Table 3.6. Hence, for these workloads, low leakage of STT-MRAM does not compensate for its high dynamic energy loss over SRAM in I-Cache. However, when STT-MRAM is used in both I-Cache and D-Cache, energy is gained for all the workloads because the low leakage of STT-MRAM applies to both the I-Cache and D-Cache. Concerning PARSEC, the best cache configuration is the STT-MRAM-based D-Cache only, thanks to the low leakage of STT-MRAM in D-Cache and the low dynamic energy of SRAM in I-Cache, which is intensively accessed for PARSEC workloads.

3.5.5 Exploration for different number of cores

This section aims at analyzing the energy impact of MRAM-based cache when the number of cores is changed: quad-core, dual-core, and single-core. The results shown in this section are the average L1/L2 energy consumption over all the simulated workloads. Performance analysis is not detailed because simulation results reveal that the execution time penalty of MRAM-based cache (compared to the baseline) does not change significantly when increasing the number of cores from one to four. It will be recalled that the L2 is shared for multi-core architectures.

Figures 3.15 and 3.16 depict respectively the total L2 energy consumption and the to-

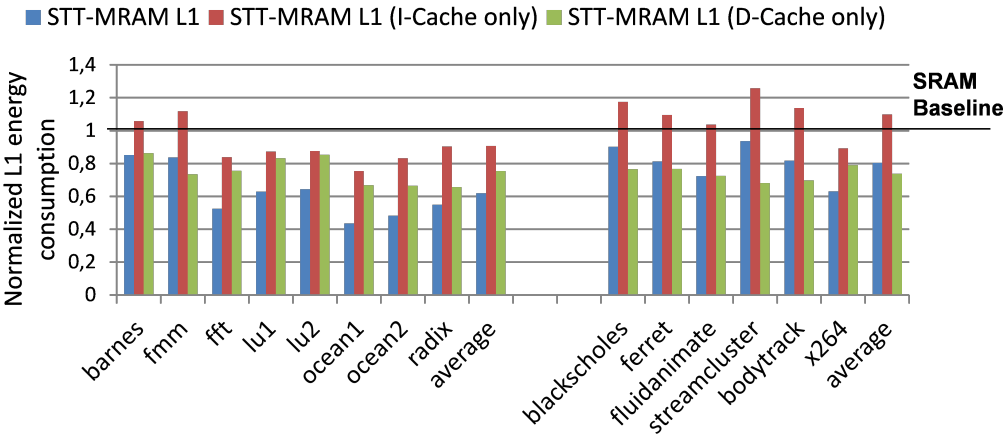


Figure 3.14: MRAM-based L1 energy consumption

tal L1 energy consumption for 4-core, 2-core and 1-core processor architectures. Substantial variations are noticed when the number of cores is changed. Regarding L2 cache in Figure 3.15, changing from 4-core to 1-core architecture increases the energy consumption gain by 14% when using TAS-MRAM instead of SRAM. On the other hand, no significant change is noticed with STT-MRAM-based L2 when the number of cores is changed.

Figure 3.16 shows that when SRAM is replaced by STT-MRAM in L1, changing from 4-core to 1-core architecture increases the energy consumption gain by 18% for STT-MRAM-based I-Cache only, by 4% for STT-MRAM-based D-Cache only, and by 24% for STT-MRAM in both I-Cache and D-Cache. To better understand this trend, i.e. the energy consumption gain over SRAM-based cache increases when the number of cores is reduced, the cache bandwidth is monitored over time (Figure 3.17) for 4-core, 2-core and 1-core architecture. The analysis is shown only for the L2 and only for one workload, since analysis for L1 and for other workloads result in the same conclusions.

Figure 3.17 shows that the L2 bandwidth is reduced by around 2 when the number of cores is decreased by 2. Thus, the dynamic part of the total L2 energy consumption is lower for 1-core than for 4-core architecture. Consequently, the loss due to the high dynamic energy of MRAM is reduced. This explains the higher energy gain using MRAM instead of SRAM in L2 for 1-core than for 2-core or 4-core architecture (Figure 3.15). This is particularly visible for TAS-MRAM because it has higher dynamic energy than SRAM for both read and write, whereas for STT-MRAM, this is the case only for writes.

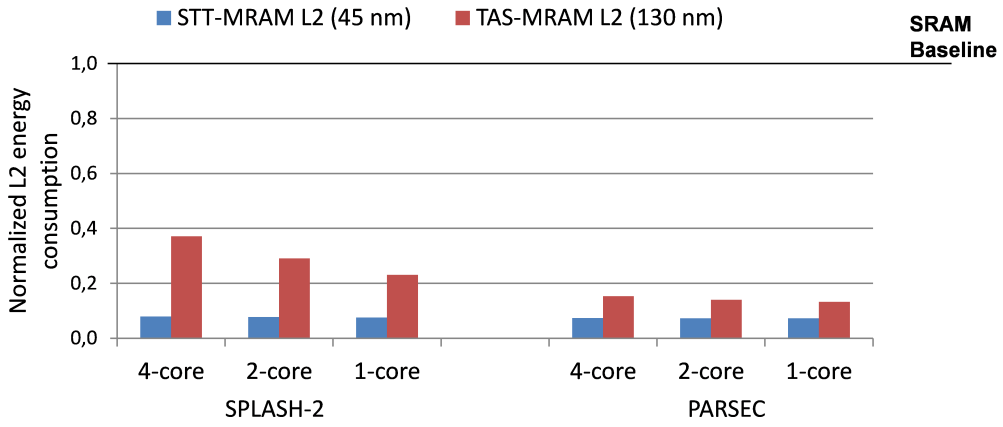


Figure 3.15: MRAM-based L2 energy consumption for different number of cores

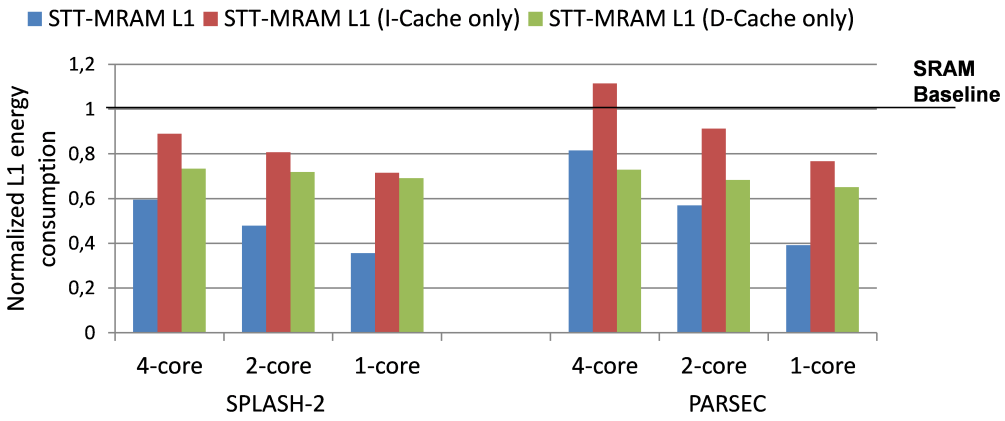


Figure 3.16: MRAM-based L1 energy consumption for different number of cores

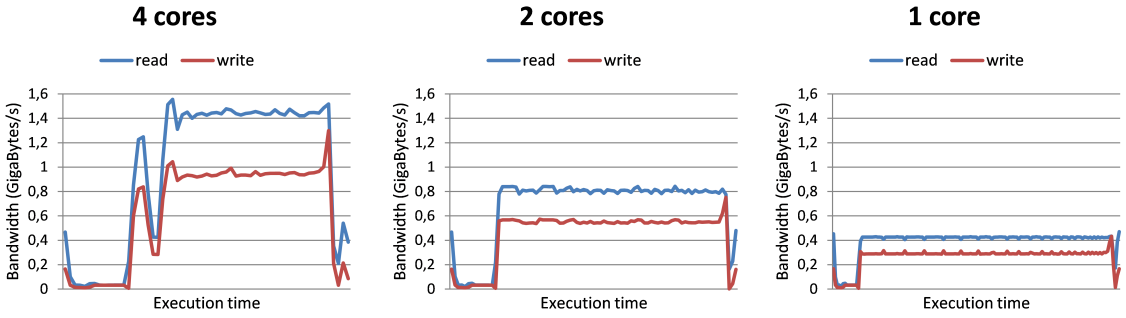


Figure 3.17: L2 bandwidth for different number of cores (lu2 workload)

3.6 Conclusion

This chapter analyzed the replacement of SRAM by TAS-MRAM and STT-MRAM respectively in L2 cache and L2/L1 caches, and evaluated the impact according to the three metrics speed/energy/area. Thanks to architecture-level and circuit-level tools, a fine-grain exploration flow has been set up considering the memory traffic of cache, e.g. the cache miss rate and the cache bandwidth, for each simulated application. The main observations of the study reported in this chapter are summarized below.

3.6.1 Area

Because of its small bit cell size compared to SRAM, MRAM allows to get higher density for the cell array of the cache memory. On the other hand, considering the peripheral circuitry, smaller area is noticed when using SRAM instead of MRAM as the latter needs large transistors for the write circuitry to generate sufficient current to write into

the MTJs. As a result, achieving smaller area by using MRAM-based cache depends on the area ratio between the cell array and the peripheral circuits. If the cell array area represents a large proportion of the cache memory, and the peripheral circuitry area is small enough compared to that of the cell array, MRAM-based cache will be denser than SRAM-base cache.

3.6.2 Speed

Regarding performance, STT-MRAM shows a lower read latency than its SRAM equivalent in L2 cache due to a smaller load capacitance for STT-MRAM. Furthermore, for the simulated workloads, L2 is much more accessed in read. As a result, despite the high write latency of STT-MRAM, very small or no penalty in execution time is observed when SRAM is replaced by STT-MRAM in L2 cache (LLC).

For TAS-MRAM, the penalty in execution time depends on the memory traffic (e.g. cache miss rate). Considering the number of accesses in L2 compared to L1 (L1/L2 access ratio), if L2 has a small impact on the total execution time of the workload, TAS-MRAM-based L2 is better than SRAM-based L2 in terms of a trade-off between performance and energy consumption.

For L1 cache, STT-MRAM is suitable for I-Cache if the read latency in terms of CPU cycles is similar to that of its SRAM equivalent, which will depend on the CPU frequency used. The long write latency of STT-MRAM reduces performance in the case of D-Cache. However, this reduction in performance can be mitigated for read intensive workloads, since STT-MRAM read latency is almost similar to that of SRAM.

3.6.3 Energy

Concerning energy, the low leakage of MRAM is extremely beneficial for lower levels of cache in the memory hierarchy, since leakage accounts for an important part of the total energy consumption of SRAM cache, as seen in Section 3.5.2. For L1 cache, because of its high number of accesses compared to L2, the energy gain due to the low leakage of STT-MRAM is significantly reduced and sometimes non-existent because of the high dynamic energy of STT-MRAM compared to SRAM. As a result, current STT-MRAM characteristics do not allow the direct replacement of SRAM by STT-MRAM in L1. Further improvements are needed to the MTJ device to achieve the same performance as SRAM. Otherwise, cache optimization techniques and/or hybrid L1 cache design using both MRAM and SRAM have to be used to mitigate the two drawbacks of MRAM: high write latency and high write energy, which were discussed in Section 4.2.

4

NON-VOLATILE MRAM-BASED EMBEDDED PROCESSOR

4.1 Introduction

The interest of promoting smart systems thanks to interconnected objects is growing fast to build up what is known as internet of things (IoT) [77]. It is assumed that 50 billion objects will be connected in 2020 [78]. IoT devices essentially do three actions: sense, process, and send. First, the object captures information from the external environment. It can be for instance movement, temperature, sound, location (via GPS), or heart rate. Second, the device stores the measured data and does some minor computation. Finally, it sends data to a data center, wirelessly. As a result, IoT objects have to comprise three main components: sensor, processing unit and communication unit.

These kinds of devices are typically battery powered. Moreover, they have to be operational for a very long duration (several months or even years). Therefore, the energy consumption is definitely a critical constraint. Generally, an IoT object is most of the time inactive, staying in low-power mode (sleep mode) and waiting for the next work to achieve. This can be periodic or dependent on external events which will wake up the device. As a result, sleep mode power will consume the largest amount of energy and battery life. Today, several Microcontrollers (MCUs) targeting low-power applications are available in the market. Commercial MCUs actually implement not only one but several power-down modes with different wake-up times, depending on from which low-power mode the MCU returns to the active mode. Thus, depending on the applications, system designers have to make a tradeoff considering the power consumption in both active and sleep modes, the wake-up time, and the ratio of time spent in active/sleep modes to ensure energy efficiency.

Within this context of energy efficiency, emerging NVM technologies have raised a new interesting computing paradigm known as *normally-off computing* [79]. This is the ability for systems to be normally powered off (there is no operation to do), momentarily

powered on (there is a need to operate). Currently, working memories of MCUs, such as registers and RAM, use volatile CMOS transistors to retain information. If the system is powered down, these kinds of memories lose data and a pretty long time (up to a few milliseconds [80]) is necessary to restore information after a new power-up. Integrating the non-volatility feature will allow systems to keep data available even after a power-down, thus significantly reducing the wake-up time. For battery-powered devices such as IoT objects, it will give the possibility to go into sleep mode more frequently, with zero leakage power since no power is required to retain the state of the system.

This chapter explores the opportunity of having a non-volatile processor by the integration of MRAM at register and main memory levels, considering the open-source Amber processor core [81], a 32-bit RISC[1] embedded processor. Two capabilities introduced by the non-volatility are investigated. First, the *instant on/off* which allows a recovery of the state of the processor after a power-down. Second, the *rollback* giving the possibility to restore a previous valid state of the processor, for instance in the case of an execution error. The cost in terms of performance and energy to save/restore the state of the processor is estimated.

The rest of the chapter is organized as follows: Section 4.2 reviews state-of-the-art on non-volatile logic circuits. Section 4.3 first describes both *instant on/off* and *rollback* capabilities, then shows simulation results of a complete backup/recovery of the state of the Amber core. Section 4.4 gives performance and energy estimations of the backup/recovery phases, and it discusses the performance and energy implications of integrating MRAM into a processor for both active and sleep modes. Section 4.6 concludes this chapter.

4.2 State-of-the-art review

Emerging NVM technologies have attracted a large part of the research community on the study of non-volatile logic circuits. Use of these memories for data storage and logic devices were and continue to be investigated due to the high interest it arouses for nanoelectronic systems such as Field-Programmable Gate Arrays (FPGAs) and processors. Thanks to their CMOS-compatibility, emerging NVM technologies allow design of hybrid CMOS/NVM logic elements capable of retaining their current state even after a power-down of the system. This section aims at giving an overview of the studies that have been done on non-volatile logic circuits.

[1] Reduced Instruction Set Computer

4.2.1 Non-volatile logic elements

Many studies explored the feasibility of designing non-volatile logic elements such as FFs and made a comparison with the counterpart CMOS-based circuits. [82] studied STT-MRAM-based FFs evaluating two different structures (merged latch and sensing circuit (MLS), separated latch and sensing circuit (SLS)) and several sensing and write circuits for the non-volatile part. [83] investigated non-volatile logic gates and possible design optimizations using compact models of STT-MRAM and Oxyde-based Resistive Random Access Memory (OxRAM) NVM technologies. In addition, a non-volatile full-adder based on these two NVM technologies has been validated by simulation in [84] and compared with its counterparts based on CMOS only. [85] evaluated a non-volatile FF based on the recent SOT-MRAM technology and compared it with a STT-MRAM-based FF. Within the context of energy-harvesting and IoT applications, [86] and [87] proposed a non-volatile FF respectively based on Ferroelectric Random Access Memory (FeRAM) and OxRAM and validated by simulation the possibility to save/restore the logic state after a power-off of the device. [88], whose results have been used in this work for TAS-MRAM-based FFs, have made an exhaustive performance/energy analysis of a set of hybrid CMOS/MTJ cells which can be used for both data storage and logic devices in SoC. [89], [87], and [90], whose results have also been used in this chapter, proposed respectively a hybrid CMOS/STT-MRAM FF, hybrid CMOS/OxRAM FF and hybrid CMOS/PCRAM FF to allow system power-off in sleep mode.

4.2.2 Non-volatile reconfigurable logic

Studies have been conducted to also explore the benefits of integrating emerging NVMs into reconfigurable logic systems such as FPGA. Major issues of such circuits are the low-power efficiency due to the high leakage current, and logic density due to the use of SRAM for the configuration storage. Moreover, the volatility of SRAM forces the system to be reprogrammed at power-up from external flash memory leading to a long start latency. [91] evaluated a FPGA architecture based on TAS-MRAM technology. In 2010, a full non-volatile FPGA has been developed using 130nm CMOS technology and Crocus 120nm TAS-MRAM [92]. [93, 94], [95] and [96] respectively explored use of STT-MRAM, PCRAM, and ReRAM into FPGA.

The main benefits of including emerging NVM technologies into reconfigurable circuits are the ability to turn-off the system and save the total power consumption thanks to the non-volatility. Moreover, a fast start-up time is possible in comparison with classical SRAM-based FPGAs. Previous studies also demonstrated new features using emerging NVM technologies for the configuration storage such as run-time reconfiguration and

multi-context configuration capabilities.

4.2.3 Non-volatile processors

Processor architectures have also been targeted by the research community to evaluate the benefits of designing non-volatile processors including emerging NVM into the registers. [97] presented the first fabricated non-volatile processor ($130nm$ CMOS process) based on ferroelectric FFs with a $3\mu s$ wake-up time. The next year, [98, 99] introduced a full non-volatile logic-based 32-bit microcontroller SoC ($130nm$ CMOS process) also using FeRAM technology. Instead of using non-volatile FFs, small FeRAM-based memory arrays are distributed throughout the SoC to backup the FFs data. A complete area/performance/energy evaluation has been made which showed a $384ns$ wake-up time capability. These works demonstrated the feasibility of designing non-volatile processors with fast save/restore times and zero leakage standby mode. However, FeRAM has not the same potential as MRAM which shows faster access latency, lower access energy and higher density [100]. A non-volatile microprocessor unit based on STT-MRAM has been evaluated by [101] using a $90nm$ CMOS process. Simulation results showed a $3\mu s$ save/restore time for the pipeline register. Nonetheless, the design simulated was simplified by implementing only 12 instructions and the capacities of the instruction/data memories were reduced to 32 words x 32 bits.

4.2.4 Summary

All the previous works clearly highlighted the high interest of designing non-volatile systems to reduce the total energy consumption, but also to integrate new interesting features thanks to the non-volatility. Although these studies validated the ability to save/restore the system state at device, circuit, and system level, they did not verified it by running a complete application or a benchmark on the processor. Even though this chapter does not present a real design of a non-volatile processor using MRAM, it validates through a real benchmark and on a full 32-bit RISC-like processor the possibility to completely save and restore the processor state via RTL simulation. In addition, this chapter also presents validation of the rollback function allowing to restore a previous valid state of the processor in the case, for instance, of an execution error.

4.3 Instant on/off and rollback features

After giving an overview of the Amber processor core, this section will focus on the description of both *instant on/off* and *rollback* concepts. Then, validation of a complete backup/recovery of the state of the Amber core via RTL simulation will be shown.

4.3.1 Amber core

The Amber core is an ARM-compatible 32-bit RISC processor fully compatible with the old ARMv2a instruction set architecture. There are two versions of the Amber core. The first is the Amber 23 which has a 3-stage pipeline, a unified instruction and data cache, a 32-bit wishbone memory bus interface, and is capable of 0.8 DMIPS[2] per MHz. The second is the Amber 25 with has a 5-stage pipeline, separate instruction and data caches, a 128-bit wishbone memory interface, and is capable of 1.0 DMIPS per MHz. Both cores are able to boot a 2.4 Linux Kernel. The Amber 23 core is a very small 32-bit core that provides good performance, whereas the Amber 25 core is a little larger and provides 15% to 20% better performance than the Amber 23 core.

Figure 4.1 depicts the Amber 23 architecture, which is used in this thesis because it is the smallest and thus has the simplest architecture.

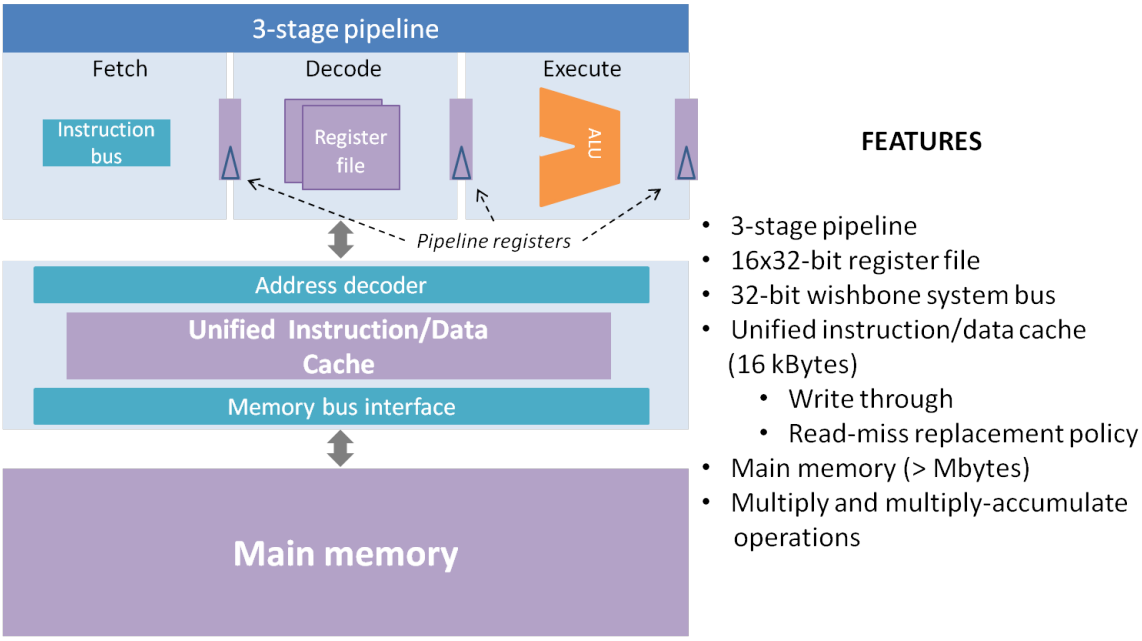


Figure 4.1: Amber 23 core architecture

4.3.2 Instant on/off

The *instant on/off* function consists in saving a complete state of the processor before a power-down, then restoring this state after a new power-up. The state of a processor is contained in both registers and main memory. At least, it is required to include the non-volatility into these two memory components to maintain the system state after a

[2]Dhrystone Millions of Instruction Per Second

power-down. In our study, 121 registers (representing 1644 flip-flops) contain the state of the Amber processor core, which include the register file, the pipeline registers, and some other internal registers.

Cache memory does not need to be non-volatile under certain conditions. The writing policy has to be a write-through[3], which is the case for the Amber core considered in this study. It is important to note that if the cache memory is kept volatile, the overall performance of the *instant-on/off* will be reduced since a warm-up period will be necessary after a power-up to restore data from main memory to cache memory. If the writing policy would have been a write-back[4], then all cache blocks marked as "dirty" would need to be written back to the main memory before a power-down to preserve the state of the processor. Thus, compared to the write-through, the write-back policy further penalizes the overall performance of the *instant-on/off*.

Figure 4.2 compares the original architecture of the Amber processor and the required architecture for a non-volatile processor with *instant-on/off* capability.

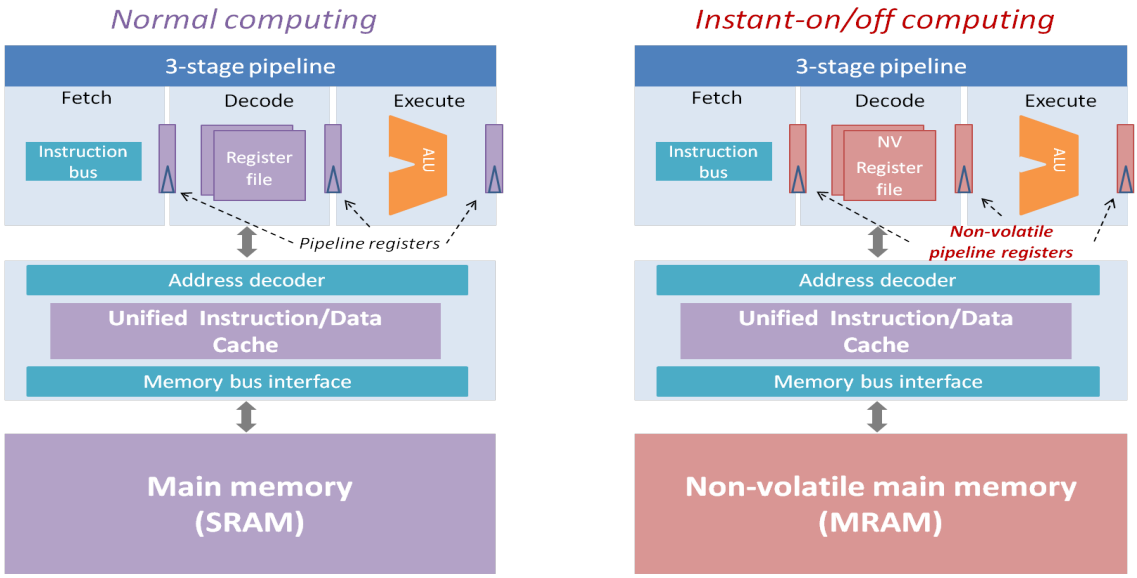


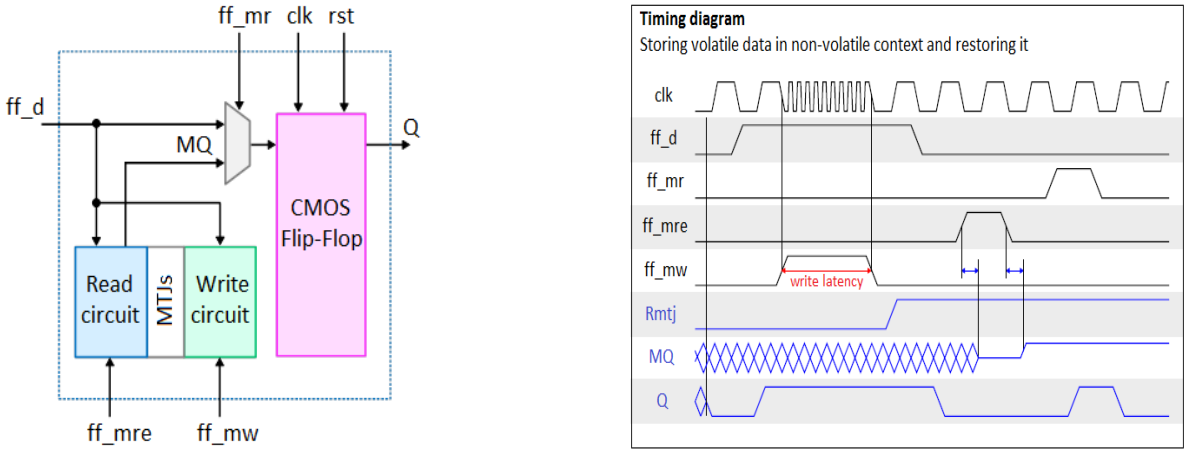
Figure 4.2: Amber architecture with instant-on/off computing: (a) Original Amber architecture (b) Amber architecture with non-volatile MRAM

[3]A scheme in which writes always update both cache and main memory, ensuring that data is always consistent between the two

[4]A scheme that handles writes by updating values only to the block in the cache, then writing the modified block to the lower level of the hierarchy when the block is replaced

Figure 4.3a shows a non-volatile flip-flop (FF) architecture based on MRAM, which can be typically used to design non-volatile registers. It consists of a standard CMOS FF for the volatile part and a MTJ for the non-volatile part. By the means of a multiplexor, the input state of the CMOS FF is either the output state of the previous stage of the circuit (ff_d in the figure) or the state of the MTJ (MQ in the figure). Also, a write circuit allows to store the state of the CMOS FF into the non-volatile MTJ.

Figure 4.3b depicts the timing diagram of the non-volatile FF. When the write signal ff_mw is activated for sufficient time (i.e. write latency), volatile data (ff_d) is stored in non-volatile context (Rmtj). Restoring this non-volatile context is performed by activating the signal ff_mre, and then the signal ff_mr.



(a) (b)

Figure 4.3: MRAM-based non-volatile flip-flop: (a) Architecture (b) Timing diagram

Assuming the processor uses this kind of non-volatile FF and the main memory is non-volatile, then the *instant on/off* procedure is described as follows:

ALGORITHM 1: Instant-on/off procedure

(1) *For each FF, save the current state by writing the value from the CMOS FF into the MTJ*;
(2) *Power down the processor. As the main memory is non-volatile, data are preserved*;
(3) *Power up the processor. As the main memory is non-volatile, data are available*;
(4) *For each FF, restore the backup data by reading the value from the MTJ into the CMOS FF.*

4.3.3 Rollback

The *rollback* is the ability to return to a previous valid state of the processor in the case for instance of an execution error. We assume that an error detection mechanism is available into the processor architecture to identify errors during the execution. The principle of

the *rollback* is shown in Figure 4.4.

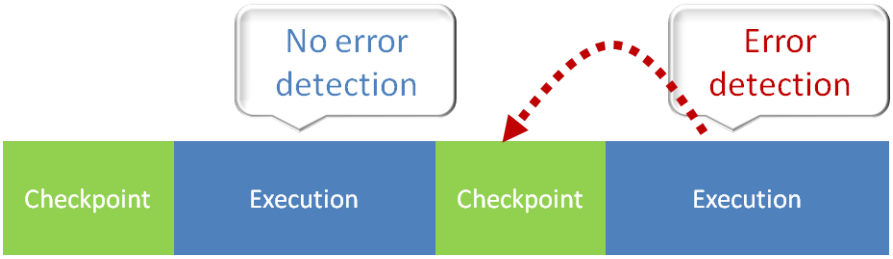


Figure 4.4: Rollback principle

Checkpoints can be created saving the state of the system either periodically or at strategic instant during the execution of the application. Then, if a system failure occurs, there is the possibility to come back to the last *checkpoint*. A *checkpoint* consists of a backup of both registers and main memory. Indeed, after each *checkpoint*, the main memory contents will most probably be modified. Therefore, it is necessary to add an additional memory (called *checkpoint memory* in the rest of this chapter) to keep a backup of the memory contents. To make it easier, the main memory is completely duplicated for the Amber processor. One memory will be used for the normal execution whereas the other one will be used to store the *checkpoint*. In a real application, the *checkpoint memory* size is smaller than the main memory size. This size depends on both the application and the interval between each *checkpoint*.

Figure 4.5 compares the original architecture of the Amber processor and the required architecture for a non-volatile processor with both *instant-on/off* and *rollback* capabilities.

Assuming the processor uses MRAM-based FFs as described in Figure 4.3a and the main memory is duplicated, then the *rollback* procedure is described as follows:

ALGORITHM 2: Rollback procedure

(1) *Create checkpoints during the execution of the application*;
(2) *A system failure is detected*;
(3) *Stall the processor*;
(4) *Restore the last checkpoint which consists in*;
- *Restoring the state of the FFs by reading the value from the MTJ into the CMOS flip-flop*;
- *Restoring the main memory contents by copying data from the checkpoint memory to the main memory*;

(5) *Take the execution of the application up again.*

4.3.4 RTL simulation

This section aims at validating the *rollback* function via RTL simulation. It is recalled that the objective is to validate the possibility to completely save/restore the state of

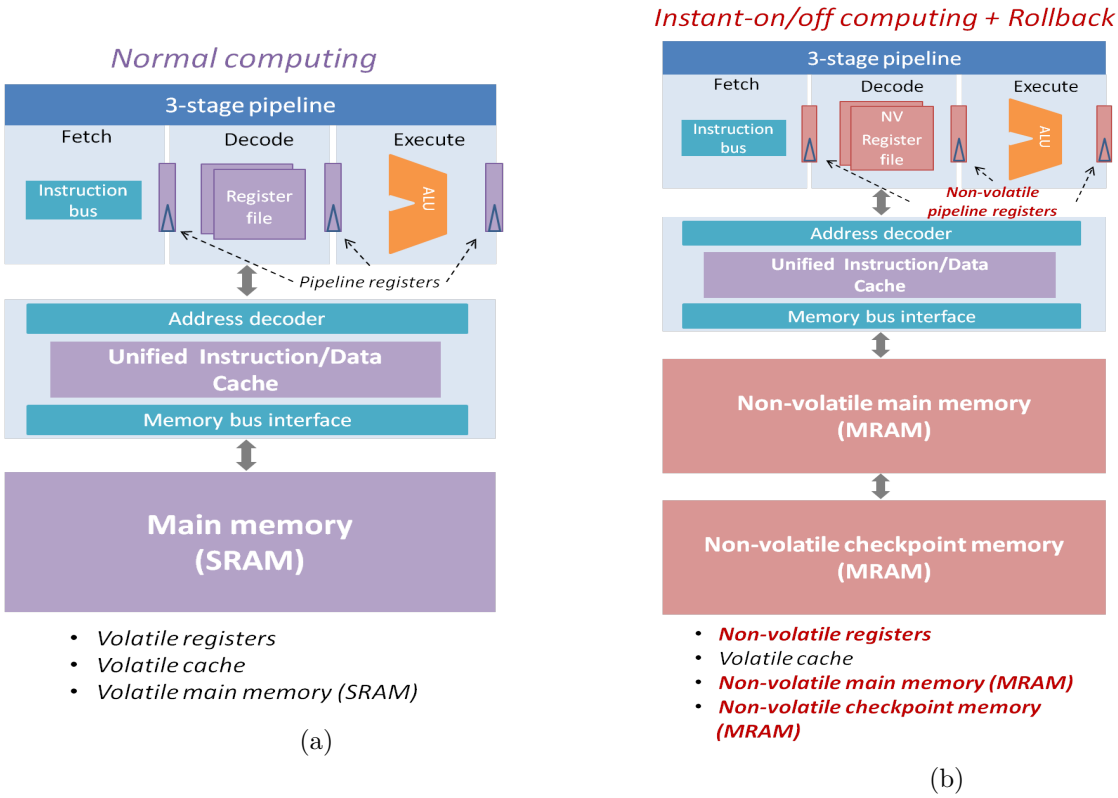


Figure 4.5: Amber architecture with instant-on/off computing and rollabck capability: (a) Original Amber architecture (b) Amber architecture with non-volatile MRAM and checkpoint memory for rollback

the processor. Simulation is made with a full application running on the processor, the Dhrystone 2.1 [102] which is included with the source code of the Amber core. For the logic implementation, the registers of the Amber processor are duplicated, as described in Figure 4.6, to emulate the non-volatile registers and save the state of the system. The original registers, named volatile registers in the figure, are used for the normal execution while the duplicated ones, named non-volatile registers in the figure, store the state of the processor.

The main memory is also duplicated to allow a backup of the memory contents. As the objective is only to validate the *rollback* functionality, a non-synthesizable main memory model is used for fast simulation purpose. At the beginning of the application, both main memory and *checkpoint memory* contain the same data. Then, during the normal execution of the application, only the main memory contents are modified. At the next *checkpoint*, only the main memory locations which were modified during the execution are copied into the *checkpoint memory* (Figure 4.7). Thus, copying all the contents of the main memory at each *checkpoint* is avoided. For that, all writes into the main memory be-

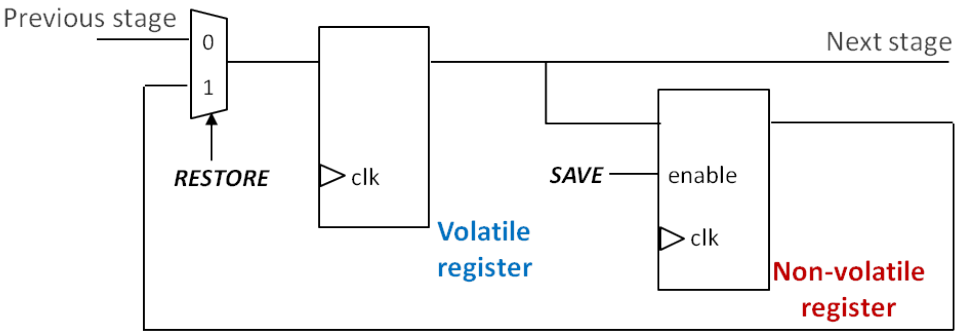


Figure 4.6: Logic implementation of the registers

tween two *checkpoints* are tracked by storing all the corresponding memory addresses into a small buffer. In the case of a system failure before the next *checkpoint*, all the memory addresses present in this buffer correspond to the main memory locations to be restored for a *rollback*. Data are restored from the *checkpoint memory* to the main memory. If the address buffer is full before the next *checkpoint*, a creation of a *checkpoint* is forced.

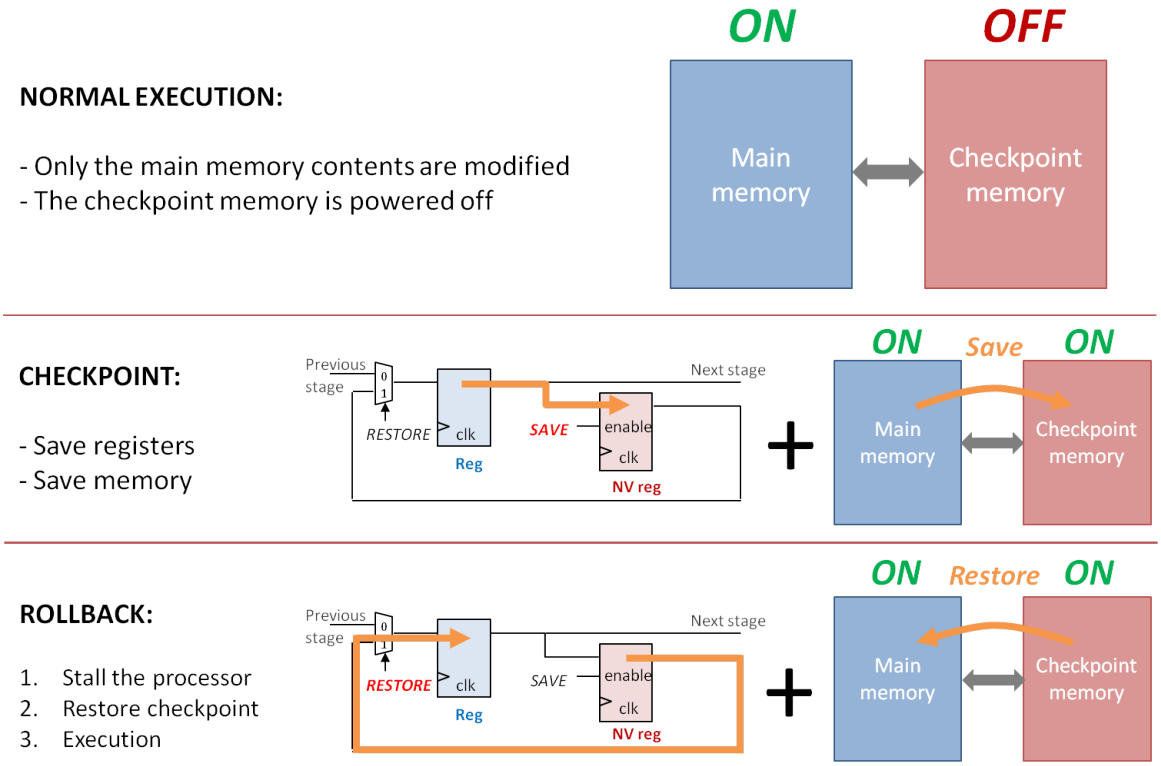


Figure 4.7: Checkpointing and rollback

Figure 4.8 shows the terminal output of the Dhrystone 2.1 application in which a

rollback is performed. This output shows when the execution starts, when the execution ends, and then prints final values of the variables used in the benchmark. The *checkpoint* is created after the execution starts and before the execution ends. Then, the same *checkpoint* is restored later when the application is printing the final values of the variables. In this simulation, only one *checkpoint* is created. The *rollback* have also been validated for the case in which several *checkpoints* are regularly created during the execution of the application. In such a case, a *rollback* restores the last *checkpoint*.

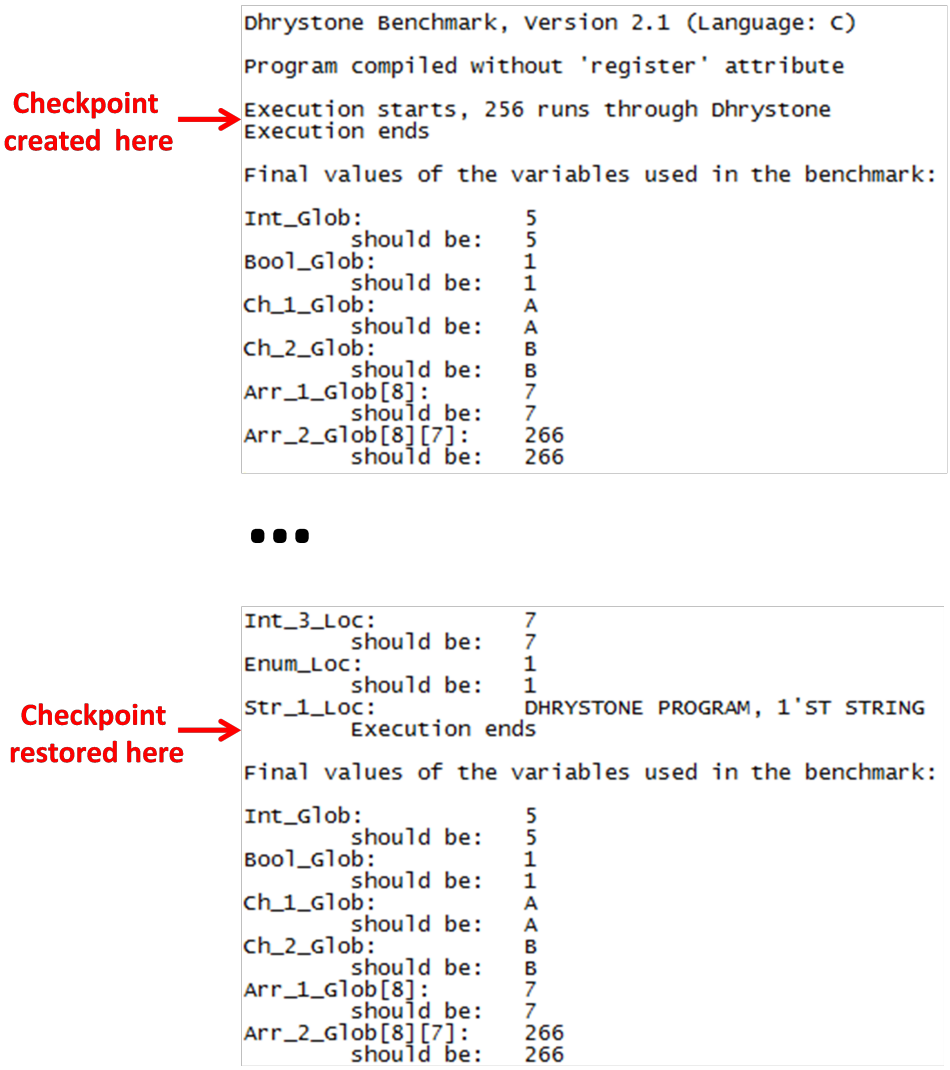


Figure 4.8: Validation of the rollback capability (terminal output of the Dhrystone application)

4.4 MRAM-based non-volatile processor: performance and energy

Many FFs based on MRAM were proposed in the literature to enable the design of non-volatile circuits such as in [82, 83]. In order to estimate overall performance of MRAM-based non-volatile processor, we consider information from the current state-of-the-art of MRAM-based FFs. For comparison purpose, the cases of OxRAM-based FFs and PCRAM based FFs are also evaluated. Table 4.1 shows the time and the energy to back-up/restore the state of a FF for the different NVM technologies.

Considering all the parameters, MRAM-based FFs show the best performance. STT-MRAM has the smallest latency and energy for the backup. On the other hand, TAS-MRAM shows better performance to restore the non-volatile data.

Table 4.1: Non-volatile flip-flops performance

Technology	**Latency (ns)**		**Energy (pJ)**	
	Restore	Back-up	Restore	Back-up
STT-MRAM [89]	0.2	4	0.012	0.5
TAS-MRAM [88]	0.13	16	0.012	5.2
OxRAM [87]	6	70	1.4	28
PCRAM [90]	370	370	7.4	463

These FFs are designed to have a dual-storage facility (hybrid). The CMOS stage of the FF uses cross-coupled inverters (latch) to store one data bit in its electrical (volatile) form. On the other hand, the magnetic stage uses a MTJ (in the case of MRAM) to store one non-volatile data bit.

In the rest of the section, the performance and energy implications of integrating MRAM into a processor architecture are discussed for the active power mode and for the transitions between active and sleep modes. Then, the next section will analyze the energy consumption profile in the case of a processor implementing the *instant-on/off*.

4.4.1 Performance

Active mode

Depending on the application, it could be very useful for the system to run at a higher speed in the active mode so that it can return quickly to low-power mode. However, running at high frequency also increases the active power. The designer has to analyze the best case for the application taking into account other factors such as the frequency at which the system needs to switch between the active and the sleep mode.

Use of fast-access registers into the processor is necessary if high speed operation is required. Hybrid CMOS/MTJ FFs are suitable to build fast-access non-volatile registers thanks to their dual-storage facility. The CMOS stage storing data in its volatile form is used during normal execution of the system. The MTJ state is only used when there is a need to back up or restore the system state. Therefore, building the registers using these hybrid CMOS/MTJ FFs will not affect the performance of the processor in active mode.

Back-up

If required, external flash memory is used in commercial MCUs to restore the program and data when going out from low-power mode to active mode. It can also be used to log data before entering sleep mode if these data have to be used later in the application. This logging phase can take several milliseconds due to the long erase/program procedure of flash memory. As this memory is usually external, additional latency will increase the back-up process due to data transfer through the serial communication interface.

Thanks to its low access latency compared to flash, MRAM is suitable to be integrated in both registers and main memory allowing a small back-up time. Since the main memory is assumed to be non-volatile, a back-up of the state of the processor corresponds to a back-up of the registers. Therefore, depending on the NVM technology used, the back-up time of the processor is the back-up time of the FF (Table 4.1).

Wake-up

The wake-up time is a key parameter for ultra-low-power devices. It informs us if the system can return from low-power mode to active mode quickly enough to accomplish the task at hand. Existing low-power MCUs include several low-power modes with different wake-up times so that the customer can choose the appropriate configuration for a given application. The components which have a significant influence on the wake-up time if they are turned off are the embedded memories, i.e. the registers and the main memory. Not retaining data into embedded memories requires the system to load data from external flash memory at each power-up, which is time and energy consuming.

In the case of MRAM-based registers, the wake-up time of the processor corresponds to the time to restore the registers states after a power-up. Assuming the main memory is non-volatile, data are already available in this memory after a power-up. Hence, the wake-up time is the latency to restore the FF state (Table 4.1).

4.4.2 Energy

Active mode

Depending on the amount of time the device remains in active mode and the frequency at which the MCU is running, the energy consumption can be more or less important. The active power consumption can be decreased if the MCU is running at low frequency. But the time to process data will increase. On the other hand, running at higher frequency will allow to quickly return to low-power mode at the cost of higher active power consumption.

As for performance, using hybrid CMOS/MTJ FFs for registers into the processor will not affect the active energy consumption since this is the CMOS part which is used during normal operation.

Back-up

In addition to the long time it takes to log data, the back-up phase using flash memory is power hungry. The required current to erase and program flash can vary from 4 to $12mA$ [103].

Figure 4.9 estimates the back-up energy of the Amber core when implementing non-volatile MRAM, OxRAM and PCRAM based registers. As already mentioned in Section 4.3.2, 1644 FFs have to be saved to retain the state of the Amber core. Therefore, the back-up energy is estimated as the energy to write into the MTJ (in the case of MRAM) times the number of FFs. As observed in Figure 4.9, use of STT-MRAM lead to around $800pJ$-backup energy, whereas the back-up energy reaches about $9nJ$ when TAS-MRAM is used. Use of OxRAM and PCRAM show respectively a back-up energy of $46nJ$ and $760nJ$

Wake-up

When estimating the average energy consumption of a system switching between active and sleep modes, the wake-up energy has to be considered. In current MCUs, this transition energy can be significantly high if returning to active mode requires data recovery from non-volatile memory (flash). If the contents of both registers and main memory are

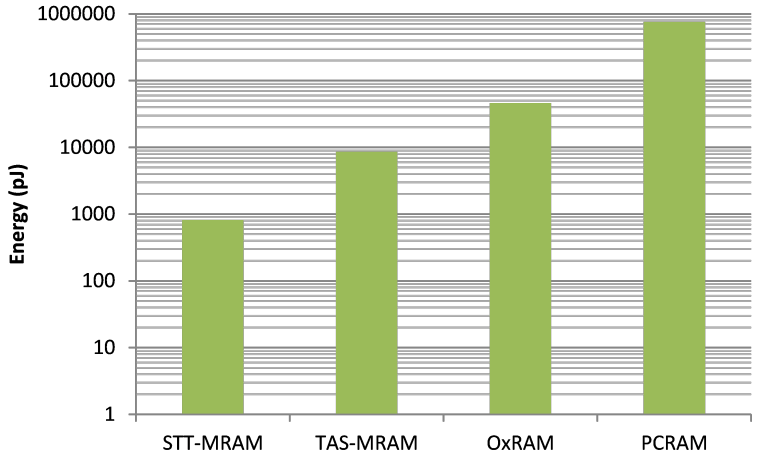


Figure 4.9: Back-up energy (logarithmic scale)

not retained during low-power mode, the boot process to run the program will also consume valuable energy. [97] showed that the energy consumption to restore 1607 FFs from off-chip flash (on-chip flash) is $1.3\mu J$ ($0.6\mu J$). Maintaining the registers and the main memory contents will decrease the wake-up time and so the wake-up energy. However, the leakage power in low-power mode will increase because of the current needed to enable data retention.

Figure 4.10 shows the wake-up energy for the Amber core if non-volatile FFs based on MRAM, OxRAM or PCRAM are used. This energy corresponds to the read energy of the MTJ (in the case of MRAM) times the number of FFs. The results show a wake-up energy of $20pJ$, $2.3nJ$, and $12.2nJ$ respectively for MRAM (both STT and TAS), OxRAM and PCRAM.

4.5 Instant-on/off and sleep mode: energy analysis

The leakage current is clearly an important factor for devices spending most of their time in low-power mode. The deeper the system sleeps (most components being turned off), the lower it consumes energy, but the longer it takes to return to active mode. Presence of leakage current in existing low-power MCUs is mainly due to the volatility of embedded memories. As already mentioned, it is necessary to keep these components turned on to allow fast wake-up time.

Integrating MRAM into the registers and the main memory for processors gives the valuable advantage to remove the power consumption when the system remains in sleep mode. As the leakage current increases dramatically with the decreasing size of the CMOS transistor, the non-volatility of MRAM is a very attractive feature which has the

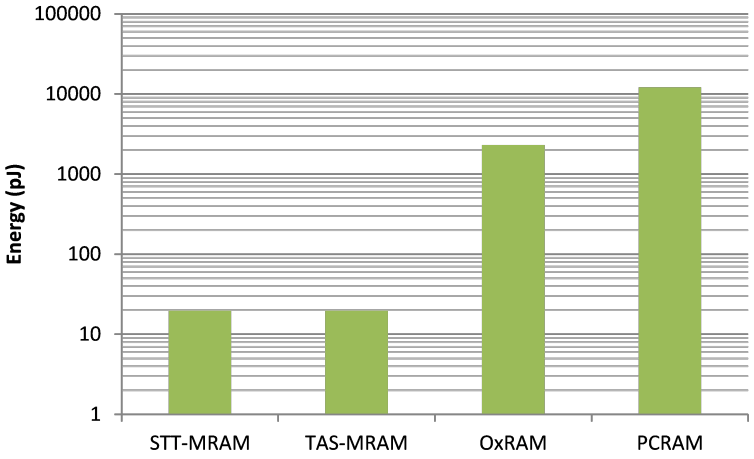


Figure 4.10: Wake-up energy (logarithmic scale)

potential to be integrated not only at flash level, but also at main memory, cache memory and register level.

Figures 4.11a and 4.11b depict the profiles of the energy consumption respectively for a classical MCU without *instant-on/off* capability and a non-volatile MCU with *instant-on/off* capability. In these figures, switching between active and sleep modes is assumed to be periodic. P_{active} (T_{active}) corresponds to the power consumption (time) in active mode. $P_{leakage}$ (T_{sleep}) is the power consumption (time) in sleep mode. T_{wakeup} is the wake-up time. For the system related to Figure 4.11a, we assume that data into the registers and the main memory are retained during the sleep mode. Hence, we also assume there is no backup energy when switching from active mode to sleep mode.

As mentioned above, the active energy consumption is not changed when replacing

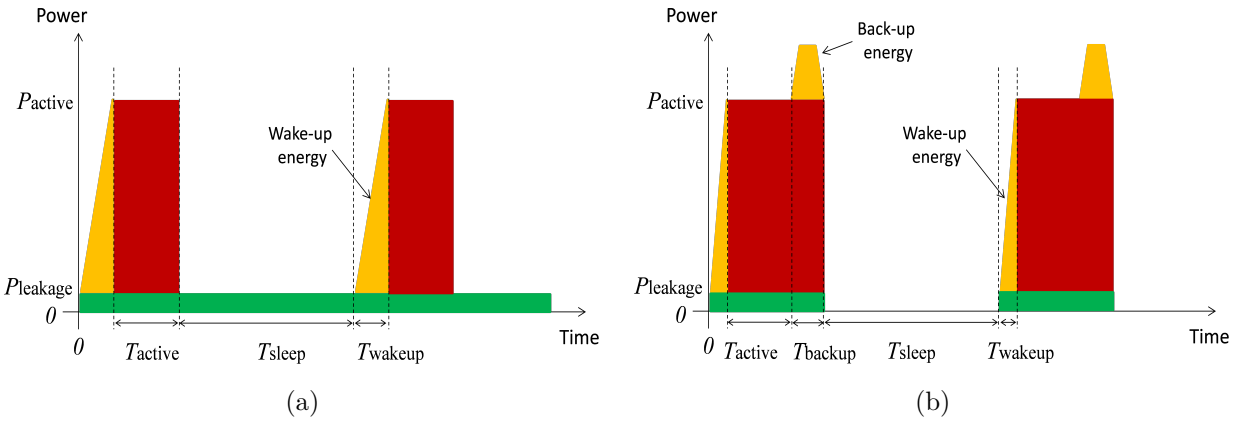


Figure 4.11: Energy profile: (a) without instant-on/off (b) with instant-on/off

CMOS-based registers by hybrid CMOS/MTJ registers. Therefore, the active power consumption is the same for both figures. The most visible energy gain when comparing the two figures is during the sleep mode. Enabling the *instant-on/off* feature thanks to MRAM removes the power consumption during this mode. However, an energy overhead is observed due to the required back-up phase before entering sleep mode. Therefore, if MRAM is used (Figure 4.11b), the system will be more energy efficient if the back-up phase consumes lower energy than the leakage energy normally consumed during the sleep mode in Figure 4.11a. Thus, the time spent in sleep mode (T_{sleep}) has to be significant enough. The condition to reduce the total energy consumption when using MRAM is given by the equation (4.1):

$$(P_{active} + P_{leakage}) \times T_{backup} + E_{backup} < P_{leakage} \times T_{sleep} \tag{4.1}$$

In this equation, note that the wake-up phase has not been considered for two reasons. First, this phase is not part of the sleep mode. Second, a classical MCU also consumes energy during this phase. Therefore, the wake-up phase has to be analyzed independently of sleep mode.

From the equation (4.1), the condition which has to be verified on the time T_{sleep} is given by the equation (4.2):

$$T_{sleep} > \frac{(P_{active} + P_{leakage}) \times T_{backup} + E_{backup}}{P_{leakage}} \tag{4.2}$$

This condition on T_{sleep} has been determined for the Amber core used in this work. The processor has been synthesized using a $65nm$ CMOS low-power HVT process. On the basis of the synthesis results, the dynamic power and the leakage power are respectively equal to $173mW$ (at $40MHz$) and $12mW$. Using the back-up time and energy estimated in this chapter, the minimum time T_{sleep} to reduce the total power consumption when using *instant-on/off* is shown in Table 4.2 for FFs based on the four considered NVM technologies. The minimum time T_{sleep} does not exceed $130ns$ ($968ns$) when the FFs are based on STT-MRAM (TAS-MRAM). These small time values, in the case of the Amber core, allows the processor to frequently go into sleep mode, and thus shows the great potential of using MRAM to significantly reduce the total energy consumption of a system thanks to *instant-on/off* computing.

Technology	STT-MRAM	TAS-MRAM	OxRAM	PCRAM
Minimum T_{sleep} (μs)	0.130	0.968	4.9	69

Table 4.2: Minimum T_{sleep} required to save energy with *instant-on/off*

It is important to note that the estimations related to the results of Table 4.2 does not consider the power down/up circuitry. Moreover, a complete study should also consider the area overhead of such a non-volatile processor, which is envisaged for the future work of this thesis. According to the state-of-the-art, a MRAM-based non-volatile FF is about 50% to 100% larger than a conventional CMOS-based FF.

Assuming such a non-volatile processor using non-volatile FFs for the registers, it is also important to note that writing at the same time into all the non-volatile parts of the FFs during the back-up phase can lead to a high peak current, which is not appreciated for the system. If we consider the write current of the STT-MRAM used in this work, which is about $100\mu A$, the peak current to write into the 1644 FFs of the Amber processor core exceeds $160mA$. Therefore, in practical, the back-up would be rather performed gradually.

4.6 Conclusion

Within the context of the IoT, this chapter investigated the use of MRAM at register level to design a non-volatile processor. Two capabilities enabled by non-volatile registers has been studied. First, the *instant-on/off* which is the ability to retain the state of a processor into non-volatile registers before a power-down. Thus, a fast wake-up is possible. Second, the *rollback* allows to restore a previous valid state of the processor, for instance in the case of a system failure. These features have the potential to reduce significantly the average power consumption of a SoC and to improve the fault tolerance. To validate the feasibility of these two capabilities, a complete backup/recovery of the processor state has been performed via RTL simulation and running the Dhrystone 2.1 application, considering a full 32-bit processor.

As part of the evaluation of such a non-volatile processor, performance and energy of the backup/restore phases has been estimated, based on the results of electrical characteristics of MRAM-based FFs, according to the state-of-the-art. For 1644 non-volatile FFs based on STT-MRAM, the estimated back-up energy is not more than $1nJ$ ($500fJ$ per FF), whereas the restore energy is as small as $20pJ$ ($12fJ$ per FF). For comparison, [99] has measured the backup/restore energy of a real prototype of a non-volatile processor based on FeRAM. Considering 2537 FFs, results showed a $7nJ$-back-up energy ($2.8pJ$ per FF) and a $2.4nJ$ restore energy ($950fJ$ per FF). Thus, estimation results presented in this chapter are quite consistent, even though going towards a real silicon prototype is necessary for more accurate evaluation.

Furthermore, considering the energy profile of a device (e.g. IoT object) with low-power mode, a processor with *instant-on/off* capability have potentially the possibility to

frequently go into sleep mode with zero leakage power thanks to the non-volatility.

Finally, to conclude this chapter, Figure 4.12 summarizes the architectural changes if the *instant-on/off* and *rollback* are implemented in the Amber processor. As the memory system is entirely volatile in the original Amber architecture (Figure 4.12a), data retention involves leakage power consumption in sleep mode. On the other hand, not retaining data results in slow wake-up time. Integrating MRAM into both registers and main memory allows *instant-on/off* computing with no leakage power in sleep mode and fast wake-up time (Figure 4.12b). However, use of non-volatile FF increases the silicon area of the processor and an energy overhead is added due to the back-up phase. Finally, at the cost of higher area overhead and execution time penalty, the processor can be more fault tolerant thanks to the *rollback* (Figure 4.12c).

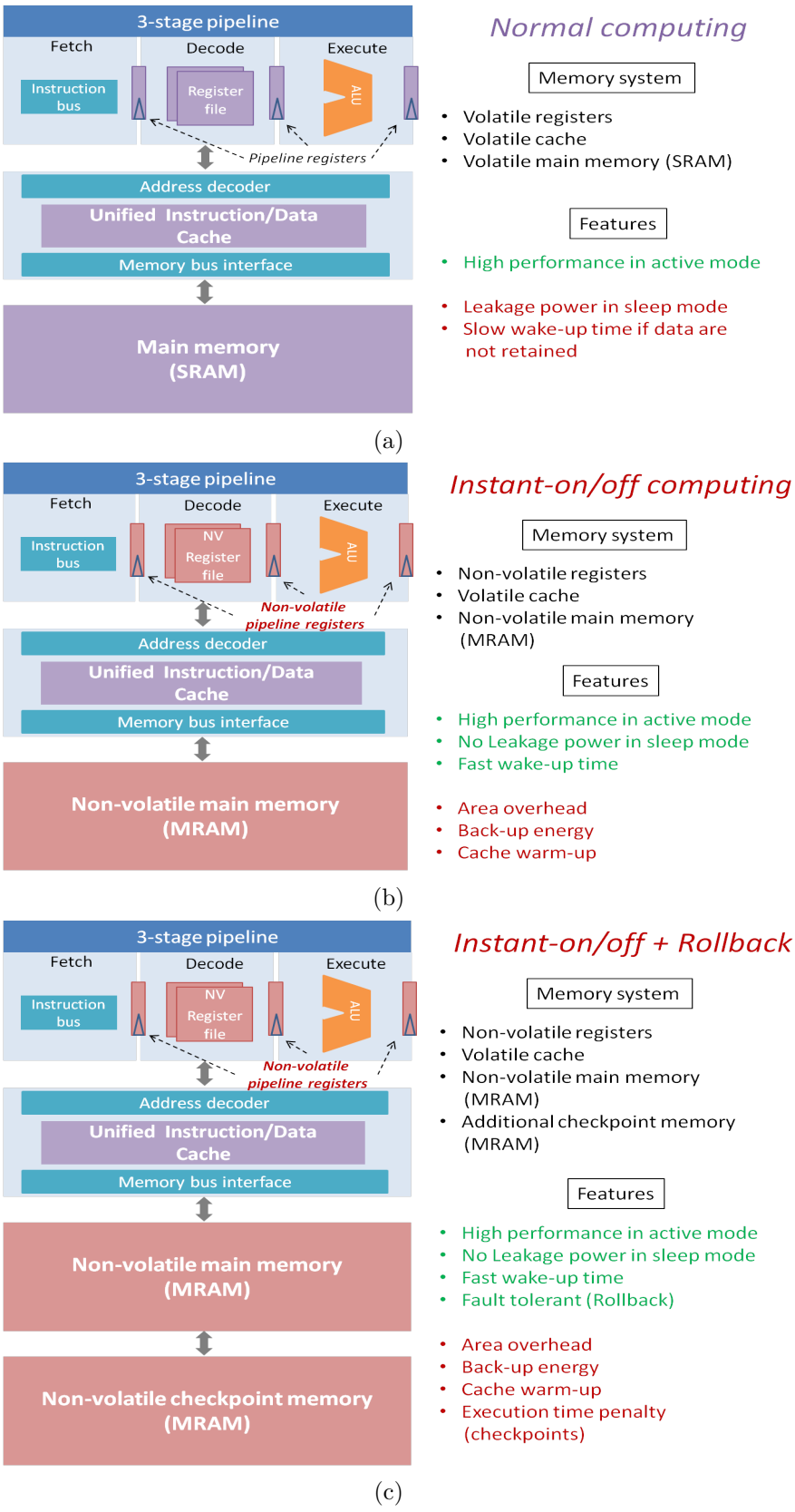


Figure 4.12: Computing paradigms: (a) Normal computing (b) instant-on/off computing (c) instant-on/off + rollback

5

CONCLUSION

Over the past few years, spintronics, a new paradigm of electronics which uses the spin of the electrons to carry information, has been attracting a large part of the research community owing to its high potential to resolve one of the most critical issue of current ICs: energy efficiency. Advances on this new area, as described in Chapter 2, lead to the birth of a new memory technology, called MRAM, combining interesting features such as non-volatility, high density, low leakage power and reasonable latency. In order to improve its overall performance, several methods has been developed to program the MRAM cell: Toggle MRAM, TAS-MRAM, STT-MRAM, MeRAM and SOT-MRAM. Today, the main issues of this memory technology, which are still under intensive investigations, are the high write latency and energy.

The major objective of this thesis, supported by Crocus Technology, was to evaluate the integration of MRAM into the memory hierarchy of processor architecture, according to the three metrics speed/energy/area. More precisely, this thesis aimed at:

- Setting up a generic exploration flow for a fine-grain evaluation at both circuit and architecture levels.
- Exploring new features than can be added thanks to the non-volatility of MRAM.

Only TAS-MRAM and STT-MRAM have been considered for this work as they are quite mature and show reasonable performance at device level compared to SRAM, used as a reference.

Chapter 3 reported a detailed evaluation of these two MRAM technologies at cache level. Regarding area, although the MRAM bit cell is smaller than its SRAM equivalent, the total area of a cache memory based on MRAM will be smaller than SRAM only if the area of the cell array is large enough compared to the area occupied by the peripheral circuitry, as the latter includes large transistors to allow high programming current for MRAM.

Concerning performance and energy, this study has clearly exposed the suitability of MRAM for lower levels of cache (i.e. L2 or LLC) in the memory hierarchy. As most of the total power consumption is static, the low leakage current of MRAM drastically reduces the total energy consumption compared to SRAM-based L2 cache (up to 90%). Running a large panel of applications on a multi-core architecture, the observed performance penalties are small and sometimes non-existent when replacing SRAM by STT-MRAM. For TAS-MRAM, even though the resulting penalty can be significant, its use in L2 cache is still a good performance/energy trade-off for some applications.

For upper level of cache (i.e. L1), current characteristics of MRAM penalize both performance and energy of the system. As it is the closest to the processor, L1 cache is intensively accessed compared to L2 and lower level of cache. Therefore, the high write latency and energy of MRAM have an important impact on the overall performance. Even though MRAM-based cache consumes lower leakage current than SRAM, it is not sufficient to compensate the high dynamic energy of MRAM.

In Chapter 4, integrating the non-volatility into the registers of a processor has been explored. The study has validated the ability to save/restore the entire state of a full 32-bit RISC-like processor (*instant-on/off*). Furthermore, running a complete application, the recovery of a previous valid state of the processor has been demonstrated (*rollback*).

The *instant-on/off* capability has been evaluated in terms of performance and energy using electrical characteristics of non-volatile MRAM-based flip-flops according to the state-of-the-art. Moreover, analyzing the energy consumption profile of low-power applications with active and sleep modes, a non-volatile processor based on MRAM is able to be frequently powered-off while preserving a fast wake-up time thanks to the non-volatility. This emerging computing paradigm, known as *normally-off computing*, is definitely a potential solution to overcome the energy efficiency issue in current embedded systems.

6

PERSPECTIVES

This thesis highlighted a few advantages of including MRAM into the memory hierarchy of processor architecture. Although it is a promising NVM technology, MRAM still needs further development, and research works are still necessary to explore how current characteristics and future advances of this memory can be used to overcome current issues of nanoelectronic systems. This chapter aims at proposing future works of this thesis.

6.1 Further exploration at cache level

Chapter 4 explored direct replacement of MRAM in both L2 and L1 cache memories with a detailed analysis at circuit and architecture levels. This study considered the same memory capacity when replacing SRAM by MRAM. An interesting work could be to not consider the same capacity but the same area, since this thesis observed a smaller area when MRAM is used for large cache memory (i.e. with large memory capacity). Moreover, an experiment recently demonstrated the multi-bit storage capability of the TAS-MRAM technology [104]. A possible study could be to run data-intensive applications on a multi-core architecture using a very large shared LLC (e.g. a few MB).

Other possible studies are to define and evaluate new architectures which takes advantages of MRAM and mitigate its drawbacks. Since this NVM technology is still not competitive with SRAM in L1 cache, hybrid SRAM/MRAM L1 cache architectures could be investigated to benefit of the high speed of SRAM and the non-volatility and low leakage of MRAM. Furthermore, many device-level and circuit-level techniques has been proposed to improve the overall performance of STT-MRAM (2.3.4), which could be considered for future MRAM-based cache explorations.

Finally, other MRAM technologies (and even other emerging NVM technologies) could be explored such as SOT-MRAM and MeRAM which show very promising performances at device level.

6.2 Extension of the NVM exploration flow

In this thesis, the NVM exploration flow only supports evaluation of cache memory. As part of the future works, this flow can be extended to explore MRAM at other level of the memory hierarchy. For instance, [105] proposed a main memory simulator designed to simulate emerging NVM at architecture level, which can be interfaced with the gem5 simulator.

Furthermore, a more generic NVM exploration flow could be investigated to evaluate performance/energy/area of full multi-core architectures including MRAM into the memory hierarchy. [106] developed McPAT, an integrated power, area, and timing modeling framework for multi-threaded, multi-core, and manycore architectures. McPAT includes models for the components of a complete chip multiprocessor, including in-order and out-of-order processor cores, networks-on-chip, shared caches, and integrated memory controllers. However, this framework does not include models for NVM technologies. As part of future work, one could integrate NVM circuit-level models such as NVSim into McPAT. Thus, the impact of MRAM-based memory hierarchy on a complete multi-core architecture can be estimated with more accuracy.

6.3 Non-volatile processor

Chapter 4 have demonstrated the possibility to save/restore a previous valid state (*rollback*) of a full 32-bit RISC processor running a full application. The rollback function requires an additional memory (*checkpoint memory*) to save the contents of the main memory when creating a *checkpoint*. Further work is needed to estimate the size of the *checkpoint memory* by running different applications.

Finally, an accurate evaluation of such a non-volatile processor using MRAM-based registers would be to, first, analyze the energy consumption of a real MCU by running a real application, second, go towards a silicon prototype to evaluate the speed/energy/area overheads.

6.4 Security

In this thesis, evaluation of MRAM was restricted to the three metrics speed/energy/area for comparison with SRAM. More metrics have to be considered if integration of MRAM into SoCs is envisaged, such as endurance, reliability and security. The latter is particularly challenging with the emergence of IoT. This section aims at showing a few perspectives of exploring MRAM for security applications.

6.4.1 Side-channel analysis

Within a security context, any potential source of information from the physical implementation of a cryptographic algorithm (i.e. algorithm to secure confidential data) is called side channel. The most common used information for side-channel attacks are timing, power consumption and electromagnetic emission. The basic idea of side-channel attacks is to retrieve the secret part of a cryptographic algorithm (i.e. the cipher key) by analyzing the correlation between the leakage information and the processed data.

Side-channel attacks and countermeasures (i.e. techniques to reduce the vulnerability of a system against attacks) have been deeply studied by the research community for the case of CMOS-based cryptosystems using both SRAM and flash memories. As MRAM might be part of future secure devices, possible future work is to explore the potential of MRAM against side-channel attacks. Crocus technology has already proposed an innovative secure function called the *match-in-place* (MIP) [4, 25] based on the MLU technology, a specific implementation of the TAS-MRAM (Section 2.3.3). A first evaluation of the MLU against side-channel attacks has been done in [107]. As another potential, MRAM is also an actual good sensor to measure electromagnetic injections into a chip, and then it can be used as an efficient countermeasure.

6.4.2 True Random number generator

Ability to generate random numbers is definitely useful for security applications since random cryptographic keys could be generated to transmit data securely. Nowadays, two kinds of random number generators (RNG) exist: pseudo-RNG (PRNG) and true RNG (TRNG). PRNGs are implemented in software and use deterministic algorithms to generate a sequence of random numbers. On the other hand, TRNGs are implemented in hardware and use non-deterministic physical event. TRNGs are clearly more appreciated for highly secure data encryption. A few solutions of TRNGs have been proposed using ring oscillators, such as in [108, 109].

[110] have demonstrated through experiment the generation of random numbers using the STT-MRAM technology. The stochastic nature of STT switching [111] has been used as a source of randomness to generate sequences of random numbers which have passed the statistical test of NIST SP-800 with the appropriate pass rate. Thus, this results are encouraging to devote more effort on the development of MRAM-based TRNGs.

6.4.3 Physically unclonable function

Physically unclonable functions (PUFs) are emerging primitives that are used for low-cost authentication and secret key storage. Current solutions for such applications are

the use of non-volatile memory (such as EEPROM[1]) or battery-backed SRAM combined with hardware cryptographic operations. However, this approach is expensive, and both area and power consuming. On the other hand, PUFs are based on simple digital circuits which exploit the innate manufacturing variability of ICs to derive a secret information. Since this variability is unique for each chip, PUFs can produce a unique signature, practically impossible to duplicate, for strong authentication. Thanks to the manufacturing variability, PUFs can also generate a secure key from the physical characteristics of the IC such as gate delay or threshold voltages. SRAM-based PUF is one of the most investigated solutions, which uses the power-on state of the memory.

Because of its potential for low power and high density SoCs, MRAM could be an attractive solution for PUF architectures. A few studies have already been done in [112, 113, 114], which showed the possibility to design MRAM-based PUFs.

[1] Electrically-Erasable Programmable Read-Only Memory

Publications

[1] **S. Senni**, L. Torres, G. Sassatelli, A. Gamatie, and B. Mussard. Non-volatile processor based on MRAM for ultra-low-power IoT devices. In *ACM Journal on Emerging Technologies in Computing*, Under submission.

[2] **S. Senni**, L. Torres, G. Sassatelli, A. Gamatie, and B. Mussard. Exploring MRAM technologies for Energy Efficient Systems-On-Chip. In *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Accepted for publication.

[3] **S. Senni**, L. Torres, and B. Mussard. Applications of Magnetic RAM for Processor Architecture. In *Leading Edge Embedded NVM Workshop (e-NVM)*, September 2015, Invited talk.

[4] **S. Senni**, L. Torres, G. Sassatelli, A. Gamatie, and B. Mussard. Emerging Non-Volatile Memory Technologies Exploration Flow For Processor Architecture. In *IEEE Computer Society Annual Symposium on Very Large Scale Integration (ISVLSI)*, 2015.

[5] **S. Senni**, L. Torres, G. Sassatelli, A. Gamatie, and B. Mussard. Potential Applications based on NVM Emerging Technologies. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, March 2015, pages 1012-1017.

[6] L. V. Cargnini, L. Torres, R. M. Brum, **S. Senni**, and G. Sassatelli. Embedded Memory Hierarchy Exploration Based on Magnetic Random Access Memory. In *Journal of Low Power Electronics and Applications (JLPEA)*, Selected Papers from Faible Tension Faibe Consommation Conference (FTFC) 2013, August 2014, volume 4, issue 3, pages 214-230.

[7] **S. Senni**, L. Torres, G. Sassatelli, A. Butko, and B. Mussard. Exploration of Magnetic RAM based memory hierarchy for multicore architecture. In *IEEE Computer Society Annual Symposium on Very Large Scale Integration (ISVLSI)*, July 2014, pages 248-251.

[8] **S. Senni**, L. Torres, G. Sassatelli, A. Butko, and B. Mussard. Power efficient Thermally Assisted Switching Magnetic memory based memory systems. In *9th Inter-*

national Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), May 2014, pages 1-6.

[9] **S. Senni**, R. M. Brum, L. Torres, and G. Sassatelli. Magnetic RAM based memory hierarchy exploration. In *GDR SoC SiP*, June 2014.

[10] L. V. Cargnini, L. Torres, R. M. Brum, **S. Senni**, and G. Sassatelli. Embedded memory hierarchy exploration based on magnetic RAM. In *Faible Tension Faible Consommation (FTFC)*, June 2013, pages 1-4.

[11] **S. Senni**, J. Azevedo, L. Torres, L. V. Cargnini, R. M. Brum, B. Jovanovic, G. Sassatelli, A. Virazel, P. Girard, A. Todri-Sanial, A. Bosio, and L. Dilillo. MRAM research @ LIRMM. In *Leading Edge Embedded NVM Workshop (e-NVM)*, October 2013.

Bibliography

[1] F. Shearer, *Power management in mobile devices*. Newnes, 2011.

[2] E. Kitagawa, S. Fujita *et al.*, "Stt-mram cuts power use by 80%," *eetimes.com*. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1280753

[3] "Crocus technology company." [Online]. Available: http://www.crocus-technology.com/

[4] S. Wolf, D. Awschalom, R. Buhrman, J. Daughton, S. Von Molnar, M. Roukes, A. Y. Chtchelkanova, and D. Treger, "Spintronics: a spin-based electronics vision for the future," *Science*, vol. 294, no. 5546, pp. 1488–1495, 2001.

[5] "The nobel prize in physics 2007: Information for the public."

[6] M. N. Baibich, J. M. Broto, A. Fert, F. N. Van Dau, F. Petroff, P. Etienne, G. Creuzet, A. Friederich, and J. Chazelas, "Giant magnetoresistance of (001) fe/(001) cr magnetic superlattices," *Physical review letters*, vol. 61, no. 21, p. 2472, 1988.

[7] P. M. Tedrow and R. Meservey, "Spin-dependent tunneling into ferromagnetic nickel," *Physical Review Letters*, vol. 26, no. 4, p. 192, 1971.

[8] M. Julliere, "Tunneling between ferromagnetic films," *Physics letters A*, vol. 54, no. 3, pp. 225–226, 1975.

[9] T. Miyazaki and N. Tezuka, "Spin polarized tunneling in ferromagnet/insulator/ferromagnet junctions," *Journal of magnetism and magnetic materials*, vol. 151, no. 3, pp. 403–410, 1995.

[10] J. S. Moodera, L. R. Kinder, T. M. Wong, and R. Meservey, "Large magnetoresistance at room temperature in ferromagnetic thin film tunnel junctions," *Physical Review Letters*, vol. 74, no. 16, p. 3273, 1995.

[11] D. Wang, C. Nordman, J. M. Daughton, Z. Qian, and J. Fink, "70% tmr at room temperature for sdt sandwich junctions with cofeb as free and reference layers," *Magnetics, IEEE Transactions on*, vol. 40, no. 4, pp. 2269–2271, 2004.

[12] W. Butler, X.-G. Zhang, T. Schulthess, and J. MacLaren, "Spin-dependent tunneling conductance of fe| mgo| fe sandwiches," *Physical Review B*, vol. 63, no. 5, p. 054416, 2001.

[13] J. Mathon and A. Umerski, "Theory of tunneling magnetoresistance of an epitaxial fe/mgo/fe (001) junction," *Physical Review B*, vol. 63, no. 22, p. 220403, 2001.

[14] S. Ikeda, J. Hayakawa, Y. Ashizawa, Y. Lee, K. Miura, H. Hasegawa, M. Tsunoda, F. Matsukura, and H. Ohno, "Tunnel magnetoresistance of 604% at 300 k by suppression of ta diffusion in cofeb/mgo/cofeb pseudo-spin-valves annealed at high temperature," *Applied Physics Letters*, vol. 93, no. 8, p. 2508, 2008.

[15] S. Yuasa and D. Djayaprawira, "Giant tunnel magnetoresistance in magnetic tunnel junctions with a crystalline mgo (0 0 1) barrier," *Journal of Physics D: Applied Physics*, vol. 40, no. 21, p. R337, 2007.

[16] B. Engel, J. Åkerman, B. Butcher, R. Dave, M. DeHerrera, M. Durlam, G. Grynkewich, J. Janesky, S. Pietambaram, N. Rizzo *et al.*, "A 4-mb toggle mram based on a novel bit and switching method," *Magnetics, IEEE Transactions on*, vol. 41, no. 1, pp. 132–136, 2005.

[17] I. Prejbeanu, M. Kerekes, R. Sousa, H. Sibuet, O. Redon, B. Dieny, and J. Nozieres, "Thermally assisted mram," *Journal of Physics: Condensed Matter*, vol. 19, no. 16, p. 165218, 2007.

[18] A. Khvalkovskiy, D. Apalkov, S. Watts, R. Chepulskii, R. Beach, A. Ong, X. Tang, A. Driskill-Smith, W. Butler, P. Visscher *et al.*, "Basic principles of stt-mram cell operation in memory arrays," *Journal of Physics D: Applied Physics*, vol. 46, no. 7, pp. 74 001–74 020, 2013.

[19] P. Gambardella and I. M. Miron, "Current-induced spin–orbit torques," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 369, no. 1948, pp. 3175–3197, 2011.

[20] K. Lewotsky, "Tech trends: Details on everspin's st-mram," *eetimes.com.* [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1280267

[21] T. W. Andre, J. J. Nahas, C. K. Subramanian, B. J. Garni, H. S. Lin, A. Omair, and W. L. Martino Jr, "A 4-mb 0.18-μm 1t1mtj toggle mram with balanced three input sensing scheme and locally mirrored unidirectional write drivers," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 1, pp. 301–309, 2005.

[22] "Everspin company." [Online]. Available: http://www.everspin.com/

[23] I. Prejbeanu, S. Bandiera, J. Alvarez-Hérault, R. Sousa, B. Dieny, and J. Nozieres, "Thermally assisted mrams: ultimate scalability and logic functionalities," *Journal of Physics D: Applied Physics*, vol. 46, no. 7, p. 074002, 2013.

[24] B. Cambou, "Match in place. a novel way to perform secure and fast user's authentication," *available online at www. crocus-technology. com*.

[25] R. Bishnoi, M. Ebrahimi, F. Oboril, and M. B. Tahoori, "Architectural aspects in design and analysis of sot-based memories," in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*. IEEE, 2014, pp. 700–707.

[26] S. Lee, K. Kang, and C.-M. Kyung, "Runtime thermal management for 3-d chip-multiprocessors with hybrid sram/mram l2 cache," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 520–533, 2015.

[27] J. Wang, X. Dong, and Y. Xie, "Oap: an obstruction-aware cache management policy for stt-ram last-level caches," in *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 2013, pp. 847–852.

[28] N. N. Mojumder, S. K. Gupta, S. H. Choday, D. E. Nikonov, and K. Roy, "A three-terminal dual-pillar stt-mram for high-performance robust memory applications," *Electron Devices, IEEE Transactions on*, vol. 58, no. 5, pp. 1508–1516, 2011.

[29] S. Kang and K. Lee, "Emerging materials and devices in spintronic integrated circuits for energy-smart mobile computing and connectivity," *Acta Materialia*, vol. 61, no. 3, pp. 952–973, 2013.

[30] X. Fong and K. Roy, "Low-power robust complementary polarizer stt-mram (cpstt) for on-chip caches," in *Memory Workshop (IMW), 2013 5th IEEE International*. IEEE, 2013, pp. 88–91.

[31] ——, "Complimentary polarizers stt-mram (cpstt) for on-chip caches," *Electron Device Letters, IEEE*, vol. 34, no. 2, pp. 232–234, 2013.

[32] H. Naeimi, C. Augustine, A. Raychowdhury, S.-L. Lu, and J. Tschanz, "Sttram scaling and retention failure," *Intel Technology Journal*, vol. 17, no. 1, pp. 54–75, 2013.

[33] P. Khalili and K. Wang, "Voltage-controlled mram: Status, challenges and prospects," *eetimes.com.* [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1280508&page_number=1

[34] J. G. Alzate, P. K. Amiri, P. Upadhyaya, S. S. Cherepov, J. Zhu, M. Lewis, R. Dorrance, J. Katine, J. Langer, K. Galatsis *et al.*, "Voltage-induced switching of nanoscale magnetic tunnel junctions," in *Electron Devices Meeting (IEDM), 2012 IEEE International*. IEEE, 2012, pp. 29–5.

[35] S. Kanai, M. Yamanouchi, S. Ikeda, Y. Nakatani, F. Matsukura, and H. Ohno, "Electric field-induced magnetization reversal in a perpendicular-anisotropy cofeb-mgo magnetic tunnel junction," *Applied Physics Letters*, vol. 101, no. 12, p. 122403, 2012.

[36] Y. Shiota, T. Nozaki, F. Bonell, S. Murakami, T. Shinjo, and Y. Suzuki, "Induction of coherent magnetization switching in a few atomic layers of feco using voltage pulses," *Nature materials*, vol. 11, no. 1, pp. 39–43, 2012.

[37] Y. Shiota, S. Miwa, T. Nozaki, F. Bonell, N. Mizuochi, T. Shinjo, H. Kubota, S. Yuasa, and Y. Suzuki, "Pulse voltage-induced dynamic magnetization switching in magnetic tunneling junctions with high resistance-area product," *Applied Physics Letters*, vol. 101, no. 10, p. 102406, 2012.

[38] P. K. Amiri, P. Upadhyaya, J. Alzate, and K. Wang, "Electric-field-induced thermally assisted switching of monodomain magnetic bits," *Journal of Applied Physics*, vol. 113, no. 1, p. 013912, 2013.

[39] W.-G. Wang, M. Li, S. Hageman, and C. Chien, "Electric-field-assisted switching in magnetic tunnel junctions," *Nature materials*, vol. 11, no. 1, pp. 64–68, 2012.

[40] K. Wang, J. Alzate, and P. K. Amiri, "Low-power non-volatile spintronic memory: Stt-ram and beyond," *Journal of Physics D: Applied Physics*, vol. 46, no. 7, p. 074003, 2013.

[41] I. M. Miron, G. Gaudin, S. Auffret, B. Rodmacq, A. Schuhl, S. Pizzini, J. Vogel, and P. Gambardella, "Current-driven spin torque induced by the rashba effect in a ferromagnetic metal layer," *Nature materials*, vol. 9, no. 3, pp. 230–234, 2010.

[42] L. Liu, C.-F. Pai, Y. Li, H. Tseng, D. Ralph, and R. Buhrman, "Spin-torque switching with the giant spin hall effect of tantalum," *Science*, vol. 336, no. 6081, pp. 555–558, 2012.

[43] F. Oboril, R. Bishnoi, M. Ebrahimi, and M. B. Tahoori, "Evaluation of hybrid memory technologies using sot-mram for on-chip cache hierarchy," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 34, no. 3, pp. 367–380, 2015.

[44] H. Noguchi, K. Kushida, K. Ikegami, K. Abe, E. Kitagawa, S. Kashiwada, C. Kamata, A. Kawasumi, H. Hara, and S. Fujita, "A 250-mhz 256b-i/o 1-mb stt-mram with advanced perpendicular mtj based dual cell for nonvolatile magnetic caches to reduce active power of processors," in *VLSI Technology (VLSIT), 2013 Symposium on*. IEEE, 2013, pp. C108–C109.

[45] K. Ikegami, H. Noguchi, C. Kamata, M. Amano, K. Abe, K. Kushida, E. Kitagawa, T. Ochiai, N. Shimomura, A. Kawasumi *et al.*, "A 4ns, 0.9 v write voltage embedded perpendicular stt-mram fabricated by mtj-last process," in *VLSI Technology, Systems and Application (VLSI-TSA), Proceedings of Technical Program-2014 International Symposium on*. IEEE, 2014, pp. 1–2.

[46] H. Noguchi, K. Ikegami, K. Kushida, K. Abe, S. Itai, S. Takaya, N. Shimomura, J. Ito, A. Kawasumi, H. Hara *et al.*, "7.5 a 3.3 ns-access-time 71.2μw/mhz 1mb embedded stt-mram using physically eliminated read-disturb scheme and normally-off memory architecture," in *Solid-State Circuits Conference-(ISSCC), 2015 IEEE International*. IEEE, 2015, pp. 1–3.

[47] R. Dorrance, J. G. Alzate, S. S. Cherepov, P. Upadhyaya, I. N. Krivorotov, J. A. Katine, J. Langer, K. L. Wang, P. K. Amiri, and D. Markovic, "Diode-mtj crossbar memory cell using voltage-induced unipolar switching for high-density mram," *Electron Device Letters, IEEE*, vol. 34, no. 6, pp. 753–755, 2013.

[48] K. Jabeur, L. Buda-Prejbeanu, G. Prenat, and G. Pendina, "Study of two writing schemes for a magnetic tunnel junction based on spin orbit torque," *International Journal of Electronics Science and Engineering*, vol. 7, no. 8, pp. 501–507, 2013.

[49] S. Rusu, S. Tam, H. Muljono, J. Stinson, D. Ayers, J. Chang, R. Varada, M. Ratta, S. Kottapalli, and S. Vora, "A 45 nm 8-core enterprise xeon processor," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 1, pp. 7–14, 2010.

[50] R. Sites, "It's the memory, stupid!" *Microprocessor Report*, vol. 10, no. 10, pp. 2–3, 1996.

[51] SEMICO, "Semico research corporation." [Online]. Available: http://www.semico.com/

[52] ITRS, "International technology roadmap for semiconductors." [Online]. Available: http://www.itrs.net/

[53] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen, "Circuit and microarchitecture evaluation of 3d stacking magnetic ram (mram) as a universal memory replacement," in *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*. IEEE, 2008, pp. 554–559.

[54] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3d stacked mram l2 cache for cmps," in *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*. IEEE, 2009, pp. 239–249.

[55] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," in *ACM SIGARCH computer architecture news*, vol. 37, no. 3. ACM, 2009, pp. 34–45.

[56] X. Wu, J. Li, L. Zhang, E. Speight, and Y. Xie, "Power and performance of read-write aware hybrid caches with non-volatile memories," in *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09.* IEEE, 2009, pp. 737–742.

[57] J. Li, C. J. Xue, and Y. Xu, "Stt-ram based energy-efficiency hybrid cache for cmps," in *VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th International Conference on*. IEEE, 2011, pp. 31–36.

[58] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for stt-ram using early write termination," in *Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*. IEEE, 2009, pp. 264–268.

[59] K.-W. Kwon, S. H. Choday, Y. Kim, and K. Roy, "Aware (asymmetric write architecture with redundant blocks): A high write speed stt-mram cache architecture," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 712–720, 2014.

[60] A. Jog, A. K. Mishra, C. Xu, Y. Xie, V. Narayanan, R. Iyer, and C. R. Das, "Cache revive: architecting volatile stt-ram caches for enhanced performance in cmps," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 243–252.

[61] E. Arima, H. Noguchi, T. Nakada, S. Miwa, S. Takeda, S. Fujita, H. Nakamura, and T. C. R. Center, "Fine-grain power-gating on stt-mram peripheral circuits with locality-aware access control," in *The Memory Forum (in conjounction with the 41 st International Symposium on Computer Architecture)*, 2014.

[62] M. P. Komalan, C. Tenllado, J. I. G. Pérez, F. T. Fernández, and F. Catthoor, "System level exploration of a stt-mram based level 1 data-cache," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 1311–1316.

[63] H. Noguchi, K. Ikegami, N. Shimomura, T. Tetsufumi, J. Ito, and S. Fujita, "Highly reliable and low-power nonvolatile cache memory with advanced perpendicular stt-mram for high-performance cpu," in *VLSI Circuits Digest of Technical Papers, 2014 Symposium on*. IEEE, 2014, pp. 1–2.

[64] F. A. Endo, D. Couroussé, and H.-P. Charles, "Micro-architectural simulation of embedded core heterogeneity with gem5 and mcpat," in *Proceedings of the 2015 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*. ACM, 2015, p. 7.

[65] P. Greenhalgh, "Big. little processing with arm cortex-a15 & cortex-a7," *ARM White paper*, pp. 1–8, 2011.

[66] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti *et al.*, "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, 2011.

[67] A. Butko, R. Garibotti, L. Ost, V. Lapotre, A. Gamatie, G. Sassatelli, and C. Adeniyi-Jones, "A trace-driven approach for fast and accurate simulation of manycore architectures," in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*. IEEE, 2015, pp. 707–712.

[68] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt, "The m5 simulator: Modeling networked systems," *IEEE Micro*, no. 4, pp. 52–60, 2006.

[69] M. M. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood, "Multifacet's general execution-driven multiprocessor simulator (gems) toolset," *ACM SIGARCH Computer Architecture News*, vol. 33, no. 4, pp. 92–99, 2005.

[70] A. Hansson, N. Agarwal, A. Kolli, T. Wenisch, and A. N. Udipi, "Simulating dram controllers for future system architecture exploration," in *Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 201–210.

[71] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 7, pp. 994–1007, 2012.

[72] S. J. Wilton and N. P. Jouppi, "Cacti: An enhanced cache access and cycle time model," *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 5, pp. 677–688, 1996.

[73] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "Cacti 6.0: A tool to model large caches," *HP Laboratories*, pp. 22–31, 2009.

[74] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The splash-2 programs: Characterization and methodological considerations," in *ACM SIGARCH Computer Architecture News*, vol. 23, no. 2. ACM, 1995, pp. 24–36.

[75] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM, 2008, pp. 72–81.

[76] C. Bienia, S. Kumar, and K. Li, "Parsec vs. splash-2: A quantitative comparison of two multithreaded benchmark suites on chip-multiprocessors," in *Workload Characterization, 2008. IISWC 2008. IEEE International Symposium on*. IEEE, 2008, pp. 47–56.

[77] T. Kaukalias and P. Chatzimisios, "Internet of things (iot)."

[78] S. Karnouskos, P. J. Marrón, G. Fortino, L. Mottola, and J. R. Martínez-de Dios, *Applications and markets for cooperating objects*. Springer, 2014.

[79] K. Ando, S. Fujita, J. Ito, S. Yuasa, Y. Suzuki, Y. Nakatani, T. Miyazaki, and H. Yoda, "Spin-transfer torque magnetoresistive random-access memory technologies for normally off computing," *Journal of Applied Physics*, vol. 115, no. 17, p. 172607, 2014.

[80] STM32L1, "Datasheet." [Online]. Available: http://www.st.com/web/en/resource/technical/document/datasheet/CD00277537.pdf

[81] C. Santifort, "Amber arm-compatible core," *OpenCores. org*, 2010.

[82] T. Na, K. Ryu, J. Kim, S.-H. Kang, and S.-O. Jung, "A comparative study of stt-mtj based non-volatile flip-flops," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 109–112.

[83] W. Zhao, M. Moreau, E. Deng, Y. Zhang, J.-M. Portal, J.-O. Klein, M. Bocquet, H. Aziza, D. Deleruyelle, C. Muller *et al.*, "Synchronous non-volatile logic gate design based on resistive switching memories," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 2, pp. 443–454, 2014.

[84] Y. Zhang, E. Deng, J. Klein, D. Querlioz, D. Ravelosona, C. Chappert, W. S. Zhao, M. Moreau, J. Portal, M. Bocquet *et al.*, "Synchronous full-adder based on complementary resistive switching memory cells," in *New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International*. IEEE, 2013, pp. 1–4.

[85] K. Jabeur, G. Di Pendina, F. Bernard-Granger, and G. Prenat, "Spin orbit torque non-volatile flip-flop for high speed and low energy applications," *Electron Device Letters, IEEE*, vol. 35, no. 3, pp. 408–410, 2014.

[86] J. Wang, Y. Liu, H. Yang, and H. Wang, "A compare-and-write ferroelectric nonvolatile flip-flop for energy-harvesting applications," in *Green Circuits and Systems (ICGCS), 2010 International Conference on*. IEEE, 2010, pp. 646–650.

[87] N. Jovanović, O. Thomas, E. Vianello, J. Portal, B. Nikolić, and L. Naviner, "Oxram-based non volatile flip-flop in 28nm fdsoi," 2014.

[88] B. Jovanovic, R. M. Brum, and L. Torres, "Comparative analysis of mtj/cmos hybrid cells based on tas and in-plane stt magnetic tunnel junctions," *Magnetics, IEEE Transactions on*, vol. 51, no. 2, pp. 1–11, 2015.

[89] D. Chabi, W. Zhao, E. Deng, Y. Zhang, N. Ben Romdhane, J.-O. Klein, and C. Chappert, "Ultra low power magnetic flip-flop based on checkpointing/power gating and self-enable mechanisms," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 6, pp. 1755–1765, 2014.

[90] J.-M. Choi, C.-M. Jung, and K.-S. Min, "Pcram flip-flop circuits with sequential sleep-in control scheme and selective write latch," *Journal of Semiconductor Technology and Science*, vol. 13, no. 1, pp. 58–64, 2013.

[91] Y. Guillemenet, L. Torres, and G. Sassatelli, "Non-volatile run-time field-programmable gate arrays structures using thermally assisted switching magnetic random access memories," *IET Computers & Digital Techniques*, vol. 4, no. 3, pp. 211–226, 2010.

[92] C. Holland, "First mram-based fpga taped-out," *Website: http://www. eetimes. com/General/DisplayPrintViewContent? contentItemId*, vol. 4200035, 2010.

[93] W. Zhao, E. Belhaire, C. Chappert, and P. Mazoyer, "Spin transfer torque (stt)-mram–based runtime reconfiguration fpga circuit," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 9, no. 2, p. 14, 2009.

[94] S. Paul, S. Mukhopadhyay, and S. Bhunia, "A circuit and architecture codesign approach for a hybrid cmos–sttram nonvolatile fpga," *Nanotechnology, IEEE Transactions on*, vol. 10, no. 3, pp. 385–394, 2011.

[95] A. Ahari, H. Asadi, B. Khaleghi, and M. B. Tahoori, "A power-efficient reconfigurable architecture using pcm configuration technology," in *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014, p. 336.

[96] O. Turkyilmaz, S. Onkaraiah, M. Reyboz, F. Clermidy, C. Anghel, J.-M. Portal, M. Bocquet *et al.*, "Rram-based fpga for "normally off, instantly on" applications," *Journal of Parallel and Distributed Computing*, vol. 74, no. 6, pp. 2441–2451, 2014.

[97] Y. Wang, Y. Liu, S. Li, D. Zhang, B. Zhao, M.-F. Chiang, Y. Yan, B. Sai, and H. Yang, "A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops," in *ESSCIRC (ESSCIRC), 2012 Proceedings of the*. IEEE, 2012, pp. 149–152.

[98] S. Khanna, S. Bartling, M. Clinton, S. Summerfelt, J. Rodriguez, and H. McAdams, "Zero leakage microcontroller with 384ns wakeup time using fram mini-array architecture," in *Solid-State Circuits Conference (A-SSCC), 2013 IEEE Asian*. IEEE, 2013, pp. 21–24.

[99] S. Khanna, S. C. Bartling, M. Clinton, S. Summerfelt, J. A. Rodriguez, and H. P. McAdams, "An fram-based nonvolatile logic mcu soc exhibiting 100% digital state retention at 0 v achieving zero leakage with 400-ns wakeup time for ulp applications," *Solid-State Circuits, IEEE Journal of*, vol. 49, no. 1, pp. 95–106, 2014.

[100] J. S. Meena, S. M. Sze, U. Chand, and T.-Y. Tseng, "Overview of emerging nonvolatile memory technologies," *Nanoscale research letters*, vol. 9, no. 1, pp. 1–33, 2014.

[101] H. Koike, T. Ohsawa, S. Ikeda, T. Hanyu, H. Ohno, T. Endoh, N. Sakimura, R. Nebashi, Y. Tsuji, A. Morioka *et al.*, "A power-gated mpu with 3-microsecond entry/exit delay using mtj-based nonvolatile flip-flop," in *Solid-State Circuits Conference (A-SSCC), 2013 IEEE Asian*. IEEE, 2013, pp. 317–320.

[102] R. P. Weicker, "Dhrystone: a synthetic systems programming benchmark," *Communications of the ACM*, vol. 27, no. 10, pp. 1013–1030, 1984.

[103] J. Borgeson, S. Shauer, and H. Diewald, "Benchmarking mcu power consumption for ultra-low-power applications," *White paper*, 2012.

[104] Q. Stainer, L. Lombard, K. Mackay, D. Lee, S. Bandiera, C. Portemont, C. Creuzet, R. Sousa, and B. Dieny, "Self-referenced multi-bit thermally assisted magnetic random access memories," *Applied Physics Letters*, vol. 105, no. 3, p. 032405, 2014.

[105] M. Poremba and Y. Xie, "Nvmain: An architectural-level main memory simulator for emerging non-volatile memories," in *VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on*. IEEE, 2012, pp. 392–397.

[106] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*. IEEE, 2009, pp. 469–480.

[107] J. Clement, B. Mussard, D. Naccache, and L. Torres, "Implementation of aes using nvm memories based on comparison function," in *VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on*. Accepted for publication.

[108] I. Vasyltsov, E. Hambardzumyan, Y.-S. Kim, and B. Karpinskyy, "Fast digital trng based on metastable ring oscillator," in *Cryptographic Hardware and Embedded Systems–CHES 2008*. Springer, 2008, pp. 164–180.

[109] A. Maiti, R. Nagesh, A. Reddy, and P. Schaumont, "Physical unclonable function and true random number generator: A compact and scalable implementation," in *Proceedings of the 19th ACM Great Lakes symposium on VLSI*. ACM, 2009, pp. 425–428.

[110] A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, and K. Ando, "Spin dice: A scalable truly random number generator based on spintronics," *Applied Physics Express*, vol. 7, no. 8, p. 083001, 2014.

[111] T. Devolder, J. Hayakawa, K. Ito, H. Takahashi, S. Ikeda, P. Crozat, N. Zerounian, J.-V. Kim, C. Chappert, and H. Ohno, "Single-shot time-resolved measurements of nanosecond-scale spin-transfer induced switching: Stochastic versus deterministic aspects," *Physical review letters*, vol. 100, no. 5, p. 057206, 2008.

[112] L. Zhang, X. Fong, C.-H. Chang, Z. H. Kong, and K. Roy, "Highly reliable memory-based physical unclonable function using spin-transfer torque mram," in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 2169–2172.

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

[114] E. I. Vatajelu, G. Di Natale, M. Indaco, and P. Prinetto, "Stt mram-based pufs," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 872–875.