Conception innovante de circuits logiques et mémoires en technologie CMOS/Magnétique

Rana Alhalabi

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **Nano-Electronique et Nano-Technologies**

Arrêté ministériel : 25 mai 2016

Présentée par

« Rana ALHALABI »

Thèse dirigée par **Ioan-Lucian PREJBEANU** et co-encadrée par **Grégory DI PENDINA** et **Etienne NOWAK**

préparée au sein des **Laboratoires CEA-LETI** et **SPINTEC** dans **l'École Doctorale EEATS de l'INP Grenoble**

Conception innovante de circuits logiques et mémoires en technologie CMOS/Magnétique

Thèse soutenue publiquement le **« 11 Octobre 2019 »**, devant le jury composé de :

Mme Lorena ANGHEL
Professeur INP Grenoble, TIMA, Présidente
Mr Gilles SASSATELLI
Directeur de Recherche CNRS, LIRMM Montpellier, Rapporteur
Mr Hassen AZIZA
Maître de Conférences Université Aix-Marseille, IM2NP, Rapporteur
Mr Jean-Baptiste RIGAUD
Maître de conférences EMSE, SAS Gardanne, Examinateur
Mr Bruno ALLARD
Professeur INSA Lyon, Laboratoire Ampère Lyon, Examinateur
Mr Ioan-Lucian PREJBEANU
Ingénieur- chercheur CEA, SPINTEC Grenoble, Directeur de thèse
Mr Gregory DI PENDINA
Ingénieur de Recherche CNRS, SPINTEC Grenoble, Encadrant Invité
Mr Etienne NOWAK
Ingénieur-chercheur CEA, LETI Grenoble, Invité

Remerciement

Par où commencer.... Certes, l'aventure doctorale n'est pas juste de longues heures de travail dans un bureau, mais aussi un fort soutien et une collaboration de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

En premier lieu, je tiens à remercier mon directeur de thèse et chef de laboratoire SPINTEC, Lucian PREJBEANU, pour la confiance et le support qu'il m'a accordée en acceptant la direction scientifique de mes travaux. Merci pour l'excellent accueil au sein du laboratoire, mais aussi pour tes conseils et tes encouragements.

J'aimerais ensuite remercier tout particulièrement mon encadrant, Gregory DI PENDINA de m'avoir parfaitement encadrée tout au long de mon travail. Merci d'avoir su trouver les bons axes de cette thèse correspondant si bien à mes attentes et à ma carrière. Merci pour ta grande disponibilité (même pendant tes vacances), tes conseils et ton respect sans faille des délais serrés de relecture des papiers et surtout du manuscrit. Merci pour ta simplicité, ta gentillesse, ton humour, ton soutien et tes remarques pertinentes dans divers sujets. Tu as eu un réel impact positif sur mon parcours professionnel et personnel. J'ai extrêmement apprécié tes qualités humaines d'écoute et de compréhension et je te suis infiniment reconnaissante pour cela... !

Je tiens à remercier également mon encadrant LETI et chef de laboratoire LCM, Etienne NOWAK, de m'avoir fait confiance sur la méthodologie utilisée dans cette thèse. Tes conseils, remarques et avis m'ont été très importants et constructifs.

Mes remerciements vont également aux membres du jury qui ont accepté avec bienveillance de juger mes travaux. Merci à Lorena ANGHEL, Gilles SASSATELLI, Hassen AZIZA, Jean-Baptiste RIGAUD et Bruno ALLARD.

Ma gratitude s'adresse spécialement à l'équipe IC design dirigée par Guillaume PRENAT. Merci Guillaume pour toutes les réunions, les discussions très intéressantes et ta bonne humeur. Je suis particulièrement reconnaissante à François DUHEM pour m'avoir aidée sur la partie numérique de ce travail, en particulier lors de cette dernière année qui a été spécialement intense. Merci de t'être impliqué gentiment dans mes travaux, merci pour ta patience et tes grandes qualités d'écoute. Je garderai longtemps le souvenir de ces heures passées à debugger les codes de Secret blaze. Sans oublier de remercier Cadence pour m'avoir appris la patience et la tolérance grâces aux superbes plantages et bug.

Mes remeciements vont encore à tous les anciens collègues de l'équipe: Kotb, Pierre et Jeremy. Les nombreux échanges que j'ai pu avoir avec vous m'ont énormément apportée.

Je n'oublie pas non plus toutes les personnes qui ont participé de près ou de loin à l'accomplissement de ces travaux. La liste est longue, et je citerai particulièrement, Luc, ancien colloc de bureau :p tu étais mon point de repère et tu ne m'as jamais déçue, merci pour tout.. ! Odilia, tu m'as fait beaucoup rire avec tes bêtises et tes expressions venues d'une autre planète ! Merci pour ton amitié et pour tous les bons moments et les sorties que l'on a partagés ensemble. Merci encore à Gilbert, Sabrina, Léa, Laura, Cécile et tous les collègues de SPINTEC et du LETI.

Je remercie également les secrétaires côté SPINTEC et côté LETI pour leur aide et leur gentillesse.

Mes remerciements vont également à mes amis et à ma famille, pour le soutient sans faille pendant la totalité de mes études. Votre soutien et votre amour étaient également un facteur déterminant dans la réussité de cette thèse. Merci maman, ma soeur, mes tantes, mon oncle et mes cousins.

Et pour finir, une pensée particulière s'adresse à mon père qui aurait été très fier de voir l'aboutissement de ce travail..!

Avec tout mon amour
Rana

Table des matières

Table des figures

Glossaire

ADC	Analog to Digital Converter
AF	AntiFerromagnétique
ALU	Arithmetic Logic Unit
ASIC	Application-Specific Integrated Circuit
BL	Bit Line
BLE	Eléments Logiques de Base
CB	Connect Box
CBRAM	Conductive Bridge RAM
CLB	Configurable Logic Block
DES	Data Encryption Standard
DRAM	Dynamic Random Access Memory
DRC	Design Rule Cheking
DUT	Device Under Test
EPROM	Erasable Programmable ROM
FeRAM	Ferroelectric RAM
FPGA	Field Programmable Gate Array
HDD	Hard Disk Drives
IP	Intellectual Property
IoT	Internet of Things
JTM	Jonction Tunnel Magnétique
LEF	Library Exchange Foramat
LUT	Look-Up Table
LVS	Layout Versus Schematic
MAD	Memory Advanced Demonstrator
MCU	microcontrôleur
MIM	Métal-Isolant-Métal
MRAM	Magnetic RAM
MPW	Multi Project Wafer
OTP	One Time Programmable
OxRAM	Oxide RAM
PCRAM	Phase Change RAM
PDK	Process Design Kit
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
ROM	Read Only Mmemory
RRAM	Resistive RAM

RTL	Register Transfer Level
RWL	Read Word Line
SA	Sense Amplifier
SAF	AntiFerromagnétique artificiel (Synthetic antiferromagnets)
SB	Switch Boxe
SCM	Storage Class Memory
SOT	Spin Orbit Torque
SRAM	Static Random Access Memory
SSD	Solid State Device
STIL	Standard Test Interface Langage
STT	Spin Transfert Torrque
TAS	Thermally Assisted Switching
TMR	Tunnel Magneto Resistance
UART	Universal Asynchronous Receiver-Transmitter
UVROM	UltraViolets Programmable Read Only Memory
WEN	Write ENable
WL	World Line
WWL	Write Word Line
2T-1JTM	2 Transistors- 1 Jonction Tunnel Magnétique
1T-1D-1JTM	1 Transistors-1 Diode-1 Junction Tunnel Magnétique
1T-1JTM	1 Transistor- 1 Jonction Tunnel Magnétique

Introduction générale

Le marché des mémoires a connu un essor exceptionnel au cours des dernières décennies. Sa réussite est à l'origine de sa capacité à répondre aux attentes des circuits intégrés électroniques comme les processeurs, les ASIC (Application Specific Integrated Circuit) ou encore les circuits logiques programmables. La technologie mémoire est un élément central de ces systèmes. Aujourd'hui, la mémoire "idéale" qui combine à la fois une endurance infinie, une densité élevée, une grande vitesse avec une consommation faible n'existe pas. Les technologies actuelles reposent sur un compromis entre ces différentes performances au regard de l'application. Deux caractéristiques s'opposent généralement : le temps d'accès et la densité. Au-delà des problématiques de miniaturisation, la consommation énergétique des mémoires occupe une part de plus en plus importante dans la consommation globale des puces. Pour diminuer cette consommation, de nouvelles technologies de mémoires intégrent depuis peu le marché des semiconducteurs. Parmi ces mémoires, la mémoire MRAM (Magnetic RAM) passe de simple « candidat potentiel » il y a quelques années à des mémoires fabriquées par de grandes industries, aujourd'hui disponibles sur le marché, suscitant un fort intérêt général dans le monde industriel et académique de la microélectronique.

Dans ce contexte, nous avons souhaité à travers cette thèse évaluer les intérêts et les atouts de cette mémoire émergente dans des systèmes électroniques différents en proposant de nouvelles architectures innovantes à base de MRAM. Plus précisement, l'objectif de cette thèse était de proposer une alternative à l'utilisation de SRAM en proposant des circuits innovants permettant de tirer avantage des atouts de la technologie magnétique. Pour cela, en plus de la partie conception, il était important d'être capable de mettre en place un flot de conception numérique hybride CMOS/MRAM permettant la conception de circuits intégrés complexes intégrant des jonctions tunnel magnétiques, et plus particulièrement des IPs de type MRAM.

Cette thèse s'est déroulée en collaboration entre deux laboratoires: d'une part Leti-LCM, Laboratoire des Composants Mémoires pour la partie monofabrication et d'autre part Spintec, laboratoire de recherche sur la spintronique pour la partie conception de circuits intégrés. C'est au laboratoire Spintec que la technologie MRAM à écriture assistée thermique (TAS-MRAM) a été inventée et développée, ce qui a donné lieu à la création de la société « Crocus Technology » en 2004 pour sa commercialisation.

Ce manuscrit présente mes travaux de recherche qui s'articulent généralement au-

tour de trois principaux axes :

- La conception de circuits hybrides CMOS/magnétique de type LUT (Look Up Table), l'élément de base d'un FPGA (Field Programmable Gate Array)
- La conception d'une mémoire embarquée en technologie magnétique SOT (Spin Orbit Torque) afin de l'intégrer dans l'architecture d'un processeur
- L'étude d'un processeur Secretblaze embarquant des mémoires magnétiques de différents types de technologie, Spin Transfert Torque (STT) et Spin Orbit Torque (SOT).

Ces parties seront décrites en détail dans les prochains chapitres.

Chapitre 1

Etat de l'art

Motivation

L'objectif de ce chapitre est de rappeler l'état de l'art des technologies de mémoires dans un système électronique. Après un bref rappel des mémoires émergentes non-volatiles, un focus un peu détaillé sera présenté sur la technologie de mémoires magnétiques, leur intérêts et leurs applications. Cette étude permet de montrer l'avantage de la technologie MRAM dans laquelle s'inscrivent les travaux de recherche menés pendant ma thèse.

1.1 Rôle des mémoires dans les systèmes électroniques

Au sein de l'unité de calcul d'un système électronique, le choix technologique des mémoires est un élément clef. Les caractéristiques et les performances des mémoires ont un impact capital sur le fonctionnement intrinsèque du système. Aujourd'hui, la mémoire idéale, autrement dit la mémoire qui combine à la fois une endurance infinie, une densité élevée, une consommation réduite, un coût très faible avec une vitesse assez importante n'existe pas. C'est pourquoi les technologies actuelles reposent sur un compromis entre ces différentes performances au regard de l'application visée. La solution est donc de ne pas se limiter à une technologie de mémoire mais d'adopter plutôt une hiérarchie de mémoire de façon bien définie.

La figure 1.1 représente la hiérarchie de mémoire que l'on peut trouver dans les systèmes électroniques, tels que les ordinateurs. Classiquement, un système informatique est formé d'une unité de calcul (tel que le processeur, ALU pour Arithmetic Logic Unit), qui va réaliser toutes les opérations et gérer toutes les tâches. Cette unité communique également avec des mémoires adjacentes où les données sont stockées afin de réaliser les opérations nécessaires. Les mémoires, opérant en lien direct avec le processeur, devraient donc avoir un temps d'accès très court afin d'assurer la rapidité de la communication.

Globalement, l'unité de calcul utilise des registres composés principalement de bascules de mémorisation type SRAM (pour Static Random Access Memory) pour sauvegarder les données nécessaires au calcul à un instant donné avec un temps d'accès < 1 ns[15]. Les mémoires de type cache de niveau 1, très rapides, et de niveaux 2 et 3, sont également des SRAMs avec un temps d'accès d'environ 1 et 10 ns respectivement. Aujourd'hui, la SRAM est la mémoire la plus rapide. Son architecture repose sur 6 transistors pour sauvegarder 1 bit de donnée, ce qui rend cette cellule gourmande en surface.

La mémoire centrale est une mémoire DRAM (pour Dynamic Random Access Memory) constituée d'un transistor et d'un condensateur pour sauvegarder 1 bit. Cette technologie de mémoire est la solution la plus bénéfique en densité avec un faible coût pour le stockage haute densité. En revanche, elle est moins rapide que la SRAM avec un temps d'accès d'environ 60 ns [16]. Etant donné que les condensateurs se déchargent, une DRAM nécessite un rafraîchissement régulier des données d'environ 15 ns pour sauvegarder l'information[17]. Ces deux types de mémoires SRAM et DRAM sont volatiles, ce qui signifie qu'elles perdent leurs données en cas de coupure de l'alimentation électrique. On note enfin les mémoires de stockage des disques durs

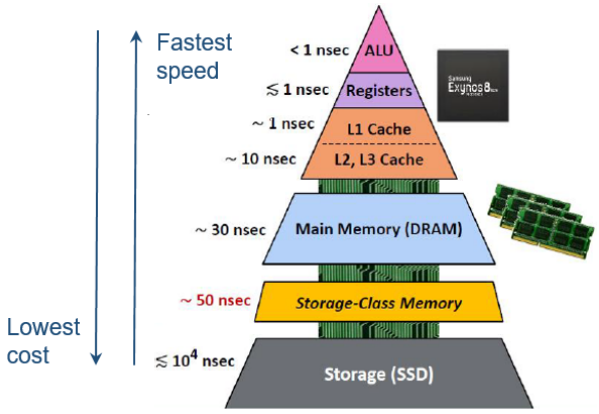


Figure 1.1: Hiérarchie mémoire standard dans des systèmes de calculs électroniques [1]

ou des mémoires Flash (SSD : Solid State Device) qui utilisent un transistor MOS possédant une grille flottante. Ces mémoires sont extrêmement denses, mais très lentes avec un temps t'accès > 50 ns et une endurance limitée[18]. Contrairement aux mémoires SRAMs et DRAMs, ces mémoires sont non-volatiles, permettant de conserver l'information même en l'absence de l'alimentation [19].
Ces caractéristiques nous permettent d'observer qu'il est en général nécessaire de combiner plusieurs types de mémoires dans une hiérarchie permettant d'associer toutes les performances. L'illustration de la figure 1.1 montre clairement que plus on est proche du cœur de calcul, plus on a besoin de mémoires rapides et peu denses, ne contenant que les données nécessaires au calcul. Au contraire, plus on est loin, les mémoires sont plus lentes mais avec une grande capacité de stockage. Cette hiérarchie est générique et souvent utilisée pour des applications haute performance. C'est pourquoi, pour des applications spécifiques de type IoT (Internet of Things) ou capteurs par exemple, la hiérarchie de mémoire est souvent modifiée.

1.2 Limitation des solutions actuelles

Au vu des besoins mémoires à l'échelle industrielle, les principaux critères d'une mémoire performante sont liés au type d'application où elle sera intégrée. On note parmi ces critères :

- Le coût de fabrication
- Le temps d'accès et le temps d'écriture : temps nécessaire pour sauvegarder un état ou une donnée
- L'énergie consommée (qui dépend en particulier des tensions et des courants ap-

pliqués liée à leur latence)

- La durée de rétention des données
- L'endurance, soit la capacité de maintenir dans le temps un état
- La compréhension des phénomènes physiques
- La miniaturisation, c'est à dire la portabilité vers un nœud technologique avancé

Jusqu'à présent, chacune des technologies citées précédemment est confrontée à des défis technologiques variés. La taille importante des SRAMs, et les courants de fuite des capacités des DRAMs qui réduisent la rétention et qui exigent un rafraîchissement régulier de ces mémoires, restent un vrai challenge. En outre, la limitation majeure de ces mémoires est la volatilité, ce qui augmente potentiellement la consommation statique du système.
Parmi les mémoires non volatiles, la technologie flash,inventée dans les années 1980[2], continue de dominer le marché. En fait, pour des cellules de petites dimensions, des limitations électriques se présentent telles que la réduction du nombre d'électrons stockés pour sauvegarder l'information ainsi que la fiabilité. Cette dernière diminue lorsque la taille de la cellule diminue [2]. Enfin, la programmation de ces mémoires exige une tension de l'ordre 3.5 à 10 V, ce qui augmente forcément la consommation totale ainsi que la surface, car des blocs spécifiques sont rajoutées afin de génerer ces tensions.

Vis- à -vis de ces limitations, changer le mode de stockage de l'information est devenu une préoccupation prépondérante. Le but est d'intégrer ces mémoires alternatives ou « émergentes » dans les systèmes embarqués pour des nœuds technologiques avancés.

1.3 Les mémoires émergentes non volatiles

Les mémoires émergentes non volatiles sont en développement depuis 15 ans avec l'objectif de prendre le relais des mémoires actuelles citées ci-dessus. Selon le cabinet Yole Développement, les technologies émergentes décollent enfin avec un marché qui devrait dépasser 1 milliard de dollars en 2019 et 7 milliards de dollars en 2023 [20]. Le développement de ces mémoires met en jeu différents principes physiques (magnétorésistance, ferroélectricité, ...) qui permettent de stocker l'information selon un état distinct de la résistance de la cellule. Ces nouvelles technologies ne s'appuient plus sur le stockage des charges. Les données sont plutôt représentées sous la forme d'un état

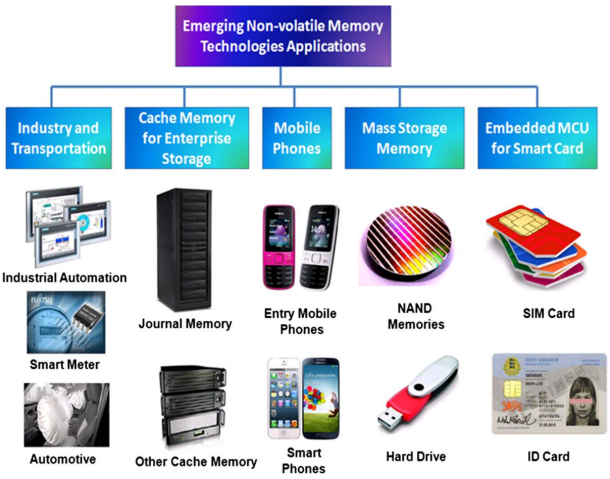


Figure 1.2: Domaine d'utilisation de mémoires émergentes non volatiles par diverses applications de l'industrie électronique [2]

de résistance selon le phénomène physique adopté.

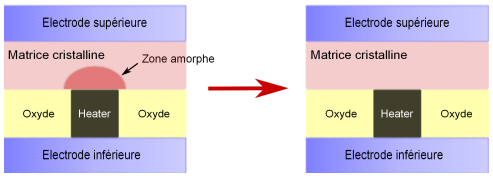
Nous présentons par la suite les technologies mémoires les plus étudiées actuellement, qui offrent au système différentes opportunités et couvrent un large spectre des applications dans le marché industriel. La figure 1.2 illustre l'utilisation de mémoires émergentes non volatiles par diverses applications de l'industrie électronique (Automobile, clés USB, smartphone, tablette numérique et ID card, ..)

1.3.1 Mémoires à changement de phase (PCRAM)

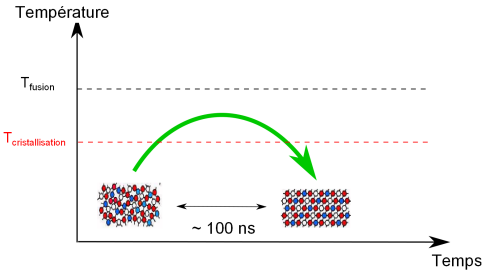
Les mémoires PCRAM pour Phase Change RAM, sont parmi les technologies les plus prometteuses pour les futures générations de mémoires non volatiles. La structure de base de la PCRAM est illustrée sur la figure 1.3a où l'information est stockée sous forme de phase soit amorphe soit cristalline d'un matériau de type chalcogénure [21]. Ces deux états représentent donc deux résistances distinctes ce qui les différencie du point de vue électrique. Si le point mémoire est amorphe, la résistance est forte et le bit sauvegardé est à '0' logique alors que si le point mémoire est cristallisé, la résistance est faible et le bit est à '1' logique.
Les transitions de phase s'effectuent grâce à une élévation locale de la température obtenue par effet Joule permettant soit la cristallisation du chalcogénure soit son amorphisation en fonction de signaux électriques de programmation.

Cette technologie de mémoire est capable de supporter une réduction drastique de ses dimensions et offre des temps d'accès courts avec une densité de stockage élevée [22]. Cependant, la vitesse de programmation est lente et sa consommation est élevée.

De plus pour des températures élevées la rétention est limitée. Parmi les entreprises qui développent cette technologie, Intel et Micron ont introduit la mémoire '3D-Xpoint' sur le marché depuis 2017 pour les applications SCM (Storage Class Memory)[4].



(a) structure de base d'une cellule PCRAM



(b) Le changement de phase de la PCRAM en fonction de la temperature appliquée

Figure 1.3: Principe de fonctionnement d'une cellule mémoire PCRAM.

1.3.2 Mémoire ferroélectrique (FeRAM)

Cette technologie est basée sur une structure simple: une capacité en série avec un transistor, similaire à celle d'une DRAM [23], [24] (voir figure figure 1.4). L'information est stockée sous forme de polarisation ferroélectrique dans une couche utilisée (HFO_2) en guise de diélectrique du condensateur, apportant à son tour la non volatilité. La commutation de la mémoire est réalisée par l'application d'un champ électrique aux bornes des électrodes du condensateur pour renverser son état de polarisation. Pour écrire l'état logique '0', quel que soit l'état initial du matériau ferroélectrique, il suffit d'appliquer un champ éléctrique positif supérieur à un certain seuil, alors qu'un champ éléctrique négatif permet de fixer la polarisation dans l'état rémanent négatif et ainsi d'inscrire un '1' logique.

La technologie FeRAM représente un bon candidat de mémoire non volatile pour des applications de l'IoT puisqu'elle possède une faible consommation avec une vitesse d'accès rapide (10 ns)[25]. Néanmoins, des limitations intrinsèques se présentent liées à la réduction de l'épaisseur de la couche ferroélectrique. Cette reduction est imposée par la diminution de la tension de programmation qui risque de dégrader ses propriétés diélectriques avec le cyclage en écriture.

1.3.3 Mémoire résistive filamentaire (RRAM)

Cette technologie regroupe deux familles de technologies, les OxRAMs (pour Oxide RAM) et les CBRAMs (pour Conductive Bridge RAM), différentiables par la physique

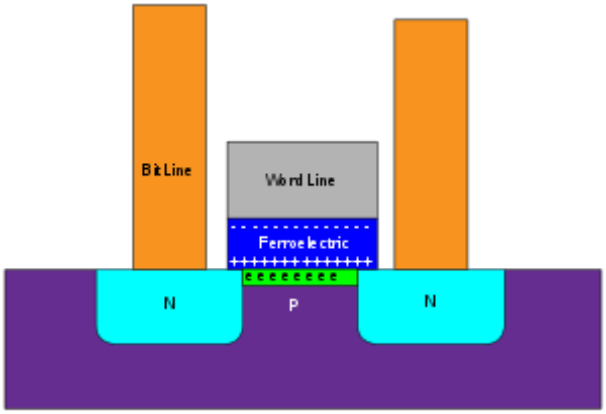


Figure 1.4: Structure de base d'une cellule FeRAM

de commutation.

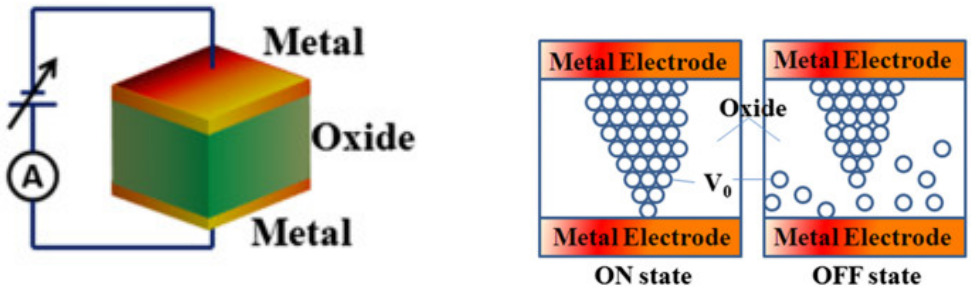
- Les OxRAM sont constituées principalement d'un oxyde métallique où le filament conducteur est à base de lacunes d'oxygène [26]
- Les CBRAM constituées principalement d'un électrolyte solide où le filament conducteur est formé d'Ag ou de Cu provenant d'une électrode active [27]

Globalement, la cellule mémoire résistive filamentaire consiste en un empilement de type métal-isolant-métal (MIM) où l'information est stockée sous forme d'une résistance électrique (voir figure 1.5a). Quelque soit le matériau utilisé, le principe de la commutation de la cellule reste similaire. Une tension est appliquée aux bornes des électrodes assurant la formation d'un filament de faible résistance dans le matériau. Une tension plus forte assure donc la rupture de ce filament et une augmentation de la résistance électrique[28].

Les atouts de cette mémoire en terme de facilité de fabrication, de fort potentiel pour la réduction de sa taille, et de sa vitesse de programmation sont particulièrement attractifs en regard des applications IoT[28]. En revanche, le principal verrou de cette mémoire est sa faible endurance, d'environ 10^6 [29]. Notons que la société Crossbar est parmi les leaders principaux de cette technologie.

1.3.4 Mémoire magnétique (MRAM)

La MRAM est le fruit de recherches menées dans le domaine de la spintronique. Elle s'est développée dans les années 90 [2]. Son principe repose sur l'orientation de l'aimantation de couches ferromagnétiques. L'élement de base de cette mémoire est une Jonction Tunnel Magnétique (JTM): deux couches ferromagnétiques séparées par une couche isolante servant de barrière tunnel, comme illustré sur la figure 1.6. En réalité, la



(a) structure de base d'une cellule RRAM

(b) Le commutation de la RRAM en formant la formation ou la rupture d'un filament (lacune)conducteur appliquée

Figure 1.5: Principe de fonctionnement d'une cellule mémoire RRAM.

monofabrication consiste à déposer une multitude de couches telles que Co, Fe, B, Pt, Te, MgO,...pour former une MTJ. Les états parallèle et antiparallèle de l'aimantation des deux couches permettent d'avoir deux états de résistance qui correspondent aux niveaux '0' et '1' logique.

Cette technologie possède un temps d'accès rapide avec une endurance élevée et une densité importante [30]. Cependant le coût de fabrication de la cellule mémoire est relativement élevée du fait de la complexité de l'empilement de couches.
Aujourd'hui, la société Everspin est le leader principal à commercialiser des produits industriels MRAM. Elle propose des mémoires de 1Go en 28nm ainsi que des mémoires commerciales de 256 Mb en 40 nm [31].

Mes travaux de recherches étant basés sur cette technologie, une partie dans la suite sera consacrée à décrire en détail le principe de fonctionnement de cette famille de mémoires non volatiles, car plusieurs déclinaisons existent.

1.4 Comparaison entre les mémoires émergentes non volatiles

Pour résumer, le tableau de la figure 1.7 montre les caractéristiques essentielles des mémoires actuelles présentées précédemment. Ces informations sont obtenues par une étude réalisée par Yole en 2016 [20].

Chaque technologie présente des éventuels avantages et inconvénients en terme d'endurance, de densité, de vitesse, de coût et de consommation. En outre, plusieurs grandes compagnies se sont positionnées sur le marché des mémoires émergentes

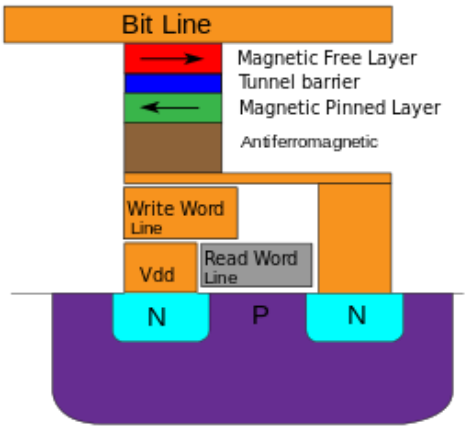


Figure 1.6: Vue de coupe de l'empilement d'une jonction tunnel magnétique du procédé CMOS/magnétique [3]

telles que Micron pour les PCRAM, Fujitsu pour les FeRAM, Everspin pour les MRAM où certaines de ces mémoires sont d'ailleurs déjà commercialisées depuis quelques années.

	Emerging Memories				Standard Memories	
	STT-MRAM	PCM « 3D XPoint »	RRAM	FeRAM	DRAM	Flash NAND
Non-Volatile	Yes	Yes	Yes	Yes	No	Yes
Endurance	High (10^{12})	Medium (10^{8})	Low (10^{6})	High (10^{14})	High (10^{15})	Low (10^{5})
Tech. node (2016)	40nm	20nm	130nm	X	1Xnm	15nm
Cell size (F^2)	Medium (6 -12)	X	Medium (6 -12)	Medium (<10)	Small (6 -10)	Very small (4)
Read latency	Fast (10 -20ns)	Fast (50 -100ns)	Medium (250ns)	Fast (~20ns)	Very fast (few ns)	Slow (100µs)
Power consumption	Medium (50pJ/bit)	Medium	Medium (6nJ/bit)	Medium (10pJ/bit)	Low	Very high
2016 price	High	Low	High	Low	Low	Very low
Suppliers	Everspin	Micron/Intel	Adesto/Fujitsu	Cypress	Samsung, Micron, SK Hynix	Samsung, Micron, Toshiba, SK Hynix, Intel

Figure 1.7: Comparaison des propriétés des pricipales technologies mémoire [4]

Il est difficile de comparer les familles de technologies émergentes au regard des performances pour choisir la mémoire universelle qui répond à tous les besoins. Selon les applications, chaque technologie peut être choisie ou combinée avec d'autres familles de mémoires pour être capable de couvrir un spectre large de performances.

Cependant, au delà du choix technologique, c'est la conception des systèmes intégrés qui va affecter ou définir les caractéristiques du système. Le choix de conception doit gérer au mieux la consommation, la vitesse ou d'autres paramètres en fonction de l'application souhaitée.

1.5 Intérêts de la MRAM

Nous avons vu qu'il n'existe pas parmi les futures mémoires émergentes la mémoire universelle qui peut être adaptée pour tous les types d'applications. Depuis plusieurs années et de plus en plus, de grandes entreprises evoluent et développent les mémoires émérgentes.

En ce qui concerne la technologie MRAM, le gain de performances de ces mémoires en terme d'endurance, de vitesse, de niveau de maturité et de consommation a rendu cette technologie une très bonne candidate parmi les mémoires non volatiles. Initialement étudiée par les chercherurs académiques, cette technologie a finalement muri et offre actuellement des prioriétés intéressantes semblables à celles présentées sur le marché.

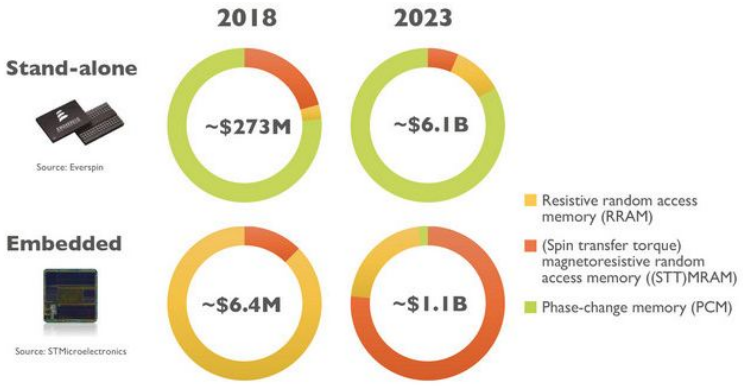


Figure 1.8: Prédiction d'évolution des mémoires émergentes non volatiles dans les applications embarquées et stand-alone [4]

L'année 2018 a été une année remarquable pour les mémoires MRAM, plusieurs investigations ont été menées par les grands acteurs du marché. Il s'agissait de Crocus Nano electronics pour une mémoire STT-MRAM (Spin Transfert Torrque) à 90 nm [32], Everspin pour son STT-MRAM 40 nm avec une capacité de stckage de 256 Mo commercial et son STT-MRAM 28 nm de 1 Gb [33]. Imec a montré la fonctionnalité d'une STT-MRAM de 5nm pour des applications de mémoires cahes [**?**]. On cite également Toshiba, Schneider et TDK qui ont montré beaucoup d'intérêts pour cette technologie.

Samsung a annoncé dès octobre 2017 que sa production de mémoires MRAM utilisant la technologie STT débuterait fin 2018[34]. Ainsi, d'autres acteurs majeurs de la microélectronique l'ont suivi, comme TSMC, GlobalFoundries, Micron et encore Qualcomm.
Selon Yole Developement, les ventes de mémoires MRAM autonomes passeront d'environ 50 millions de dollars en 2018 à environ 500 millions de dollars en 2023 [35], [20]. Les ventes de mémoires MRAM embarquées augmenteront beaucoup plus rapidement, passant de seulement 1 million de dollars en 2018 à plus de 800 millions de dollars en 2023 [35].

1.6 Fonctionnement général des MRAM

Les mémoires magnétiques MRAM représentent l'une des applications de la spintronique ou l'électronique de spin avec les capteurs magnétiques, HDD (Hard Disk Drives) ou les oscillateurs RF. Cette dernière est une branche de la physique de la matière condensée qui étudie la propriété physique du spin des électrons dans le but de stocker l'information.

1.6.1 Structure de base de la MTJ

La cellule de base d'une MRAM est constituée d'une Jonction Tunnel Magnétique (JTM) qui stocke l'information sous la forme d'une orientation d'une aimantation. En fait, une MTJ peut être représentée comme une résistance dont sa valeur dépend de l'orientation relative des 2 couches ferromagnétiques. Cette jonction est obtenue en empilant 3 couches nanométriques:

- Une couche ferromagnétique de référence dans laquelle l'aimantation est fixée soit par un couplage avec une couche antiferromagnétique (AF), soit par couplage antiferromagnétique artificiel (SAF).
- Une barrière tunnel isolante (MgO, AlO...) d'épaisseur $\sim nm$ pour que l'effet tunnel puisse avoir lieu. L'effet tunnel est un processus résultant de la nature ondulatoire de l'electron, dérivant strictement de la mécanique quantique. Cet effet se produit lorsque les électrons traversent des espaces qui leur sont interdites en physique classique, résultant de barrières de potentiel [36].
- Une couche ferromagnétique de stockage (CoFeB) dans laquelle l'aimantation peut être parallèle ou antiparallèle à celle de la couche de référence selon si une information '0' ou '1' est stockée.

La valeur de cette résistance peut s'exprimer selon l'équation suivante :

$$R(\theta) = R_p + \Delta_R \cdot \frac{(1 - cos(\theta))}{2} \tag{1.1}$$

Où θ est l'angle entre l'orientation des deux couches ferromagnétiques.

Lorsque les aimantations sont dans le même sens, $\theta = 0 \Rightarrow R = R_p$
Et lorsque les aimantations sont dans le sens opposé, $\theta = 180 \Rightarrow R = R_p + \Delta_R = R_{ap}$

Figure 1.9: Empilement schématique de trois couches rentrant dans la configuration d'une MTJ

La variation de résistance dans les jonctions tunnel magnétiques est exprimée en pourcentage par la TMR: (Tunnel Magneto Resistance), selon l'equation suivante :

$$TMR(\%) = \frac{\Delta_R}{R_p} = \frac{R_{ap} - R_p}{Rp} \tag{1.2}$$

R_p / R_{ap} représentent la résistance parallèle/ anti-parallèle de la MTJ quand les aimantations sont dans le même sens/ sens opposé.

Le fait d'avoir une TMR élevée permet une lecture fiable et rapide. Autrement dit, ceci permet de pouvoir lire et décoder de façon facile et stable les niveaux logiques '0' et '1'. La valeur de la TMR dépend principalement des matériaux utilisés dans la MTJ et du procédé de nanofabrication, elle peut atteindre 600% à basse température [37]. Elle est typiquement autour de 150 à 200%.

1.6.2 Lecture d'une cellule MRAM

Le principe de lecture consiste à mesurer la résistance de la cellule (faire circuler un courant de quelque µA à travers la jonction pour déterminer si la résistance de celle-ci est forte ou faible). Dans les applications mémoires, on compare ce courant, converti ou non en tension, à une référence grâce à un amplificateur de lecture. Comme le montre la figure 1.10, illustrant le principe de lecture:

- Si $V_{MTJ} < V_{ref}$, un niveau logique à '0' sera detecté par l'amplificateur
- Si $V_{MTJ} > V_{ref}$, un niveau logique à '1' sera donc detecté par l'aplificateur

Dans les applications numériques, la structure différentielle est souvent utilisée pour avoir une meilleure fiabilité. Les états des MTJ sont alors toujours opposés. Une des MTJ contient la donnée alors que l'autre contient le bit complémentaire. Le courant dans chaque branche est donc different. C'est ce que nous illustrons à travers la figure 3.5b avec les deux courbes en bleue et en verte.
Pendant la phase de lecture, la sortie Q a une valeur légèrement supérieure à Vdd/2 alors que la sortie complémentée $\overline{Q}$ a une valeur légèrement inférieure à Vdd/2. Les noeuds Q et $\overline{Q}$ prennent alors la valeur la plus proche de celle de Vdd ou GND. Le niveau de la TMR est donc important pour avoir une différence de potentiel ΔV élevée, afin de bien distinguer les deux états. Plus la TMR est élevée, plus le niveau de détection sera loin de Vdd/2, aussi bien sur les axes des noeuds Q et $\overline{Q}$. Dans ce type d'applications, la robustesse de lecture dépend largement de la valeur de TMR ainsi que du dimensionement des transistors.

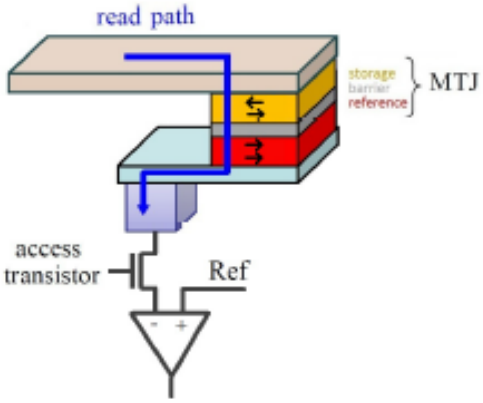
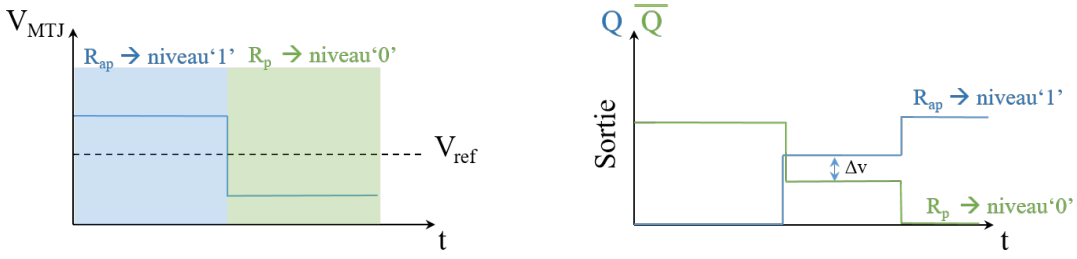


Figure 1.10: Schéma de lecture d'une cellule MRAM, basé sur un amplificateur de lecture pour déterminer l'état logique sauvegardé dans la cellule mémoire

1.6.3 Écriture d'une cellule MRAM

Les mémoires MRAM mettent en jeu différentes technologies d'écriture dont chacune possède ses propres caractéristiques, qui les rendent toutes adaptées pour différentes types d'application. Nous présentons par la suite de ce manuscrit les technologies les plus récentes soit utilisées sur le marché de mémoires MRAM, soit en cours de developpement.



(a) Principe de lecture dans les applications mémoires (comme décrit dans le texte)

(b) Principe de lecture dans les applications numériques

Figure 1.11: Représentation schématique de l'opération de lecture

1.6.3.1 Écriture TAS: Thermally Assisted Switching

L'écriture TAS proposée par Spintec et développée par la société Crocus technology, startup créée en 2006 par le laboratoire Spintec, consiste à chauffer pendant la phase d'écriture la cellule afin de pouvoir changer plus facilement l'orientation de son aimantation. L'échauffement est alors assuré par un courant électrique qui passe à travers la jonction (la température doit être au-delà de la température de blocage, environ 150°C [38]). Ce courant passe tout d'abord par la couche antiferromagnétique placée au-dessus de la couche de stockage qui a pour rôle d'assurer la stabilité à l'interface entre ces 2 couches. Lorsque la jonction est chauffée, les spins de la couche antiferromagnétique sont désordonnés. Ainsi, un faible champ magnétique extérieur est suffisant pour changer l'orientation de l'aimantation de la couche de stockage. Ce champ est généré par un courant électrique injecté dans une piste passant proche de la MTJ. La phase de refroidissement se produit après quelques nanosecondes où la jonction n'est plus chauffée car le courant électrique ne la traverse plus.

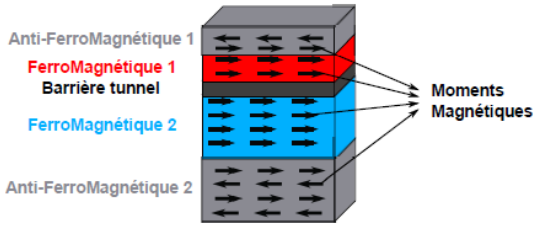


Figure 1.12: Empilement d'une jonction TAS

La technique d'écriture TAS permet de résoudre les problèmes de sélectivité rencontrés dans des autres techniques non citées dans ce manuscrit (comme FIMS et Toggle FIMS). Elle consiste à chauffer la jonction qui doit être sélectionnée par un transistor

de sélection. Les limitations de cette technique sont le temps de latence dû au temps de chauffage et refroidissement de la jonction, environ 10 ms. Même si le courant de chauffage de jonction est relativement faible, le courant permettant de générer le champ reste relativement élevé.
Notons que cette technique est développée et brevetée par le laboratoire SPINTEC en 2001 et commercialisée actuellement par la société Crocus Technology.

1.6.3.2 Écriture STT: Spin Transfer Torque

L'écriture par STT utilise un courant de polarisation injecté à travers la MTJ pour commuter l'aimantation de la couche de stockage. Dans un courant électrique non polarisé, la polarisation des spins des electrons est aléatoire, par contre, lorsque ce courant passe à travers un matériau ferromagnétique, il acquiert une polarisation effective, dépendante du taux de polarisation de ce matériau. Alors, il est possible d'utiliser cette polarisation pour retourner l'aimantation d'un autre matériau ferromagnétique. La couche de référence assure le rôle de polariseur dans un empilement de type STT. Dans cette approche, un courant bidirectionnel passe à travers la jonction permettant l'écriture de niveau logique '1' ou '0' selon le sens de courant appliqué.

- Dans le cas où les électrons passent d'abord par le polariseur (couche de référence): Les spins majoritaires traversent la MTJ et font retourner l'aimantation de la couche de stockage dans le sens parallèle de celle de l'aimantation de la couche de référence. Par contre, les spins minoritaires sont réfléchis hors de la jonction. On obtient donc un niveau logique à '0'.

- Dans le cas où les électrons passent d'abord par la couche de stockage. Les spins minoritaires sont réfléchis dans la couche de stockage. Par contre dans ce cas ils font retourner l'aimantation de cette couche dans un état antiparallèle à celle de la couche de référence alors que les spins majoritaires traversent la JTM. On obient alors un niveau logique à '1'.

Cette mémoire présente aujourd'hui de très bonnes propriétés permettant aux grandes sociétés, notamment IBM, Samsung, Toshiba, TSMC et Everspin de developper des puces STT-MRAM[20]. Elle offre une vitesse de programmation de quelques ns, une endurance prouvée à 10^{12} cycles et une densité élevée qui peut atteindre 10 nm de diamètre de la MTJ[36]. De nombreux travaux ont également montré que l'introduction des STT-MRAM dans les niveaux de caches ou dans la logique (registres par exemple) permette de diminuer la consommation en mode veille[39].

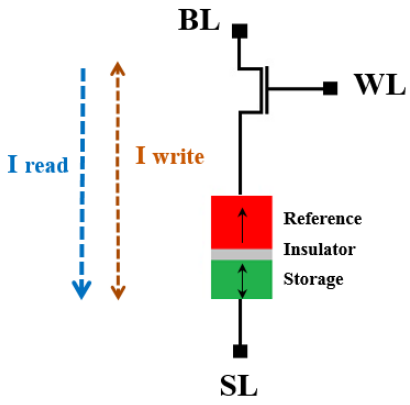


Figure 1.13: Schéma d'une cellule STT-MRAM formé d'une MTJ en série avec un transistor d'adressage. Les flèches bleu et marron montrent les chemins de lecture et d'écriture respectivement

Cependant on peut voir sur la figure figure 1.13 que les chemins de lecture et d'écriture sont les mêmes, ce qui peut aboutir à des écritures non désirées pendant la phase de lecture si ce courant de lecture est trop important en cas de lecture rapide. Ceci limite l'intégration de la STT-MRAM dans les caches de niveau L1. En plus l'écriture asymétrique des deux états représente un inconvénient par rapport aux performances liées aux aspects timing. Finallement, le courant d'écriture passant à travers la MTJ limite l'endurance à environ 10^{13} cycles pour des impulsions de courants de 10 ns [40].

1.6.3.3 Écriture SOT: Spin Orbit Torque

Plus récemment, un nouveau phénomène de retournement de l'aimantation de la MTJ a été proposé au laboratoire SPINTEC menant à obtenir des dispositifs à trois terminaux [41]. Il a été montré qu'il est possible de retourner l'aimantation de la couche ferromagnétique en applicant un courant dans la piste adjacente à la MTJ. Ce courant injecté dans la piste composée d'un métal lourd (le tantale ou platine typiquement), génère des couples spin-orbite[42]. Ceux-ci peuvent retourner l'aimantation selon deux phénomènes: l'effet Hall de spin (Spin Hall Effect) qui correspond à un effet volumique, et l'effet Rashba qui correspond à un effet surfacique [42].

En effet, l'injection du courant dans la couche métallique non ferromagnétique placée en-dessous de la couche de stockage conduit à l'accumulation des spins entre les deux couches. Les spins accumulés vont alors se diffuser dans la couche de stockage en retournant l'aimantation de cette dernière dans un sens ou l'autre selon le sens de courant. Aucun courant ne circule alors à travers la MTJ pendant son écriture. Il s'agit

plutôt des effets d'interface entre les couches. Ceci résout le problème d'endurance auquel est confronté actuellement la technologie MRAM. En outre, le risque d'écrire la jonction pendant la phase de lecture est largement diminué, vu que les chemins d'écriture et de lecture sont totalement séparés (voir figure 1.14).

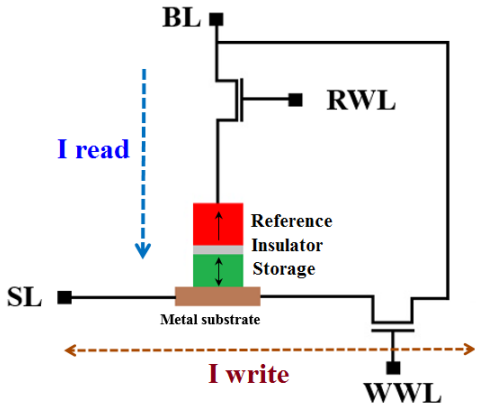


Figure 1.14: Schéma d'une cellule SOT-MRAM, formée d'une couche métallique surmontée de la MTJ dans laquelle le courant d'écriture passe. Les deux voies d'écriture et de lecture sont totalement séparées comme indiqué par les flèches orange et bleue respectivement

En comparant avec la STT-MRAM, la technologie SOT offre une endurance plus élevée à environ 10^{15} avec une écriture rapide [43]. Typiquement, des centaines de picosecondes ont été démontrées expérimentalement (210ps) par l'IMEC [44]. La majeure limitation de cette technologie comparée à celle des STT-MRAM est la densité de la cellule mémoire puisqu'un terminal de plus est rajouté.
Ces avantages font de la SOT-MRAM une mémoire particulièrement intéressante pour les applications à haute vitesse. Typiquement, elle peut être intégrée dans les caches de haut niveau ainsi que dans les applications de l'IoT [45], [46].

1.7 Applications des mémoires MRAM

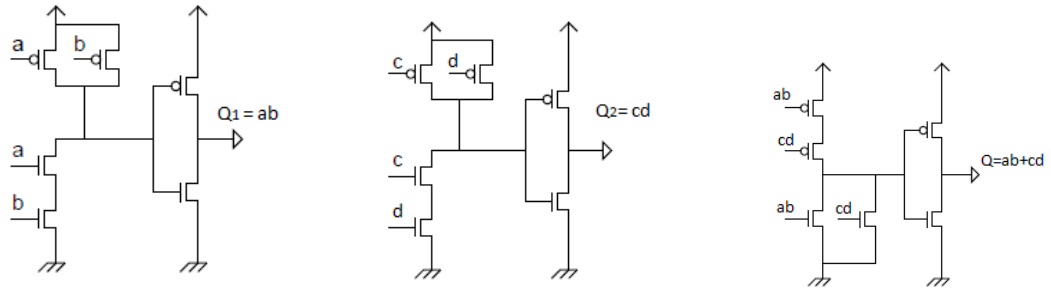
Toutes les caractéristiques mentionnées ci-dessus des différentes technologies des mémoires MRAM, les rendent certainement adaptées à un large spectre d'applications. Notamment la possibilité de réduire la consommation d'énergie en mode veille, c'est à dire le concept "Normally off computing", abordé en détail au cours de ce chapitre.
Nous présentons dans la suite quelques domaines d'intégration des mémoires MRAM.

1.7.1 Circuits logiques hybrides CMOS/MTJ

Le concept « logic-in-memory » correspond à une architecture dans laquelle des éléments de mémoire sont répartis dans les unités de calcul. Le plus souvent au niveau des cellules standards, permettant l'amélioration des performances en termes d'énergie par rapport à l'architecture classique de Von-Neumann [47]. Depuis l'apparition des mémoires non-volatiles, plusieurs unités logiques ont été conçues et fabriquées à base de logic-in-memory telles que des half-adder[48], full adder[49], flip-flop [39], Look-Up Table (LUT) [50] et beaucoup d'autres.

1.7.1.1 Portes logiques magnétiques NAND/NOR/XOR

Les codes et les programmes stockés dans n'importe quel type de mémoire sont exécutés et implémentés à l'aide de portes logiques. Pour implémenter une fonction logique à l'aide de transistors, il suffit de respecter un mécanisme précis des procédés CMOS qui permet de réaliser la fonction souhaitée. Cela consiste donc à connecter les transistors NMOS en série pour réaliser des fonctions type "ET" (AND) ou de les connecter en parallèle afin de réaliser des fonctions type "OU" (OR), et inversement pour les PMOS. Prenons l'exemple d'une fonction logique "ab+cd". Pour implémenter cette fonction, il faut utiliser 2 portes logiques "ET" et une porte "OU", chacune à deux entrées. Notons qu'en technologie CMOS, la sortie est complémentée ce qui nécessite l'ajout d'un inverseur pour obtenir chacune des fonctions élémentaires comme illustré sur la figure 1.15c.



(a) schéma d'une porte logique "ET" pour realiser la fonction Q_1= ab

(b) schéma d'une porte logique "ET" pour realiser la fonction Q_2= cd

(c) schéma d'une porte logique "OU" pour realiser la fonction Q= ab+cd

Figure 1.15: Portes logiques pour réaliser la fonction "ab+cd"

W. Black et B. Das ont été les premiers à développer des architectures logiques combinant jonctions tunnel magnétiques et technologie CMOS (architecture hybride) [51].

D'autes auteurs ont proposé de réaliser des portes logiques magnétiques, dans le but d'avoir des entrées non-volatiles réalisées par des MTJ [52]. Le but de ces portes est d'accélérer le temps de calcul, c'est à dire améliorer les délais dûs aux interconnexions entre l'unité de calcul et les mémoires, étant donné que l'information dans ce cas est stockée directement dans les portes logiques.

1.7.1.2 Registres magnétiques de type flip-flop

Contrairement aux circuits combinatoires (portes ET, OU,..) où la sortie dépend uniquement des entrées, les systèmes séquentiels sont basés sur le principe que la fonction de sortie dépend à la fois des variables d'entrées et de l'état antérieur des sorties. En d'autres termes, la logique séquentielle utilise la notion mémoire de stockage, alors que la logique combinatoire n'en a pas.
L'élément de base de la logique séquentielle est la bascule (flip flop)qui peut être asynchrone (bascule RS par exemple) ou synchrone (bascule D ou bascule JK). La bascule est utilisée pour sauvegarder les données au niveau local dans le CPU (Central Processing Unit), souvent utilisées pour former les registres.

Cependant, avec la miniaturisation de la technologie CMOS, les bascules classiques (de type SRAM) souffrent d'une consommation d'énergie statique croissante, causée par le courant de fuite [53] et d'un nombre important de transistors. C'est pourquoi l'introduction de la non-volatilité dans les bascules permet de réduire la consommation d'énergie en mode veille, en coupant l'alimentation des parties inactives sans perte de données de stockage. Plusieurs auteurs ont proposé des bascules magnétiques hybrides CMOS/ MRAM qui fonctionnent en mode standard (volatil) si la partie magnétique est désactivée, et en mode non volatile intégrant les MTJ [54]. Pour un fonctionnement performant, le courant d'écriture des jonctions magnétiques devrait être relativement grand pour assurer une programmation rapide. Un exemple d'une bascule en technologie SOT est illustré sur la figure 1.16, proposé par [5] dans le but d'améliorer les performances en terme de vitesse et de consommation d'énergie.

1.7.1.3 Full adder magnétiques

L'unité de base pour effectuer des opérations arithmétiques dans le CPU est l'additionneur (Full adder). Comme son nom l'indique, c'est un circuit logique qui permet de réaliser une addition. Le circuit de base est formé de 3 entrées (A_i,B_i et la retenue) et deux sorties (voir la figure 1.17). La relation entre les entrées et les sorties est réalisée à l'aide des portes logiques (XOR, AND, OR) via des transistors CMOS dans les architectures conventionnelles. Il est possible d'enchainer plusieurs blocs afin d'effectuer le calcul des mots de tailles importantes dans les fonctions complexes.

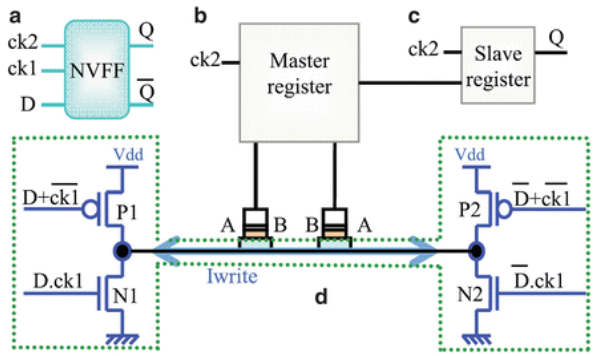


Figure 1.16: Bascule en technologie SOT-MRAM formée de 2 blocs élémentaires: le "Master register" connecté à deux MTJ pour assurer la non-volatilté, en série avec le bloc "slave register". L'écriture des MTJ est assurée par les 4 transistors P1, N1, P2, N2 [5]

Plusieurs additionneurs non volatils basés sur la technologie MRAM ont été proposés. Deux MTJ en mode complémentaire sont connectées pour chaque entrée, permettant d'obtenir des portes non-volatiles avec une vitesse de calcul relativement grande. Un additionneur en technologie STT a été proposé par [55], montrant une très basse consommation avec une densité élevée.

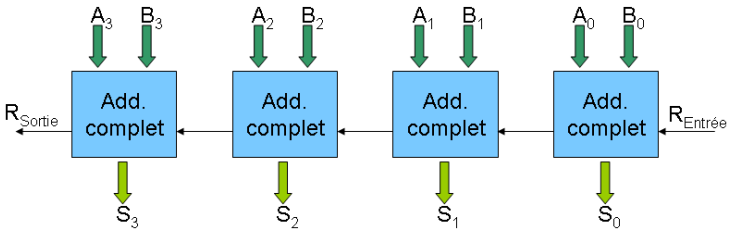


Figure 1.17: Association d'une chaîne de full adder

1.7.2 Circuits reconfigurables: MRAM based FPGA

Cet acronyme de Field Programmable Gate Array désigne littéralement une 'Matrice/réseau de portes logiques programmables. Ces circuits sont programmables électriquement et peuvent être déstinés à impementer n'importe quel circuit numérique. Ils ont comme principal atout d'être reprogrammables après fabrication, selon l'application ou la fonction qui leur est assignée.

Les FPGA sont composés essentiellement de plusieurs milliers de LUT qui servent à implémenter des équations logiques ayant généralement 4 à 6 entrées et une sortie. Les LUT peuvent toutefois être considérées comme un ensemble de points mémoires qui contienne les sorties possibles d'une fonction logique.

Prenons l'exemple Xilinx Virtex-5 [56], ce FPGA ne contient pas moins de 20 000 cellules LUT. Ces dernières sont connectées entre elles par une matrice de routage configurable. Cependant, la complexité d'implémentation d'un FPGA est relative à l'application qui implique une augmentation du nombre de cellules LUT ainsi que sa consommation d'énergie. La plupart des grands FPGA modernes sont fondés sur des cellules SRAM, qui sont généralement assez rapides à programmer. Néanmoins, dès que la surface du système devient critique, la technologie SRAM remet en cause les solutions proposées. On constate donc des structures SRAM magnétiques composées d'une association d'une SRAM avec une cellule magnétique formée par deux MTJ où l'information est stockée [57]. Une approche de type MRAM a été montré dans [58] pour apporter de la non volatilité aux LUT et réduire la surface occupée. Beaucoup d'autres équipes de recherche ont proposé des architectures innovantes CMOS /MRAM permettant un fonctionnement performant qui répond à un certain nombre de besoins [59].

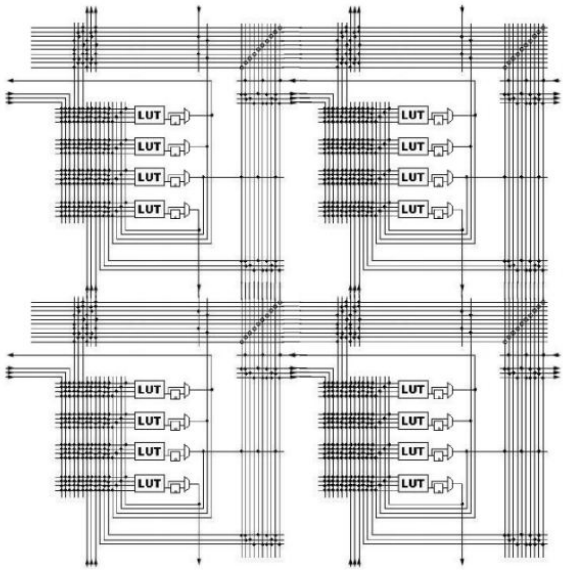


Figure 1.18: Architecture d'un FPGA connectant plusieurs LUT via un réseau d'interconnexions

1.7.3 Processeurs embarqués

Apporter de la non-volatile dans un système électronique ouvre la voie vers un nouveau concept appelé "Normally Off Computing". Sa mise en oeuvre touche la plupart des systèmes, surtout dans le domaine de l'IoT. Une étude menée en 2014 [39], [60] montre que 2 tiers de l'énergie consommée dans un microcontrôleur (MCU) est dépensée en mode veille. Autrement dit, cette énergie est dépensée lorsque le système est inactif mais sous alimentation. Pour résoudre ce problème de perte, les MCU commerciaux mettent en oeuvre plusieurs méthodes de mise hors tension avec des temps de

réveil différents. Or, selon l'application, il faut trouver le bon compromis entre le rapport de l'énergie consommée en mode veille et l'énergie nécessaire au réveil, celui-ci étant directement lié au temps requis (nombre de cycles d'horloges).

Cependant, le potentiel que dévoile les mémoires émergentes, incluant les MRAM, en terme de réduction d'énergie, permet d'intégrer ces mémoires avec la technologie CMOS classique. En coupant l'alimentation pendant la phase d'inactivité du système, l'énergie consommée en mode veille est quasiment à zéro (quasi zero leakage). En effet, il subsiste un certain faible courant de fuite dans les dispositifs de coupure, basés sur des transistors ultra low leakage ayant un fort V_t (tension de seuil). Comme illustré sur la figure 1.19, il est important de modéliser l'énergie dans le mode conventionnel et dans le mode non volatil afin qu'on puisse les comparer [6].

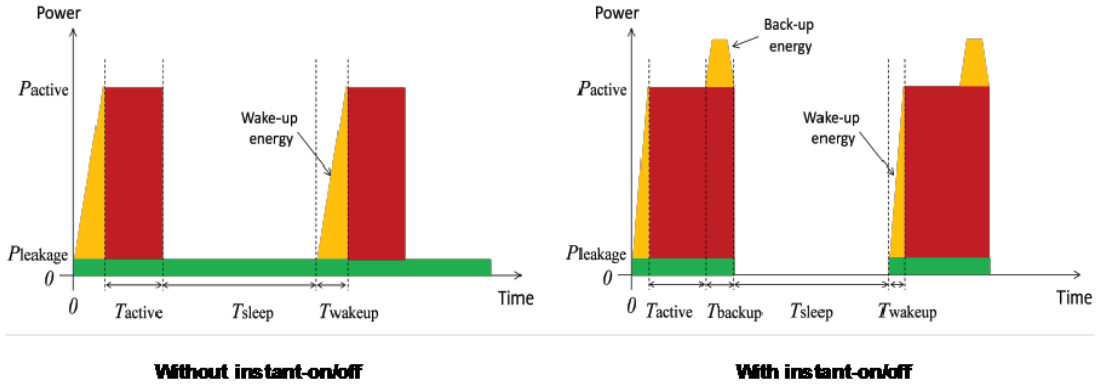


Figure 1.19: Consommation d'un système éléctronique CMOS vs CMOS/ Magnétique [6]

Peu importe le mode de fonctionnement, lorsque le circuit est actif, la puissance dynamique consommée est la même (partie en rouge). En revanche, la puissance statique est constante dans le temps en mode conventionel CMOS. Par contre, cette puissance est présente uniquement pendant les phases actives en mode non volatil. En ce qui concerne le réveil du circuit, une puissance de restauration de l'information se rajoute à chaque fois qu'on le réactive. On constate que cette puissance est plus importante en mode conventionnel. Cependant, une puissance supplémentaire se rajoute en mode non volatil uniquement, correspondant à la puissance nécessaire pour sauvegarder les variables d'état du système. Donc, pour assurer l'efficacité énergétique des mémoires émergentes non volatiles dans un système:

$$(P_{statique} x T_{sauvegarde}) + E_{sauvegarde} < T_{endormi} x P_{statique} \tag{1.3}$$

Donc,

$$\frac{P_{statique} \times T_{sauvegarde} + E_{sauvegarde}}{E_{statique}} < T_{endormi} \tag{1.4}$$

Dès que le temps d'inactivité du système est supérieur à une certaine valeur, on s'aperçoit qu'il devient intéressant d'utiliser les mémoires émergentes en mode "Normally off computing". L'intégration de ces mémoires dans un processeur peut être réalisée par deux voies: soit locale dans le CPU (au niveau des registres, LUT et portes logiques comme décrit ci-dessous), soit par un simple remplacement des mémoires caches par une mémoire MRAM.
On remarque sur la figure 1.20 que la technologie MRAM peut remplacer les caches de haut niveau. C'est le cas de la technologie SOT qui, éventuellement, peut remplacer le cache de niveau L1 alors que la STT se situe aux niveaux L2 et L3, car sa vitesse est limitée par le chemin commun d'écriture et de lecture[61].

L'idée d'un processeur non volatil est d'un intérêt grandissant pour plusieurs centres de recherche. Sur le plan national on note Spintec, l'université de Montpellier(LIRMM) et l'université Paris-Sud. Dans ce contexte, une collaboration entre Spintec et le LIRMM a été mise en place afin d'étudier les performances de la technologie STT au niveau d'un processeur "SecretBlaze" [62] crée au sein du LIRMM.
Ce processeur est également le sujet de la deuxième partie de ma thèse pour lequel un chapitre y est consacré.

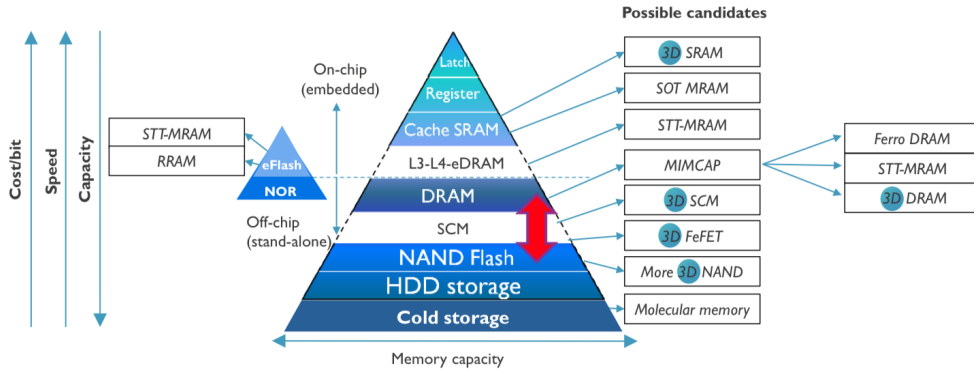


Figure 1.20: Évolution de la hiérarchie mémoire en intégrant les candidats de mémoires possibles afin de remplacer les technologies traditionnelles [**?**]

Il est également important de mentionner les autres solutions apportées par les mémoires émergentes au niveau des processeurs. Le principe "Chekpointing/rollback" par exemple, qui consiste à revenir à un état antérieur valide du processeur en cas de la détection d'une erreur, comme proposé par [39]. Le principe de la restauration est illustré sur la figure 1.21. Pour éviter de réinitialiser l'application du système en cas d'une erreur intervenue pendant l'exécution, il est possible de créer des points de contrôle en sauvegardant l'état de processeur soit de manière périodique, soit d'une manière dépendante de l'application désirée. Ce principe est assuré par des registres

non volatils qui permettent de stocker l'information d'une façon locale afin d'éviter le déplacement vers la mémoire centrale (c'est le cas des processeurs volatils SRAM). La mise en place du mécanisme "Chekpointing/rollback" et sa stratégie d'utilisation dépend entièrement de l'application souhaitée.

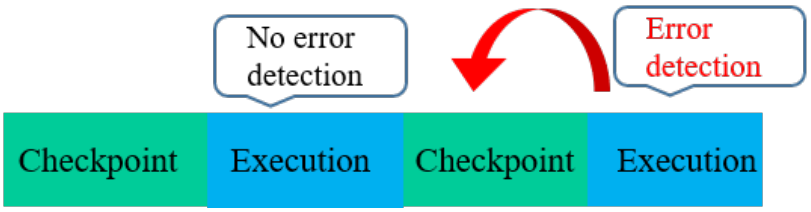


Figure 1.21: Principe Chekpointing/rollback [6]

1.8 Positionnement du sujet de thèse

Les mémoires embarquées traditionnelles (SRAM, DRAM, Flash,...) se confrontent à des problèmes technologiques liés soit à la difficulté de miniaturisation, soit à sa forte consommation (qui est difficilement surmontable). C'est pourquoi, les recherches s'orientent actuellement vers des nouvelles technologies émergentes afin de surmonter ces limitations. Le besoin est considérable, considérant le nombre gigantesque d'objets connectés à venir dans les prochaines années.

Nous avons vu qu'il n'existe pas, parmi les mémoires émergentes, la technologie universelle qui répond à toutes les attentes. Chaque mémoire présente ses propres caractéristiques adaptées à l'application souhaitée.

Dans ce contexte, mes activités de recherches se positionnent dans le domaine de la mémoire MRAM qui se distingue par son degré de maturité, sa faible consommation, sa vitesse et son endurance par rapport aux autres technologies. Plusieurs applications basées sur la technologie MRAM sont mises en jeu, tels que les circuits hybrides pour "logic in memory", les circuits reconfigurables, les applications dédiées aux processeurs... Mes travaux de recherches se sont articulés donc autour de 3 axes qui seront décrits en détails dans les prochains chapitres:

- La conception des circuits hybrides CMOS/magnétique de type LUT, dans le but de réaliser un démonstrateur.
- La conception d'une mémoire complète en technologie SOT afin de l'intégrer dans un processeur volatil SecretBlaze

- L'étude d'un processeur embarquant des mémoires non volatiles STT-MRAM et SOT-MRAM.

Chapitre 2

Conception d'un générateur de fonctions logiques hybrides CMOS magnétique

Motivation

Ce chapitre s'intéresse à la conception d'un générateur de fonctions logiques hybrides CMOS magnétique (LUT:Look Up Table), l'élément de base d'un FPGA. Après une présentation du rôle des LUT, une description de la conception full custom sera abordée. Plusieurs versions des architectures de LUT seront proposées par la suite dans le but d'intégrer ces circuits sur une puce de silicium, dans le cadre d'un projet interne MAD (Memory Advanced Demonstrator).

2.1 Architectures reconfigurables

Les architectures programmables configurables sont des circuits intégrés anciens à nos yeux, proposées depuis la fin des années 1970 [63]. Ces composants sont formés de portes logiques (type AND, OR, XOR) interconnectées entre elles pour réaliser une fonction souhaitée par le concepteur. L'évolution technologique de ces architectures a permis la possibilité de reconfigurer ces composants plusieurs fois, ce qui n'était pas le cas pour les circuits UltraViolets Programmable Read Only Memory (UVROM) et One Time Programmable (OTP). Cette évolution a modifié l'architecture interne de ces circuits, ne reposant pas forcément sur des portes logiques dont la programmation est figée d'une façon définitive, mais plutôt sur des éléments logiques reconfigurables (LUT:Look Up Table), placés sur un réseau de routage.

En 1985 Xilinx, entreprise américaine de semi-conducteurs, était à la tête des compagnies qui ont commercialisés les circuits reconfigurables appelés Field-Programmable Gate Arrays (FPGA)[64]. Les FPGA occupent aujourd'hui le devant du marché des circuits intégrés ayant les avantages d'avoir un prototypage rapide, une possibilité de reconfiguration dynamique et un coût de production relativement faible par rapports aux ASIC (Application-Specific Integrated Circuit). Ces derniers comme leurs noms l'indiquent, sont des circuits intégrés spécialisés pour une fonction déterminée, programmée une fois uniquement lors de la fabrication, ce qui leur donne l'avantage d'être plus performant et de répondre parfaitement à des spécifications imposées.

Il est évident que les FPGA sont utilisés aujourd'hui dans un large spectre d'applications, notamment dans les systèmes embarqués. Leurs architectures ont évolué au fil du temps, et leur niveau de complexité dépend des applications désirées. L'architecture générale des FPGA comme illustrée sur la figure 2.1, est composée de 3 blocs principaux :

- Configurable Logic Block (CLB) : ces blocs génèrent les fonctions logiques combinatoires ou séquentielles via des portes logiques reconfigurables.
- Connect Box (CB) : Ces blocs permettent de connecter les entrées/sorties des blocs logiques (CLB) aux réseaux.
- Switch Boxe (SB) : Ces blocs assurent la connectivité entre les lignes horizontales et verticales des blocs logiques entre eux via les blocs de connexions.

La programmation d'un FPGA utilise un flot de conception numérique où les différentes fonctionnalités sont codées à l'aide d'un langage de description matérielle

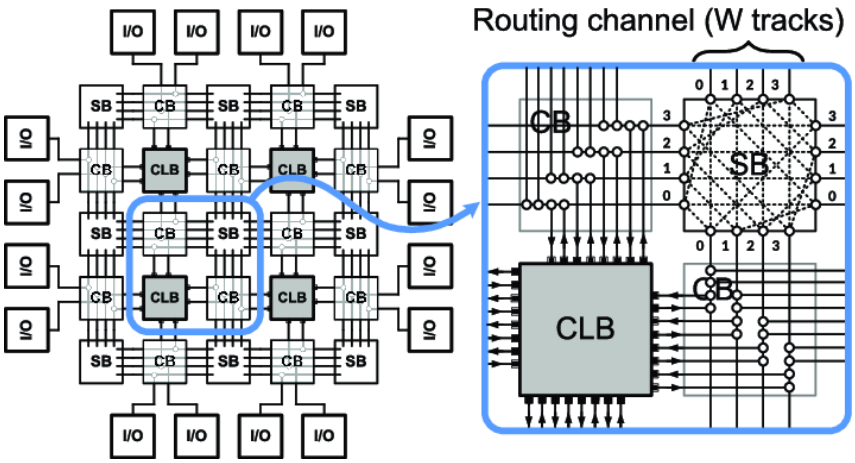


Figure 2.1: Architecture générale d'un FPGA [7]

(Verilog, VHDL). Ces codes sont synthétisés par la suite et donc convertis en composants afin d'implémenter la fonction souhaitée.

2.2 FPGA à base de LUT

Les FPGA actuels représentent l'équivalent de plusieurs milliers de tables de correspondance (LUT), qui constituent l'élément de base des CLB et qui servent à implémenter n'importe quelle fonction logique. On trouve par exemple jusqu'à 400 000 CLB dans les FPGA Stratix V de Altera [65] et 1 956 000 CLBs pour les FPGA Virtex 7 de Xilinx [66].
Nous présentons par la suite le fonctionnement des LUT ayant un rapport direct avec mes recherches de la première année de la thèse.

2.2.1 Principe de fonctionnement d'une LUT

Une LUT est le composant élémentaire d'un FPGA. Elle permet d'implémenter une fonction numérique en sauvegardant la table de vérité de la fonction dans des cellules mémoires. Dans certaines applications, le calcul des fonctions complexes peut se révéler récurrent et par conséquent ralentir la vitesse d'excution. Pour régler cela, l'application va précalculer au démarrage les valeurs dont il a besoin, et va les sauvegarder dans des LUT. Pendant le calcul et à chaque fois que le système aura besoin d'une valeur, il pourra effectivement restaurer cette valeur sauvegardée dans la LUT la plus proche [67].
Globalement, une LUT est caractérisée par le nombre d'entrées k, 2^k bits de sélection et une sortie. Le nombre d'entrées varie généralement entre 2 et 6 selon la complexité

de la fonction à implémenter. En outre, chaque fonction est programmable, c'est-à-dire que l'on est libre de programmer (écrire) les cellules mémoires plusieurs fois afin de réaliser la fonction attribuée.

Prenons l'exemple de la fonction suivante F= $\overline{IN0}$.IN1 + IN1.IN2 + $\overline{IN0}$.$\overline{IN1}$.IN2

Pour réaliser cette fonction à l'aide de portes logiques, nous avons besoin de 4 portes ET, 2 portes OU et 3 inverseurs connectés comme le montre la figure 2.2.

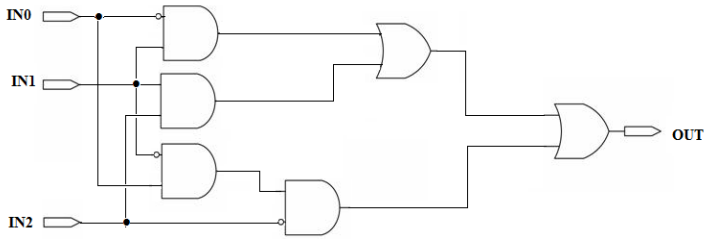
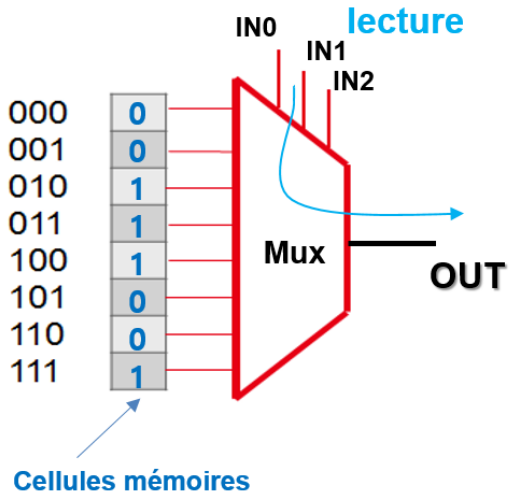


Figure 2.2: Réalisation d'une fonction F= $\overline{IN0}$.IN1 + IN1.IN2 + $\overline{IN0}$.$\overline{IN1}$.IN2 à partir de porte logiques

Généralement, rechercher une valeur en mémoire est souvent plus rapide qu'effectuer un calcul complexe. C'est pourquoi une LUT peut simplifier cette implémentation en précalculant toutes les sorties possibles et en les stockant dans des cellules mémoires. Une LUT n'est plus qu'une structure de données stockées en mémoire, dont une représentation est montrée en figure 2.3b.

IN0	IN1	IN2	OUT
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(a) Table de vérite de la fonction F= $\overline{IN0}$.IN1 + IN1.IN2 + $\overline{IN0}$.$\overline{IN1}$.IN2



(b) LUT à 3 entrées formée d'un multiplexeur et de 8 cellules mémoires

Figure 2.3: Circuit de base d'une LUT.

Comme le montre le schéma de la figure 2.3a, des cellules mémoires contenant la table de vérité de la fonction F sont placées aux entrées du multiplexeur. Ensuite, pendant le fonctionnement du CLB, on applique l'entrée de la fonction sur les bits de sélection afin de lire l'information contenue à la bonne adresse. Dans notre exemple, si la valeur d'entrée est '101', le multiplexeur va donc transmettre la valeur '0' en sortie sans perdre le temps à recalculer la sortie de cette fonction de nouveau. De même pour '111', la sortie à '1' est transmise et ainsi de suite pour toutes les autres combinaisons possibles.

Cependant, pour des fonctions de plus en plus complexes le nombre d'entrée « k » augmente, ce qui fait augmenter d'une façon considérable le nombre de cellules mémoires (2^k). C'est pourquoi le fabricant doit faire le choix lors de la conception de la taille des LUT en prenant en considération deux paramètres :

+ Plus k augmente, plus on a la possibilité d'implémenter des fonctions logiques complexes avec moins de surface car on aura moins de blocs logiques (même si la taille du bloc en lui même augmente). De plus, le chemin critique entre les différents LUT diminue permettant d'améliorer la rapidité.

− Plus k augmente, plus le nombre de cellules mémoires par bloc augmente (2^k), ce qui augmente le délai de propagation dans la LUT et donc diminue la vitesse de lecture et d'accès de la LUT. Par ailleurs, plus la taille du bloc augmente plus la consommation est élevée.

On constate alors qu'il faut faire un compromis entre les 3 axes qui définissent les performances d'un bloc logique: vitesse, surface et consommation.

2.2.2 Évaluation de la taille des LUT

Au début de leur introduction sur le marché, les FPGA étaient une option pour les circuits ayant un faible volume de prototypage. Au fil du temps et avec les améliorations apportées en terme de performances, les FPGA sont devenus les circuits incontournables dans les systèmes numériques complexes contenant de la mémoire, des processeurs et d'autres fonctionnalités. Indépendamment de la technologie utilisée, beaucoup d'études ont été réalisées par des équipes de recherches industrielles et académiques afin d'améliorer l'architecture des FPGA, soit au niveau du routage, soit dans la taille des LUT. Cette dernière a été le centre de recherche de plusieurs études, qui ont etudié l'effet de la taille des LUT sur la surface et la vitesse du FPGA.

Les études réalisées dans [68] et [69] ont montré que la taille optimale d'une LUT est 4. En outre, il a été montré dans [70] et [10] que l'utilisation des LUT de 5 à 6 entrées

est le meilleur compromis entre surface et vitesse pour former un bloc logique optimal. En revanche, l'étude menée par [8] a révélé que l'utilisation des LUT à 2 et 3 entrées est équivalent au LUT à 4 entrées en termes de surface mais beaucoup moins contraignante en vitesse. Ce qui n'était pas le cas dans [11] qui considère qu'une LUT entre 4 à 6 entrés est le bon compromis entre surface et rapidité, surtout pour les circuits de fonctionnalité complexe.

Afin de bien appréhender le choix de la taille d'une LUT, nous présentons ci-après l'impact de ce choix sur les performances d'un FPGA.

2.2.2.1 Dépendance entre la taille des LUT et la surface totale occupée

Cette partie décrit donc le compromis concernant la taille des LUT étudié dans la littérature, en particulier dans [8]. Ce compromis dépend principalement des deux facteurs, k : le nombre d'entrées d'une LUT, et N : le nombre de LUT par CLB.
La structure d'un CLB est illustrée sur la figure 2.4. Chaque CLB contient N éléments logiques de base (BLE) où chaque BLE est formé d'une LUT à k entrées associé à une bascule afin de réaliser des circuits séquentiels comme des registres. Un multiplexeur est généralement ajouté pour utiliser ou non la bascule. I représente le nombre d'entrées par CLB, qui est typiquement égal à 50 % du nombre total d'entrées possibles du produit : kxN, et qui vaut :

$$I = (N + 1) \cdot \frac{k}{2} \tag{2.1}$$

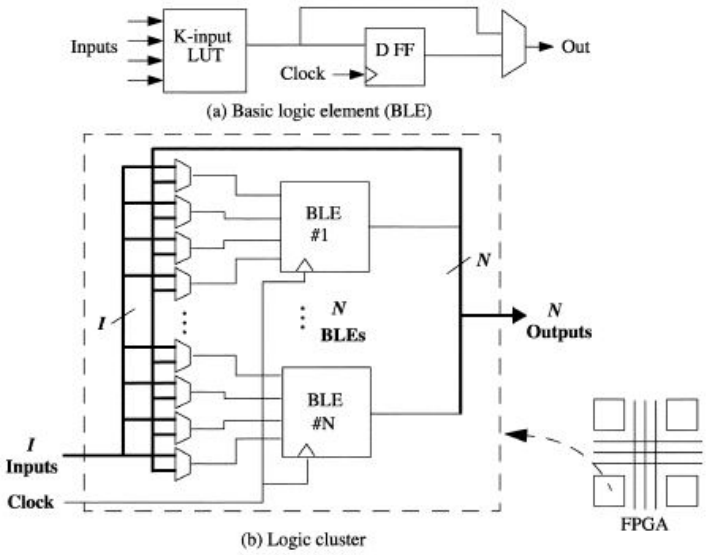


Figure 2.4: Architecture d'un CLB, l'élément principal d'un FPGA [8]

La surface d'un FPGA en fonction de N et k est représentée sur le graphe de la figure 2.5. Elle est calculée en utilisant la largeur minimale des transistors pour avoir la surface la plus optimisée. On constate que les LUT de 4 à 5 entrées présentent un gain notable permettant d'avoir une surface minimale d'un FPGA.

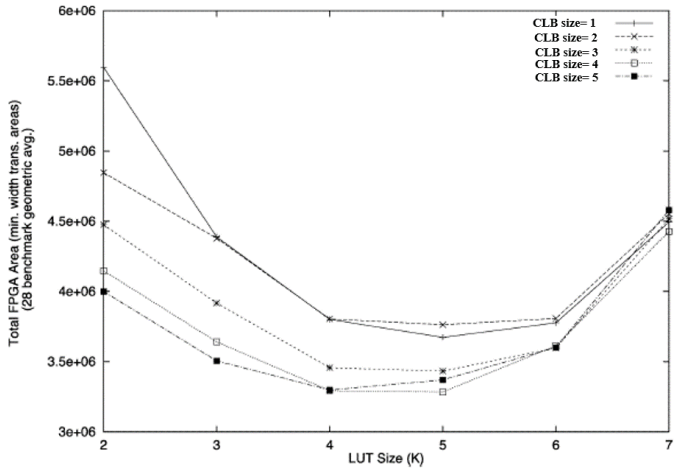


Figure 2.5: La surface totale d'un FPGA en fonction de la taille d'une LUT [8]

Or, il est important de distinguer les deux types de connexions, qui ont un impact direct sur la surface totale d'un FPGA : l'interconnexion entre les différents blocs de CLB et l'intraconnexion (à l'intérieur de CLB) entre les différentes LUT. Il suffit de multiplier ces deux types de connexion pour obtenir la surface moyenne totale d'un FPGA.
On constate toujours d'après la figure 2.5 que l'augmentation de la taille de CLB engendre une faible augmentation de la surface totale. Ceci est tout à fait logique car le réseau de routage est moins complexes par rapport à un bloc ou il y a plus de CLB. Dans les FPGA modernes, les CLB contiennent plusieurs BLE qui partagent les même entrées, ce qui facilite le réseau d'interconnexions entre les différents CLB.

2.2.2.2 Dépendance entre la taille des LUT et la vitesse

Il est également important de distinguer deux types de délais qui interviennent dans le calcul du délai total du chemin critique : inter-délai et intra-délai.

L'intra-délai défini le délai à l'intérieur des CLB. Ce délai croît avec l'augmentation de k, alors que l'inter-délai défini le délai de propagation entre les différents blocs logiques CLB. Comme nous l'avons déjà décrit, plus la taille k d'une LUT augmente, plus la possibilité d'implémenter des fonctions logiques complexes avec moins de blocs logiques augmente. L'augmentation de k engendre donc d'une part une augmentation de l'intra-délai et d'autre part une diminution de l'inter-délai. Ceci est dû

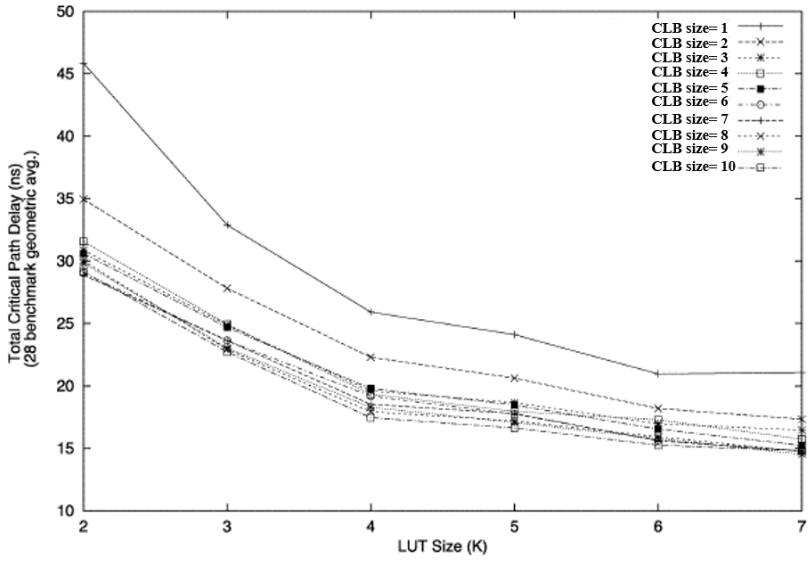


Figure 2.6: Evaluation du délai total du chemin critique des blocs logiques en fonction de la taille d'une LUT [8]

au fait que le chemin critique demande moins de blocs pour implémenter une fonction logique. Ce compromis en rapidité a été également étudié dans [8].

On constate d'après la figure 2.6 que le chemin critique entre LUT est de plus en plus faible avec l'augmentation de k. Le gain en rapidité est significatif jusqu'à une taille k de 6 et pour un nombre de LUTs > 3.

2.3 Différentes structures de LUT

Il existe plusieurs types de FPGA dont chaque type offre ses propores avantages et inconvénients selon la technologie utilisée. Nous présentons ci-après quelques structures de LUT sur différentes technologies, en abordant plus en détail celles à base de MRAM.

2.3.1 LUT à base d'une SRAM

La technologie SRAM est largement utilisée pour la plupart des FPGA du marché, fabriqués par les grandes entreprises telles que Xinlinx, Altera et Lattice [71]. La raison pour laquelle cette technologie est la plus utilisée depuis les dernières décennies est sa simplicité de fabrication, car elle est composée de transistors CMOS uniquement, ainsi que sa vitesse élevée. L'utilisation d'une cellule mémoire SRAM ne nécessite aucune fabrication technique spéciale et elle est moins couteuse que les technologies qui utilisent des fusibles ou les EPROM (Erasable Programmable ROM) [72].

Comme illustrée sur la figure figure 2.7, chaque cellule mémoire est constituée de deux inverseurs tête-bêche permettant de conserver 1 bit de donnée contenu dans la LUT. La donnée est écrite et lue via les deux transistors d'accès M5 et M6. Ces transistors sont commandés par la ligne "World Line" (WL) qui permet de charger l'information via les lignes "Bit Line" (BL et son complément $\overline{BL}$). Généralement chaque cellule mémoire est formée de 6 transistors. Pour une LUT de 5 entrées, on aura donc besoin de (2^5 x 6) transistors pour l'ensemble des cellules mémoires sans compter les transistors constituant le multiplexeur. Ceci reste toutefois un inconvénient majeur du point de vue de la surface et augmente la surface totale d'un FPGA.

En outre les deux principaux inconvénients des LUT à base de SRAM sont: leurs caractéristiques volatiles, c'est-à-dire qu'elles doivent être reconfigurées après chaque coupure de l'alimentation. C'est pourquoi une mémoire Flash est souvent intégrée en supplément pour apporter l'aspect non-volatil, la limitation en miniaturisation pour des nœuds technologiques avancés [73] liée à une décroissance de la charge de commutation.

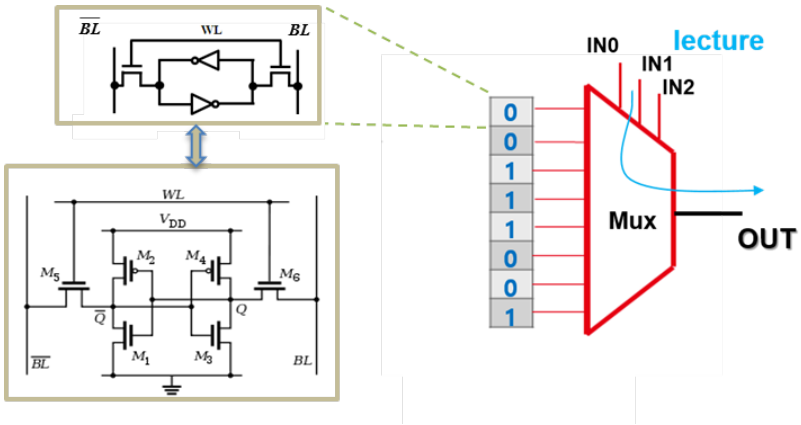


Figure 2.7: Schéma d'une LUT en technologie SRAM

2.3.2 LUT hybride à base d'une SRAM/Magnétique

W.C. Black et B.Das ont été les premiers a proposé la technologie hybride SRAM-Magnétique dans le but de combiner les avantages de chacune des deux technologies [9]. La cellule mémoire est constituée de deux jonctions magnétiques, représentées par deux valeurs différentes en opposition: R_{min} et R_{max}, connectées en série avec les transistors M3 et M4 de la bascule (figure 2.8). L'avantage de cette structure par rapport à une SRAM classique est la non volatilité apportée par les cellules magnétiques, qui sauvegarde l'information même en absence d'alimentation. Les JTM en états complé-

mentaires stockent la donnée à sauvegarder de façon permanente et la cellule SRAM lit la donnée pour la restaurer. Le transistor M5 est utilisé pour la phase de lecture. Il permet un pseudo court-circuit entre pour les deux branches du latch. Chacune des tensions dépend de la différence de résistance entres les lignes Q et $\overline{Q}$ pendant la lecture. Le signal "auto zero" est d'abord activé pendant la phase d'évaluation pour assurer l'équilibre entre Q et $\overline{Q}$. Ensuite ce signal est désactivé permettant à un courant de passer à travers les 2 résistances. La sortie va donc basculer permettant de lire un état stable à '1' ou '0' logique.

Cette architecture hybride est intéressante du point de vue vitesse car les JTM n'influent pas sur les phases d'écriture et de lecture de la partie CMOS. En revanche, elle est contraignante en termes de surface, où chaque point mémoire exige 5 transistors et 2 JTM. De plus, des transitors supplémentaires sont ajoutés pour programmer et sélectionner les JTM.

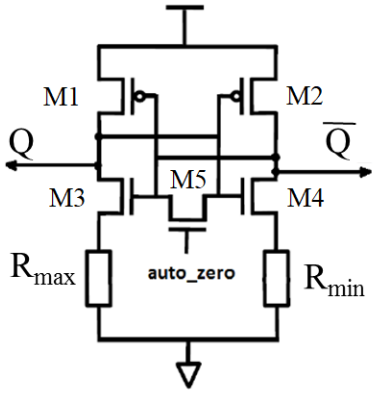


Figure 2.8: Cellule mémoire en technologie SRAM/magnétique [9]

2.3.3 LUT à base d'une MRAM

Pour réduire la consommation statique de la SRAM ainsi que sa surface, plusieurs études ont été proposées dans la conception des LUT en technologie MRAM. Une approche a été montrée dans [74]. Elle repose sur l'utilisation de deux JTM en série avec un amplificateur de lecture pour chaque cellule mémoire. Donc, une LUT à 4 entrées exige 2^4 x 2 JTM et 24 amplificateurs de lectures. L'information est sauvegardée sous la forme de deux états complémentaires, d'où la nécessité de 2 JTM/bit).

Cette structure a pour principal avantage d'être assez robuste, car le courant de lecture est détecté par l'amplificateur de lecture avant qu'il soit affecté par l'arbre de

codage. La taille de l'amplificateur de lecture dépend généralement du degré de robustesse. En revanche, la structure proposée ne résout pas le problème de densité rencontré par la technologie SRAM. Ce problème a été également étudié dans [10], qui propose une LUT à 6 entrées robuste, et probablement aussi dense par rapport à la structure présentée dans figure 2.9. Leur structure repose sur l'utilisation d'un seul amplificateur de lecture quel que soit le nombre d'entrées. La figure 2.9 illustre le schéma de principe de la LUT proposée qui est formée de deux parties : le réseau des JTM où la table de vérité est sauvegardée et la partie référence dans laquelle on compare l'état de l'information désirée, s'il est à '0' ou à '1'.

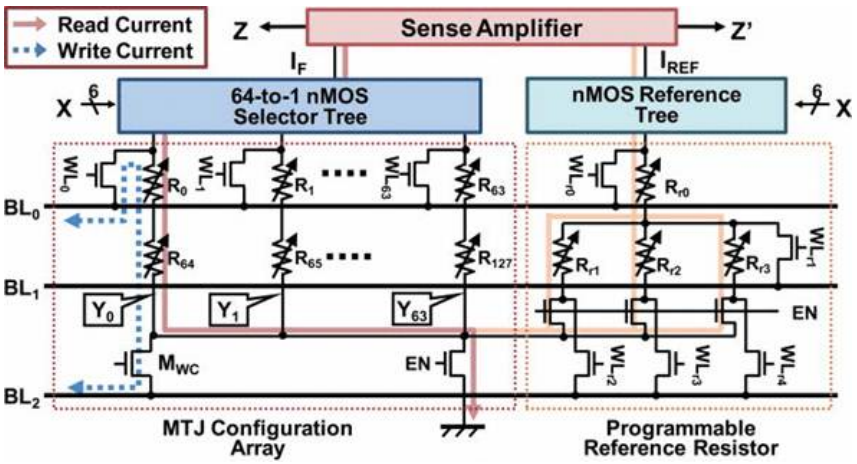


Figure 2.9: Configuration d'une LUT à 6 entrées proposées dans [10]

On retrouve également un amplificateur de lecture 'sense amplifier' qui permet de lire l'information sauvegardée, un arbre de transistors CMOS permettant de sélectionner la cellule mémoire selon une connexion en série de six transistors. Chaque branche de cet arbre (64 branches au total) est connectée à 2 JTM représentées par la résistance R_n. Chaque JTM est programmée via un transistor connecté en parallèle avec chaque branche (WL_n). Pour lire l'information contenue dans les cellules mémoires, une branche de transistors nMOS est sélectionnée permettant ainsi à un courant I_F de circuler à travers les 2 jonctions (R_0, R_{64} par exemple). Par conséquent, ce courant sera comparé à I_{REF} à l'aide de l'amplificateur de lecture afin de lire la valeur en sortie Z.

La raison principale pour laquelle deux JTM sont utilisées par cellule mémoire, est de garantir une lecture fiable. La présence de 6 transistors en série entre les jonctions et l'amplificateur constitue une forte résistance. Ce qui réduit les marges de détection pour l'amplificateur de lecture et augmente sa sensibilité face à la variation du courant de lecture.

Par ailleurs, bien que cette structure paraisse plus dense par rapport à une LUT en technologie SRAM ou une LUT hybride, la présence de deux JTM par cellule mémoire

augmente forcément la surface de la partie magnétique. En outre, cette structure exige des transistors d'écriture ayant des tailles très importantes afin d'écrire deux JTM en série en même temps pour chaque branche de l'arbre nMOS.

Dans le but d'améliorer toujours la densité d'une LUT en technologie MRAM, les études portées dans [11] consistent à proposer une LUT très dense à 6 entrées en utilisant un mode de lecture non différentiel. Le principe repose sur l'utilisation d'un amplificateur de lecture 'single-ended' permettant effectivement de supprimer la partie de référence, comme illustré sur la figure 2.10.

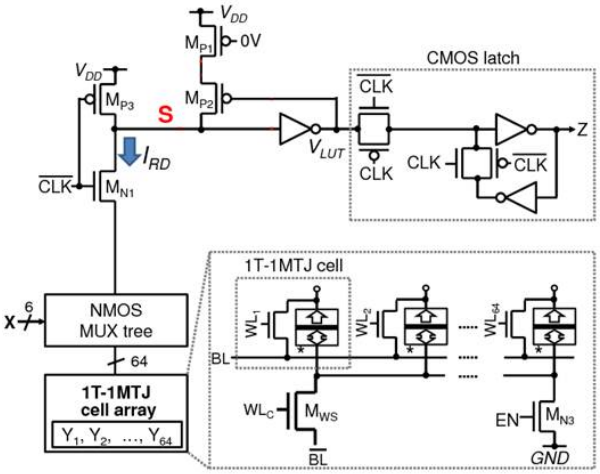


Figure 2.10: LUT à 6 entrées en technologie STT-MRAM formée par un arbre logique CMOS, des cellules mémoires JTM et un amplificateur de lecture 'single-ended' [11]

Le mode de fonctionnement est divisé en deux phases : phase de fonctionnement et phase de maintien. Pendant la phase de fonctionnement, CLK = '0' permettant d'activer le transistor MN1 et désactiver MP3. Cela a pour effet de lire l'information sauvegardée dans la JTM via le courant I_{RD}, permettant ainsi d'avoir une tension V_S qui correspond à un niveau haut ou bas de la JTM sélectionnée. Si V_S est faible, la tension V_{LUT} à la sortie de l'inverseur devient élevée et au contraire si la chute de tension V_S est faible, V_{LUT} devient faible permettant d'activer le transistor MP2 et maintenir une tension élevée au noeuds S. Dans les deux cas, V_{LUT} est amplifié dans l'élément de mémorisation de type "latch" CMOS. Pendant la phase de maintien CLK='1', la valeur de sortie Z est alors maintenue dans le latch CMOS.

D'un point de vue surface, cette architecture semble plus intéressante par rapport aux autres structures étudiées, mais elle est très sensible aux variations de procédés de fabrication magnétique notamment. Pour une faible TMR (Tunnel Magneto Resis-

tance), la variation de l'amplitude du courant à travers la JTM est faible, ce qui induit une difficulté de distinguer un niveau haut d'un niveau bas et donc un taux d'erreur à la lecture qui peut être elevé.

Or, on se confronte très vite avec une telle architecture, comme avec toutes les structures proposées dans l'état de l'art, à un problème de délai de transmission de l'information. On a vu dans toutes les LUT proposées que l'arbre CMOS en série entre l'amplificateur de lecture et les JTM constitue une résistance importante qui réduit la marge de lecture. En outre, le délai de propagation du signal contenant l'information est directement proportionnel au nombre de transistors d'accés en série avec la jonction, ce qui induit alors la réduction de la vitesse de propagation.

Nous présentons donc par la suite la conception d'une LUT innovante à 6 entrées pour laquelle plusieurs versions ont été étudiées en technologie STT-MRAM. La conception de ces LUT s'inscrit dans le cadre du projet interne MAD.

MAD pour Memory Advanced Design est un projet qui intègre plusieurs technologies de mémoires émergentes non volatiles sur des wafers CMOS 200mm en technologie 130nm en configuration MPW (Multi Project Wafer). L'objectif principal de ce projet est de faire une évaluation fine des caracéristiques éléctriques de plusieurs cellules mémoires emergentes (PCRAM, OxRAM, MRAM,..).

2.4 Architecture proposée

Pour pallier au problème de vitesse et de surface des LUT présentées dans la section précédente, nous avons proposé dans ce travail une LUT innovante permettant une lecture fiable indépendamment de la taille de la LUT tout en maintenant une vitesse de la lecture constante indépendante de k. L'innovation est apportée par une séparation de la partie logique, c'est-à-dire l'arbre CMOS, et de la partie magnétique où les informations sont stockées, ce qui n'etait pas le cas dans l'architecture conventionnelle précédente.

Selon la combinaison des entrées possibles, une seule sortie est activée permettant de sélectionner un seul transistor placé en série entre l'amplificateur de lecture et les jonctions. Le courant passe à travers la jonction via ce seul transistor de sélection quel que soit le nombre d'entrées.

Décrivons alors les différents blocs de l'architecture proposée et présentée sur la figure 2.11

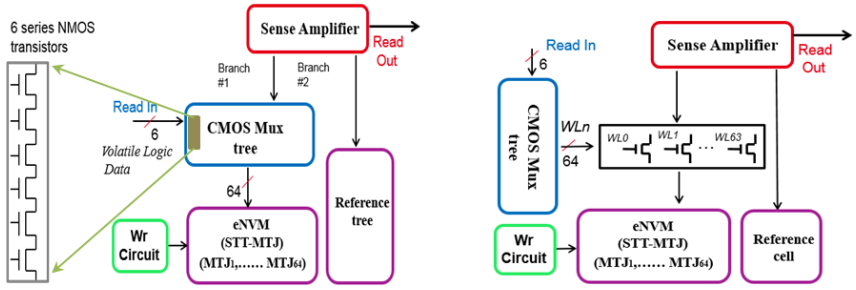


Figure 2.11: Scéma bloc d'une LUT conventielle (sur la partie gauche) et de la structure innovante proposée (sur la partie droite)

2.4.1 Schéma de décodage

Quelle que soit l'architecture de la LUT étudiée précédemment, le schéma de décodage est formé de transistors d'accès en série avec la jonction, appelé décodage par arbre. Ce type de décodage est évidement avantageux en termes de densité car pour k entrées, on aura besoin de $2(2^k\text{-}1)$ transistors d'accès. Cependant, le délai de propagation est directement lié au nombre de ces transistors en série avec la jonction. Ceci augmente le délai de propagation pour lire l'information et rend l'amplificateur de lecture plus sensible aux problèmes de sélectivités.

Donc pour pallier à ces problèmes, nous avons proposé de séparer la partie décodage de la partie cellules mémoires où les informations sont stockées. Le décodeur réalisé en transistors d'accès sert alors à sélectionner un transistor placé en série entre l'amplificateur de lecture et les cellules mémoires. Ce transistor est donc commandé par le décodeur (parmi 2^k branches) alors que les autres transistors sont désactivés. Pour cela, il est donc nécessaire de bien dimensionner la taille des transistors du décodeur ainsi que les transistors de sélection afin d'obtenir une meilleure sélectivité.

2.4.2 Amplificateur de lecture

Le choix de l'amplificateur de lecture est lié à l'application ou plutôt aux performances souhaitées. C'est pourquoi, nous avons décidé d'utiliser l'amplificateur de détection de pré-charge qui offre un bon compromis entre la fiabilité, la consommation et la vitesse.

L'amplificateur de lecture utilisé dans notre architecture est présenté sur la figure 2.12. La phase de lecture comprend deux phases: la phase de pré-charge et la phase d'évaluation. Globalement, les valeurs de résistance dans les JTM sont compatibles avec la technologie CMOS, ce qui facilite la détection de niveau logique. Pendant

la phase de pré-charge, "CLK" vaut '0', les nœuds P1 et P2 sont connectés à Vdd à travers les transistors MP2 et MP3 (les flèches en bleu), pendant que MN2, MN3 et MN4 restent désactivés. Cependant, lors de la phase d'évaluation, "CLK" vaut '1'ce qui active les transistors NMOS (MN0, MN1, MN3, MN4) et désactive à son tour MP2 et MP3. En raison de la différence de résistance entre les 2 branches, le courant de décharge circulant à travers les jonctions est donc différent. La branche ayant la résistance faible atteint plus rapidement la tension de seuil du transistor MP0/MP1. A ce moment là l'autre branche passera à Vdd (le niveau '1' logique). Les flèches rouges et vertes sur la figure 2.12 représentent les deux cas possibles de décharge des nœuds P1 et P2 en fonction de la résistance de JTM.
La sortie et son complément sont ensuite amplifiés à l'aide des deux inverseurs qui jouent le rôle d'un buffer.
Notons d'ailleurs que les transistors MN3 et MN4 jouent un rôle essentiel de l'isolation de la partie lecture pendant la phase d'écriture. Durant cette dernière, ces transistors sont désactivés pour bloquer la circulation du courant dans l'amplificateur de lecture. Nous détaillerons le bloc d'écriture par la suite dans ce chapitre.

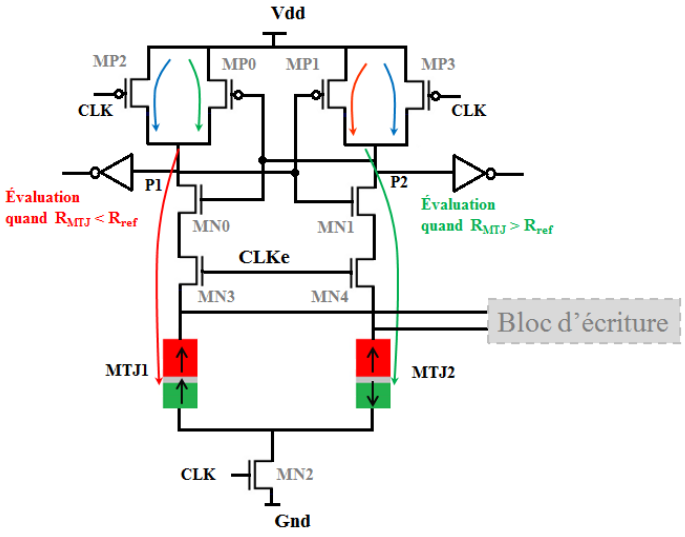


Figure 2.12: Amplificateur de lecture

La figure 2.13 montre une opération de lecture où JTM1 est en état anti-parallèle et JTM2 en état parallèle (états complémentaires). Pendant la phase de "pré-charge", les deux sorties P1 et P2 sont chargées à Vdd à 1.8V. Puis, durant la phase "d'évaluation", "CLK" est activée et le courant de décharge passe à travers la jonction dans chaque branche.
On constate un léger retard de décharge au début de cette phase dû au temps de montée du signal "CLK". A partir d'un certain temps t1, la branche P2 atteint plus rapide-

ment la tension de seuil et continuera à se décharger au fur et à mesure. Tandis que le nœuds P1 bascule dans l'autre sens et commence à se charger pour atteindre le niveau '1' logique.

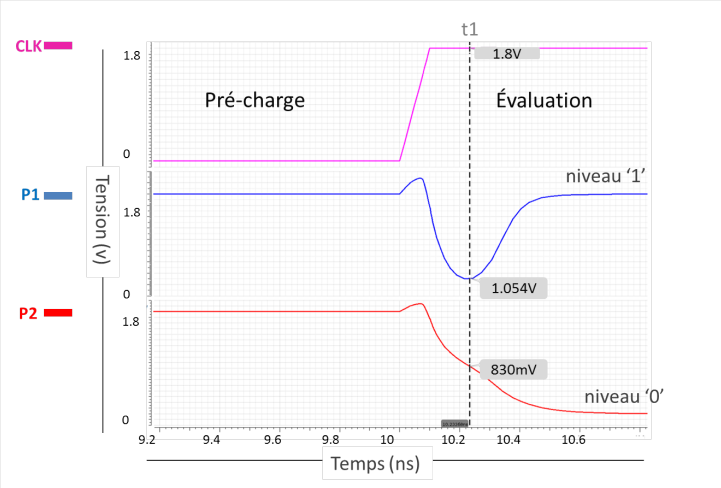


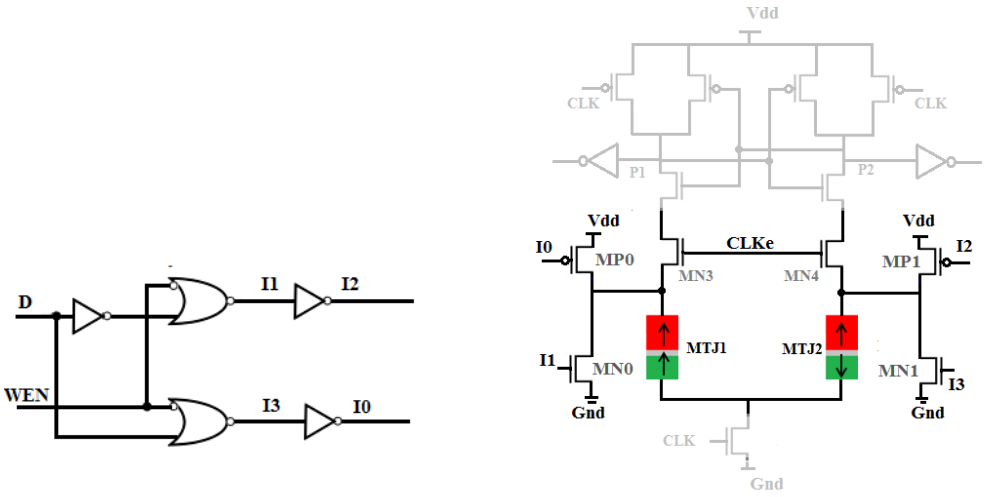
Figure 2.13: Opération de lecture montrant les 2 phases: pré-charge et évaluation

Comme nous venons de l'illustrer, cette opération de détection est très rapide. On constate un temps inférieur à 500ps, ce qui rend cet amplificateur adapté aux applications logiques.

2.4.3 Circuit d'écriture

Retourner l'aimantation de la couche de stockage dans la JTM permet une écriture à '0' ou à '1' logique selon le sens du courant injecté à travers la jonction. C'est le principe d'écriture de la mémoire type STT-MRAM comme détaillé dans le chapitre précédent. Par conséquence, l'étude des circuits d'écriture est importante pour la conception des circuits hybrides CMOS / JTM, car deux facteurs importants y interviennent : la puissance et la vitesse. C'est pourquoi nous présentons par la suite le circuit d'écriture utilisé dans nos circuits LUT en technologie STT-MRAM.

L'écriture de la jonction dans un état parallèle ou antiparallèle consiste à imposer le courant soit dans un sens, soit dans l'autre, selon 4 transistors MP0, MP1, MN0 et MN1, comme illustré sur la figure figure 2.14b Ces transistors sont activés ou non, selon un circuit d'écriture contrôlé par 2 signaux d'entrée D et WEN. D représente la Donnée à écrire et WEN (Write ENable) ative l'opération de programmation (écriture). Ce circuit est composé de 2 portes logiques OU en série avec un inverseur chacune, de façon à avoir une sortie à Vdd et une sortie complémentée à Gnd. Ces derniers vont à leur tour activer/désactiver les 4 transistors d'écriture de manière réversible (MP0/MN1 ou MP1/MN0). On distingue alors 3 cas possibles:



(a) schéma de contrôle des signaux d'écriture afin de créer un courant bidirectionnel

(b) association de 4 transitors d'écriture avec les JTM pour retourner l'aimentataion de de la couche de stockage

Figure 2.14: Principe d'écriture des JTM.

- Lorsque WEN ='0', aucun transistor n'est activé $\Longrightarrow$ I0=I2=1 et I1=I3=0 .

- Lorsque WEN = '1' et D = '0' , les sorties I2,I3 seront donc chargées à Vdd alors que I0, I1 seront à Gnd. Les deux transistors MP0 et MN1 seront donc activés permettant à un courant de circuler à travers les deux jonctions dans un sens. Bien que ces jonctions soient placées en série, le courant d'écriture les traversant a un effet différent. Ce courant traverse la jonction JTM1 en passant en premier par sa couche de référence, permettant d'avoir JTM1 en état antiparallèle. Inversement, le courant traverse JTM2 en passant par sa couche de stockage en premier permettant le retournement l'aimantation de cette couche en état parallèle.

- Lorsque WEN = '1' et D = '1', les jonctions sont écrites dans le sens opposé à la configuration précédente puisque le courant circule dans le sens opposé, c'est à dire JTM1 en parallèle et JTM2 en état antiparallèle. Ceci est dû à l'activation de deux transistors MP1 et MN0 contrôlés par I2 et I1 respectivement.

Indépendamment du sens du courant, les transistors d'écriture devraient avoir une résistance faible afin de générer le maximum de courant. Donc, ils devraient fonctionner dans la zone linéaire où leur résistance peut être exprimée approximativement par les équations ci-dessous.

La résistance de transistor NMOS:

$$R_n = \frac{1}{\mu_n C_{ox} \frac{W}{L}(V_{GS} - V_{TH})} \tag{2.2}$$

La résistance de transistor PMOS:

$$R_p = \frac{1}{\mu_p C_{ox} \frac{W}{L}(V_{GS} - V_{TH})} \tag{2.3}$$

Donc, la valeur de courant d'écriture peut s'exprimer selon l'équation (1.4)

$$I_{oP} = \frac{Vdd}{R_p + R_{ap} + R_n + R_p} \tag{2.4}$$

μ_n est la mobilité des électrons, μ_p est la mobilité des trous, C_{ox} est la capacité associée à l'oxyde de grille, W et L sont la largeur et la longueur de grille respectivement, V_{GS} étant la tension de polarisation de grille et V_t la tension de seuil de charge.

L'ensemble des cas présentés ci-dessus a été validé par simulation électrique (figure 2.15). Cette illustration montre clairement que les opérations d'écriture respectent le fonctionnement décrit ci-dessus.

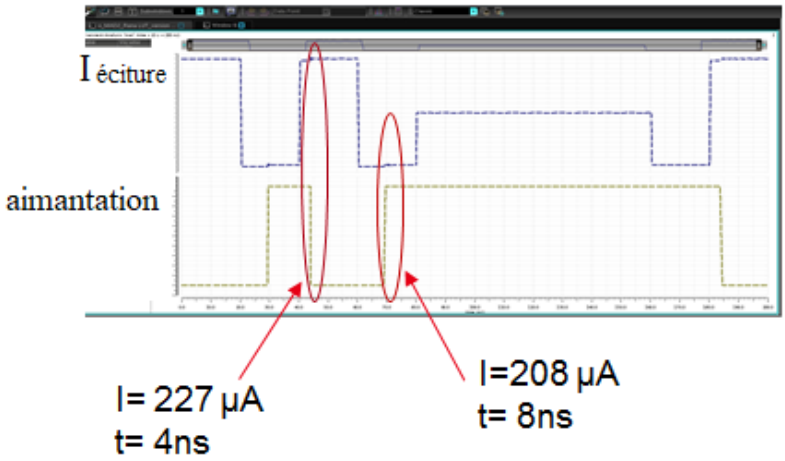


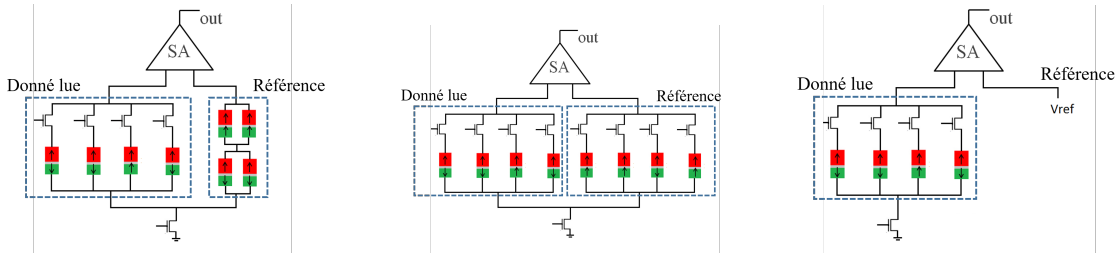
Figure 2.15: Opération d'écriture d'une JTM

La figure 2.15 montre plusieurs opérations d'écriture d'une jonction de 90nm de diamètre. Par ailleurs, afin de pouvoir écrire rapidement, il est évident d'augmenter l'intensité du courant en augmentant la largeur du canal des transistors CMOS tout en gardant une tension aux bornes des JTM inférieure à leur tension de claquage, soit

environ 1V. Cependant, il existe une forte dépendance entre la surface et la vitesse. Il s'agit donc à nouveau de trouver le compromis en fonction des contraintes. Il est possible d'écrire la jonction en 1.5ns au lieu de 4ns avec un courant deux fois plus important avec un transistor 4 fois plus grand.

2.4.4 Cellule de référence

Dans le but d'intégrer notre architecture sur un démonstrateur silicium et garantir un large spectre de fonctionnement, en prenant en compte la variation du procédé de fabrication, nous avons décidé d'intégrer trois types de référence présentées sur la figure 2.16.



(a) Référence A (b) Référence B (c) Référence C

Figure 2.16: Trois types de référence pour lire la JTM

- **Référence A**

  Cette référence est formée par l'association de plusieurs jonctions (soit deux, soit quatre, voire plus) connectées de telle manière à avoir l'état de cette référence fixée à un niveau moyen résistif entre l'état parallèle et l'état antiparallèle. C'est pourquoi le courant traversant cette référence pendant la lecture peut s'exprimer selon l'équation (2.5)

$$I_{ref} = \frac{I_p + I_{ap}}{2} \tag{2.5}$$

  I_{read} représente la valeur de la différence de courant entre la jonction lue et la référence. En effet, le courant traversant une jonction étant soit dans un état parallèle soit dans un état antiparallèle peut s'exprimer suivant cette relation:

$$I_{read} = |I_{ref} - I_p| = |I_{ref} - I_{ap}| = \frac{I_P - I_{ap}}{2} \tag{2.6}$$

$$I_p = \frac{V_{JTM}}{R_p}, \qquad I_{ap} = \frac{V_{JTM}}{R_{ap}}, \qquad I_{ap} = \frac{V_{JTM}}{R_p(1+TMR)} \tag{2.7}$$

V_{JTM} représente la tension appliquée aux bornes de la jonction. Par ailleurs, la variation relative entre le courant de lecture I_{read} et le courant de référence I_{ref} peut être définie selon l'équation ci-dessous:

$$\frac{I_{read}}{I_{ref}} = \frac{I_p - I_{ap}}{I_p + I_{ap}} = \frac{TMR}{TMR + 2} \tag{2.8}$$

On remarque alors que la variation entre les deux courants chute d'une façon remarquable. Dans notre architecture, avec une TMR de 120%, le rapport de courant vaut 37.5%, ce qui rend cette cellule de référence la moins robuste face aux variations de procédé parmi les 3 références étudiées. De ce fait, cette cellule est utilisée plutot pour les applications à haute densité, vu qu'elle occupe une surface relativement compacte.

- **Référence B**

  Cette approche utilise deux cellules pour stocker chaque bit de data. Chacune d'elles étant toujours dans l'état magnétique opposé à son homologue, elles sont donc systématiquement complémentaires. Cette approche est totalement différentielle pour laquelle la variation de courant peut s'exprimer suivant la relation suivante:

$$\frac{I_{read}}{I_{ref}} = \frac{I_p - I_{ap}}{I_p} = \frac{TMR}{TMR + 1} \tag{2.9}$$

  Pour la même valeur de TMR à 120%, la variation de courant vaut 54.5%. Ceci donne 17% d'avantage supplémentaire en terme de robustesse face aux variations de procédés par rapport à l'approche précédente. Bien que cette cellule de référence soit clairement plus robuste que la précédente en termes de variations de procédé , elle est plus contraignante en termes de surface. En effet, le nombre de jonctions est dupliqué afin d'assurer l'aspect différentiel et occupe donc une place supplémentaire.

- **Référence C**

  Cette méthode consiste à utiliser une référence générée par un bloc analogique dans le but d'offrir la valeur exacte de référence vis-à-vis des variations de procédé. Cette référence pourra donc être modifiée dynamiquement afin de se placer au plus précis entre I_p et I_{ap}.
  Dans le cadre de notre démonstrateur, aucun bloc analogique n'a été conçu car les hypothèses sur le procédé étaient trop vastes. Nous avons alors opté pour un signal de référence externe généré via le testeur Diamand D10. C'est la solution

qui paraissait le plus sûr et permettant un maximum de flexibilité et de chance de succès au moment du test.
Nous aborderons la description de ce testeur dans le chapitre suivant.

2.4.5 Cellules mémoires

En ce qui concerne les cellules magnétiques, la technologie STT a été utilisée. La taille choisie pour les JTM étaient de 90 nm, taille qui offrait le meilleur compromis entre rendement et performance dans le cadre du projet auquel nous avons pu participer. Le tableau ci-dessous présente les paramètres essentiels de la jonction en technologie CMOS 130nm et sous une tension d'alimentation de 1.8v.

RA	7,5 [ohm.µm^2]
Rp	1.2 [kΩ]
Rap	2.6 [kΩ]
TMR	120%
$I_{switch(p\rightarrow ap)\ t=1.5ns}$	478 [µA]
$I_{switch(ap\rightarrow p)\ t=1.5ns}$	353 [µA]

Figure 2.17: Caractéristiques de la jonction magnétique utilisée

2.4.6 Simulations électriques

L'ensemble des blocs présentés, ci-dessus, ont été simulés séparément afin de bien dimensionner l'intégralité des transistors. En outre, l'amplificateur de lecture et le circuit d'écriture ont été simulés par des simulations paramétriques, sur de nombreuses itérations. En effet, le but étant de concevoir un démonstrateur, nous avons dimensionné notre circuit d'une façon à ce qu'il soit le plus robuste vis-à-vis des variations de procédé, lors de la phase de fabrication, et à prendre suffisamment de marge pour couvrir ces variations.

Nous avons choisi de présenter les résultats de simulation d'une LUT à 6 entrées basée sur cette architecture innovante.

La simulation électrique permettant de valider le fonctionnement de notre LUT hybride est illustrée sur la figure 2.18. Elle illustre plusieurs phases de fonctionnement de la LUT :

- **Phase de programmation:** Il s'agit de la phase d'écriture durant laquelle le signal WE est maintenu à '1'. Les informations à écrire sont codées par le signal Data

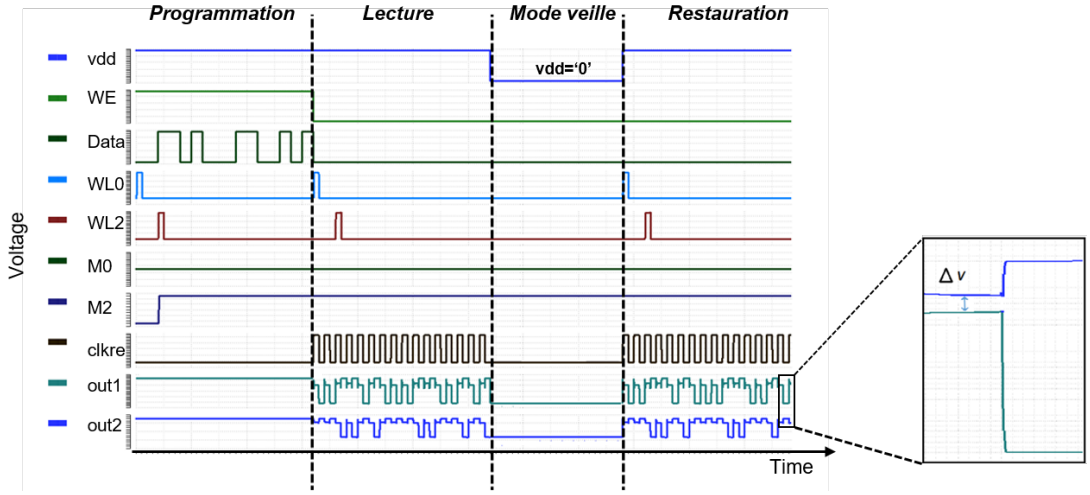


Figure 2.18: Simulation électrique d'une LUT à 6 entrées sous l'environnement Cadence

qui vaut '001101000110010'. WL_n permet de sélectionner la jonction souhaitée. A la fin de chaque sélection, une jonction est écrite à '0' ou '1' logique selon la valeur de "Data". C'est le cas des 2 jonctions JTM0 et JTM2, écrites à '0' et '1' respectivement (M représente l'aimantation de la jonction). Enfin, les sorties out1 et out2 sont maintenues à Vdd (comme décrit précédemment).

- **Phase de lecture:** Cette étape permet de s'assurer de l'état de la jonction écrite lors de la phase de programmation. On retrouve également les valeurs écrites par "Data" dans les 2 jonctions JTM0 et JTM1 ainsi que dans toutes les autres. La sortie out1 vaut donc '001101000110010' et son complément out2 qui est égal à '110010111001101'.

- **Phase en mode veille:** Il s'agit de la mise en veille du circuit pour laquelle l'alimentation est coupée (Vdd vaut zero). On remarque qu'il ne se passe rien pendant toute cette phase et que toutes les sorties sont à '0'.

- **phase de restauration:** Il s'agit d'une phase de lecture pendant laquelle on restaure la valeur stockée préalablement dans les jonctions.

Cette simulation permet de valider d'une part le bon fonctionnement de la LUT et montre d'autre part l'aspect non-volatile. Le fait d'être capable de lire la même information après une coupure de l'alimentation n'engendre aucun dysfonctionnement dans les jonctions magnétiques. Par ailleurs, cela permet de réduire la consommation totale des LUT en les intégrant dans un circuit complet d'un FPGA.

2.4.7 Évaluation des performances

Dans cette partie, nous avons évalué plusieurs caractéristiques de performances de notre architecture en comparaison avec celle de l'état de l'art [12] pour différentes configurations, notamment le nombre d'entrées k. En effet, nous avons présenté précédemment que ce paramètre est un paramètre important pour l'efficacité du FPGA.

Tout d'abord, nous avons évalué le nombre de transistors nécessaires pour plusieurs tailles de LUT, en comparant notre architecture proposée avec une cellule hybride non volatile SRAM de l'état de l'art. On peut constater que l'écart en nombre de transistors devient de plus en plus important lorsque la taille de LUT augmente. Pour une LUT à 6 entrées par exemple, on obtient une réduction de 64% du nombre de transistors NMOS (figure 2.19). Or, il était important de respecter certains dimensionnements de transistors de façon à ce que cela ne soit ni surdimensionné au risque d'engendrer une augmentation de surface, ni sous-dimensionné qui risque évidement de réduire les performances du point de vue de la vitesse. La longueur minimale du canal de transistors utilisés dans nos circuits est de 0.35 µm alimenté en 1,8V.

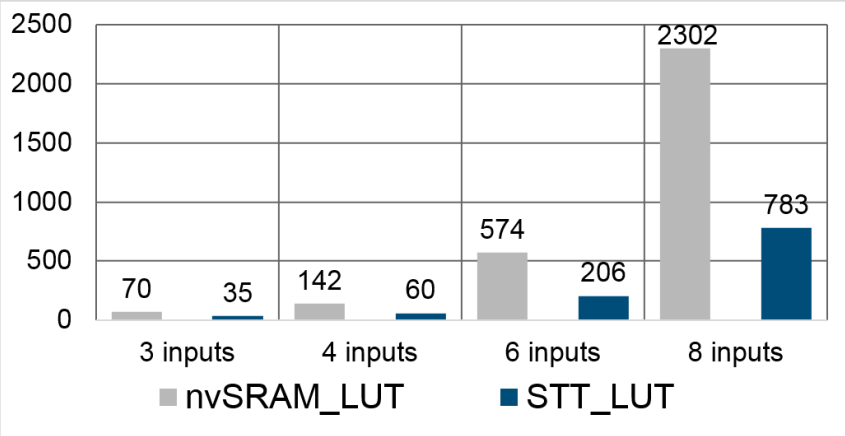


Figure 2.19: Évaluation du nombre de transistors

Par ailleurs, nous avons également évalué l'aspect de consommation dans notre architecture. Généralement, plus une LUT est grande, plus elle consomme. Une grande partie de sa puissance est consommée par la partie magnétique pendant la phase d'écriture des JTM, et par la partie CMOS qui influe en particulier sur sa puissance statique. Or dans notre architecture, la partie logique contenant l'arbre CMOS est séparée du chemin d'écriture et de lecture, ce qui décroît considérablement les courants de fuites. Cette partie fonctionne en mode tension permettant de contrôler les transistors de sélection, donc un très faible courant circule dans ces transistors. Ceci est un avantage par rapport aux architectures étudiées dans l'état de l'art.

La figure 2.20 illustre l'aspect de la consommation pour les 2 types de référence A et B, pendant une opération d'écriture et de lecture. Il est évident que la puissance consommée par une cellule différentielle est beaucoup plus importante par rapport à la référence A, car tout simplement pour chaque phase de programmation il faut écrire 2 jonctions au lieu d'une seule. Il est vrai que pour écrire une jonction, la consommation parait un peu excessive, car dans notre cas les jonctions utilisées sont de diamètre de 90nm. Il existe une forte dépendance entre la taille de la jonction et le courant de programmation. Plus la taille est importante, plus on a besoin de courant pour retourner l'aimantation de la couche de stockage. Le courant critique est proportionnel au volume de la JTM, soit au carré du diamètre, comme on peut le comprendre sur l'équation (2.10)

$$I_c \propto H \times V \tag{2.10}$$

$$V = A \times t = \pi r^2 \times t = \pi \frac{d^2}{4} \times t \tag{2.11}$$

avec H le champ d'anisotropie magnétique perpendiculaire des matériaux ferromagnétiques de la jonction, V le volume de la JTM équivalent à A, le produit de la surface de la JTM, multiplié par t, l'épaisseur de sa couche libre. A vaut πr^2 dans le cas d'une JTM de forme circulaire où r et d représentent son rayon et son diamètre respectivement.

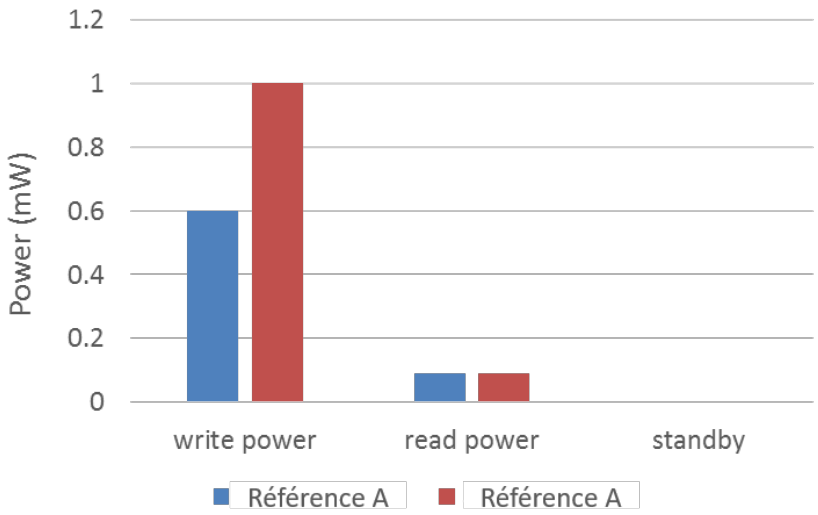


Figure 2.20: Evaluation de la puissance d'écriture et de lecture dans deux types de référence

Enfin, la puissance de lecture est la même dans les différents types de référence. Notons que plus la TMR est grande, plus la lecture est fiable et plus le risque d'erreur lors de la lecture est faible.

En effet, comme nous l'avons cité précédemment, le but est d'avoir un délai de propagation indépendent du nombre d'entrées, ce qui va permettre d'avoir une lecture fiable et rapide tout en augmentant les possibilités de codage de fonction dans une LUT. Le tableau ci-dessous compare le délai de propagation pour plusieurs tailles de LUT entre l'architecture typique étudiée dans l'etat de l'art [12] et notre structure. On constate dans notre cas que le délai de propagation est constant quelle que soit la taille de la LUT, vu qu'un seul et unique transistor est sur le chemin de lecture.

	Sense delay (ps)	
Number of inputs	Typical circuit	Proposed circuit
3	160	103
4	190	103
6	228	103
8	287	103

Figure 2.21: Estimation du délai de propagation entre l'architecture conventionnelle [12] et la structure proposée

A ce stade de la conception, il était important de vérifier si notre structure est assez robuste pour suffisamment tolérer les variations de process (magnétique et CMOS). Pour cela, nous avons effectué des analyses Monte-Carlo en faisant varier plusieurs paramètres de fabrication. Le principe est de lancer plusieurs simulations en même temps et c'est le simulateur qui gère d'une façon aléatoire et automatique la variation de chaque paramètre, comme par exemple la longueur, la largeur et l'épaisseur de grilles des transistors, ainsi que de nombreux autres paramètres technologiques. Pour la partie magnétique, ces variations ont été imposées de façon manuelle et dans le modèle compact des JTM.

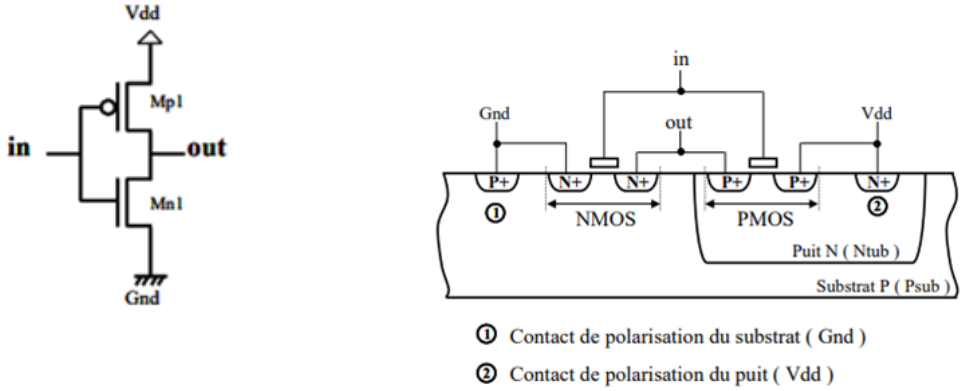
Généralement, ces analyses sont lourdes et consomment énormément de temps. C'est pourquoi nous avons lancé une simulation sur 100 itérations avec plusieurs configurations afin de s'assurer de la robustesse de notre circuit. Les résultats ont révélé un fonctionnement correct sur toutes ces itérations (100/100), permettant une écriture et une lecture fiable du niveau logique pour les 3 types de référence.

2.4.8 Dessin des masques

Après avoir validé une certaine robustesse de notre architecture vis-à-vis des variations en vue de la fabrication du démonstrateur pour lequel ce travail a été effectué, nous avons pu passer à l'étape du dessin des masques, appelée 'layout'. Contrairement aux circuits numériques où les masques sont générés automatiquement à partir

d'une bibliothèque de cellules standard, içi chaque layout est dessiné individuellement dans les circuits full custom. Cela nécessite donc d'importer tous les composants et les interconnecter manuellement en respectant les règles de dessin de la technologie.

Avant de présenter la vue globale de notre layout, nous allons rappeler brièvement la structure physique d'un inverseur afin de montrer la correspondance avec son layout.



(a) Vue symbolique d'un simple inverseur fomé d'une connexion en série des deux transitors NMOS et PMOS

(b) Vue de coupe d'une structure physique d'un inverseur sur un substrat de type P

Figure 2.22: Structure symbolique et physique d'un inverseur.

La figure 2.22a présente la vue symbolique d'un inverseur composé de deux transistors de types différents (PMOS et NMOS) connectés en série. On retrouve dans sa structure physique le substrat de type P sur lequel sont gravés les transistors NMOS. Généralement, un NMOS est formé à partir de deux diffusions de type N dopé (+) qui sont le drain et la source, et d'une grille conductrice en polysilicium séparée par un isolant du substrat. Alors que le PMOS est constitué dans un puit en semi-conducteur de type N dans lequel deux diffusions de type P dopé (+) sont prises pour obtenir le drain et la source. La grille est également séparée du substrat par un oxyde isolant. Comme le montre la vue symbolique, le drain du PMOS est connecté à l'alimentation (Vdd), sa source est connectée au drain du NMOS connecté aussi à la sortie (out). Sa grille est connectée à la grille du NMOS formant l'entrée de l'inverseur et la source de NMOS est finalement connectée à la masse (Gnd). On retrouve de même une prise de contact de polarisation du substrat de type (P+) à Gnd et un contact de polarisation dans le puit de type (N+) à Vdd.

La figure 2.23 illustre une vue de layout de l'inverseur où l'on retrouve toute la description présentée ci-dessus.

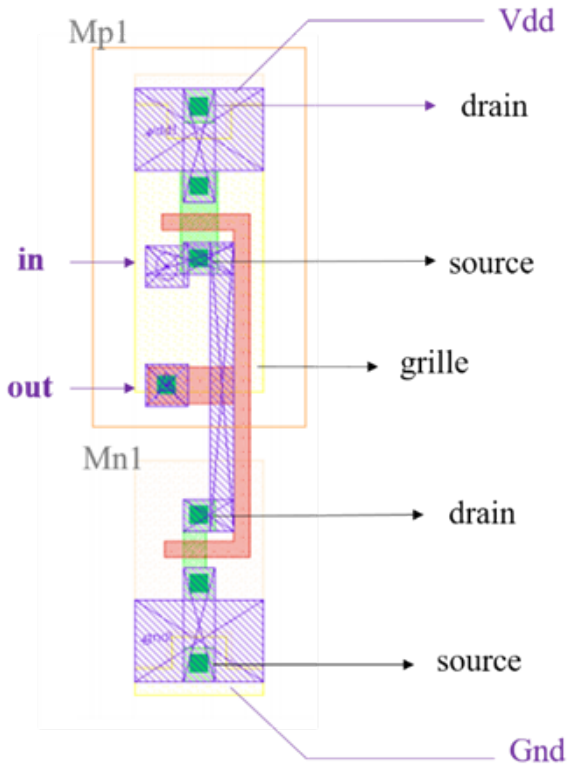


Figure 2.23: Vue de layout d'un inverseur

Etant moi-même inexpérimentée dans ce domaine, le fait de comprendre le layout d'une porte logique simple telle que l'inverseur, et par la suite de cellules plus complexes, a été mon point de départ pour réaliser le layout de notre cellule. C'est pourquoi l'étape de layout était importante et enrichissante dans mon travail car tout a été fait de façon manuelle, ou dit full custom
La figure 2.24 illustre une vue de layout de l'ensemble des niveaux superposés de notre LUT hybride CMOS/magnétique pour 4 entrées.

En vue de la fabrication d'un démonstrateur, qui était un des objectifs principaux de nos travaux, nous avons décidé d'intégrer plusieurs cellules de LUT, ce qui va permettre de multiplier les chances de fonctionnement de nos circuits sur silicium. Nous avons donc décidé de mettre en place vingt cellules, dont les différentes variantes sont présentées ci-dessous:

- LUT à 2 et 4 entrées sans la partie décodeur. Le choix d'une telle cellule est de pouvoir tester sur silicium la fonctionnalité de la partie magnétique (lecture, écriture) en contrôlant les transistors de sélection via le testeur. Un tel choix peut être très utile dans le cas d'un problème dans le décodeur au niveau des transistors d'accès.

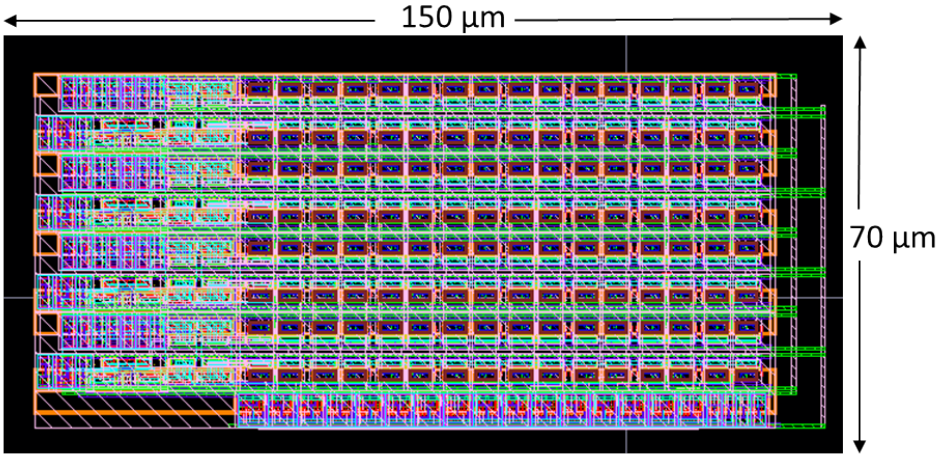


Figure 2.24: Dessin des masques de la LUT CMOS/magnétique à 4 entrées

- LUT à 2 et 4 entrées incluant la partie décodeur.

Chacune de ces 4 cellules a été intégré avec deux tailles différentes de JTM, 60 nm et 90nm, afin d'étudier l'ensemble des performances en fonction de la taille des jonctions, et donc dimensionées en conséquence.

De plus, chaque cellule a été intégrée avec trois types de référence (A, B et C) comme présentées dans la partie précédente afin de pouvoir tester la robustesse sur silicium et surtout d'augmenter fortement les chances d'avoir des blocs fonctionnels, sans privilégier la surface à la consommation.

Ceci donne au total 20 structures différentes dessinées individuellement. Effectivement, les étapes de conception ne s'arrêtent pas là, il est important de vérifier si la vue layout de chaque cellule est conforme au schématique qui a été simulé et qui a permis de valider l'architecture et le dimensionnement. En plus, il est nécessaire de vérifier si les règles de dessin fournies par le fondeur de la technologie ont été bien respectées. C'est pourquoi nous présentons par la suite les vérifications LVS et DRC réalisées.

2.4.8.1 Vérification DRC (Design Rule Cheking)

Cette vérification assure la conformité du circuit intégré aux règles de fabrication du fondeur. Ces règles sont définies dans un manuel appelé DRM, pour Design Rule Manual. Généralement, ces règles de conception spécifient certaines restrictions géométriques des formes et des connexions, afin de garantir le bon fonctionnement du circuit en tenant compte des variations de procédé lors de la fabrication. On retrouve dans ces règles par exemple, la largeur minimale de toute les formes (ligne de métal, via, contact, ...),

ainsi la distance minimale entre deux objets adjacents.

Dans le cadre de ce projet MAD, des règles spécifiques à la technologie MRAM ont été implémentées imposant des contraintes supplémentaires, comme par exemple les tailles des vias des JTM qui étaient 7 fois plus grandes que les vias de la technologie CMOS ainsi que la taille de la JTM par rapport à la taille de la piste de connexion.

2.4.8.2 Vérification LVS (Layout Versus Schematic)

Cette vérification vise à garantir que le layout dessiné est bien conforme au schéma départ du point de vue électrique. Le logiciel de conception reconnait toutes les formes dessinées qui représentent les composants électriques du circuit ainsi que les connexions entre eux et forme un fichier appelé 'netlist'. Celle-ci est comparée par la suite à une netlist similaire issue du schématique. En effet, si un terminal est mal défini ou une connexion est mal réalisée ou manquante par exemple, le LVS indiquerait évidement des erreurs.
A nouveau, le PDK hybride (Process Design Kit) fourni permet de faire ces vérifications tout en incluant les JTM au flot de conception.

Ces deux vérifications sont des étapes cruciales dans la conception full custom. Les premières vérifications DRC, LVS réalisées ont fournies un nombre énorme d'erreurs, donc il était nécessaire de traiter ces erreurs au fur et à mesure. Avec de la patience envers ces multiples erreurs, nous avons obtenu à la fin une fenêtre qui indique que l'analyse est réussie à 100%
Nous avons réalisé ces vérifications pour les vingt structures individuellement, en s'assurant à chaque fois que l'analyse ne contenait plus d'erreurs et qu'elle était correcte à 100%.

Nous avons obtenu au final vingt cellules dessinées et vérifiés séparément, la figure figure 2.25 illustre une vue d'ensembles de ces cellules

2.5 Conclusion

Le fait d'être capable de réaliser la conception full custum d'un circuit intégré de A à Z était l'un de mes objectifs de cette thèse. L'opportunité d'inclure mes propres circuits, depuis l'idée et l'innovation d'une architecture jusqu'au dessin des masques et la réalisation d'un démonstrateur, en passant par toutes les étapes standards de ce type de conception, était une excellente opportunité et a été très formateur.

Figure 2.25: Vue de l'ensemble des cellules intégrées sous forme de barrettes

D'une part, sur le plan scientifique cela m'a permis de mieux comprendre l'architecture et le fonctionnement global d'un FPGA et en particulier sa cellule de base, la Look Up Table. Comme présenté précédemment dans ce chapitre, la taille de la LUT a une influence sur 3 facteurs essentiels: la vitesse, la surface et la consommation. Le problème majeur des architectures étudiées dans l'état de l'art était la forte dépendance entre le délai de propagation et la taille de LUT alors qu'idéalement les FPGA auraient besoin de LUT à plus d'entrées pour pouvoir réaliser des fonctions plus complexes dans un espace réduit. Malheureusement, plus la taille d'une LUT est importante, plus le délai de propagation et la marge de détection du signal de lecture sont affectés à cause de la forte résistance de la partie logique dans les architectures standars de l'état de l'art. C'est pourquoi nous avons proposé de séparer la partie logique de la partie magnétique de sorte qu'un seul et unique transistor soit sur le chemin de lecture. Ceci a pour énorme avantage que la vitesse ne dépend plus du nombre d'entrées de la LUT tout en assurant le même mode de fonctionnement. De plus, la conception en vue de la fabrication exige de concevoir plusieurs versions de circuits de façon à garantir un maximum

de flexibilité vis-à-vis les variations de procédé de fabrication. C'est pourquoi notre choix de l'intégration de plusieurs cellules LUT. L'environnement de test dans lequel ces cellules devront être testées est présenté par la suite dans le chapitre 3.

D'autre part, sur le plan personnel cela m'a permis d'acquérir beaucoup de patience, surtout sur la partie layout pour être capable de finir les 20 structures en respectant une date limite. C'était également une immense opportunité, car cela m'a permis d'acquérir de l'expérience typique à une entreprise, en respectant un cahier des charges et un planning imposé pour fournir le travail prévu.

L'expérience et les compétences acquises dans ces travaux sont très bénéfiques. Elles m'ont permis d'approfondir mon domaine de recherche et d'augmenter la nature de mes travaux dans la suite de ma thèse. Nous aborderons ci-après dans ce manuscrit le deuxième axe de travail, qui est la conception numérique d'un ASIC.

Chapitre 3

Test du demonstrateur

Motivation

Le chapitre précédent a donné un résumé des différentes étapes de la conception full custom. Une attention particulière a donc été accordée aux circuits LUT auxquels s'inscrivent mes travaux de recherche de la première partie de cette thèse. Ce chapitre 3 a pour but de présenter l'environnement de test dans lequel des circuits numériques, analogiques et mixtes peuvent être testés après fabrication.

Nous présentons en détails le test d'un filtre numérique en technologie CMOS/STT-MRAM, conçu par Spintec dans le cadre du projet MAD. L'objectif de ce test était pour moi de me familiariser avec les méthodes et l'environnement de test dans le but de tester mes propres circuits LUT après fabrication.

3.1 Du sytème au silicium

La fabrication d'un circuit intégré est précédée d'une phase de conception durant laquelle s'élaborent les caractéristiques du circuit sur la base de ses spécifications fonctionnelles. La figure 3.1 illustre les différentes relations qui s'enchaînent entre la phase de conception et celle de la mise en boîtier. La réalisation d'un démonstrateur est une phase cruciale, elle permet de passer du niveau "système" au niveau "silicium". Cependant la moindre variation ou erreur de fabrication durant ce passage peut générer des défauts et entraîner un dysfonctionnement du circuit après fabrication. C'est pourquoi il est important d'anticiper et de vérifier le fonctionnement par simulation à chaque phase de la conception car il se peut que la taille des composants après fabrication soit différente de quelques pourcents par rapport à celles dessinées. Il en est de même pour les courts-circuits ou les circuits ouverts. En effet, chaque fondeur donne pour chaque niveau physique une marge dans laquelle il garantit la variation, ce qui fixe la gamme de fonctionnement.

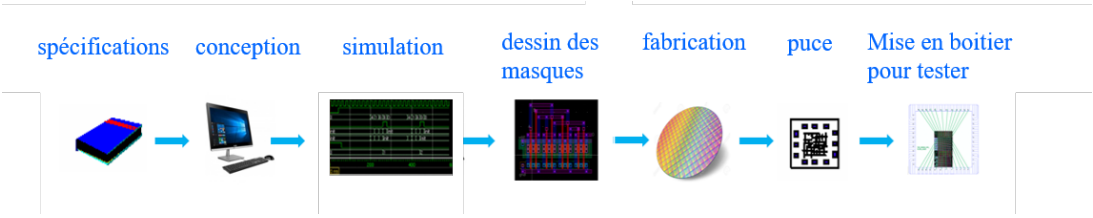


Figure 3.1: Méthodologie de conception

3.2 Démonstrateur: projet MAD

Comme mentionné dans le chapitre précédent, l'objectif principal du projet MAD est l'intégration de plusieurs technologies de mémoires émergentes non volatiles sur une plateforme en technologie 130nm, dans le but de tester les performances électriques de celles-ci dans un environnement hybride sur silicium. Plusieurs versions de ce projet ont été réalisées permettant d'améliorer à chaque fois les caractéristiques électriques obtenues et d'intégrer des circuits de plus en plus complexes. Dans ce cadre, nous avons eu l'opportunité d'intégrer nos circuits innovants dans le deuxième démonstrateur de ce projet, qui est sorti de fabrication en 2018. Il s'agit de l'architecture LUT présentée dans le chapitre 2 avec ses différentes versions, notamment de références. Quant au premier lot, des circuits hybrides en technologie CMOS/STT ont été intégrés avant le début de cette thèse. Il s'agit notamment d'un filtre numérique passe-bas en technologie 130nm, conçu au sein de l'équipe "Spintronic IC design" de Spintec.
Dans la mesure où j'ai réalisé le test de ce circuit, nous présentons ci-après son fonc-

tionnement ainsi que l'environnement de test mis en place. En effet, pour pouvoir réaliser ces tests, il a fallu que je prenne en main son fonctionnement à travers des documentations et en faisant plusieurs simulations.

3.3 Filtre numérique passe-bas CMOS/STT-MRAM

3.3.1 Fonctionnement

Un filtre numérique passe-bas est un filtre qui laisse passer les fréquences inférieures à la fréquence de coupure et qui attenue les fréquences qui sont supérieures à celle-ci jusqu'à les filtrer complètement. L'expression analytique du filtre est donnée par l'équation suivante:

$$S_{out}(n) = \sum_{i=0}^{N_1} a(i) \times S_{in}(n-i) \tag{3.1}$$

- S_{in}: le signal d'entrée
- a(i): le coefficient du filtre
- n: le nombre total de coefficients
- S_{out}: le signal filtré en sortie

Le filtre testé est un filtre numérique à réponse impulsionnelle finie de 32 coefficients (figure 3.2). Il reçoit en entrée "Filter_in", la valeur du signal échantillonné par un convertisseur analogique / numérique (ADC) et délivre en sortie un signal filtré "Filter_out" à un convertisseur numérique / analogique (DAC).

Plusieurs blocs définissent le fonctionnement du filtre:

- Une machine à état « FSM », qui a pour rôle de contrôler le séquencement de calculs dans chaque cycle et de générer le signal Filter_out vers le DAC.
- Un bloc de registre à décalage « Delay line » qui a pour rôle de charger les échantillons fournis par l'ADC.
- Un bloc « RAM coeff » pour stocker les 32 coefficients du filtre passe-bas. Ces coefficients sont calculés au préalable à partir du logiciel Matlab.
- Un bloc « multiplieur » pour multiplier les valeurs échantillonnés par les 32 coefficients selon l'équation (3.1)

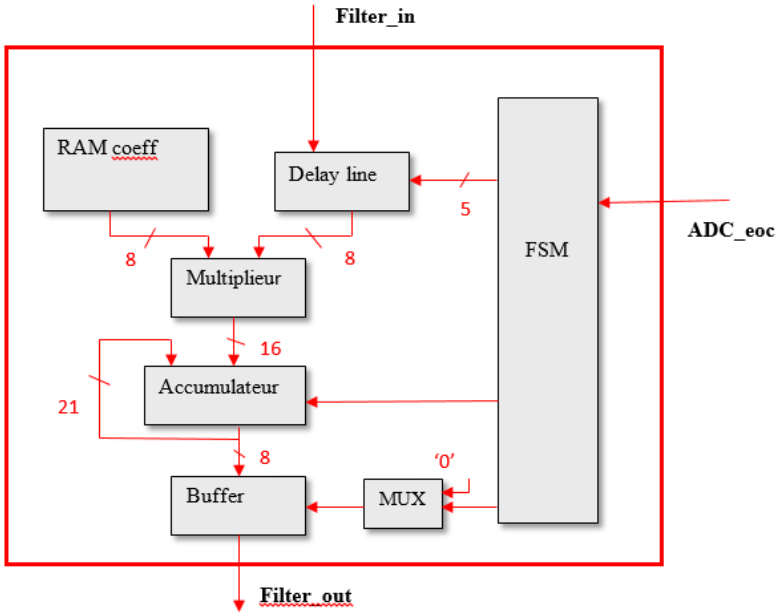


Figure 3.2: Architecture du filtre numérique

- Un bloc « accumulateur » qui permet d'accumuler le résultat des 32 opérations de multiplication par la même valeur échantillonnée, mais décalées par le bloc « Delay line ».
- Un bloc « Buffer » qui mémorise et délivre la valeur de sortie à la fin de calcul seulement et non pas les valeurs intermédiaires.
- Un bloc « multiplexeur » qui assure le contrôle du fonctionnement du buffer pour synchroniser le résultat en sortie.

Le fonctionnement est le suivant :

Une sinusoïde de fréquence variable, échantillonnée à une fréquence f_e = 862 kHz est appliquée à l'entrée du filtre. Cette fréquence correspond au double de la fréquence maximale f_{max} du signal d'entrée selon le theorème de Shannon ($f_e \geqslant f_{max}$).
Le filtre est activé à l'état '1' du signal reset. A chaque fois que le signal ADC_eoc (end of conversion) passe à '0', l'échantillon de l'entrée est chargé puis le signal ADC_eoc reprend la valeur '1'. Ensuite, l'ensemble des coefficients sont multipliés un à un par les 32 coefficients successivement pour donner les 32 valeurs intermédiaires qui seront ajoutées à chaque cycle de période pour donner la somme globale. La valeur de sortie (Filter_out), synchronisée par le buffer, est délivrée à la fin du calcul. Le temps nécessaire donc pour un nouveau chargement de l'échantillon est 35 cycles d'horloge, là où le signal ADC_eoc reprend de nouveau la valeur'0' et ainsi de suite. Le résultat de

simulation est présenté sur la figure 3.3 et detaillé ci-après.

3.3.2 Simulation

Une sinusoïde de fréquence variable est donc appliquée à l'entrée du filtre. La fréquence de coupure mesurée est 89.6 kHz où l'amplitude du signal de sortie (Filter_out) est égale à 70% d'amplitude maximum (128) qui correspond à -3 dB. Cette valeur est obtenue pour une fréquence d'horloge de 50MHz ainsi qu'une période de 1140 ns du signal ADC_eoc (rappelons que cette durée correspond au temps nécessaire de calcul de chaque échantillon par les 32 coefficients).
En diminuant cette période en-dessous de 35 cycles d'horloge, le signal de sortie sera perturbé. Cette perturbation résulte de certaines conversions de valeurs d'entrées qui ne seront pas chargées. Dans ce cas, la valeur de sortie est transmise avant la somme des 32 valeurs intermédiaires.

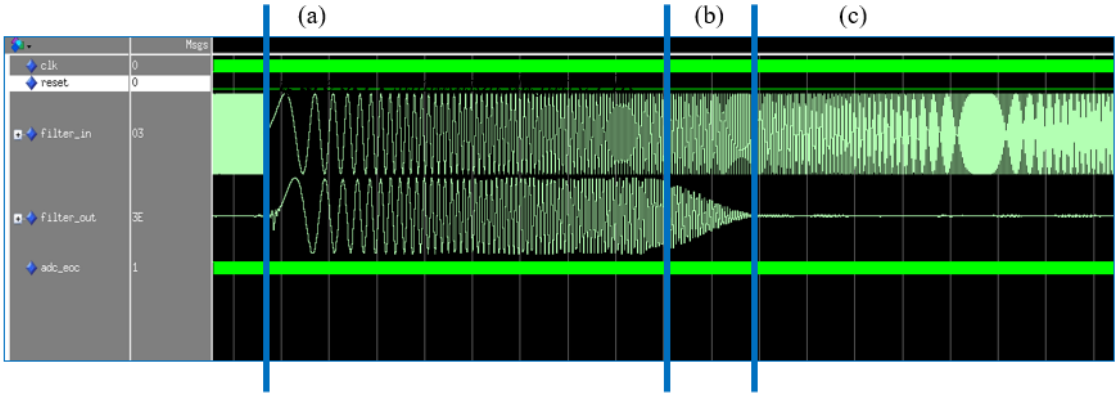


Figure 3.3: Simulation du filtre

La simulation numérique valide donc le bon fonctionnement du filtre passe bas, dans lequel les basses fréquences sont transmises en sortie (zone (a)), les fréquences moyennes sont atténuées (zone (b)) et les hautes fréquences sont coupées par le filtre (zone (c)).

3.3.3 Intégration de jonctions tunnel magnétiques

Le but de l'intégration de jonctions tunnel magnétiques dans le filtre est de sauvegarder de façon non-volatile tous les registres pendant toutes les phases du calcul. Ceci permet de sécuriser les données en apportant de la fiabilité au circuit. En cas de coupure brusque de l'alimentation pendant la phase de calcul, on peut revenir au

dernier état qui est sauvegardé dans les jonctions magnétiques sans refaire tout le calcul de nouveau. De plus, on est capable de sauvegarder la sortie du filtre à n'importe quel moment, de façon non volatile et de la restaurer plusieurs fois en gardant la même valeur tant que les jonctions ne sont pas réecrites de nouveau. La transformation du filtre numérique CMOS en filtre numérique CMOS / Magnétique non volatil est assurée par l'intégration des jonctions dans des bascules de type flip-flop.

3.3.4 Simulation du filtre en technologie CMOS/Magnétique

La simulation de la figure 3.4 montre les deux phases de sauvegarde et de restauration que nous avons souhaité valider. Lors de la phase de sauvegarde, un état est écrit dans les flip-flops, notamment dans leur partie magnétique. Durant la phase de restauration, on remarque que la sortie prend la même valeur que celle sauvegardée. Cela permet donc de restaurer un état connu et stable à n'importe quel moment. On constate également que l'intégration des flip-flops magnétiques ne pertube pas le fonctionnement global du filtre. Le filtre continue tout de même de fonctionner normalement entre la phase de sauvegarde et celle de restauration. Cette simulation valide donc que la fonction de filtrage en mode non volatil est assurée.

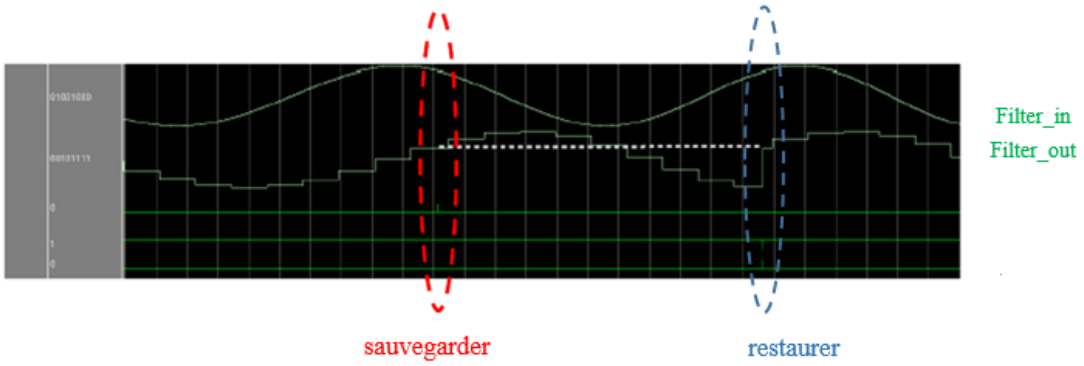


Figure 3.4: Simulation d'un filtre numérique non volatil en technologie CMOS /Magnétique

L'objectif pour lequel ce filtre a été conçu est de valider sur silicium le fonctionnement global de la partie CMOS ainsi que la partie magnétique. Nous rappelons que cette architecture a été conçue par un ancien postdoc de Spintec. Il était donc important de comprendre toutes les parties du fonctionnment du filtre et d'étudier de même le comportement de tous les signaux pour la mise en place de l'environnement de test. Cela m'a également permis d'approfondir mes connaissances dans l'utilisation des outils de simulation numérique, ce qui m'a beaucoup servi par la suite en troisième année.

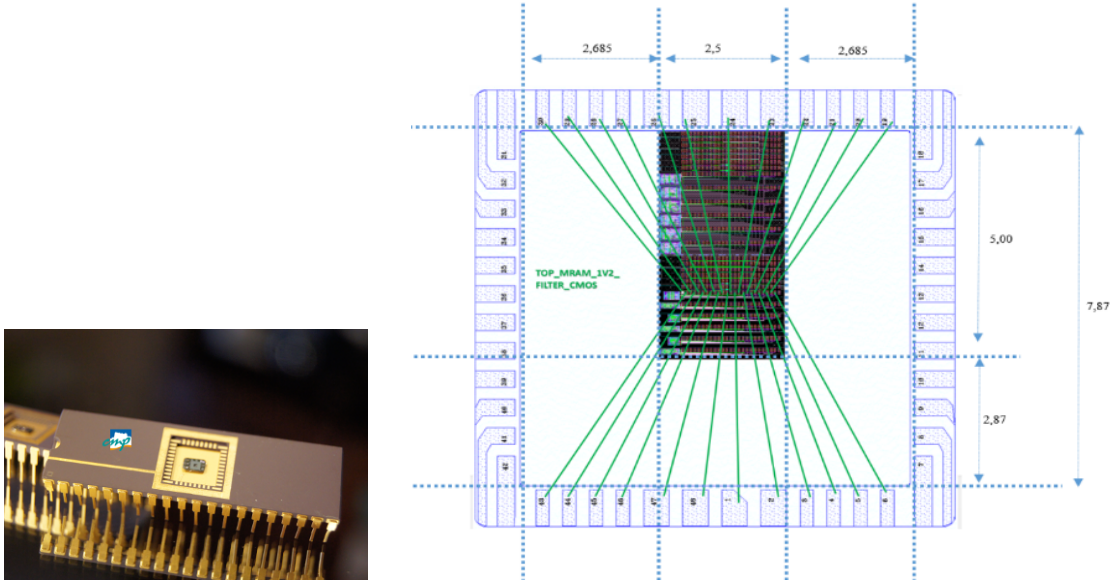
3.4 Environnement de test

Pour que je puisse préparer l'ensemble de l'environnement de test, il est indispensable de connaître de quelle façon le circuit intégré va être conditionné et accessible. Il faut donc au préalable définir le type de boîtier et le moyen de communication entre le boîtier et le testeur.

3.4.1 Découpe et câblage

Pour éviter de faire un PCB, nous avons choisi le boîtier DIL48 pour l'assemblage. Grâce à ce type de boîtier, nous avons pu utiliser une plaque à trous sans soudure et ainsi connecter toutes les pins vers les cartes du testeur.

Le plan de câblage a été réalisé pour les deux versions du filtre numérique en fonctionnement CMOS classique et magnétique. Cette étape consiste à relier tous les plots de la puce aux broches du boîtier à l'aide de fils de quelques dizaines de microns de diamètre. Après avoir pris en main les règles de câblage concernant la longueur des fils et la symétrie par exemple, nous avons fourni à une société spécialisée dans la découpe de wafer et le packaging, un modèle au format word tout en indiquant les données nécessaires importantes pour réaliser le câblage (voir figure 3.5b).



(a) DIL48 de type Ceramic

(b) Plan de câblage du Filter_CMOS dans un boitier DIL48

Figure 3.5: Plan de câblage du circuit à tester

Ce travail était nouveau pour moi. Il m'a permis de mieux comprendre des étapes standard dans la réalisation d'un démonstrateur.

3.4.2 Description du testeur

Les tests se sont déroulés sur le testeur Diamond D10 de chez LTX Credence[13] que Spintec possède .

Figure 3.6: Testeur Diamond de LTX Credence

Ce testeur est destiné au test de circuits numérique, analogiques et signaux mixtes. Il peut gérer 96 signaux numériques et 16 alimentations. Il peut contenir 10 cartes de tests en fonction des besoins. A Spintec, nos besoins nécessitent l'utilisation de seulement deux cartes :

- VIS16 pour les alimentations : c'est une carte analogique qui possède 16 sources de courant/tension quatre quadrants indépendantes. Les tensions max sont +/- 20V avec +/- 300mA et +/- 60V avec 100mA.

- DPIN96 pour les signaux numériques: cette carte peut tester jusqu'à 96 signaux numériques. Pour chaque canal, les niveaux de tension, de courant, les timing, les formats et les paramètres de mesure peuvent être contrôlés indépendamment. Les vecteurs de test peuvent avoir une fréquence de 100 MHz maximum. La

tension maximale des signaux est 12V. Chaque canal peut stocker jusqu'à 32M vecteurs en mémoire.

3.4.3 Présentation du langage test

Le langage STIL (Standard Test Interface Langage) [75] a été spécifiquement développé pour indiquer au testeur l'environnement de test nécessaire en termes d'alimentation, entrées/sorties et leur emplacement sur la carte. Plusieurs programmes utilisant ce langage sont nécessaires pour configurer le test. Ceux-ci sont présentés ci-après.

3.4.3.1 Déclaration des signaux

Cette portion de code se décompose essentiellement en 2 parties. La première partie concerne la déclaration des signaux comme étant une entrée 'In' ou 'Out' et la deuxième partie permet de définir les regroupements de signaux pour faciliter leurs appels dans les stimuli. En effet, les timings, les tensions d'entrée correspondant aux niveaux logiques 0 et 1 ainsi que les tensions seuils des sorties sont indiquées dans d'autres fichiers au format 'stil'.

3.4.3.2 Déclaration des stimuli

Dans cette portion de code sont généralement définies sous formes de vecteurs les valeurs que peuvent prendre les différentes entrées sous forme de groupes de signaux. Ces valeurs peuvent prendre deux niveaux logiques '0' ou '1', quel que soit le type de signal.
Les entrées à appliquer sont : Filter_in à 8 bit, clk, reset et ADC_eoc à 1 bit chacun. Pour définir l'entrée Filter_in qui est une sinusoïde de fréquence variable, un code sous 'Matlab' a été généré afin de convertir l'entrée analogique en entrée numérique sur 8 bits. Après exécution du code, une sinusoïde à temps discret est obtenue et à une fréquence d'échantillonnage de 860 kHz (fréquence maximale de l'entrée) ainsi qu'une liste des vecteurs de 8 bits. Ces vecteurs sous forme de '0' et '1' ont été importés par la suite dans un fichier au format stil afin d'être interprétés par le testeur.

3.4.3.3 Définition des macros

Une macro est l'association d'un ensemble de vecteurs, appelée au moment de l'exécution d'une séquence de test. L'usage d'une macro permet d'appeler un enchaînement de vecteurs définis comme le montre la figure 3.7. La principale difficulté que nous avons rencontré lors de la création de macros était la synchronisation entre les différents signaux d'entrée. Filter_in, clk et ADC_eof ont chacun une fréquence différente. La possibilité de créer des variables ou de séparer les stimulis dans les vecteurs est non acceptée

par le language STIL. Après plusieurs essais effectués pour gérer les timings des signaux, la solution proposée était la génération des plusieurs macros qui décrivent les différentes fréquences. A chaque macros, nous avons donc associé plusieurs instructions de type 'loop'. Une 'loop' permet d'exécuter un ensemble de vecteurs, répété à « n » itérations. A chaque période de filter_in la valeur de « n » est différente dans le but de générer une fonction en escalier non linéaire d'une sinusoïde.
De cette manière nous avons généré plus de 2560 vecteurs pour afficher plusieurs fréquences différentes. Ces macros sont appelées séquentiellement, permettant ainsi d'obtenir l'entrée souhaitée.

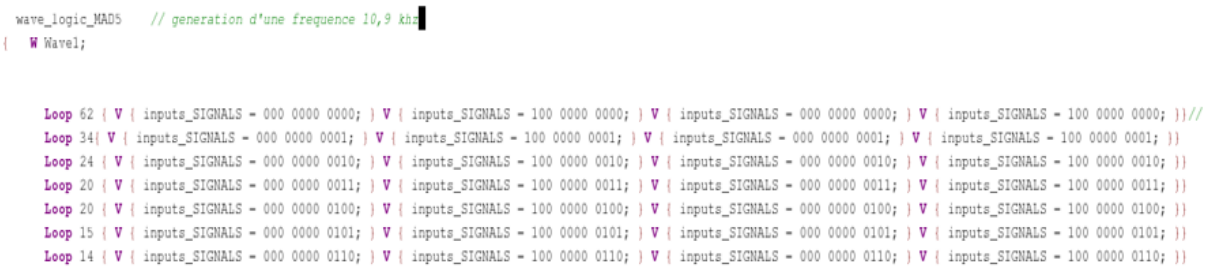


Figure 3.7: Extrait d'une macro décrivant la définition des vecteurs

3.4.3.4 Description des vecteurs de test

La séquence de vecteurs de test est incluse dans une structure « PatternBurst ».

On peut y définir l'ordre dans lequel on souhaite injecter les vecteurs ou les macros dans le système. Un ensemble de macros forme une 'wave' comme le montre la figure 3.8. Plusieurs waves sont nécessaires pour définir la séquence complète à exécuter. Elles doivent être préalablement définies. Des waves qui décrivent les entrées du filtre pour un fonctionnement CMOS classique ainsi que des waves pour décrire le comportement magnétique, c'est-à-dire lecture et écriture des jonctions magnétiques.
Les figures 1.9 et 1.10 illustrent le résultat obtenu sur le testeur.

Le test de façon globale est piloté grâce à un code écrit en langage C. Il permet de côntroler toutes les étapes du test, c'est-à-dire la mise en route, l'extinction des alimentations, le lancement de l'ensemble des stimulis et le type de test (continuité, fonctionnalité, caractérisation,etc).

3.5 Les différents tests

La mise en place du test du filtre a donc débuté par l'écriture du programme qui indique au testeur les ressources matérielles utilisées (entrées, sorties, niveau d'alimentation,

```
Pattern pattern_FILTER
  W Wavel;
// V { inputs_SIGNALS = 000 0000 0000 ;} // on ne fait rien
V { inputs_mag = 0   00 0000 0000  001 ;} //
  Macro wave_logic_MAD0;     // reset
  Macro wave_logic_MAD1;    // Generation d une frequence 3,6 khz
  Macro wave_logic_write;
  Macro wave_logic_MAD2;    // Generation d une frequence 3,6 khz
  Macro wave_logic_read;
  Macro wave_logic_MAD3;    // Generation d une frequence 5,4 khz
  Macro wave_logic_MAD4;    // Generation d une frequence 5,4 khz
  Macro wave_logic_MAD7;    // Generation d une frequence 8,1 khz
  Macro wave_logic_MAD8;    // Generation d une frequence 8,1 khz
  Macro wave_logic_MAD5;    // Generation d une frequence 10,9 khz
  Macro wave_logic_MAD6;    // Generation d une frequence 10,9 khz
  Macro wave_logic_MAD9;    // Generation d une frequence 18 khz
  Macro wave_logic_MAD10;   // Generation d une frequence 18 khz
  Macro wave_logic_MAD11;   // Generation d une frequence 25 khz
  Macro wave_logic_MAD12;   // Generation d une frequence 25 khz
  Macro wave_logic_MAD13;   // Generation d une frequence 56 khz
  Macro wave_logic_MAD14;   // Generation d une frequence 56 khz
  Macro wave_logic_MAD15;   // Generation d une frequence 118 khz
```

Figure 3.8: Extrait de code décrivant la séquence des macros à exécuter pour génener une sinusoïde à fréquence variable

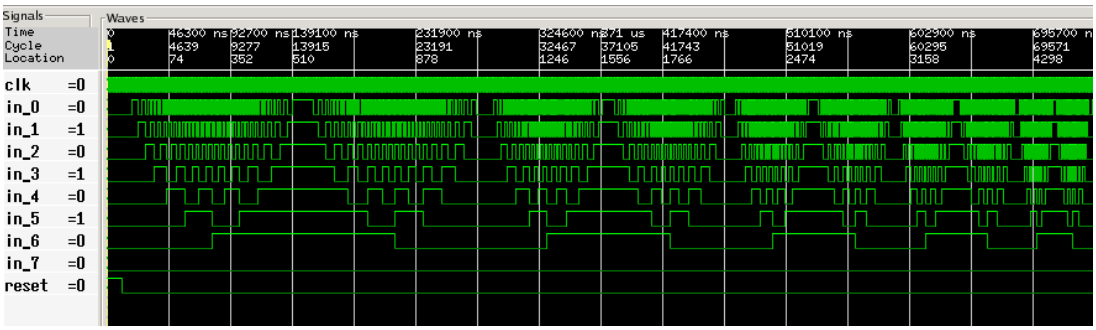


Figure 3.9: Interface graphique indiquant les signaux d'entrée

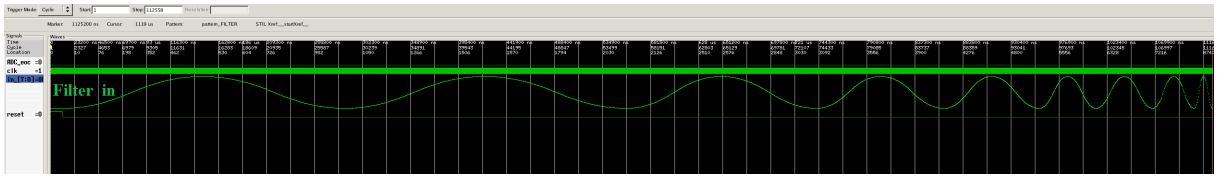


Figure 3.10: Sinusoïde à fréquence variable programmée à partir des vecteurs

niveau de tension et niveau de courant,...). Toutes ces informations ont donc été écrites en plusieurs fichiers au format STIL et dans un fichier complémentaire en langage C qui contrôle toutes les étapes de test. Une fois la partie programmation faite, l'étape suivante était la configuration du testeur en connectant physiquement avec des fils chaque pin du circuit aux cartes DPIN96 et VIS16 du testeur (voir figure 3.12).

3.5.1 Test de continuité

Pour vérifier que la connexion physique est assurée, nous avons d'abord commencé à mettre en place un test de continuité sur tous les signaux.

Principe: Ce test consiste à envoyer un courant dans les plots de connexion et à mesurer une tension. Si la connexion est ouverte il n'y a pas de tension alors que si elle est correcte on lit la tension de seuil des diodes de protection ESD intégrées aux plots du DUT (Device Under Test). Le schéma de principe est illustré en figure 3.11. La tension de seuil est comprise entre 0.2 et 0.9V.

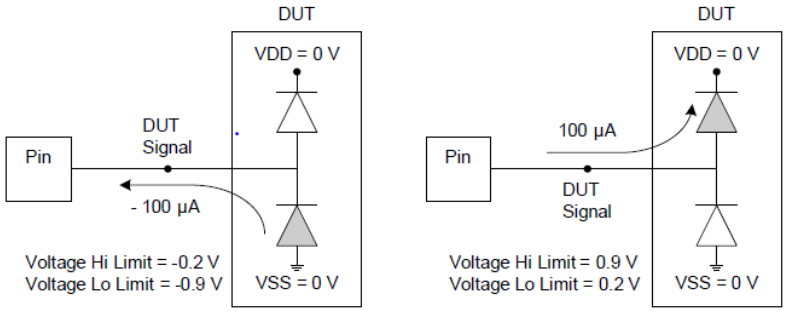


Figure 3.11: Schéma de principe d'un test de continuité [13]

Résultats: Ce test a permis d'identifier un problème concernant de nombreuses mauvaises connexions sur les signaux d'entrées. Nous avons alors pu corriger l'ensemble de ces erreurs de connexion. Après vérification que la connexion physique est bien assurée, nous avons essayé de faire la simulation électrique sur Cadence avec les diodes de protection ESD pour connaître la tension de seuil ainsi que les différentes valeurs de tension/ courant nécessaires pour mesurer les diodes utilisées lors de la conception. Après la modification dans le programme de test de continuité la marge du courant de polarisation des diodes, nous n'avons plus rencontré ce type de problème.

3.5.2 Test fonctionnel

La deuxième partie du test consistait à exécuter les tests fonctionnels pour le circuit pur CMOS.

Principe: Appliquer en entrée la sinusoïde à fréquence variable pour observer en sortie le filtrage à partir d'une certaine fréquence (fréquence de coupure)

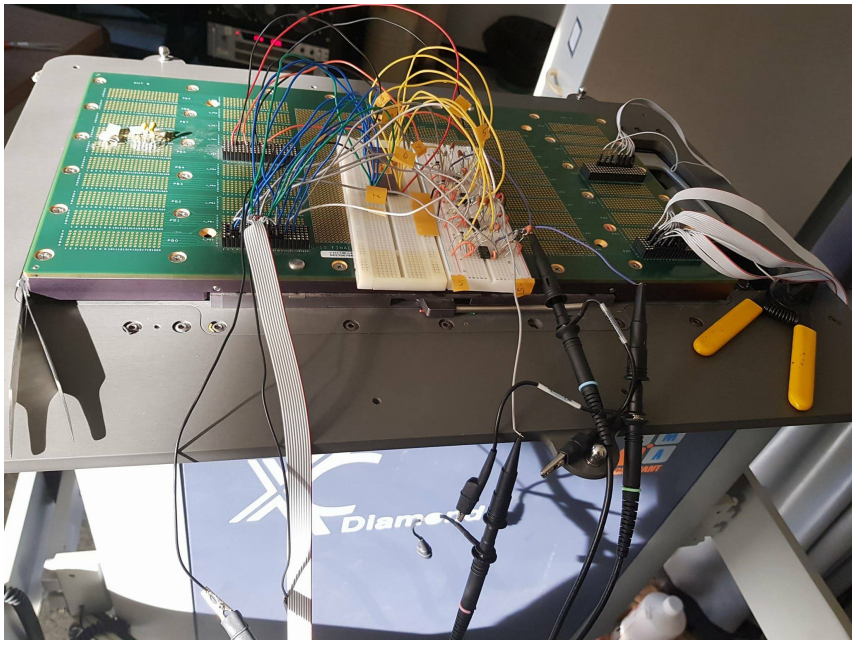


Figure 3.12: Environnement de test

Résultats: Au cours des premiers tests, nous avons remarqué qu'aucun signal ne bougeait en sortie. Nous avons alors regardé les signaux à l'oscilloscope. Nous nous sommes aperçus de la présence d'un fort bruit sur les signaux de sorties. Ces derniers présentant de nombreux pics de tension que l'on peut attribuer à du bruit présent sur l'environnement de test. En effet, les fils qui connectent les signaux du testeur au boîtier sont très longs ce qui favorise l'apparition du bruit. Pour régler ce problème, nous avons décidé de mettre en place un filtre actif de second ordre de type Butterworth pour couper le bruit à haute fréquence (voir figure 3.13). Ceci a pu résoudre partiellement le problème des glitchs sur les entrées et les sorties. Nous avons beaucoup travaillé sur le dimensionnement du filtre Butterworth (valeurs des résistances et des capacités R,C) afin d'obtenir des signaux propres, mais malheureusement des parasites se manifestaient encore. Pour diminuer encore ce problème, nous avons décidé par la suite de remplacer les fils de connexion individuels par des nappes, tel que le recommande le fabricant du testeur et comme c'est l'usage. Nous avons alors intégré une plaque à trous métalliques en soudant les pins d'entrée et de sortie afin de brancher la nappe directement sur la carte d'interface de testeur (voir figure 3.14).

A l'issu de ces améliorations, nous avons réussi à obtenir sur le testeur et son environnement graphique une réponse répétitive avec un bruit très faible, ce qui n'était pas le cas auparavant.

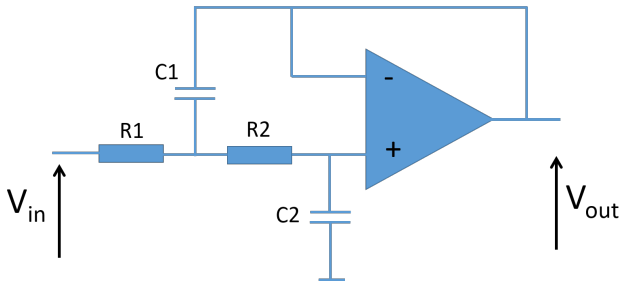


Figure 3.13: Filtre actif passe bas de second ordre de type Butterworth

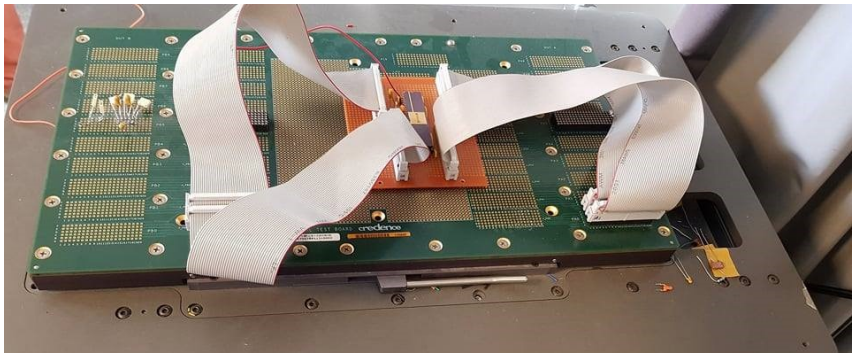


Figure 3.14: Intégration des nappes de connexion sur la carte de testeur

Cependant, les différentes versions de filtres ne fonctionnaient pas parfaitement, les 8 bits de sorties variaient de façon identique, ce qui n'est le cas dans le fonctionnement attendu. La figure 3.15 montre un chronogramme de la phase de test mesurée sur l'oscilloscope. Nous remarquons qu'une sortie parmi les quatre varie d'une manière correcte à une frequence variable, ce qui n'est pas le cas pour les autres. Les trois autres sorties changent d'états d'une façon identique, ce qui n'est pas tout à fait cohérent avec la réponse attendue.

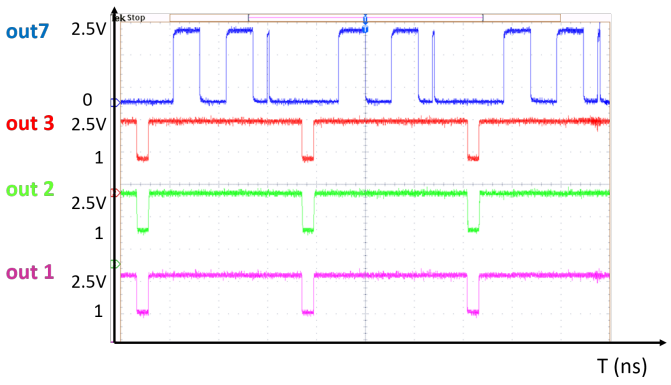


Figure 3.15: Réponse du filtre mésurée à l'oscilloscope

Nous avons beaucoup travaillé sur les stimuli d'entrée afin de diminuer la fréquence du signal filter_in en dessous de la fréquence de coupure. Mais malgré cela, nous nous sommes aperçus du même comportement des signaux de sortie. Nous avons décidé ensuite de vérifier le câblage au microscope, mais aucune coupure sur les fils n'a pu être identifiée. Ceci est logique car si c'était le cas, nous aurions obtenu des sorties figées à un état stable '0' ou '1'. Nous avons donc cherché à comprendre les raisons pour lesquelles le filtre ne fonctionnait pas.
D'abord nous avons essayé d'augmenter la tension d'alimentation de 2.5V à 3V jusqu'à 5V. Les résultats de test étaient toujours identiques aux précédents.
Ensuite nous avons pensé aux buffers de sortie intégrés dans le filtre pour le signal filter_out (de 8 bits). En effet, chaque sortie est pilotée par un buffer pour restaurer le signal et capable de supporter de fortes capacités de charge comme les plots de bonding par exemple.

Effectivement il était impossible de faire les tests sans les buffers car le circuit à tester n'est qu'une "boîte noire" dans laquelle se trouvent des centaines de cellules. Lors de la conception, il n'avait pas été prévu de signaux de contrôle interne. Nous avons vérifié le layout des buffers pour s'assurer qu'aucun court-circuit n'est établi. Malgré cela, nous n'avons pas pu comprendre la raison pour laquelle les sorties ne répondaient pas correctement. Nous nous sommes alors retournés vers la version magnétique en gardant beaucoup d'espoir que le comportement du filtre ne soit pas le même.

Dans un premier temps, nous avons testé la partie CMOS uniquement en inhibant les signaux magnétiques. Malheureusement nous avons rencontré le même problème qui se manifestait toujours pour quelques pins de sortie du signal filter_out. L'activation des signaux magnétiques ne nous a pas trop permis de mieux comprendre si une écriture ou une lecture avait lieu. Il était difficile d'interpréter le résultat car quelques sorties étaient toujours figées à une fréquence constante, donc impossible de retracer une forme sinusoïdale avec moins de la moitié des bits du mot de sortie. Cependant, il s'est avéré dans les deux versions que les fréquences supérieures à la fréquence de coupure sont bien filtrées car les sorties étaient complètement à zero pour des hautes fréquences. Pour des fréquences proches de la fréquence de coupure, la fréquence de sortie était cohérente pour quelques signaux seulement. Notons que parmi les 8 bits de sortie, seul 3 signaux ont révélé un comportement correct. Ceci nous a alors permis de vérifier que la fonction de filtrage est quasi fonctionnelle même pour le circuit hybride. En ce qui concerne la partie magnétique, il n'a pas été possible de montrer que l'en arrivait à écrire et lire des jonctions.

3.6 Conclusion

Ce travail a été une période formatrice au cours de laquelle j'ai acquis une grande expérience. Le but étant de se familiariser avec un nouveau langage de programmation (STIL) afin de tester mes propres circuits conçus, les LUT. Cela m'a appris une certaine méthodologie et des concepts de base du test. En ce qui concerne le test du filtre, il était difficile de tirer une conclusion certaine, la controlabilité du circuit n'était pas assez importante. Ceci est étrange et laisse des interrogations à ce jour. C'est vrai que plus le circuit est complexe, plus le procédé de fabrication hybride est relativement sensible aux variations. Il suffit qu'un via soit mal connecté ou deux pistes en court-circuit par exemple pour rendre certains signaux non fonctionnels. Il est tout de même possible et il y a une forte chance que le problème réside dans les buffers de sortie pour lesquels nous avons plusieurs interrogations. Ce qui peut expliquer pourquoi les diodes de protections n'étaient pas polarisées aux bonnes valeurs lors des premiers tests de continuité.

Cependant la conclusion que l'on peut tirer c'est que la fonction de filtrage est quasi bien assurée. Les basses fréquences sont apparemment transmises en sortie et les hautes fréquences sont coupées par le filtre en version CMOS et CMOS/STT-MRAM. Ce qui peut être une phase très importante pour les prochains circuits à tester.
La difficulté de la phase de test de nos circuits est d'une part la complexité du circuit qui comporte des centaines de portes logiques, et d'autre part que nous n'avons pas accès aux signaux internes, mais seulement aux entrées/sorties de la "boîte noire". La compréhension des résultats de test devient fastidieuse lorsque le circuit est de plus en plus complexe avec une controlabilité faible. Ceci a d'ailleurs été un point important auquel nous avons essayé de nous méfier pour la conception des LUT. Comme présenté dans le chapitre précédant, nous avons intégré plusieurs configurations de circuits afin de multiplier les chances de fonctionnalité de ces circuits sur silicium.
Cette partie de test en début de thèse m'aura fortement aidé à me poser les bonnes questions lors de la conception de mes propres architectures et dans l'implémentation sur le démonstrateur.

Nous pouvons mentionner qu'à ce jour, le procédé de fabrication de la partie magnétique des LUT est toujours en cours. En effet, ce procédé est confronté à des verrous technologiques ne permettant pas d'aboutir facilement à un demonstrateur hybride à partir d'un procédé dans un environnement de recherche académique, comme cela pourrait être le cas pour un procédé industriel.

Chapitre 4

Conception d'une mémoire embarquée en technologie SOT-MRAM

Motivation

Les travaux présentés dans ce chapitre concernent la réalisation d'une mémoire complète de 32kbits en technologie SOT-MRAM intégrant tous les blocs numériques et analogiques périphériques. La première partie de ce chapitre sera consacrée à une étude de quelques structures de mémoires proposées en technologie SOT-MRAM afin d'évaluer leurs eventuels gains au regard de l'application souhaitée. La deuxième partie s'intéressera à la description d'un processeur, une large section etant consacrée à une étude de l'integration des mémoires MRAM conçues dans l'architecture de ce processeur.

4.1 Mémoires à semi-conducteurs

On peut distinguer deux types d'applications parmi les mémoires à semi-conducteurs: la mémoire autonome (Stand-alone memory) et la mémoire embarquée (Embedded memory). La mémoire autonome est, comme son nom l'indique, une mémoire autosuffisante en termes de ressources. Elle est utilisée dans la plupart des produits électroniques tels que les disques durs, les clés USB, etc. Actuellement, elle couvre une grande partie du marché des mémoires à semi-conducteurs avec une grande capacité de stockage (>1Gb). Contrairement à la mémoire autonome, la mémoire embarquée est intégrée sur la puce d'un circuit logique à haute performance, en géneral un processeur ou un microcontrôleur. Elle possède généralement une faible capacité de stockage, typiquement de quelques dizaines à quelques centaines de Mbits. Le choix d'embarquer une technologie de mémoire est dicté par l'application.

La bibliographie présentée au cours du premier chapitre a montré que les fabricants de semi-conducteurs développent les mémoires émergentes afin de surmonter les limitations des mémoires standard de type SRAM, DRAM, Flash.

Dans ce contexte, nous avons décidé d'aborder dans le deuxième axe de nos recherches la conception d'une mémoire embarquée non volatile en technologie SOT-MRAM. Le but principal était d'intégrer cette mémoire dans un processeur afin de voir si cela peut améliorer ses performances par rapport à l'utilisation de mémoires volatiles SRAM.

4.2 Architecture d'une mémoire embarquée

Indépendamment de la technologie utilisée, une mémoire intégrée est conçue selon la combinaison de différents blocs, comme illustré sur la figure 4.1. Chaque bloc a une fonction spécifique ainsi qu'un impact direct sur les performances et les caractéristiques finales de la mémoire.
Nous présentons, ci-après, chacun de ces blocs et nous décrivons les caractéristiques essentielles de notre mémoire SOT-MRAM conçue selon le flot "full custom" en technologie 28 nm FDSOI.

4.2.1 Matrice mémoire

Le bloc capital formant l'architecture d'une mémoire est la matrice de bitcells (point mémoire) où les données sont stockées. L'organisation de la matrice des bitcells est un facteur important ayant un impact direct sur les performances et la taille de la mémoire. Une matrice est généralement organisée en n lignes et m colonnes pour obtenir une capacité de stockage de n x m bits (voir figure 4.2). A chaque intersection (ligne,

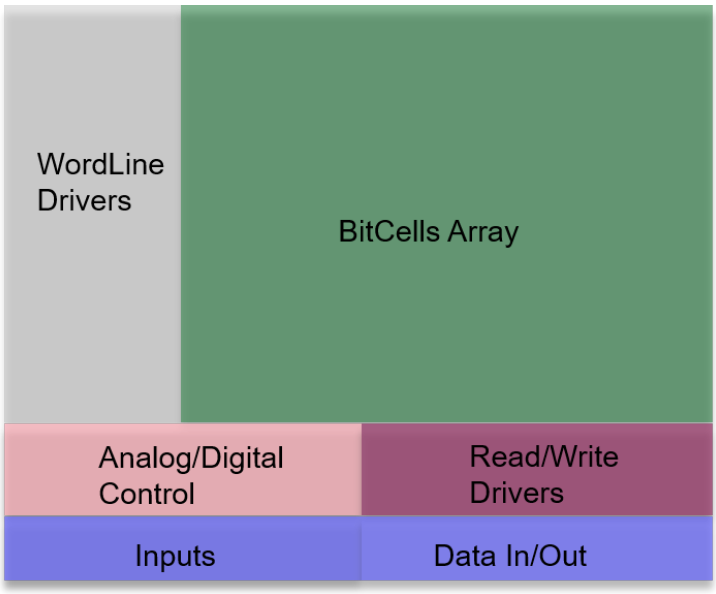


Figure 4.1: Architecture d'une mémoire embarquée

colonne) on trouve une bitcell qui contient un seul bit de donnée. Dans le cas d'une architecture SRAM, la structure d'une bitcell est illustrée par le zoom de la figure 4.2. Elle est formée de 6 transistors, parmi lesquels 4 transistors constituent 2 inverseurs montés tête-bêche, et 2 autres transistors d'accès. L'information à sauvegarder est chargée par la ligne BL et son complément (BLB). Elle est stockée sur deux nœuds de stockage complémentaires, l'un qui contient l'information à '0' alors que l'autre contient son complément '1'. L'écriture d'une bitcell consiste à imposer l'état du point (n,m) alors que la lecture consiste à récupérer l'information stockée.

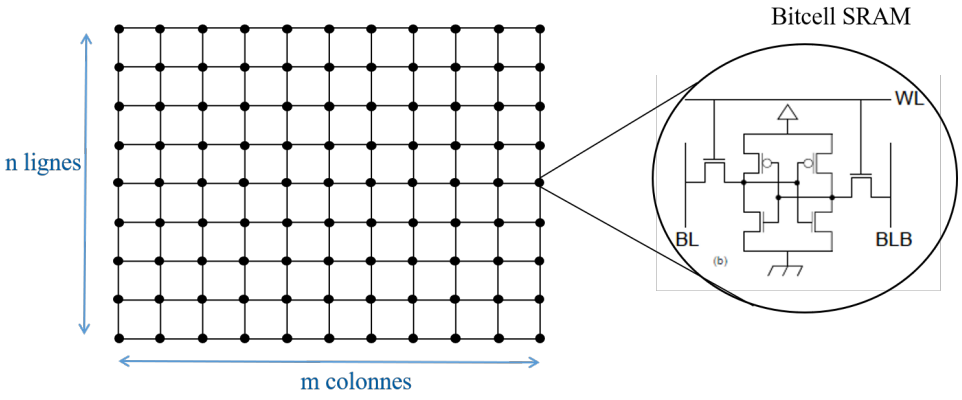


Figure 4.2: Organisation matricielle des bitcells en technologie SRAM

La gestion des lignes et des colonnes nécessite différents blocs internes tels que : le décodeur de lignes, le multiplexeur de colonnes et la logique de contrôle. Cette organisation en lignes et en colonnes peut cependant s'avérer complexe. Nous détaillerons

cette partie dans le paragraphe suivant.
Notons ainsi que le dimensionnement des transistors d'une bitcell dépend entièrement de l'application souhaitée. Ils ne doivent ni être sous dimensionnés pour ne pas pénaliser les performances telles que la vitesse, ni être surdimensionnés ce qui peut produire une surface importante ainsi qu'une surconsommation.

4.2.2 Décodage d'adresses

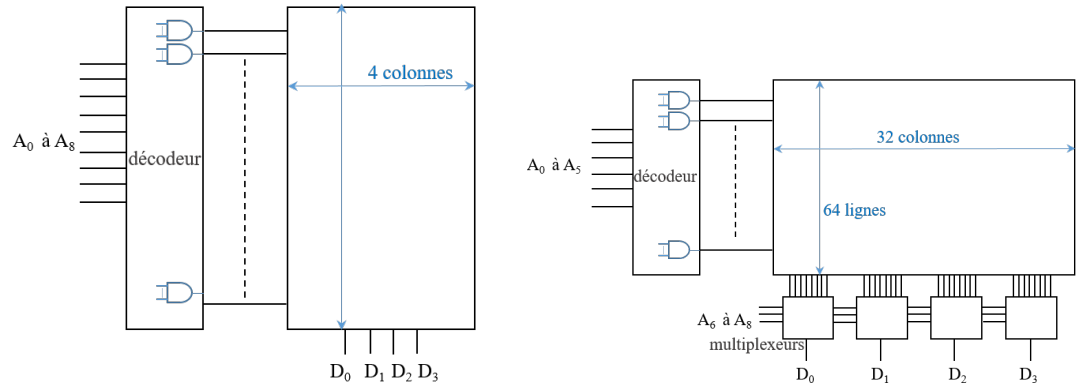
L'adressage des cellules à l'intérieur de la matrice nécessite un certain nombre de connexions entre les bitcells et les portes logiques, formant les décodeurs. Globalement, cet adressage se fait en deux temps : premièrement la sélection d'une ligne et ensuite la sélection d'une colonne.
L'organisation des cellules adressables est une étape cruciale dans la conception d'une mémoire. Elle a un impact direct sur la performance de la mémoire du point de vue vitesse et consommation. C'est pourquoi, une architecture matricielle des bitcells aussi carrée que possible est préférée.

Prenons par exemple, une mémoire contenant 512 mots de 4 bits, soit 2048 bits (figure 4.3a). Pour réaliser le décodeur de lignes, il faut 512 portes ET sans avoir besoin de décodeur de colonnes, car l'information est accessible directement via l'une des 4 colonnes. Une réduction du nombre de portes logiques peut être obtenue en organisant la matrice en 64 lignes et 32 colonnes (2048= 512 $\times$ 4 = 64 $\times$ 32) (voir figure 4.3b). Dans ce cas seulement 64 portes ET sont donc nécessaires pour le décodage ligne et seulement 4 multiplexeurs pour l'adressage colonne. Ces multiplexeurs permettent de sélectionner une colonne parmi les 8. Il faut donc compter 8 portes ET et une porte OU par multiplexeur. On obtient au total 64 + (9 $\times$ 4) = 100 portes logiques contre 512 pour une architecture 512 $\times$ 4.
Cela représente cinq fois moins de portes par rapport à la configuration de la figure 4.3a avec des pistes de connexion plus courtes et donc un délai de propagation des signaux réduit. Ce délai est dû au fait que les capacités parasites des lignes sont proportionnelles à la longueur de la piste. La valeur d'une capacité parasite entre deux lignes de métal est définie par la relation ci-dessous :

$$C = \varepsilon_r.\varepsilon_0.\frac{A}{d} \tag{4.1}$$

ε_0: permittivité relative du vide
ε_r: permittivité relative de l'isolant
A: surface du condensateur

d: épaisseur entre les 2 armatures de la capacité



(a) Organisation d'une matrice mémoire de 2048 bits (512 × 4)

(b) Organisation d'une matrice mémoire de 2048 bits (64 × 32)

Figure 4.3: Schémas d'adressage d'une mémoire de 2048 bits

4.2.3 Le bloc de contrôle

Le mécanisme de contrôle est mis en œuvre à l'aide d'un ensemble de registres et de circuits de contrôle formés de cellules de bibliothèques standards, et d'autres circuits eux-mêmes conçus au niveau transistors.

Ce bloc est en quelque sorte le cerveau de la mémoire dans la mesure où il va piloter tous les signaux de contrôle et les interfaces d'entrées-sorties principales de la mémoire. Les parties d'écriture et de lecture sont systématiquement pilolées par ce bloc. Celui-ci a pour rôle donc de: i) générer les aspects de contrôle des différents signaux internes de la mémoire, ii) décoder les adresses, iii) piloter les différents sous-blocs, iv) synchronisation entre les différents signaux d'entrées à l'aide de registres.

4.2.4 Amplificateurs de lecture

Pendant la phase de restauration, l'utilisation d'un amplificateur de lecture 'SA' (Sense Amplifier) est primordiale. Il est le composant le plus critique dans la périphérie de la mémoire. Son rôle est d'amplifier la tension d'entrée différentielle, ΔBL, développée entre les deux bitlines d'une bitcell. Dans le cas d'une SRAM, les 2 signaux BL et BLB sont connectés à l'amplificateur et sont préchargés à Vdd (voir figure 4.4). Ainsi, lorsque les transistors d'accès sont activés par le signal WL, la ligne BL connectée au bit stocké '0' va se décharger à travers le transistor d'accès dans un des 2 inverseurs.

Ceci provoque une chute de tension qui sera détectée par l'amplificateur de lecture. Cette lecture se fait donc de façon différentielle entre les signaux BL et BLB.

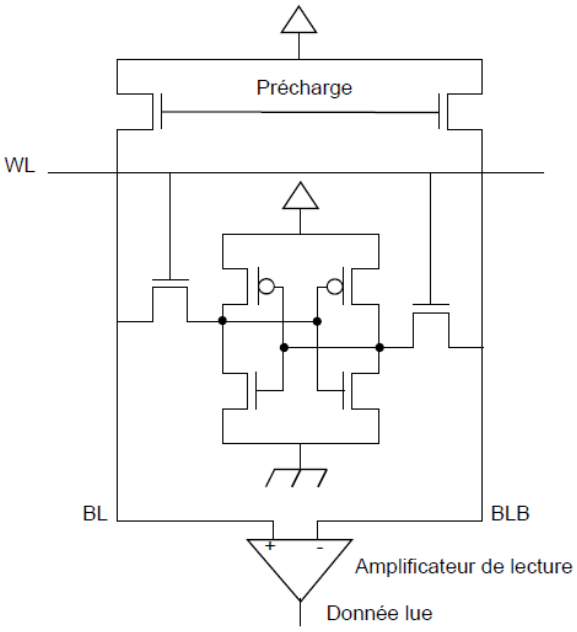


Figure 4.4: Cellule SRAM connectée à un amplificateur de lecture

En ce qui concerne une mémoire MRAM où l'information est sauvegardée sous forme de deux états résistifs distincts de la JTM, il existe deux modes pour lire l'information sous forme binaire : le mode courant ('current sensing') où on lit le courant traversant la JTM pour une tension de polarisation donnée, et le mode tension ('voltage sensing') où l'on récupère la tension aux bornes de la JTM pour un courant donné. Quel que soit la méthode appliquée, le paramètre récupéré (tension ou courant) est comparé à une valeur de référence dans l'optique de déterminer l'état magnétique du dispositif considéré et donc sa résistance.

Le schéma de la figure 4.5 montre une structure de lecture d'une bitcell en technologie SOT-MRAM dans laquelle un miroir de courant formé par les transistors PMOS, et un amplificateur de lecture sont présents. Les 3 lignes BL, BL_p et BL_{ap} sont tout d'abord préchargées à Vdd. Ceci permet à 3 courants de circuler à travers les transistors de lecture NMOS (Re) et donc à travers les 3 jonctions. R_p et R_{ap} forment une résistance de référence, elles n'ont pas les mêmes valeurs car elles ne sont pas dans le même état magnétique. Le courant dans chaque branche est donc différent. A cet instant de la phase de lecture, les nœuds V_+ et V_- ont des valeurs distinctes de tension qui seront détectées par l'amplificateur SA.

La conception de l'amplificateur de lecture doit tenir compte de l'impact du dimen-

sionnement des transistors d'accès du décodeur sur le chemin de lecture car ce chemin est aussi emprunté lors de la lecture de la mémoire. L'arbre du décodeur a donc un impact direct sur la tension aux bornes de la JTM nécessaire pour la lecture.

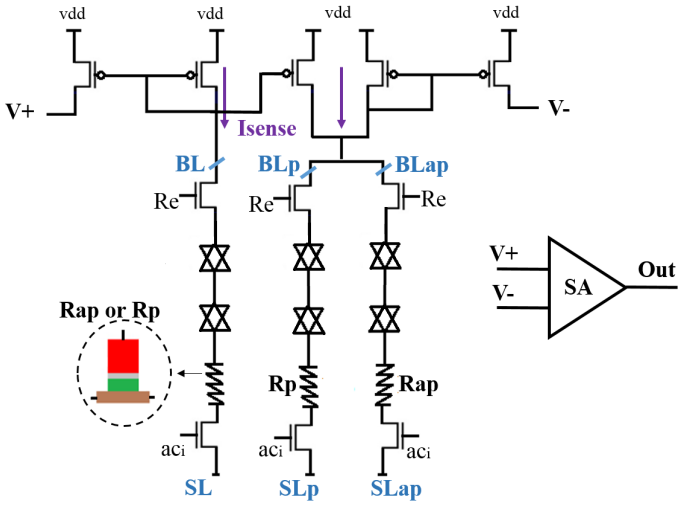


Figure 4.5: Schéma de lecture d'une cellule SOT-MRAM

4.2.5 Circuit d'écriture

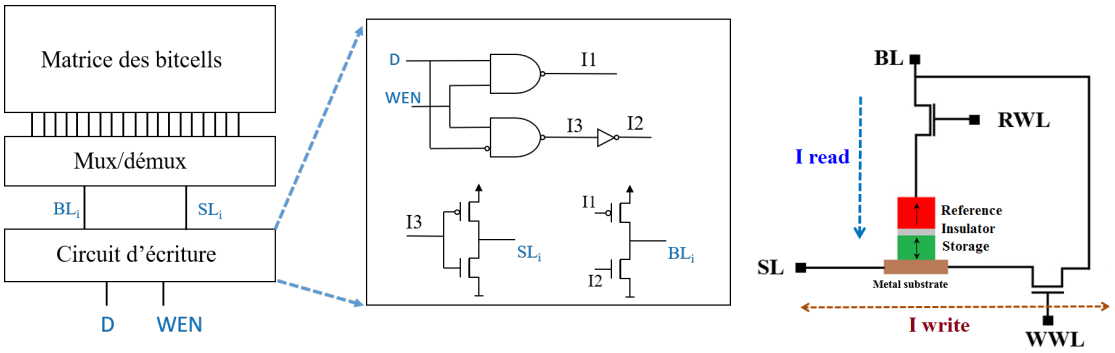
Lorsque l'on veut écrire et sauvegarder un mot dans une mémoire, l'adresse du mot est envoyée en entrée du décodeur de ligne et la valeur du mot (entrée des données) est envoyée sous forme d'une tension via un circuit d'écriture. Ce dernier charge les longues lignes d'interconnexion (BL et BLb), en fonction de la valeur des nouvelles données. Dans le cas d'une SRAM, 1 seul bit est stocké alors que 2 nœuds de stockage complémentaires l'un de l'autre sont nécessaires.

Le principe de programmation est dépendant de la technologie de mémoire. Dans le cas d'une cellule SOT-MRAM, un chemin de courant bidirectionnel est établi pour chaque bitcell. L'écriture se fait en préchargeant les bitlines BL et les source lines SL respectivement à Vdd et à Gnd, ou inversement selon l'information à programmer (voir figure 4.6a). Le circuit d'écriture est composé de deux portes ET et de deux inverseurs permettant d'imposer le sens du courant. Ce courant doit donc circuler à travers la piste métallique placée en dessous de la JTM de la cellule cible. On distingue 3 cas de fonctionnement possibles :

- Lorsque WEN(Write Enable) ='0', aucun courant d'écriture ne circule.
- Lorsque WEN ='1', D(Data)= '1' et WWL = '1', les sorties I1 et I2 seront chargées à Gnd permettant de précharger BL à Vdd et SL à gnd.

- Lorsque WEN ='1', D= '0' et WWL = '1', les sorties I1 et I2 seront chargées à Vdd permettant de précharger BL à gnd et SL à vdd.

La programmation de l'ensemble de la mémoire se fait en configurant les multiplexeur/démultiplexeur pour véhiculer les électrons dans les lignes BL et SL souhaitées.



(a) Schéma d'écriture d'une bitcell (b) cellule SOT-MRAM cible

Figure 4.6: Schéma d'écriture d'une bitcell en technologie SOT-MRAM

4.2.6 Les entrées/sorties

Les registres entrées/sorties permettent l'interface entre la mémoire et le circuit dans lequel elle sera intégrée, le processeur dans notre étude. Les entrées et les sorties de la mémoires sont synchronisées à l'aide de bascules D. Certaines bascules ne réagissent que sur un seul type de front de l'horloge. Cette synchronisation est importante pour compenser les délais de propagation des signaux dans les différents blocs de la mémoire.

Il était donc indispensable de connaître et comprendre tous ces blocs présentés précédemment formant l'architecture d'une mémoire embarquée afin de maîtriser et optimiser la conception et l'utilisation de mémoire SOT-MRAM. C'est pourquoi, nous avons mis en place et developpé l'ensemble de ces blocs pour concevoir une mémoire embarquée en technologie magnétique selon le flot full custom. Un de nos objectifs était d'intégrer cette mémoire dans un système numérique afin d'étudier l'intérêt des technologies magnétiques émergentes par rapport à la mémoire volatile de type SRAM.

Dans ce contexte, nous avons choisi d'étudier plusieurs architectures de mémoire à base de SOT-MRAM, qui est une des mémoires émergentes les plus premetteuses

pour l'intégration dans un microprocesseur. Le but est de surmonter notamment les limitations de la technologie STT en termes de vitesse de lecture, comme présenté dans le chapitre 1, et de proposer une alternative à l'utilisation de SRAM.

4.3 Etude de mémoires magnétiques selon plusieurs architectures en technologie SOT-MRAM

Comme décrit précédemment dans le chapitre 1, la technologie SOT propose un nouveau mécanisme de retournement de l'aimantation de la cellule mémoire, permettant de s'affranchir des problèmes de vitesse de lecture et d'endurance des mémoires STT. Cette mémoire écrite par couple de Spin Orbit comprend un métal lourd (tantale ou platine typiquement) adjacent à une jonction tunnel magnétique. Les caractéristiques et les paramètres essentiels de la cellule mémoire SOT utilisée dans notre étude sont donnés sur le tableau 4.1:

Table 4.1: *Caractéristiques et propriétés essentielles d'une cellule SOT-MRAM*

JTM (diamètre)	40nm
TMR	150%
Piste métallique (L × l × e)	(100 × 50 × 4) nm
Résistance parallèle R_p	5KΩ
Technologie CMOS	28 nm FDSOI
Alimentation	1 V

Notons d'ailleurs que le courant critique pour écrire une donnée et le temps de commutation de la JTM dépendent essentiellement de la taille ainsi que des paramètres physiques de la bitcell. Le courant de programmation dépend fortement des caractéristiques de la piste métallique placée en dessous de la JTM. Sa résistance peut s'exprimer selon l'équation suivante :

$$R = \frac{R_{mt} \times l_s}{a_r} \tag{4.2}$$

$$a_r = w_s \times t_s \tag{4.3}$$

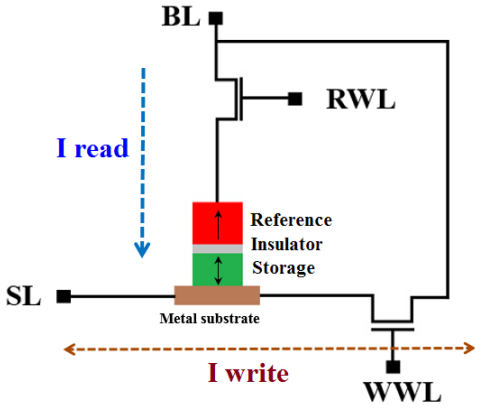
R_{mt}: résistivité éléctrique du matériau, platine dans notre cas (20 μΩ.cm)
l_s: longueur de la piste (100 nm)
a_r: surface latérale de la piste
w_s: largeur de la piste (50 nm)

t_s: épaisseur de la piste (4nm)

Nous étudierons par la suite l'architecture d'une mémoire de 32 kbits basée sur le principe d'une cellule mémoire 2T-1JTM. Nous proposons également deux autres configurations de mémoire en technologie SOT dont une a fait l'objet d'un dépôt de brevet d'invention.

4.3.1 Structure 2T-1JTM

La figure 4.7a montre la structure de base d'une bitcell classique 2T-1JTM.



(a) Schéma d'une cellule SOT-MRAM contenant 2 transistors et une JTM

Operations	*Write '1'/'0'*	*Read*
WWL	V_{DD}	0
RWL	0	V_{DD}
BL	V_{Write} / 0	I_{sense}
SL	0 / V_{Write}	0

(b) Signaux de contrôle des opérations de lecture et d'écriture

Figure 4.7: Schéma d'une bitcell SOT, 2T-1MTJ

Le chemin d'écriture étant différent de celui de la lecture, un transistor de plus est ajouté par rapport à une bitcell STT. Le courant électrique nécessaire pour écrire un '0' ou un '1' logique est alors contrôlé par un transistor NMOS d'écriture commandé par sa grille WWL (Write Word Line) comme illustré sur la figure 4.7b. Une résistance faible ou forte est donc obtenue, selon le sens du courant passant à travers la piste métallique placée en dessous de la jonction.

Quant à la lecture, un courant électrique passe à travers la jonction, contrôlé par le transistor NMOS de lecture, commandé par sa grille RWL (pour Read Word Line). En effet, dans cette structure un seul des 2 transistors est passant à la fois, ce qui permet de séparer les deux opérations de lecture et d'écriture. Cela permet d'augmenter la vitesse de lecture par rapport à une JTM de type STT car il n'y a pas le risque d'écriture non souhaitée pendant la phase de lecture. La limitation sera alors dans la tension imposée à la barrière tunnel, qui doit être en dessous de la tension de claquage, typiquement

autour de 1V.

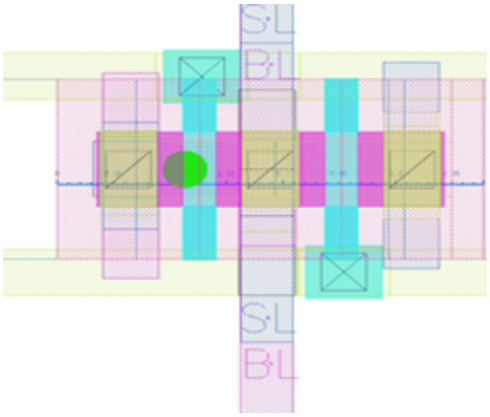


Figure 4.8: Vue du layout d'une bitcell en technologie hybride SOT (2T-1JTM) de taille 0.378 $\times$0.192 = 0.072 μm^2

La figure 4.8 montre le layout de la structure 2T-1MTJ qui révèle une surface de 0.072 μm^2. Cette surface peut effectivement varier selon les caractéristiques de la cellule mémoire et des performances souhaitées. Dans notre étude, les transistors d'écriture et de lecture sont dessinés avec la taille minimale autorisée par la technologie.

La figure 4.9 a pour but de valider le fonctionnement de cette structure. On peut remarquer qu'après avoir injecté un courant de 90 μA dans la piste d'écriture de la cellule SOT (BL=1; SL=0), le basculement de l'aimantation de P vers AP (Parallèle vers Anti-Parallèle) se fait en < 1ns. On remarque également que pour écrire l'état opposé, AP vers P (BL=0; SL=1), le temps nécessaire est similaire. Ceci est une caractéristiques des JTM SOT que n'ont pas les JTM STT.

Afin de pouvoir étudier les performances de cette bitcell unitaire au niveau d'une mémoire embarquée complète, il était nécessaire d'adapter et de mettre à jour tous les blocs de la mémoire décrits précédemment dans ce chapitre. En effet, chaque bloc a été simulé individuellement afin de vérifier le fonctionnement et les performances souhaités. La figure 4.10 illustre une vue simplifiée d'une matrice (4 $\times$ 4) pour l'architecture 2T-1JTM.

4.3.1.1 Evaluation et performances

La figure 4.11 montre un résultat de simulation de la mémoire complète de 32kb où l'on remarque le comportement de la JTM pendant les deux phases, programmation et lecture.

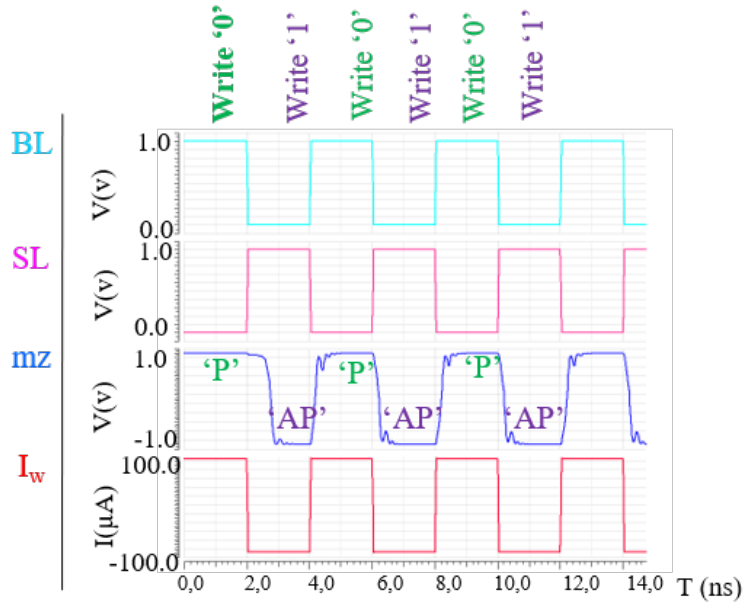


Figure 4.9: Opération d'écriture d'une bitcell SOT-MRAM

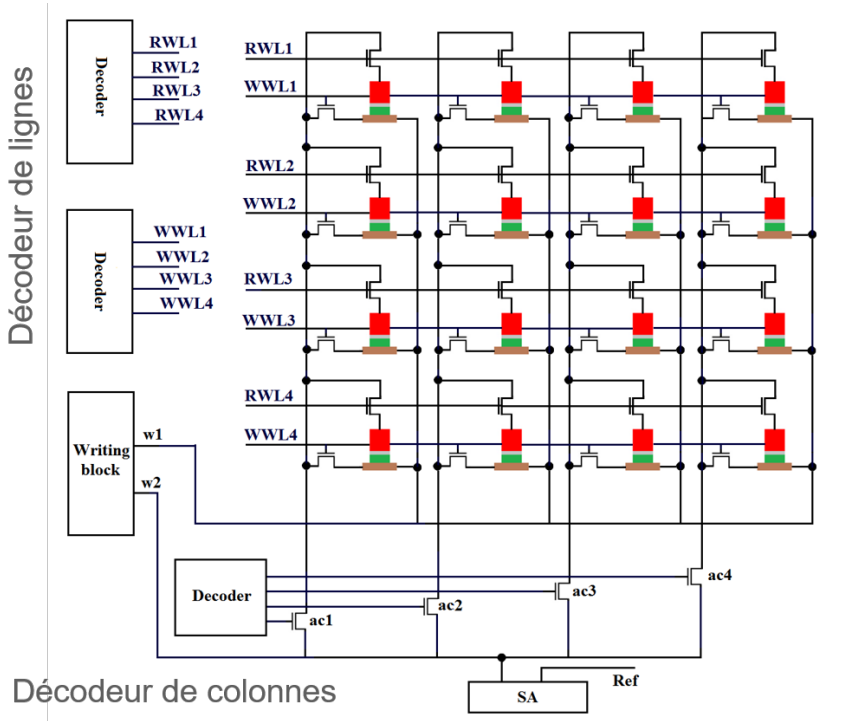


Figure 4.10: Matrice (4 × 4) pour l'architecture 2T-1JTM

Pendant la phase de programmation, la JTM cible est écrite à '0' ou à '1' selon l'information à coder (signal D pour Data). Lorsque la commande de grille de tran-

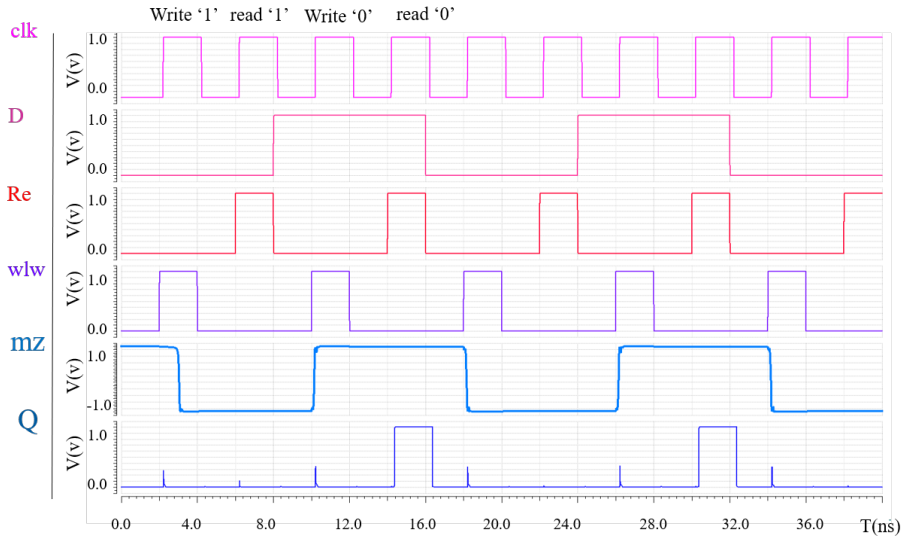


Figure 4.11: Résultats de simulation de la mémoire complète de 32kb

sistor d'écriture (wlw) est activée, l'aimantation bascule dans un sens ou dans l'autre en fonction des lignes BL et SL et lors d'un front montant de clk. Lors de la phase de restauration le signal Re (pour Read) est activé sur un front montant de clk permettant de lire la valeur attendue en sortie (signal Q).

A l'issu de cette simulation, nous avons décidé de générer une horloge interne qui réduit la consommation dynamique durant ces deux phases. Le principe est illustré sur la figure 4.12. En effet, clk_i sera généré pendant le temps nécessaire de commutation de la JTM pour la phase de programmation et pendant le temps de réponse de l'amplificateur de lecture lors de la phase de restauration. De cette façon, le courant est inhibé quand clk_i est désactivé. Le temps d'activation de l'horloge est programmé dans le bloc de contrôle et déterminé selon les performances souhaitées de la mémoire. De cette façon la consommation dynamique est réduite d'un facteur 2 pour une $T_{clk} = 2 \times T_{clki}$.
Notons d'ailleurs que la fréquence d'horloge maximale atteignable par la mémoire est égale à f_{clki}. Comme mentionné précédemment, ce temps dépend des paramètres physiques de la bitcell, qui dépendent essentiellement de l'application.

Les caractéristiques et les paramètres essentiels de la mémoire que nous avons conçue sont illustrés dans le tableau 4.2

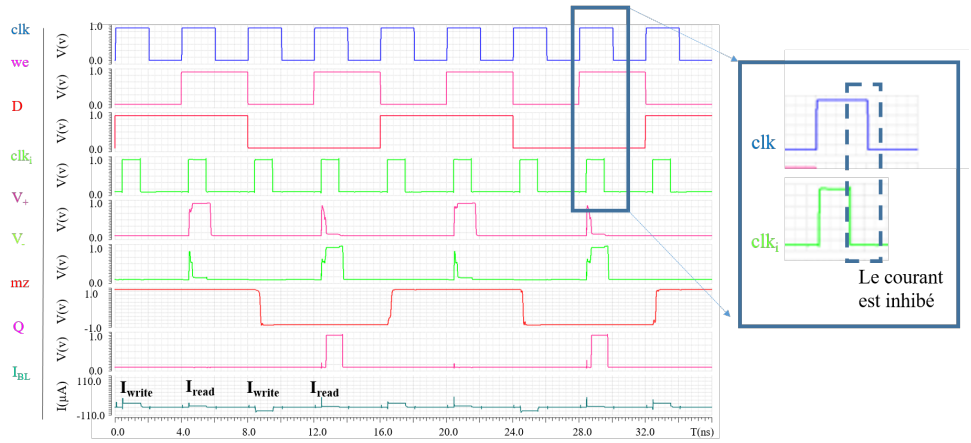


Figure 4.12: Optimisation de la consommation dynamique

Table 4.2: *Caractéristiques et propriétés essentielles de la mémoire 2T-1MTJ*

Courant critique de programmation	90 µA < 1ns
Capacité	32 kbits
Architecture bitcell	2T-1JTM
Horloge	4ns
Entrée/ sortie	8 bits
Technologie CMOS	28 nm FDSOI
Densité	80 F^2

4.3.2 Structure 1T-1D-1JTM

La cellule de base à 2T-1JTM a pour principal avantage d'être très rapide tout en étant moins dense que la technologie STT. C'est pourquoi, nous étudions dans cette section le comportement d'une cellule mémoire ayant un seul transistor et une diode. L'utilisation des diodes au lieu des transistors d'accès a été étudiée dans [76] pour une mémoire "crossbar" en technologie STT. Cette étude montre que la densité de la bitcell est largement augmentée (une densité sub-1F^2 peut être réalisée). L'objectif que nous nous sommes fixés est d'étudier l'intérêt du point de vue de la consommation et de la densité de l'intégration de la diode dans une bitcell SOT-MRAM. Une matrice 4 × 4 est illustrée sur la figure 4.13.

4.3.2.1 Fonctionnalité

Dans cette configuration, le transistor d'accès de lecture est remplacé par une diode comme le montre la figure figure 4.13. Le sens du courant est conforme au fonctionnement souhaité, unidirectionnel en lecture et bidirectionnel en écriture. Pour écrire la jonction, il faut tout d'abord activer le transistor d'écriture (wr) et désactiver tous les transistors de multiplexeur (t_i). Ensuite il faut sélectionner la colonne / ligne souhaitée en activant l'un des quatre transistors d'accès ac_i / WWL_i. Ceci permet donc de programmer la cellule souhaitée soit à '1' soit à '0' logique selon le sens du courant injecté dans la ligne. Cependant, pour lire le bit stocké dans la JTM, il faut d'abord adresser la colonne / ligne souhaitée via les transistors ac_i / t_i et désactiver les transistors d'écriture (wr et WWL_i). Cela a pour effet de générer un courant, imposé par les lignes BL et SL, permettant de lire la valeur attendue en sortie.
La circulation unidirectionnelle du courant via la diode a l'immense avantage d'inhiber les courants de fuites dans les cellules adjacentes.

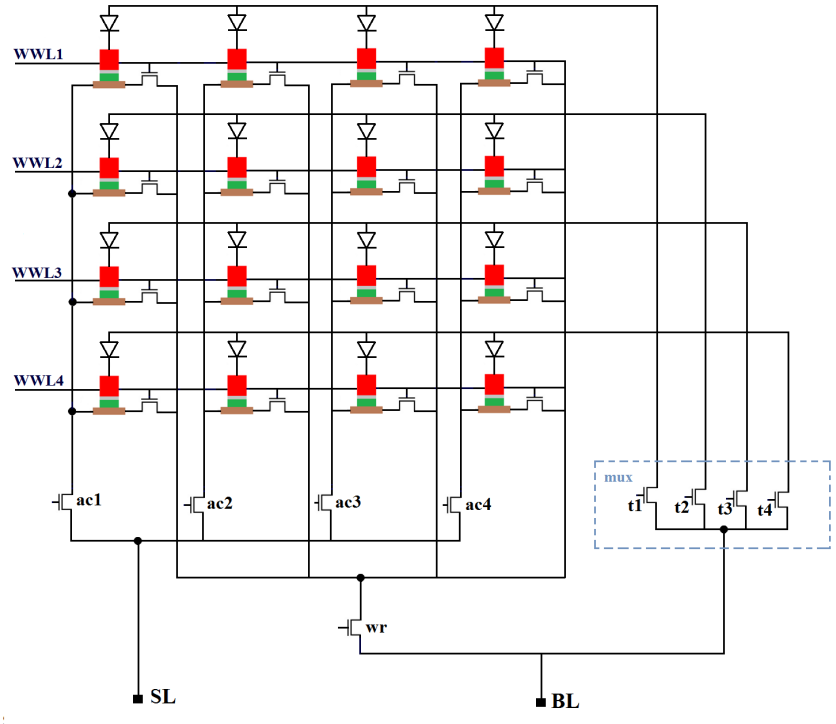


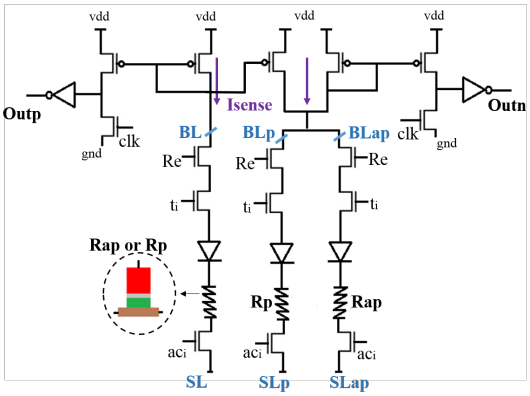
Figure 4.13: Matrice de points mémoires en technologie SOT, 1T-1D-1JTM

4.3.2.2 Evaluation

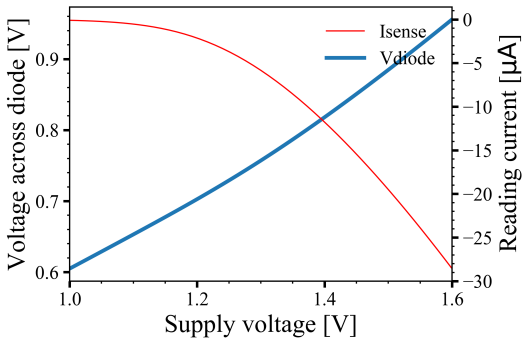
Le principal intérêt pratique d'une diode est qu'elle laisse passer le courant dans un sens uniquement. Celui-ci est conforme au comportement que nous souhaitons avoir pour la phase de lecture.

En effet, l'intégration de la diode sur le chemin de lecture diffère d'un simple transistor, dans la mesure où elle nécessite une certaine tension de seuil, relativement élevée. Reprenons alors les caractéristiques d'une diode à semi-conducteur. En polarisation inverse le courant inverse est très faible. Au contraire, en polarisation directe et au-delà de la tension de seuil (V_T), la diode est conductrice.

Etudions donc le régime de la diode lors de la phase de lecture, comme illustré sur la figure 4.14b. On constate que pour une tension de détection de diode inférieur à V_T (0.7 V pour les diodes au silicium), le courant de lecture est trop faible. Cependant, au-delà de V_T, le courant traversant la diode est de plus en plus important et devient suffisant. Ceci nécessite alors une augmentation de la tension d'alimentation au-delà de la tension nominale de 1V.



(a) Circuit de lecture d'une cellule 1T-1R-1TMR



(b) Caractéristique courant / tension en mode lecture appliquée

Figure 4.14: Principe de fonction d'une cellule mémoire MRAM.

En ce qui concerne le chemin de lecture, la tendance serait alors de diminuer la résistance de ce chemin afin d'augmenter le courant circulant dans la diode. Cependant, le modèle de simulation de la SOT en mode lecture montre que plus la résistance de la JTM augmente, plus la variation d'amplitude est importante (ΔV : différence entre la tension de la cellule mémoire lue et celle de référence : V_+ - V_-). On constate également que cette variation est encore plus importante pour des valeurs de TMR élevées. ΔV de l'ordre de 50mV est atteignable pour une TMR de 250% et pour des valeurs de résistances de JTM supérieures à 8KΩ. Rappelons tout de même que le besoin d'avoir une ΔV élevée est lié au degré de robustesse de l'amplificateur de lecture vis-à-vis des variations des procédés des technologies CMOS et magnétique.

Cependant l'augmentation de la résistance parallèle de la JTM augmente la puissance nécessaire pour lire l'information (P= R $\times$ I^2). En effet, la consommation à la lecture est deux fois plus élevée comparée avec la bitcell classique 2T-1JTM. Ceci est

due à l'augmentation de la tension d'alimentation afin de rendre la diode dans un état passant. Ceci implique, qu'il y a un compromis à trouver entre la consommation et la robustesse en mode restauration pour une valeur de résistance choisie. Afin de trouver le bon compromis, nous avons étudié l'évolution de ΔV en fonction de R_p (figure 4.15). Pour une valeur R_p égale à 5kΩ par exemple, la puissance de lecture est deux fois plus élevée que la puissance d'une bitcell 2T-1JTM. Le tableau 4.3 résume les caractéristiques essentielles d'une bitcell 1T-1D-1JTM par rapport à celle 2T-1JTM.
Notons que cet inconvénient d'avoir à augmenter la tension d'alimentation sur le procédé 28nm n'en est pas un pour un procédé moins avancé qui utiliserait une tension d'alimentation de 1.8V par exemple.

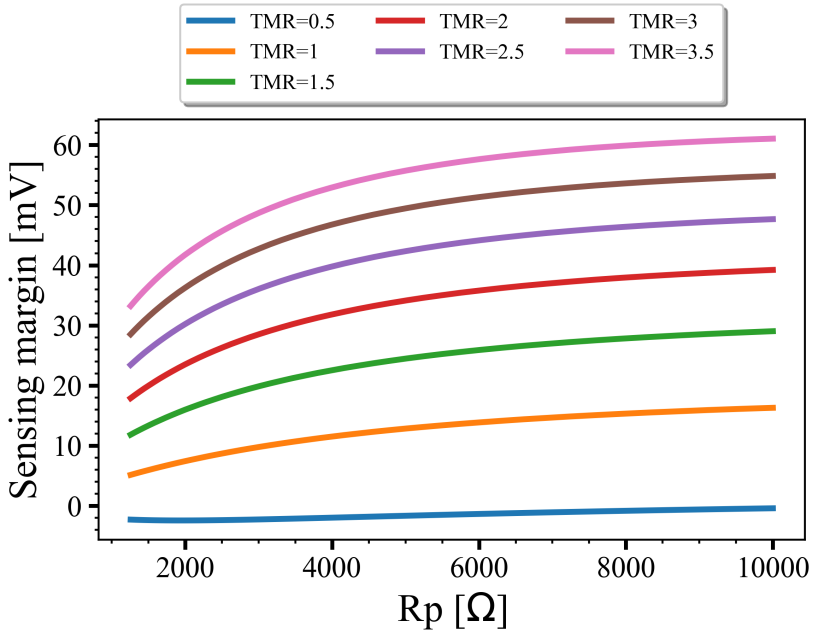


Figure 4.15: Evolution de ΔV en fonction de la résistance parallèle de la JTM pour différentes valeurs de TMR

Table 4.3: *Caractéristiques et propriétés essentielles d'une structure 1T-1D-1JTM*

	2T-1JTM	1T-1D-1JTM
Technologie	28nm	28nm
Densité	80 F^2	68F^2
Nombre de transistors pour une matrice (n×m)	2(n×m)	m (n + 1)
Consommation en mode écriture	99 µW	99 µW
Consommation en mode lecture	22 µW	40 µW

En ce qui concerne la phase d'écriture, on remarque que la consommation de la bitcell est la même dans les deux approches, vu que l'intégration de la diode n'intervient pas sur le chemin d'écriture. Quant à la densité, on constate que l'approche 1T-1R-1JTM offre une meilleure densité qui permet de diminuer la surface d'une bitcell d'environ 15% par rapport à la structure classique 2T-1JTM (la figure 4.16 montre le layout de cette cellule). Le nombre de transistors est également beaucoup plus faible dans cette approche. Pour une matrice de 32kbits, le nombre de transistors est réduit d'environ 45%. Cette réduction implique aussi une réduction du nombre de portes logiques dans la partie décodage du fait que le transistor d'accès dans la bitcell est supprimé.

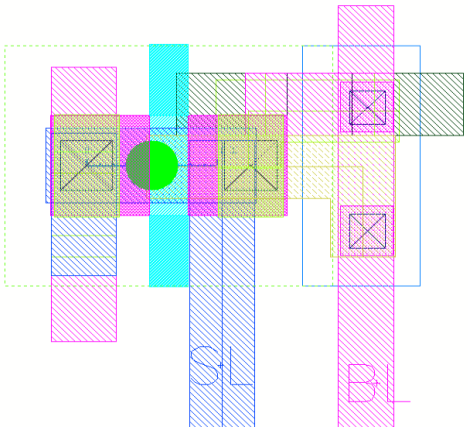


Figure 4.16: Vue du layout d'une bitcell en technologie hybride SOT (1T-1D-1JTM) de taille 0.319 $\times$ 0.192 = 0.061 μm^2

4.3.3 Structure hybride STT/SOT : 2T-2MTJ

La demande croissante de capacité de stockage de données dans les applications modernes conduit à un important effort de développement des nouvelles technologies de mémoires non volatiles, comme déjà présentée dans le chapitre 1. Dans ce contexte, notre étude a pour but d'évaluer l'éventuel intérêt d'une architecture hybride STT-SOT. Il s'agit d'une bitcell hybride qui comporte au moins deux mémoires dont une mémoire à deux terminaux. La cellule que nous proposons possède deux mémoires magnétiques, l'une à trois terminaux et l'autre à deux terminaux en technologie SOT et STT respectivement (voir figure 4.17).

Afin d'augmenter la densité de la bitcell, un transistor est partagé (t2) entre les mémoires STT et SOT qui sert à la lecture/ écriture de la STT et aussi à la lecture de la SOT. Du point de vue schématique, aucun transistor de plus n'est ajouté en comparaison avec une bitcell SOT standard 2T-1JTM. Pour séparer le fonctionnement des 2 mémoires, les 2 lignes BL0 et SL0 sont pilotées par la partie contrôle de la mémoire

par 2 interrupteurs CMOS uniquement (S1 et S2). Ces interrupteurs seront partagés par toutes les bitcells de la même colonne d'une matrice.

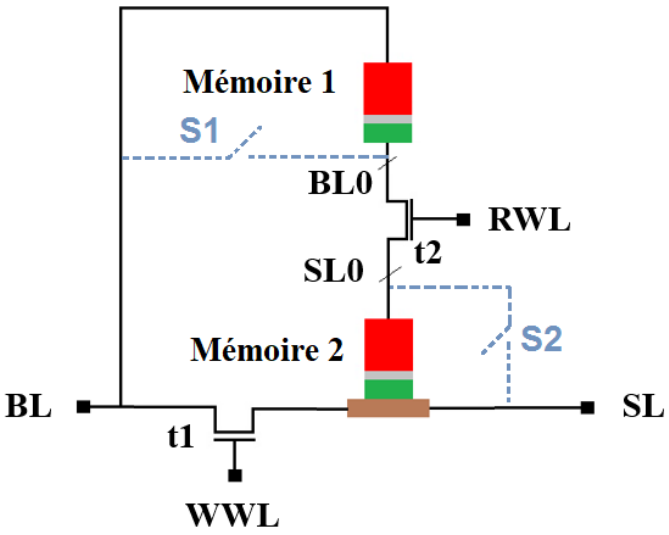


Figure 4.17: Cellule hybride proposée en technologie STT/SOT

Du point de vue de la fabrication, l'intégration de ces 2 procédés magnétiques reste compatible avec un procédé CMOS et l'integration est objectivement envisageable. Comme illustré sur la figure 4.18, l'intégration des JTM se fait entre des niveaux de métal différents. On retrouve la STT entre le métal 3 et le métal 4 par exemple alors que la SOT se trouve entre des niveaux de métal supérieurs, métal 5 et le métal 6. La figure 4.18 donne un exemple de la vue de coupe de l'ensemble de ce procédé hybride CMOS/STT/SOT.

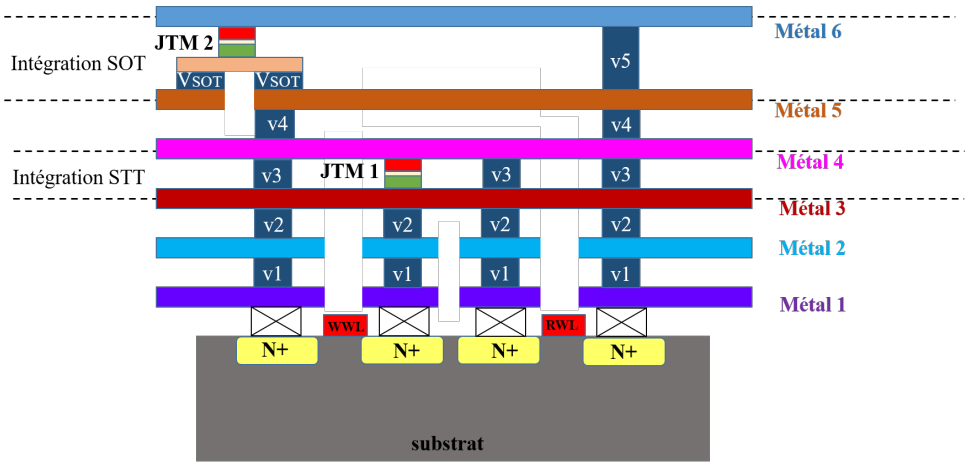


Figure 4.18: Vue transverse de la structure hybride CMOS / STT /SOT

4.3.4 Fonctionnement

Le mode de fonctionnement de cette architecture présentée sur la figure 4.17 est le suivant:

Pour la mémoire JTM 2 en technologie SOT:

- Pour écrire la jonction, RWLl vaut '0' alors que WWL vaut '1' permettant d'activer le transistor t1. Le courant circule donc dans un sens ou dans l'autre dans la piste métallique placée en dessous de la jonction selon la tension appliquée sur les lignes BL et SL.
- Pour lire la jonction, WWL vaut '0' alors que RWL vaut '1'permettant d'activer le transistor t2 uniquement. L'interrupteur S1 doit être fermé permettant de relier BL0 à BL afin d'inhiber le passage de courant dans la mémoire STT (qui sera court-circuitée dans ce cas). Le courant passe donc à travers la jonction SOT qui sera détecté par l'amplificateur de courant décrit précédemment.

Concernant la JTM 1 en technologie STT:

- Pour l'écriture et la lecture, RWL='1' et WWL='0', l'interrupteur (S2) est fermé permettant de relier SL0 à SL. La mémoire SOT est court-circuitée dans ce cas. Le courant passe à travers la jonction STT soit dans un sens soit dans l'autre pour écrire la jonction.
- Une tension moins importante que celle de l'écriture est appliquée pendant la phase de lecture, permettant tout de même à un courant de passer à travers la jonction.

Ces deux modes de fonctionnement sont illustrés sur le chronogramme de la figure 4.19

Notons d'ailleurs que "l'état indéterminé" signifie que les nœuds internes BL0 ou SL0 ne sont pas connectés à la sortie des interrupteurs et ont un niveau de tension analogique. Pendant les phases de lecture de la JTM 2, la tension de ligne BL prend aussi un niveau analogique "n" n'étant pas proche de celui de la programmation.

Dans les applications de type microprocesseur ou microcontrôleur, les bitcells sont organisées de façon matricielle, comme décrit précédemment. Cela veut dire que les interrupteurs S1 et S2 seront partagés pour toute la matrice, et non pas au niveau de la bitcell. Il n'y a donc pas de pénalité sensible sur la surface. Ceci est bien illustré sur la figure 4.20. En effet, on remarque l'intégration de diodes de protection qui permettent d'isoler le passage de courant entre les bitcells adjacentes.

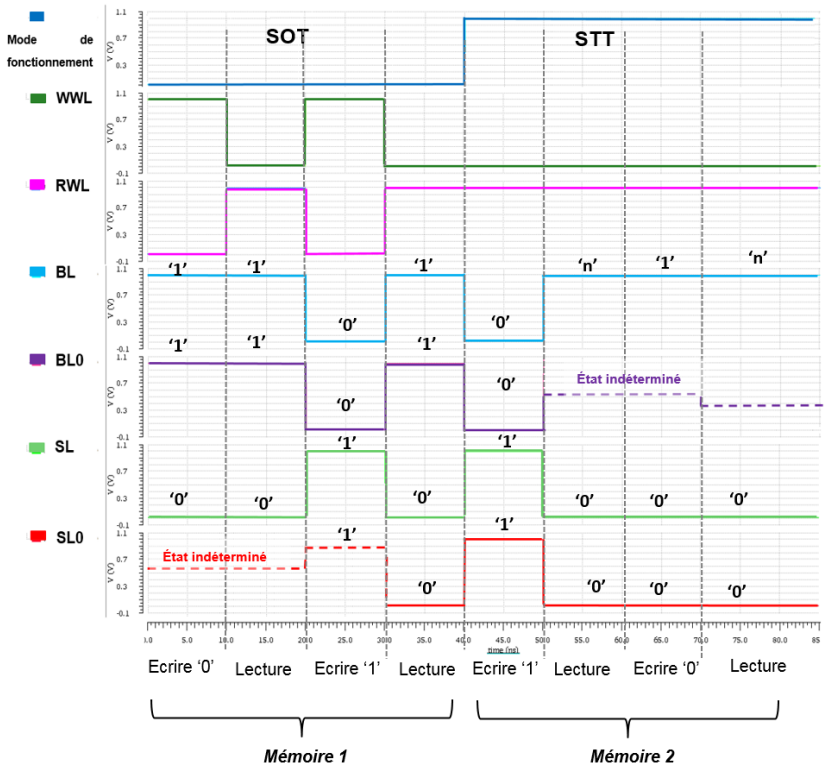


Figure 4.19: Mécanisme d'écriture et de lecture de la cellule hybride proposée

4.3.5 Evaluation

L'architecture hybride proposée possède le même nombre de transistors en comparaison avec une cellule standard d'une SOT 2T/1JTM, mais avec une capacité de stockage deux fois plus grande. L'intérêt technique de la présente architecture est la mutualisation des commandes, ce qui permet de maintenir la surface de la bitcell tout en doublant la capacité de stockage. Ceci est tout à fait différent des cellules hybrides proposées dans la littérature où la surface de la bitcell doublerait par rapport à la surface de chaque cellule individuelle.
La taille de la structure proposée a révélé un gain de 34% environ (0,0927 / 0,140μm^2) de surface par bitcell (2 bits) par rapport à la taille de deux bitcells adjacentes (2 $\times$ 1bit) en technologie SOT, pour obtenir la même capacité de stockage (figure 4.21). Dans notre étude, nous avons choisi des paramètres très réalistes par rapport à la littérature. En ce qui concerne les JTM, dans les deux technologies le diamètre est fixé à 40 nm. Dans notre cas nous avons combiné les deux aspects, la haute densité de la STT et la haute vitesse de la SOT afin d'obtenir une cellule innovante à double performance, et non pas en se contentant de doubler la capacité de stockage.

Cette architecture cumule donc beaucoup d'avantages, en termes de densité et de vitesse assurées par les deux technologies magnétiques. Cependant l'intégration des

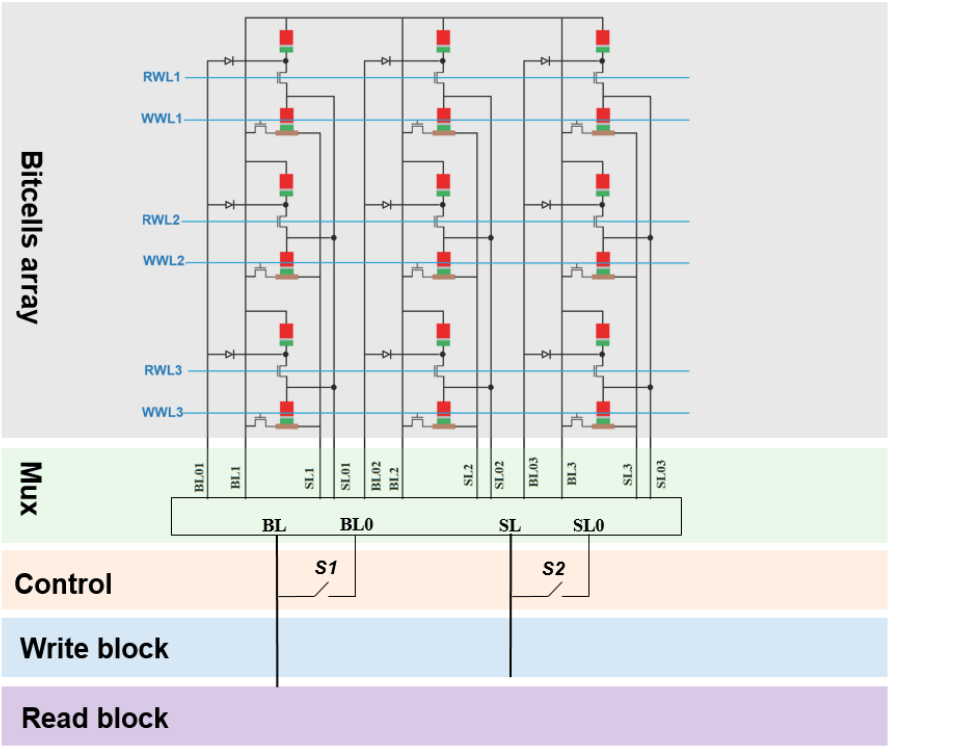


Figure 4.20: Architecture d'une mémoire embarquée en technologie hybride STT/SOT

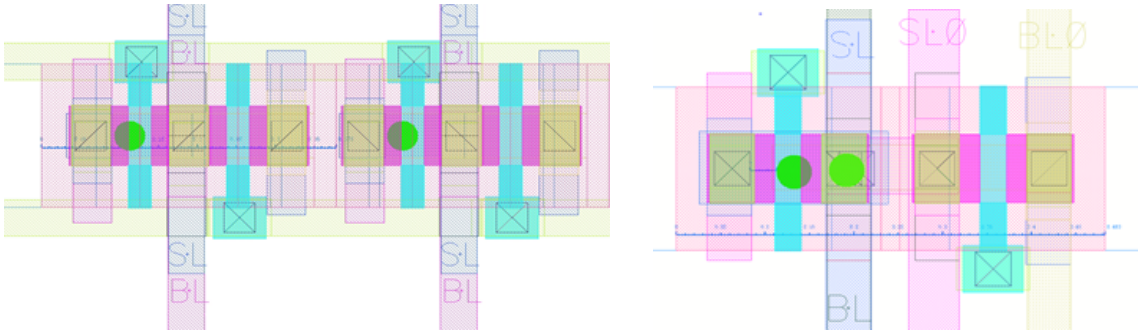
diodes de protection entre les bitcells adjacentes affecte l'opération de lecture de la SOT. Une tension de lecture supérieure à la tension seuil des diodes doit être imposée (comme décrit dans la section précédente). Ceci augmente par conséquent la consommation en mode lecture de la SOT. En revanche, la consommation en mode écriture est inchangée par rapport à une SOT standard car le courant de programmation ne passe pas par la diode. Il en est de même pour les opérations de lecture et d'écriture de la STT où leur consommation est indépendante de l'ajout des diodes.

La conception de cette cellule innovante a aboutit à un brevet d'invention déposé au niveau national en juillet 2018 et étendu à l'international en juillet 2019.

4.3.6 Comparaison entre les différentes architectures

Le tableau 4.4 résume les différents aspects analysés pour toutes les architectures décrites précédement ainsi qu'une cellule en technologie STT.

On constate que chaque architecture offre de très intéressantes performances adaptées aux différents types d'applications. La structure classique 2T-1MTJ a pour principal avantage d'être dédiée à des applications basses consommation et hautes vitesses (< 1ns pour programmer une bitcell). Cependant, elle nécessite 2 transistors par bitcell,



(a) 2 bitcells adjacentes (2 × 1 bit) en technologie SOT(2T-1JTM) de taille 0.728 ×0.192 = 0.140 μm^2

(b) 1 bitcell (2 bits) en technologie hybride SOT/STT (2T-2JTM) de taille 0.482 × 0.142 = 0.0927 μm^2

Figure 4.21: Dessin des masques de deux bitcells en technologie standard SOT et d'une bitcell en technologie hybride SOT/STT, les deux de même capacité de stockage

Table 4.4: *Caractéristiques et propriétés essentielles des différentes architectures*

	SOT	SOT	STT	SOT/STT
	2T-1JTM	1T-1D-1JTM	1T-1JTM	2T-2JTM
Technologie	28nm	28nm	28nm	28nm
Densité	80 F^2	68F^2	58 F^2	102 F^2
Consommation en mode écriture	99 µW	99 µW	131 µW	99 / 131 µW
Consommation en mode lecture	22µW	40 µW	45 µW	40/45 µW
Nb de transitor / matrice (n×m)	2(n×m)	m(n + 1)	n×m	2(n×m)
Temps de programmation	< 1 ns	< 1 ns	< 3 ns	< 1 ns / < 3 ns
Capacité de stockage/bitcell	1bit	1bit	1bit	2bits

ce qui engendre une augmentation de la surface de 28 % par rapport à la technologie STT. En effet, l'utilisation d'une diode à la place du transistor de lecture paraît moins contraignant en termes de surface par rapport à la structure 2T-1JTM. Comme nous venons de le présenter, pour une matrice de 32kbits, le nombre de transistors est réduit de 45%. Par ailleurs, bien que la réduction de surface soit l'avantage le plus significatif, une augmentation de la consommation de lecture s'ajoute également. Ceci est due à l'augmentation de la tension de lecture afin de rendre la diode dans un état passant.
Le tableau 4.4 montre clairement que parmi les différentes architectures, la technologie STT est la plus dense grâce à l'utilisation d'un seul transistor d'accès pour les deux opérations de lecture et d'écriture. Cette mémoire sera particulièrement utilisée dans les applications nécessitant une haute capacité. Cependant, sa vitesse de lecture reste un inconvénient majeur pour certaines applications. De plus la structure hybride SOT/STT présente un intérêt grandissant en termes de densité, capacité de stockage

et mutualisation des deux aspects, haute densité de la STT et la haute vitesse de la SOT.

Ce gain en surface et en coût peut être considérable dans des applications de type cache. En effet, en plus d'avoir une double capacité de stockage dans une surface réduite, cette structure permet d'obtenir une double performance.

Dans notre étude nous nous sommes intéressés à intégrer les technologies magnétiques non volatiles dans un processeur dans le but d'étudier le gain, principalement en consommation, par rapport aux mémoires SRAM. Le choix des architectures implémentées visant une basse consommation sont la structure 2T-1JTM pour la technologie SOT et 1T-1JTM pour la technologie STT.

4.4 Etude d'un processeur: Secretblaze

Le processeur Secretblaze est un système robuste développé au sein du laboratoire de recherche LIRMM de Montpellier. Secretblaze est un clone open source du Microblaze (un softcore proposé par Xilinx pour ses FPGA) dont le principal objectif est de mettre à disposition des chercheurs une architecture ouverte permettant l'étude des aspects sécurité de microprocesseurs.
Comparé aux autres processeurs de type "open source" tels que MB-lite et OpenRISC 1200 comme illustré sur la figure 4.22, le Secretblaze apparaît comme un bon compromis surface / performance [14]. Il implémente un ensemble de contremesures dans le but de garantir une sécurité matérielle avec le moindre coût. Il a été utilisé dans divers projets auxquels le laboratoire SPINTEC a participé. En effet, une collaboration entre le LIRMM et SPINTEC a été mise en place afin d'étudier les performances de la technologie STT au niveau du Secretblaze. Ceci a d'ailleurs été la forte motivation derrière le choix de ce processeur, qui dispose d'une architecture moins complexe que d'autres processeurs et dont l'équipe a déjà une connaissance approfondie.

4.4.1 Description du Secretblaze

Secretblaze est la version robuste du processeur OpenScale, de type Havard [14]. C'est un processeur softcore 32 bits à jeu d'instructions réduit RISC (Reduced Instruction Set Computer) sécurisé pour les systèmes embarqués. L'objectif principal pour lequel ce processeur est conçu consiste à fournir une implémentation de hautes performances supérieures aux solutions développées par d'autres processeurs [77]. De plus, une configuration permettant aux concepteurs la liberté de personnaliser leurs structures pour une application donnée a été développée. Cette flexibilité rend le cœur du Secretblaze

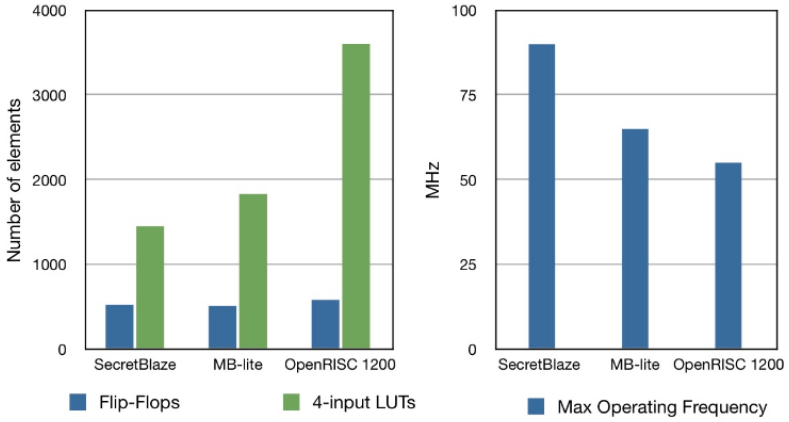


Figure 4.22: Comparaison du Secretblaze, MB-lite et Open Risc 1200 [14]

adapté pour répondre aux exigences d'une variété d'applications.

Ce processeur possède cinq étages de pipeline (Fetch, Decode, Execute, Memory Access, Write-Back), comme illustré sur la figure 4.23.

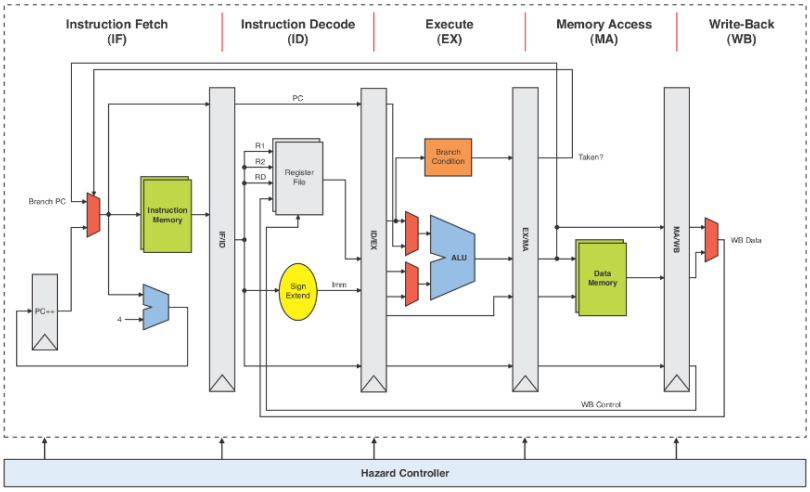


Figure 4.23: Architecture du coeur du Secretblaze [14]

- Instruction Fetch (*Chercher*): gère la récupération des nouvelles instructions à partir de la mémoire d'instructions
- Instruction Decode (*Décoder*): décode les instructions et adresse les registres
- Execute (*Exécuter*): supervise l'exécution de l'instruction par les unités de calcul tels que les unités arithmétiques et logiques afin de fournir les meilleures performances

- Memory Access (*Accès mémoire*): gère la communication entre les registres où l'instruction est exécutée et la mémoire de données. Elle assure un transfert depuis un registre vers la mémoire dans le cas d'une instruction type 'store' (accès en mode écriture) et de la mémoire vers un registre dans le cas d'un 'load' (accès en mode lecture)

- Write Back (*Ré-écriture*): Dans cette dernière étape du pipeline, les résultats sont écrits dans les registres.

La communication du Secretblaze est basée sur le bus Wishbone [78] qui assure la communication entre le processeur, la mémoire interne ainsi que les autres éléments périphériques (contrôleur d'interruption, contrôleur de port série UART: Universal Asynchronous Receiver-Transmitter). Les mémoires de données et d'instructions sont accessibles via des bus séparés comme le montre la figure 4.24.

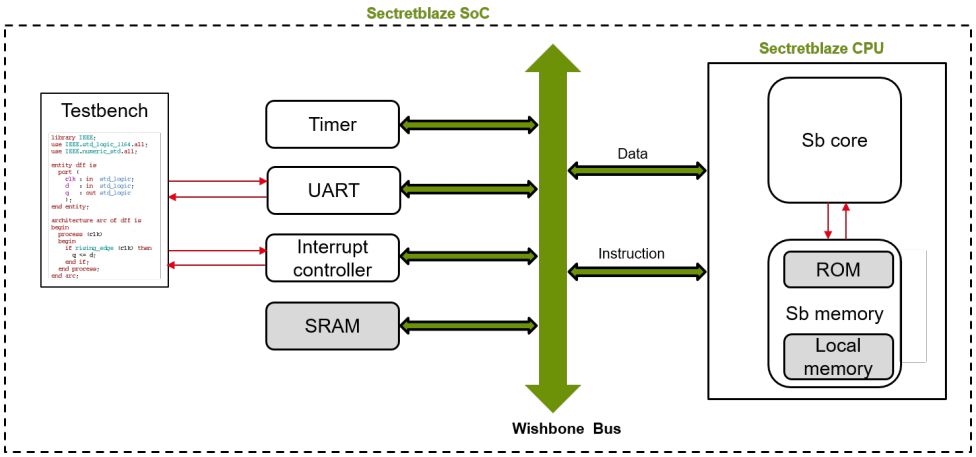


Figure 4.24: Représentation de l'architecture complète du SoC

On distingue trois types de mémoire dans l'architecture:

- La mémoire ROM, interne au CPU, qui est utilisée durant la séquence d'initialisation du SoC

- La mémoire locale du CPU, avec deux ports d'accès concurrents pour les instructions et les données

- Une mémoire partagée connectée au bus Wishbone et donc accessible à tous les périphériques, avec un seul port d'accès en lecture/écriture.

Du point de vue du processeur, la possibilité d'utiliser la mémoire locale ou la mémoire partagée permet d'étudier différents partitionnements de données pour optimiser les performances d'une application donnée.

Des travaux ont été menés dans le cadre de plusieurs projets afin d'améliorer les performances du Secretblaze et de prouver les mécanismes de protection contre les attaques [14], [79]. Parmi ces travaux, une collaboration entre Spintec et le LIRMM a eu lieu dans le cadre du projet H2020 GREAT [80]. Dans ce contexte, j'ai eu l'opportunité d'avoir accès aux codes sources RTL du processeur en VHDL. Les travaux menés dans le cadre du projet GREAT consistaient à implémenter sur silicium une mémoire non volatile en technologie CMOS 180 nm afin d'apporter la non volatilité au SoC et ainsi mettre en évidence les gains en énergie rendus possibles par de tels systèmes. Cette implémentation a été réalisée sur deux aspects: intégration de bascules non volatiles en technologie STT dans le cœur du Secretblaze, et implémentation d'une mémoire en technologie STT afin de remplacer uniquement la mémoire partagée SRAM.

Ce processeur a été basé sur un benchmark pour l'application cryptographique DES (Data Encryption Standard) [81] dans le but de garantir la confidentialité, l'authenticité et la sécurité des données. En effet, la cryptographie s'appuie sur des outils mathématiques qui permettent de chiffrer des messages afin de les rendre inintelligibles à un observateur externe.

4.4.2 Fonctionnement du SoC

Ayant accès aux codes sources, j'ai tout d'abord commencé à m'immerger dans ce domaine relativement nouveau pour moi. J'ai dû comprendre l'ensemble de son architecture et la fonctionnalité de chacune des parties le constituant dans toute sa complexité. La maîtrise du fonctionnement était indispensable pour pouvoir apporter des modifications, principalement en termes de remplacement de mémoires à semi-conducteurs par des MRAM. De plus, ayant pour objectif d'analyser et comparer les performances des différentes versions, une compréhension globale était nécessaire.

Pour comprendre le fonctionnement du SoC, il était donc indispensable de réaliser quelques simulations au niveau RTL (Register Transfer Level). Ceci m'a permis de mieux de comprendre les différentes opérations du SoC ainsi que le scénario implémenté.

Le SoC démarre à partir d'un programme initial écrit en assembleur et stocké dans une ROM (Read Only Memory), dans lequel la série d'instructions est prédéfinie. Comme le montre la figure figure 4.25, l'architecture initiale de mes travaux était celle développée dans le projet GREAT, dans laquelle seule la mémoire partagée a été remplacée par une mémoire STT-MRAM, en technologie CMOS 180 nm.

Le code de démarrage attend un octet de configuration provenant de l'interface UART. Les variables disponibles dans cet octet sont les suivantes : **MEM SELECT** pour sélectionner le type de mémoire utilisée: la mémoire partagée (SRAM ou STT-MRAM) ou la mémoire locale, **DOWNLOAD SIZE** pour définir la taille du programme applicatif à stocker dans le type de mémoire utilisé et **ENABLE RESTORE** pour choisir entre le démarrage normal ou la restauration depuis les JTM en mode non volatil. Une fois que l'octet de configuration est envoyé, une des mémoires, soit la mémoire locale soit la mémoire partagée, prend le relais de l'exécution du code applicatif. La séquence du démarrage du SoC est illustrée sur la figure 4.25:

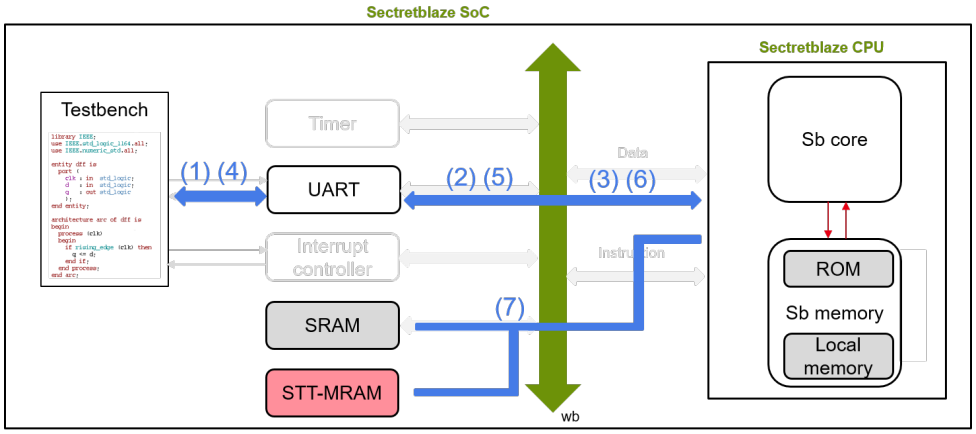


Figure 4.25: Les différentes étapes de boot depuis la ROM

- (1) : le testbench envoie l'octet de configuration à l'UART.
- (2) : l'UART met à jour le contenu des registres internes du processeur pour mémoriser l'octet de configuration
- (3) : le CPU lit la valeur de cet octet et donne la main à la mémoire souhaitée
- (4) : le testbench envoie les instructions du code applicatif à l'UART
- (5) : l'UART met à jour le contenu des registres
- (6) : le CPU lit les instructions envoyées par l'UART
- (7) : le CPU écrit ces instructions dans le type de mémoire choisi pour être exécutées par la suite.

Le processeur fonctionne selon ce scénario d'exécution décrit en assembleur. C'est finalement à travers ce code que s'ordonne l'exécution du code applicatif qui déclenche par la suite des écritures et des lectures de données.

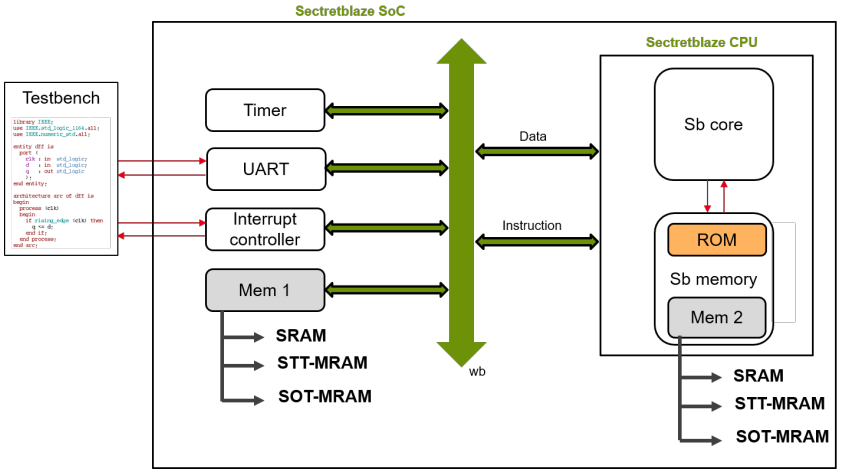


Figure 4.26: Architecture du SoC en implémentant deux type de mémoire Mem 1 et Mem 2

Dans le but de diminuer la consommation du SoC, notre approche a été d'implémenter deux mémoires de natures différentes dans le SoC, comme illustré sur figure 4.26: une mémoire partagée (Mem 1) et une mémoire locale (Mem2), chacune de ces deux mémoires étant soit une SRAM, soit une STT, soit une SOT. En effet, pendant l'initialisation du SoC un seul type de mémoire (Mem 1 ou Mem 2) sera sélectionné.

Nous présentons à travers la deuxième partie de ce chapitre le flot de conception numérique permettant d'intégrer les mémoires embarquées MRAM de 32 kbits dans l'architecture du SoC.

4.4.3 Flot de conception numérique

Le but étant d'évaluer les intérêts des technologies magnétiques principalement en termes de consommation au niveau du SoC, il était important de les intégrer dans un flot de conception numérique comme illustré sur la figure 4.27. Nous décrivons ci-après les spécificités de chaque étape d'un flot numérique.

4.4.3.1 Description comportementale au niveau RTL

Contrairement à la conception full custom, décrite dans le chapitre 2 où les circuits sont conçus au niveau transistor, la conception numérique consiste à décrire le fonctionnement d'un circuit à l'aide d'un langage de description matériel (HDL - Hardware Description Language). Dans le cas du Secretblaze, la description a été réalisée à l'aide du langage VHDL pour chaque bloc, comme le CPU, les éléments de la péripherie, le bus WB et bien d'autres. Ces blocs sont, au final, instanciés au plus haut niveau de la hiérarchie et connectés entre eux afin d'assurer le fonctionnement souhaité.

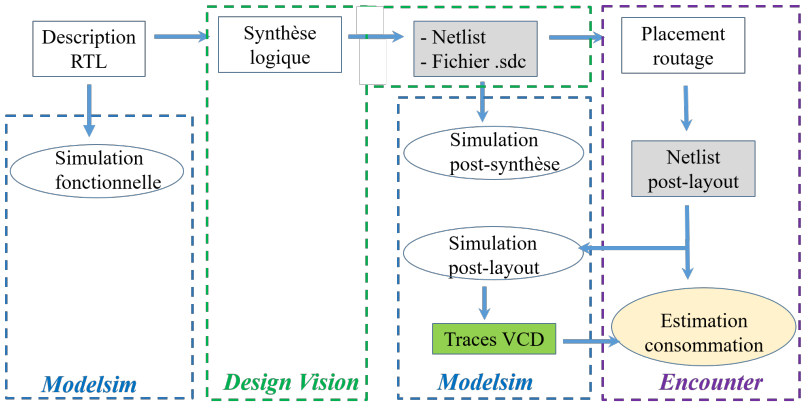


Figure 4.27: Flot de conception numérique

Pour intégrer les mémoires MRAM en technologie SOT et STT dans l'architecture du SoC, il a fallu décrire le comportement de ces mémoires au niveau RTL tout en précisant les entrées/ sorties et le nombre de bits par signal.
Dans le cas d'utilisation des mémoires CMOS, nous avons eu accès via le fondeur à l'ensemble des fichiers technologiques nécessaires pour chaque étapes du flot de conception.

Il était également important de modifier le code de démarrage stocké dans la ROM du processeur dû au fait que plusieurs types de mémoires sont implémentées dans l'architecture du SoC. Notre approche repose sur l'utilisation d'un seul type de mémoire (locale ou partagée) pour trois types de technologies disponibles: SRAM, STT ou SOT. Pour la mise en oeuvre de cette approche, deux variables ont été ajoutées dans l'octet de configuration permettant de sélectionner la mémoire désirée.

- **MEM SELECT** pour choisir le type de mémoire à implémenter, soit la mémoire locale en double port soit la mémoire partagée en simple port
- **MEM TYPE** pour choisir la technologie de mémoire, soit la SRAM, soit la STT soit la SOT

4.4.3.2 Simulation comportementale

Comme représenté par le diagramme du fonctionnement du SoC figure 4.25, celui-ci prend en entrée un fichier de simulation écrit en langage VHDL. En effet, simuler un circuit numérique consiste tout d'abord à lui associer un fichier de simulation (ou testbench) qui définit une structure ou un scénario souhaité.

La figure 4.28 montre un résultat de simulation par l'outil de simulation Modelsim, où l'on voit toutes les étapes du scénario implémentées pour l'application 'DES'. On

distingue 4 phases de fonctionnement du SoC :

- **Phase A :** L'octet de configuration est envoyé par l'UART dans lequel le type de mémoire est indiqué pour la mise en charge du code applicatif.
- **Phase B :** L'écriture des instructions du code applicatif (DES) dans la mémoire souhaitée (8 instructions de 4 octets chacune)
- **Phase C :** l'envoi des données de chiffrement par l'UART.
- **Phase D :** Le CPU est en charge de réaliser le chiffrement et d'envoyer les résultats via l'UART.

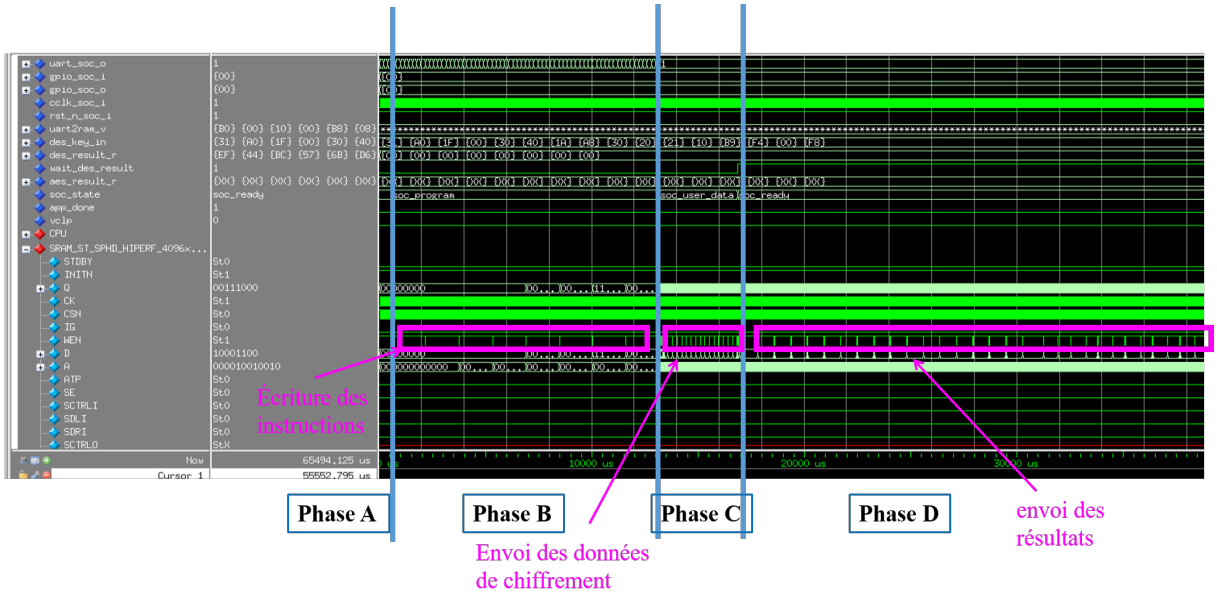


Figure 4.28: Les différentes étapes de fonctionnement du SoC

Après avoir décrit le comportement des mémoires implémentées au niveau RTL, il était important de vérifier le fonctionnement de toutes les versions. Pour cela 6 simulations logiques ont été mises en place : simulation du SoC en intégrant la mémoire partagée en SRAM, STT et SOT puis en intégrant la mémoire locale dans les 3 technologies également.
Cette étape a été une source de problèmes et d'investigations longues notamment pour les modèles de mémoire en SRAM, formés de plus de deux mille lignes de code. Nous avons donc cherché à comprendre les modèles du fondeur au format Verilog et les raisons pour lesquelles les mémoires SRAM ne fonctionnaient pas comme souhaité. Il a fallu inhiber tous les signaux supplémentaires définis dans ces modules (signaux de *scan chain*, *BIST mode*, *power gating* et bien d'autres). De plus, il a fallu synchroniser le signal de reset des mémoires avec le signal reset du SoC pour obtenir des simulations

conformes au fonctionnement attendu du SoC.
De nombreuses modifications ont été apportées au niveau des codes Verilog afin d'obtenir des simulations pré-synthèse correctes et conformes au fonctionnement du SoC. Nous avons obtenu au final 6 simulations similaires en mode de fonctionnement quelle que soit la technologie de mémoire implémentée.

4.4.3.3 Synthèse logique

La synthèse logique consiste à convertir la description comportementale en VHDL en un réseau de portes logiques interconnectées entre elles. Cette étape de conception permet de générer une netlist qui décrit l'architecture dans la technologie choisie.
Dans notre cas, les fichiers technologiques (.lib) des 2 mémoires SRAM (simple et double ports) ont été fournis par STMicroelectronics. Dans ces fichiers, tous les composants, les terminaux, la surface et la consommation sont déclarés. Il était donc important de comprendre globalement la structure de ces fichiers afin de créer des fichiers similaires pour les 4 mémoires non volatiles, interne et partagée, pour les deux technologies STT et SOT. Nous avons donc fait de nombreuses simulations électriques au niveau transistor afin d'extraire toutes les données nécessaires pour constituer ces 4 fichiers technologiques.

La simulation d'une mémoire complète avec tous les périphériques est très longue (environ une journée). C'est pourquoi toutes les simulations électriques ayant pour objectif la caractérisation des mémoires ont été réalisées sur le chemin critique uniquement (comme présenté sur la figure 4.29) de chaque configuration. Nous avons donc identifié la bitcell ayant le chemin le plus long et le temps de propagation maximal pour la caractériser.

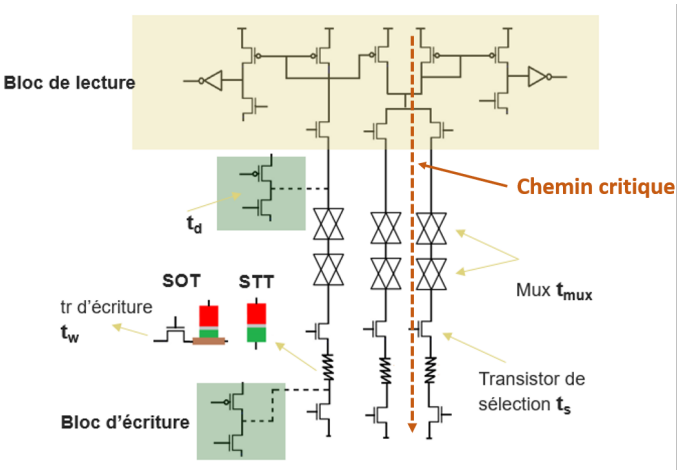


Figure 4.29: Chemin critique d'une mémoire non volatile en technologie STT et SOT

Comme le montre la figure 4.29, plusieurs éléments (transistors de sélection, bloc

d'écriture, bloc de lecture) ont un impact direct sur les performances de la mémoire. Nous avons donc réalisé des simulations paramétriques sur le chemin critique selon une gamme de variation du dimensionnement des différents transistors (transistor d'écriture, transistor de sélection, transistor de décodage). Le tableau de la figure 4.30 montre un exemple de résultats avec une mémoire partagée en technologie STT.

t_s	Ecriture '0' ($R_P \rightarrow R_{AP}$)				Ecriture '1' ($R_{AP} \rightarrow R_P$)				Lecture R_{AP}			Lecture R_P		
	I_s (µA)	T_s (ns)	E_s/bit (fJ)	E_s/sc (fJ)	I_s (µA)	T_s (ns)	E_s/bit (fJ)	E_s/sc (fJ)	I_r (µA)	T_r (ns)	E_s/sc (fJ)	I_r (µA)	T_r (ns)	E_s/sc (fJ)
W_s=80n	68,3	2,6	11	177	57,3	1,5	11,2	86	38,5	0,9	32	44,9	0,9	40,4
W_s=90n	75,2	1,8	9,6	135	61,9	1,2	10,5	74	40,5	0,9	36	47,9	0,9	43,1
W_s=100n	81,8	1,4	8,8	114	66,2	1	9,9	66	42,3	0,9	38	50,4	0,9	45,3
W_s=110n	88,1	1,15	9	100	84,6	--	--		44	0,9	40	52,8	0,9	47,5
W_s=120n	94,2	1	9,5	94	90	--	--		45,5	0,9	41	55	0,9	49,5

20% 46% 35% 33% 23%

Figure 4.30: Caractérisation d'une mémoire STT en fonction de la largeur du transistor de sélection

On peut déduire de ce tableau que les performances de la mémoire en écriture et en lecture dépendent largement de la taille des transistors de sélection en série avec la JTM. La largeur du transistor W_s est limitée à 110nm pour une écriture rapide. Au-delà de cette valeur, la JTM est écrite pendant la phase de lecture avec un courant de 53 µA environ.

On ne rencontre plus ce type de comportement avec la technologie SOT ou les deux voies de lecture et d'écriture sont séparées. Le tableau dans la figure 4.31 représente un exemple de la caractérisation avec la mémoire partagée en SOT. On remarque dans ce tableau que la consommation est réduite d'un facteur de 20% avec un transistor d'écriture 30% plus grand. La JTM est écrite beaucoup plus rapidement avec un courant important, ce qui est tout à fait logique.

En ce qui concerne l'estimation de la surface de ces 4 configurations de mémoires, il était difficile de faire le layout complet intégrant tous les périphériques pour chaque version individuellement. Nous avons alors réalisé le layout de la matrice de bitcells des différentes versions MRAM tout en gardant de façon similaire aux mémoire SRAM la taille des périphéries. Nous avons observé à partir des fichiers technologiques des SRAM (voir figure 4.32) que les périphériques (décodeurs, bloc de contrôle, etc) occupent environ 40% de la surface totale de la mémoire, ce qui représente une partie importante de la mémoire embarquée.

Le tableau 4.5 donne un ordre de grandeur de la surface estimée de l'ensemble

t_W	Ecriture				Lecture		
	I_S (µA)	T_S (ns)	E_S/bit (fJ)	E_S/sc (fJ)	I_r (µA)	T_r (ns)	E_r/sc (fJ)
W_S=80n	65	1,4	0,5	91	22	0,9	20
W_S=90n	75	1,1	0,6	82	22	0,9	20
W_S=100n	84	0,9	0,65	76	22	0,9	20
W_S=120n	92	0,8	0,7	74	22	0,9	20
W_S=200n	136	--			22	0,9	20

33% 63% 20%

Figure 4.31: Caractérisation d'une mémoire SOT en fonction de la largeur du transistor d'écriture

Table 4.5: *Estimation de la surface des différentes configurations de mémoire*

Type de mémoire	Mémoire partagée (Mem 1)			Mémoire locale (Mem 2)		
Type de technologie	SRAM	STT	SOT	SRAM	STT	SOT
Taille de la matrice (μm^2)	5360	1806	3645	9432	3612	7290
Taille totale de la mémoire (μm^2)	9000	3034	6124	34230	7224	14580

des 6 mémoires. On remarque sur le tableau que le choix de la technologie joue un rôle capital et influe sur la surface totale du SoC. La surface est significativement plus petite de 32% et 66% pour des mémoires SOT et STT respectivement par rapport à la mémoire SRAM en simple port pour la mémoire partagée.

La figure 4.33 montre un exemple de synthèse basée sur l'outil de synthèse Design Vision [82]. A l'issu de cette phase, on peut distinguer chaque bloc de la hiérarchie du SoC.

Lors de la synthèse, différents types de fichiers sont générés :

- *Un fichier ".v"*: fichier au format Verilog qui correspond à la netlist du SoC au niveau portes logiques, à base de cellules standards de bibliothèques de la technologie
- *Un fichier ".sdf"*: fichier "Standard Delay Format" qui détermine les délais entre les différentes portes logiques connectées entre elles.
- *Un fichier ".sdc"*: fichier "Synopsys Design Constraints" qui définit les contraintes du design
- *Un fichier ".txt"*: rapport de synthèse qui permet de quantifier une estimation des

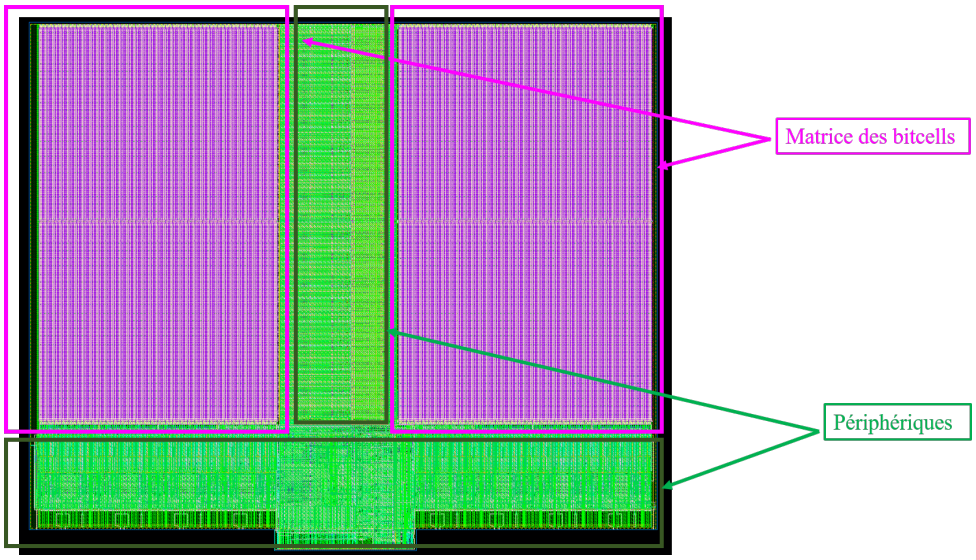


Figure 4.32: Vue layout d'une mémoire SRAM simple port (pour une mémoire partagée) en technologie CMOS 28nm FDSOI.

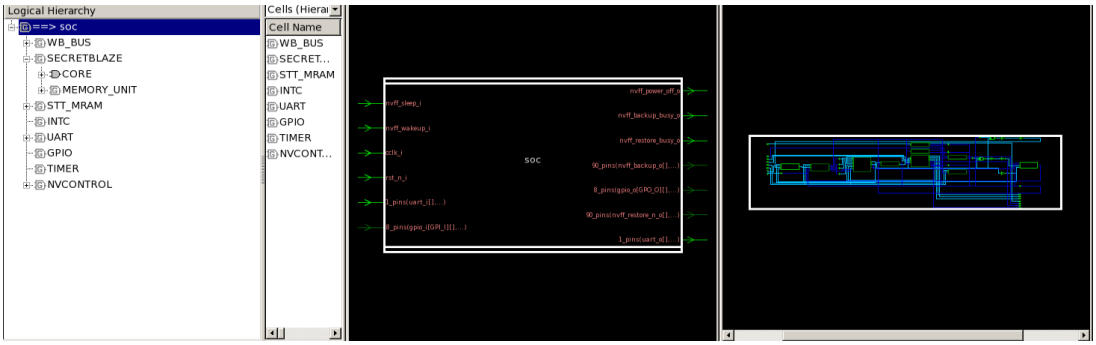


Figure 4.33: Synthèse logique du SoC

caractéristiques du circuit généré (analyse du chemin critique, consommation de puissance, surface,..).

4.4.3.4 Simulation après synthèse

Afin de vérifier le fonctionnement du SoC après synthèse, il était nécessaire de réaliser les simulations post-synthèses des six configurations de mémoires. L'outil de simulation Modelsim utilise dans cette étape la "netlist" générée par la synthèse qui décrit le comportement exact de chaque cellule, ainsi que le fichier .sdf qui contient les delais induits par les portes logiques.
Cette étape a révélé beaucoup de messages d'erreurs notamment au niveau des mémoires MRAM. Pour corriger ces erreurs, plusieurs itérations ont été nécessaires entre modification des codes RTL, simulations comportementales et synthèses. Cette étape a été fastidieuse et chronophage car le fait de modifier un code dans le SoC implique une

série de modification des plusieurs fichiers en parallèle. Nous avons aussi identifié lors des premiers essais quelques constructions RTL qui n'étaient pas synthétisables par le compilateur mais qui fonctionnaient parfaitement en simulation comportementale. En outre nous avons identifié que l'utilisation d'une sortie à 3 états dans la programmation VHDL est synthétisable mais ne produit pas le résultat voulu. Ceci était difficile à identifier facilement, car en ce qui concerne la description RTL les codes étaient complètement synthétisables. Après chaque modification apportée, nous avons dû générer une nouvelle netlist post-synthèse afin de lancer une nouvelle simulation. Une des difficultés provenant aussi du fait que l'essentiel du code a été écrit par une autre personne et qu'il est difficile de déceler les imperfections.
La simulation d'une "netlist" après synthèse implique un temps relativement long, ce qui nécessite beaucoup de temps pour obtenir les résultats. Après plusieurs reprises, nous avons réussi à valider le fonctionnement souhaité pour les 6 configurations de mémoire en simulation post-synthèse. Cela nous a permis également de nous assurer de la conformité de chaque netlist de mémoire avant de passer à l'étape finale dans le flot de conception numérique.

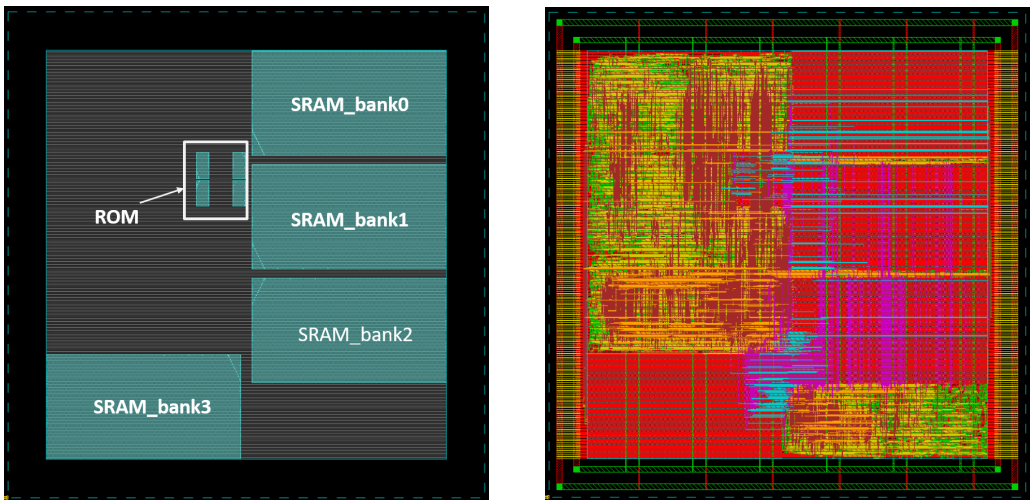
4.4.3.5 Placement et Routage

Lors de cette étape, l'outil Encounter utilise certains fichiers générés lors de la synthèse qui décrivent le SoC à partir des cellules standard connectées entre elles. Les principales étapes à effectuer lors du placement/routage sont:

- Définir l'espace alloué, appelé "floorplan", tout en précisant la taille et la géométrie du circuit (figure 4.34a)
- Définir le réseau d'alimentation pour les rails Vdd et Gnd en indiquant leur dimensionnement.
- Placer et répartir toutes les cellules dans l'ensemble du "flooplan"
- Router les signaux en interconnectant les entrées et les sorties des cellules

L'outil de placement routage utilise pour cela des fichiers technologiques ".lef" (Library Exchange Format) qui comportent pour chaque cellule la description de l'ensemble des polygones définis sur chaque niveau de métal, ainsi que l'ensemble des ports d'entrées/soties pour la connexion de chaque cellule. Ces fichiers ont été fournis par STMicroelectronics pour les mémoires SRAM. En ce qui concerne les mémoires MRAM, comme précisé précédemment, nous avons réalisé le layout de la matrice mémoire uniquement. Nous avons donc créer pour chacune des 4 mémoires MRAM un fichier .lef en considérant 40% de plus que la matrice en terme de surface, correspondant aux périphériques, pour rester dans les proportions des 2 SRAM CMOS.

Une fois l'étape de placement routage terminée (figure 4.34b), les mémoire MRAM en STT et SOT sont considérées comme une boite noîre dont on ne considère que les ports d'entrées et de sorties.



(a) Floorplan (b) SoC placé routé

Figure 4.34: Placement et routage du SoC complet

4.4.3.6 Simulation post-layout

Une fois l'étape de placement routage terminée, une netlist issue du layout est obtenue. Il est donc important de simuler à nouveau le circuit pour s'assurer du bon fonctionnement tout en prenant en compte les effets parasites. Après avoir résolu les erreurs rencontrées lors de simulations post-synthèse, les simulations post-layout étaient conformes au fonctionnement attendu sans trop de difficultés. Les simulations ont été réalisées pour les 6 configurations de mémoires. Un exemple de simulation post-layout du SoC intégrant la mémoire partagée en SRAM est illustré sur la figure 4.35. On retrouve le scénario applicatif implémenté. La mémoire est remplie/initialisée lors de la première phase par les instructions envoyées par l'UART (comme présenté dans le paragraphe 4.4.2). Lors de la deuxième phase, le programme est exécuté en autorisant des écritures et de lectures des données dans la mémoire.

4.4.3.7 Estimation de consommation du SoC

Cette phase repose sur toutes les étapes du flot numérique présentées ci-dessus. Une fois la simulation post-layout validée, on obtient à partir de l'outil de simulation un fichier (.wlf) qui contient le chronogramme de l'ensemble des signaux et des noeuds

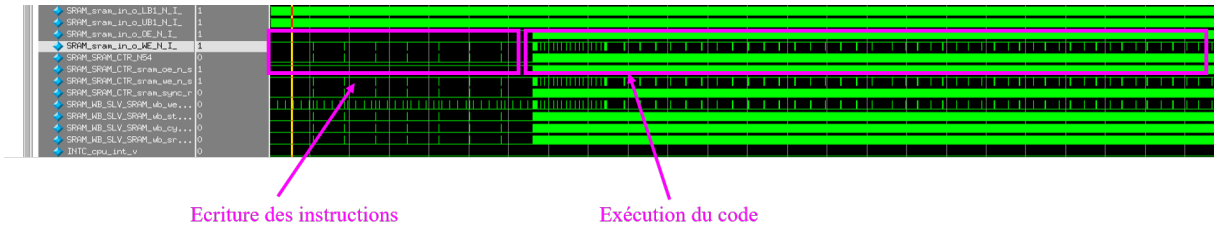


Figure 4.35: Simulation post-layout du SoC intégrant la mémoire partagée SRAM

du SoC. Il est important de convertir ce fichier en un fichier au format (.vcd) pour qu'il soit interprété par l'outil Encounter permettant l'analyse de consommation. Ce fichier décrit toutes les transitions des signaux numériques. Globalement ce fichier est lourd car il contient beaucoup d'informations.

En effet, le fichier (.vcd) est utilisé, en plus de la netlist post-layout, comme entrée pour l'outil Encounter par lequel l'estimation de la consommation est réalisée. Lors de cette estimation, l'outil Encounter est capable d'extraire plusieurs informations concernant la puissance du circuit. On distingue trois composantes de puissance :

- La puissance dynamique interne, ou puissance de court-circuit qui découle du fait que les commutations des signaux ne sont pas instantanées. Pendant un court-instant, un chemin de conduction entre la masse et l'alimentation est généré, résultant de la conduction simultanée des transistors NMOS et PMOS et donc un court-circuit.

- La puissance dynamique externe est dûe à la charge et à la décharge des différentes capacités dans le circuit.

- La puissance statique correspond à la puissance dissipée quand le circuit est au repos (sans aucune activité de commutation).

Un fichier listant toutes les cellules de base utilisées ainsi que toutes les consommations par composant est obtenu.

Le premier tableau 4.6 de notre étude compare les 3 composantes de la puissance consommée pour les 3 versions de la mémoire partagée. On peut remarquer que la puissance totale du SoC décroit de 11% et 16.5% avec les mémoires STT et SOT respectivement par rapport à la mémoire SRAM. En effet, cette décroissance est à l'origine de deux facteurs, la puissance dynamique interne et la puissance statique.

En ce qui concerne la puissance statique, on remarque de ce tableau que les valeurs sont tout de même relativement proches pour les 3 versions. Il est clair dans notre

étude que les mémoires MRAM n'apportent pas un large intêret du point de vue de la consommation statique, ce qui n'est pas très juste en réalité dans les noeuds technologiques avancés. Cependant, les valeurs obtenues restent tout à fait cohérentes par rapport à notre étude car la consommation statique des versions MRAM n'était pas calculée à partir des techniques de 'power gating', comme c'était le cas pour les mémoires SRAM. Dans le cas des MRAM, elle était calculée pendant la phase d'inactivité de la mémoire où le bloc d'écriture et de lecture sont inactifs. Ceci n'est pas très précis dans la mesure où la puissance dissipée dans les périphériques CMOS n'est pas prise en compte, et donc pas vraiment comparable. Les gains pourraient être bien supérieurs.

En ce qui concerne la puissance interne, on remarque qu'elle est plus importante en SRAM que dans les mémoires magnétiques. Ceci est dû au fait qu' il y a plus de portes logiques dans le SoC en intégrant la mémoire SRAM, comme le montre le tableau 4.7. En ce qui concerne la mémoire partagée par exemple, une SRAM occupe 50% de la surface totale du SoC, contre 30% et 42% pour les mémoires STT et SOT respectivement. Ce gain en surface est à l'origine du gain en consommation pour les mémoires magnétiques.

En comparant les deux technologies STT et SOT, on peut déduire qu'un gain de 6% est atteignable en termes de puissance totale grâce à la mémoire SOT par rapport à celle de la STT. Ceci s'explique par une consommation plus faible de la SOT pendant les opérations de lecture et d'écriture comme présenté précédement dans les tableaux des figures 4.30 et 4.31.

En ce qui concerne la mémoire locale en double port, l'estimation de la consommation sera évaluée dans les prochaines semaines.

Table 4.6: *Estimation de la consommation des différentes configurations de mémoire en simple porte*

Type de mémoire	SRAM	STT	SOT
Puissance dynamique interne (mW)	0.359	0.2466	0.2494
Puissance dynamique externe (mW)	0.003965	0.003833	0.003824
Puissance statique (mW)	0.1354	0.1919	0.1627
Puissance totale (mW)	0.4983	0.4424	0.4159

Table 4.7: *Estimation de la surface des différentes configurations de mémoire*

Type de mémoire	Mémoire partagée (Mem 1)			Mémoire locale (Mem 2)		
Type de technologie	SRAM	STT	SOT	SRAM	STT	SOT
Nombre de cellules	34823	29794	32541	36208	31254	34798
S_{totale}: Surface totale du SoC (μm^2)	71824	40401	58564	120409	69913	99856
Surface de mémoire/S_{totale}	50%	30%	42%	57%	37%	48%

4.5 Conclusion

A travers ce chapitre, deux grandes parties ont été présentées.

Dans la première partie, nous avons présenté l'architecture d'une mémoire embarquée intégrant tous les blocs numériques et analogiques périphériques. Nous avons étudié plusieurs configurations de mémoire en technologie SOT-MRAM afin d'évaluer les intêrets et les gains de chaque configuration au regard de l'application souhaitée. Un comparatif de 3 architectures selon plusieurs critères a été présenté, montrant les avantages et les inconvénients de chacune.
Ce travail a permis de comprendre et maîtriser les différents blocs formant l'architecture d'une mémoire à semi-conducteur. En outre, cela m'a appris une certaine méthodologie et des concepts de base du design pour une technologie avancée.

Dans la deuxième partie, nous avons intégré les architectures de mémoires conçues en technologie SOT et STT dans un processeur "Secretblaze" dans l'optique d'étudier le gain en consommation par rapport aux mémoires SRAM. Un flot de conception numérique spécifique a été mis en place afin de réaliser cette évaluation.
Cette étude a permis de montrer d'une part que la puissance totale du SoC est réduite de 11% et 16.5% avec les mémoires STT et SOT respectivement par rapport à la mémoire SRAM, et d'autre part qu'un gain de puissance de 6% est atteignable avec la mémoire SOT par rapport à l'utilisation de STT. En ce qui concerne la surface, cette étude montre que la SRAM occupe 50% de la surface totale du SoC, alors qu'une réduction de 16% est atteignable avec la mémoire SOT. La STT reste la mémoire la plus dense parmi ses technologies permettant une réduction de 60% de la surface du SoC par rapport à la mémoire SRAM.

Cette étude s'est avérée relativement difficile du fait que la maîtrise du fonctionnement du SoC était indispensable pour pouvoir intégrer les 6 versions de mémoires. Dans le cas de mémoires CMOS, l'ensemble des fichiers technologiques (.lib) ont été fournis par le STMicroelectronics. La principale difficulté était de n'avoir que très peu

de ressources disponibles en information pour ces mémoires. Nous n'avons pas eu accès à leur architecture au niveau transistor, ce qui a été compliqué pour comprendre plusieurs parties des fichiers .lib. En outre, ces mémoires ont été conçues à échelle industrielle avec des techniques et outils de conception avancés permettant d'avoir des résultats plus pertinents et précis que ce que nous avons pu faire pour les MRAM.

En ce qui concerne les mémoires MRAM, les fichiers technologiques ont été créés d'une façon similaire à celle des SRAM. De nombreuses simulations électriques au niveau transistor ont été réalisées sur le chemin critique uniquement, afin d'extraire les données nécessaires pour constituer ces fichiers. Ceci explique les résultats obtenus en termes de puissance statique qui n'étaient pas vraiment comparable avec la SRAM.

Pour conclure, ce travail a permis la mise en place d'un flot numérique complet en technologie 28nm FDSOI. Les résultats présentés dans ce chapitre illustrent surtout ce que l'on peut faire avec des outils de conception dans un environnement de recherche académique. Dans ce cadre, j'ai appris toute la méthodologie de la conception de circuit numériques, notamment la caractérisation d'IPs.

Conclusions et perspectives

L'objectif essentiel de cette thèse était l'évaluation du potentiel des mémoires émergentes non valatiles MRAM dans des circuits intégrés différents en proposant des architectures innovantes à base de ces mémoires. Cette étude s'est principalement focalisée sur deux grandes familles de circuits intégrés. Les circuits programmables (FPGA) et les circuits spécifiques (ASIC, mémoires), pour lesquelles deux flots de conception ont été mis en place dans cette étude. Les étapes de chaque flot sont illustrées sur la figure 4.36.

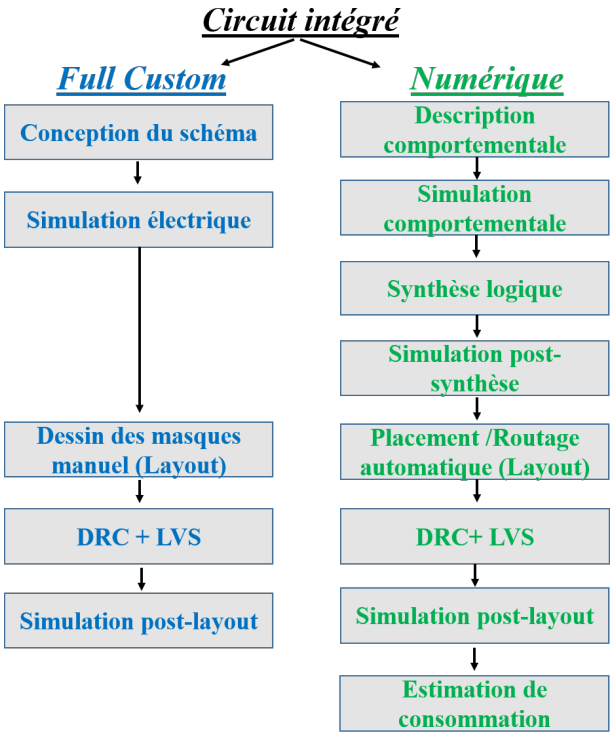


Figure 4.36: Flot de conception full custom et numérique d'un circuit intégré

La première partie de cette étude était consacrée à la conception full custom d'un générateur de fonctions logiques hybrides CMOS magnétique (LUT:Look Up Table), l'élément de base d'un FPGA. Plusieurs versions d'architectures de LUT ont été proposées dans le but d'intégrer ces circuits sur une puce de silicium, dans le cadre d'un projet interne MAD (Memory Advanced Demonstrator). Le problème majeur des architectures étudiées dans l'état de l'art était la forte dépendance entre le délai de propagation et la taille des LUT alors qu'idéalement les FPGA auraient besoin de LUT à plus d'entrées pour pouvoir réaliser des fonctions plus complexes dans un espace réduit. Dans ce contexte, nous avons proposé des LUT innovantes à 6 entrées ayant pour

énorme avantage que la vitesse ne dépend plus du nombre d'entrées de la LUT tout en assurant le même mode de fonctionnement. La conception de ces circuits a été réalisée de A à Z en passant par toutes les étapes de conception d'un flot full custom comme présenté dans la figure 4.36. Malheureusement, nous n'avons pas pu prouver les résultats électriques de ces circuits sur silicium car le procédé de fabrication de la partie magnétique des LUT est toujours en cours. Ces circuits seront testés dans les prochains mois.

En parallèle, j'ai eu l'opportunité de tester sur silicium le fonctionnement d'un filtre numérique passe-bas en technologie CMOS/STT-MRAM, conçu par Spintec dans le cadre du premier lot du projet MAD. Il était difficile de tirer une conclusion certaine à propos du test de ce circuit car sa contrôlabilité n'était pas assez importante. Cependant la conclusion que l'on peut affirmer c'est que la fonction de filtrage est partiellement bien assurée. Les basses fréquences sont apparemment transmises en sortie et les hautes fréquences sont coupées par le filtre en version CMOS et CMOS/STT-MRAM. Nous avons pu observer cela sur 3 bits parmi 8 du bus de sortie. En ce qui concerne la partie magnétique, il n'a pas été possible de montrer que l'on arrivait à écrire et lire des jonctions.

Ce travail a été une période très formatrice au cours de laquelle une certaine méthodologie et des concepts du test ont été acquis, ce qui aura été un atout majeur pour la suite de mes travaux.

La deuxième partie de notre étude a été consacrée à la réalisation d'une mémoire complète de 32kbits en technologie SOT-MRAM en 28nm FDSOI intégrant tous les blocs numériques et analogiques périphériques. Quelques structures de mémoire ont été proposées en technologie SOT-MRAM afin d'évaluer leurs éventuels gains au regard de l'application souhaitée. Parmi ces structures, nous avons proposé une cellule mémoire innovante qui a abouti à un brevet d'invention déposé au niveau national en juillet 2018 et étendu en juillet 2019.
Au cours de cette seconde partie j'ai acquis une expérience en design mémoire. J'ai conçu une forte mémoire SOT en adaptant tous les blocs périphériques de façon spécifique à cette technologie à 3 terminaux. J'ai également conçu une mémoire STT à partir d'une architecture classique développée à Spintec. Tout cela a été fait sur une technologie avancée 28nm CMOS FDSOI, ce qui a aussi été très formateur.

Par ailleurs, dans notre étude nous nous sommes intéressés à intégrer les technologies magnétiques non volatiles dans un processeur à travers ces mémoires conçues, dans l'optique d'étudier les gains par rapport à l'utilisation de mémoire SRAM. C'est

pourquoi, une large section est consacrée à l'étude d'un processeur "Secretblaze" en intégrant les mémoires STT-MRAM et SOT-MRAM en technologie CMOS 28 nm FDSOI, en comparaison des mémoires standards SRAM.
Un flot de conception numérique intégrant toutes les étapes (comme présenté sur la figure 4.36) a été mis en place.
Dans ce cadre, j'ai appris toute la méthodologie de la conception de circuits numériques, notamment la caractérisation d'IPs. En effet, pour intégrer les 4 versions de MRAM, j'ai dû caractériser chacune d'elles et créer les fichiers technologiques correspondants, permettant chacune des étapes (simulation, synthèse, placement/ routage, évaluation de consommation).

Cette étude a permis de montrer d'une part que la puissance totale du SoC est réduite de 11% et 16.5% avec les mémoires STT et SOT respectivement par rapport à l'utilisation d'une mémoire SRAM, et d'autre part qu'un gain de puissance de 6% est atteignable avec une mémoire SOT par rapport à l'utilisation d'une STT. En ce qui concerne la surface, cette étude montre que la SRAM occupe 50% de la surface totale du SoC, alors qu'une réduction de 16% est atteignable avec la mémoire SOT. La STT reste la mémoire la plus dense parmi ses technologies permettant une réduction de 60% de la surface du SoC par rapport à la version avec la mémoire SRAM.
Cette étude s'est avérée relativement difficile du fait que nous avons comparé les mémoires magnétiques conçues lors de cette thèse dans un environnement académique avec des modèles SRAM conçus à une échelle industrielle avec des techniques et des outils de conception avancés permettant d'avoir des résultats plus pertinents, notamment en intégrant des aspects de "power gating" dans la consommation statique.

Les avantages liés à cette partie sont grandissants. D'une part, ce travail a permis la mise en place d'un flot numérique complet en technologie 28nm FDSOI et d'autre part l'évaluation des intérêts des deux technologies STT et SOT dans un système complet de type SoC basé sur des résultats et caractérisations électriques. Ceci a également permis d'ouvrir de très intéressantes perspectives pour les prochaines années.

Cette étude du processeur "Secretblaze" a été essentiellement réalisée pour une application cryptographique basse consommation. L'implémentation de cette application repose sur un scénario spécifique permettant d'autoriser un certain nombre d'opérations de lecture et d'écriture pour la mémoire. On peut imaginer que pour un autre scénario applicatif, les performances du processeur peuvent être largement améliorées et davantage bénéfiques.
En termes de consommation, il serait intéressant d'étudier le gain en puissance en ajoutant les techniques de "power gating" pour les mémoires MRAM afin de réduire

voire inhiber l'énergie consommée en mode veille. En effet, cette étude permet de comparer de façon efficace l'utilisation de mémoires STT et SOT entre elles, mais de façon moins efficace qu'avec l'utilisation de mémoires SRAM du fait que la caractérisation des IPs n'ait pas été faite suivant le même protocole. Il est fort probable que l'utilisation de MRAM au sens large pourrait apporter d'autres avantages que celle de la consommation, comme la densité par exemple, dans des applications autre que cryptographie. Ce travail ouvre la porte à d'autres études du même genre.

En tant que perspective à court terme, il est essentiel de réfléchir à l'intégration de la structure hybride STT/SOT proposée dans le chapitre 4 dans l'architecture du processeur. Cette structure innovante combine les deux aspects, la haute densité de la STT et la haute vitesse de la SOT avec une capacité de stockage deux fois plus grandes que l'architecture standard SOT, 2T-JTM.
Ceci était un de nos objectifs pendant la thèse mais la conception d'une mémoire complète de 32kbits intégrant cette structure s'est avérée longue en modifiant tous les périphériques de mémoire. C'est pourquoi, nous avons privilégié l'intégration des architectures standards en STT et SOT dans l'étude du SoC.
Cette structure hybride trouvera aussi sa place dans des applications nécessitant une haute capacité de stockage. Cependant, l'intégration de diodes entre les bitcells adjacentes reste à améliorer. Des sélécteurs type diode, pourraient être utilisés dans l'empilement des JTM afin de remplacer les diodes de protection, ce qui pourrait être un fort avantage pour des mémoires crossbar à base de JTM.

Les perspectives de ces travaux sont donc nombreuses et il est fort probable que l'intégration et l'utilisation appropriée de nouvelles architectures MRAM amènent à des résultats prometteurs dans les années à venir.

Brevets et Publications

Brevet

COMPACT MAGNETIC STORAGE MEMORY CELL, Rana ALHALABI, Gregory DI PENDINA

Publications

Alhalabi, R., Nowak, E., Prejbeanu, I. L., & Di Pendina, G. (2018, October). High density SOT-MRAM memory array based on a single transistor. In 2018 Non-Volatile Memory Technology Symposium (NVMTS) (pp. 1-3). IEEE.

Kharbouche-Harrari, M., Alhalabi, R., Postel-Pellerin, J., Wacquez, R., Aboulkassimi, D., Nowak, E., ... & Di Pendina, G. (2018, November). MRAM: from STT to SOT, for security and memory. In 2018 Conference on Design of Circuits and Integrated Systems (DCIS) (pp. 1-6). IEEE.

Alhalabi, R., Di Pendina, G., Prejbeanu, I. L., & Nowak, E. (2017, August). High speed and high-area efficiency non-volatile look-up table design based on magnetic tunnel junction. In 2017 17th Non-Volatile Memory Technology Symposium (NVMTS) (pp. 1-4). IEEE.

Bibliography

[1] Fig. 1 Typical structure of a computer memory hierarchy., . URL https://www.researchgate.net/figure/Typical-structure-of-a-computer-memory-hierarchy_fig1_281805561.

[2] Jagan Meena, Simon Sze, Umesh Chand, and Tseung-Yuen Tseng. Overview of emerging nonvolatile memory technologies. *Nanoscale Research Letters*, 9 (1):526, 2014. ISSN 1556-276X. doi: 10.1186/1556-276X-9-526. URL http://nanoscalereslett.springeropen.com/articles/10.1186/1556-276X-9-526.

[3] schéma de lecture d'une MRAM - Recherche Google, . URL https://www.google.com/search?q=sch%C3%A9ma+de+lecture+d%27une+MRAM&client=firefox-b-d&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjdgoux-s_jAhUE3uAKHa2LAm8Q_AUIEigC&biw=1280&bih=607&dpr=1.5#imgrc=3_waM3UkENGhOM:.

[4] yole, . URL https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjC-Mbi1crjAhWEzoUKHbwvAeoQFjAAegQIAxAB&url=https%3A%2F%2Fwww.usinenouvelle.com%2Farticle%2Fles-technologies-emergentes-de-memoires-non-volatiles-prennent-enfin-leur-esso
N778059&usg=AOvVaw0g5IOIDvztccv5e2gljnSI.

[5] Kotb Jabeur, Gregory Di Pendina, Fabrice Bernard-Granger, and Guillaume Prenat. Spin Orbit Torque Non-Volatile Flip-Flop for High Speed and Low Energy Applications. *IEEE Electron Device Letters*, 35(3):408–410, March 2014. ISSN 0741-3106, 1558-0563. doi: 10.1109/LED.2013.2297397. URL http://ieeexplore.ieee.org/document/6714403/.

[6] Lionel Torres and Sophiane Senni. From Embedded World to High Performance Computing using STT-MRAM. page 28.

[7] architecture FPGA - Recherche Google, . URL https://www.google.com/search?q=architecture+FPGA&client=firefox-b-d&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiEq5e3_8_jAhUrAGMBHbleAP8Q_AUIESgB&biw=1280&bih=607#imgrc=TmtmwogbzL1tDM:.

[8] E. Ahmed and J. Rose. The effect of LUT and cluster size on deep-submicron FPGA performance and density. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12(3):288–298, March 2004. ISSN 1063-8210, 1557-9999. doi: 10.1109/TVLSI.2004.824300. URL http://ieeexplore.ieee.org/document/1281800/.

[9] William C. Black and Bodhisattva Das. Programmable Logic Using Giant-Magnetoresistance and Spin-Dependent Tunneling Devices. *Journal of Applied Physics*, 87:6674–6679, May 2000. doi: 10.1063/1.372806.

[10] D. Suzuki, M. Natsui, T. Endoh, H. Ohno, and T. Hanyu. Six-input lookup table circuit with 62% fewer transistors using nonvolatile logic-in-memory architecture with series/parallel-connected magnetic tunnel junctions. *Journal of Applied Physics*, 111(7):07E318, April 2012. ISSN 0021-8979, 1089-7550. doi: 10.1063/1.3672411. URL http://aip.scitation.org/doi/10.1063/1.3672411.

[11] Daisuke Suzuki and Takahiro Hanyu. Design of an MTJ-based nonvolatile lookup table circuit using an energy-efficient single-ended logic-in-memory structure. In *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4, Fort Collins, CO, USA, August 2015. IEEE. ISBN 978-1-4673-6558-1. doi: 10.1109/MWSCAS.2015.7282195. URL http://ieeexplore.ieee.org/document/7282195/.

[12] Daisuke Suzuki, Yuhui Lin, Masanori Natsui, and Takahiro Hanyu. A 71%-Area-Reduced Six-Input Nonvolatile Lookup-Table Circuit Using a Three-Terminal Magnetic-Tunnel-Junction-Based Single-Ended Structure. *Japanese Journal of Applied Physics*, 52(4S):04CM04, April 2013. ISSN 0021-4922, 1347-4065. doi: 10.7567/JJAP.52.04CM04. URL http://stacks.iop.org/1347-4065/52/04CM04.

[13] Diamond Digital Applications Student Guide (PN: 068-0112-09). page 454, .

[14] Pascal Benoit. Contribution à la conception de systèmes numériques adaptatifs. page 205.

[15] Duan Zongtao, Zhang Yanni, and Duan Zongyuan. An Overview of Data Bandwidth Hierarchy for an Embedded Stream Processor. In *2009 International Forum on Computer Science-Technology and Applications*, pages 34–36, Chongqing, China, 2009. IEEE. ISBN 978-1-4244-5422-8 978-0-7695-3930-0. doi: 10.1109/IFCSTA.2009.14. URL http://ieeexplore.ieee.org/document/5385141/.

[16] M. Aoki, Y. Nakagome, M. Horiguchi, H. Tanaka, S. Ikenaga, J. Etoh, Y. Kawamoto, S. Kimura, E. Takeda, H. Sunami, and K. Itoh. A 60-ns 16-Mbit CMOS DRAM with a transposed data-line structure. *IEEE Journal of Solid-State*

Circuits, 23(5):1113–1119, October 1988. ISSN 00189200. doi: 10.1109/4.5932. URL http://ieeexplore.ieee.org/document/5932/.

[17] Donghyuk Lee, Yoongu Kim, Gennady Pekhimenko, Samira Khan, Vivek Seshadri, Kevin Chang, and Onur Mutlu. Adaptive-latency DRAM: Optimizing DRAM timing for the common-case. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 489–501, Burlingame, CA, USA, February 2015. IEEE. ISBN 978-1-4799-8930-0. doi: 10.1109/HPCA.2015.7056057. URL http://ieeexplore.ieee.org/document/7056057/.

[18] Thomas Parnell, Celestine Dunner, Thomas Mittelholzer, Nikolaos Papandreou, and Haralampos Pozidis. Endurance limits of MLC NAND flash. In *2015 IEEE International Conference on Communications (ICC)*, pages 376–381, London, June 2015. IEEE. ISBN 978-1-4673-6432-4. doi: 10.1109/ICC.2015.7248350. URL http://ieeexplore.ieee.org/document/7248350/.

[19] Zhiping Zhang, Young Yang Liauw, Chen Chen, and S. Simon Wong. Monolithic 3-D FPGAs. *Proceedings of the IEEE*, 103(7):1197–1210, July 2015. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2015.2433954. URL http://ieeexplore.ieee.org/document/7118639/.

[20] L'Usine Nouvelle. Les technologies émergentes de mémoires non volatiles prennent enfin leur essor - Electronique. December 2018. URL https://www.usinenouvelle.com/article/les-technologies-emergentes-de-memoires-non-volatiles-prennent-enfin-leur-essor.N778059.

[21] Charles Henry Sie. Memory cell using bistable resistivity in amorphous As-Te-Ge film. page 64.

[22] Junji Tominaga, Xiaomin Wang, Alexander V. Kolobov, and Paul Fons. A reconsideration of the thermodynamics of phase-change switching. *physica status solidi (b)*, 249(10):1932–1938, October 2012. ISSN 03701972. doi: 10.1002/pssb.201200350. URL http://doi.wiley.com/10.1002/pssb.201200350.

[23] E. M. Philofsky. FRAM-the ultimate memory. In *Proceedings of Nonvolatile Memory Technology Conference*, pages 99–104, June 1996. doi: 10.1109/NVMT.1996.534679.

[24] D. Takashima. Overview of FeRAMs: Trends and perspectives. In *2011 11th Annual Non-Volatile Memory Technology Symposium Proceeding*, pages 1–6, November 2011. doi: 10.1109/NVMTS.2011.6137107.

[25] Toshikazu Fukuda, Koji Kohara, Toshiaki Dozaka, Yasuhisa Takeyama, Tsuyoshi Midorikawa, Kenji Hashimoto, Ichiro Wakiyama, Shinji Miyano, and Takehiko Hojo. 13.4 A 7ns-access-time 25μW/MHz 128kb SRAM for low-power fast wake-up MCU in 65nm CMOS with 27fa/b retention current. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 236–237, San Francisco, CA, USA, February 2014. IEEE. ISBN 978-1-4799-0920-9 978-1-4799-0918-6. doi: 10.1109/ISSCC.2014.6757415. URL http://ieeexplore.ieee.org/document/6757415/.

[26] E. Vianello, D. R. B. Ly, S. L. Barbera, T. Dalgaty, N. Castellani, G. Navarro, G. Bourgeois, A. Valentian, E. Nowak, and D. Querlioz. Metal Oxide Resistive Memory (OxRAM) and Phase Change Memory (PCM) as Artificial Synapses in Spiking Neural Networks. In *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 561–564, December 2018. doi: 10.1109/ICECS.2018.8617869.

[27] D. A. Robayo, C. Nail, G. Sassine, J. F. Nodin, M. Bernard, Q. Raffay, G. Ghibaudo, G. Molas, and E. Nowak. Statistical analysis of CBRAM endurance. In *2018 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, pages 1–2, April 2018. doi: 10.1109/VLSI-TSA.2018.8403856.

[28] Mme Anne Kaminski. Caractérisation électrique et modélisation de la dynamique de commutation résistive dans des mémoires OxRAM à base de HfO2. page 194.

[29] Ting-Chang Chang, Kuan-Chang Chang, Tsung-Ming Tsai, Tian-Jian Chu, and Simon M. Sze. Resistance random access memory. *Materials Today*, 19(5):254–264, June 2016. ISSN 13697021. doi: 10.1016/j.mattod.2015.11.009. URL https://linkinghub.elsevier.com/retrieve/pii/S1369702115003843.

[30] Dmytro Apalkov, Bernard Dieny, and J. M. Slaughter. Magnetoresistive Random Access Memory. *Proceedings of the IEEE*, 104(10):1796–1830, October 2016. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2016.2590142. URL https://ieeexplore.ieee.org/document/7555318/.

[31] . URL https://www.everspin.com/spin-transfer-torque-mram-products.

[32] Crocus Nano Electronics successfully tests its 90 nm pMTJ STT-MRAM tech | MRAM-Info, . URL https://www.mram-info.com/crocus-nano-electronics-successfully-tests-its-90-nm-pmtj-stt-mram-tech.

[33] Spin-transfer Torque MRAM Products | Everspin. URL https://www.everspin.com/spin-transfer-torque-mram-products.

[34] Samsung | MRAM-Info, . URL https://www.mram-info.com/mram_memory_makers/samsung.

[35] URL https://www.mram-info.com/tags/market_reports.

[36] Monsieur Frédéric Petroff. Ultimate scalability of STT-MRAM: storage layer with perpendicular shape anisotropy. page 187.

[37] Jérémy Alvarez-Hérault. Mémoire magnétique à écriture par courant polarisé en spin assistée thermiquement. page 136.

[38] Bi Wu, Yuanqing Cheng, Jianlei Yang, Aida Todri-Sanial, and Weisheng Zhao. Temperature Impact Analysis and Access Reliability Enhancement for 1t1mtj STT-RAM. *IEEE Transactions on Reliability*, 65(4):1755–1768, December 2016. ISSN 0018-9529, 1558-1721. doi: 10.1109/TR.2016.2608910. URL http://ieeexplore.ieee.org/document/7585072/.

[39] Djaafar Chabi, Weisheng Zhao, Erya Deng, Yue Zhang, Nesrine Ben Romdhane, Jacques-Olivier Klein, and Claude Chappert. Ultra Low Power Magnetic Flip-Flop Based on Checkpointing/Power Gating and Self-Enable Mechanisms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(6):1755–1765, June 2014. ISSN 1549-8328, 1558-0806. doi: 10.1109/TCSI.2013.2295026. URL http://ieeexplore.ieee.org/document/6701399/.

[40] P. Wang, G. Jan, L. Thomas, A. Wang, T. Zhong, T. Torng, Y. Lee, H. Liu, J. Zhu, S. Le, S. Serrano-Guisan, R. Tong, J. Haq, J. Teng, D. Shen, R. He, and V. Lam. Development of STT-MRAM for embedded memory applications. In *2017 IEEE International Magnetics Conference (INTERMAG)*, pages 1–1, Dublin, Ireland, April 2017. IEEE. ISBN 978-1-5386-1086-2. doi: 10.1109/INTMAG.2017.8007930. URL http://ieeexplore.ieee.org/document/8007930/.

[41] Murat Cubukcu, Olivier Boulle, Nikolai Mikuszeit, Claire Hamelin, Thomas Bracher, Nathalie Lamard, Marie-Claire Cyrille, Liliana Buda-Prejbeanu, Kevin Garello, Ioan Mihai Miron, O. Klein, G. de Loubens, V. V. Naletov, Juergen Langer, Berthold Ocker, Pietro Gambardella, and Gilles Gaudin. Ultra-Fast Perpendicular Spin–Orbit Torque MRAM. *IEEE Transactions on Magnetics*, 54(4):1–4, April 2018. ISSN 0018-9464, 1941-0069. doi: 10.1109/TMAG.2017.2772185. URL http://ieeexplore.ieee.org/document/8291048/.

[42] Claire Hamelin. *Couples de spin-orbite en vue d'applications aux mémoires cache*. PhD thesis, October 2016.

[43] Tetsuo Endoh and Hiroaki Honjo. A Recent Progress of Spintronics Devices for Integrated Circuit Applications. *Journal of Low Power Electronics and Applications*, 8(4):44, November 2018. ISSN 2079-9268. doi: 10.3390/jlpea8040044. URL http://www.mdpi.com/2079-9268/8/4/44.

[44] K. Garello, F. Yasin, S. Couet, L. Souriau, J. Swerts, S. Rao, S. Van Beek, W. Kim, E. Liu, S. Kundu, D. Tsvetanova, K. Croes, N. Jossart, E. Grimaldi, M. Baumgartner, D. Crotti, A. Fumemont, P. Gambardella, and G.S. Kar. SOT-MRAM 300mm Integration for Low Power and Ultrafast Embedded Memories. In *2018 IEEE Symposium on VLSI Circuits*, pages 81–82, Honolulu, HI, June 2018. IEEE. ISBN 978-1-5386-4214-6. doi: 10.1109/VLSIC.2018.8502269. URL https://ieeexplore.ieee.org/document/8502269/.

[45] Liang Chang, Z. Wang, Yuqian Gao, W. Kang, Y. Zhang, and W. Zhao. Evaluation of spin-Hall-assisted STT-MRAM for cache replacement. In *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pages 73–78, July 2016. doi: 10.1145/2950067.2950107.

[46] Fabian Oboril, Rajendra Bishnoi, Mojtaba Ebrahimi, and Mehdi B. Tahoori. Evaluation of Hybrid Memory Technologies Using SOT-MRAM for On-Chip Cache Hierarchy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(3):367–380, March 2015. ISSN 0278-0070, 1937-4151. doi: 10.1109/TCAD.2015.2391254. URL http://ieeexplore.ieee.org/document/7008441/.

[47] I. O., John von Neumann, and A. H. Taub. John von Neumann Collected Works:. *Journal of the American Statistical Association*, 59(307):981, September 1964. ISSN 01621459. doi: 10.2307/2283131. URL https://www.jstor.org/stable/2283131?origin=crossref.

[48] D. N. Yadav and P. L. Thangkhiew. Towards an In-Memory Reconfiguration of Arithmetic Logical Unit using Memristor Crossbar Array. In *2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6, March 2018. doi: 10.1109/CONECCT.2018.8482399.

[49] J. Talafy and H. R. Zarandi. Soft error analysis of MTJ-based logic-in-memory full adder: Threats and solution. In *2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 207–208, July 2017. doi: 10.1109/IOLTS.2017.8046221.

[50] W. Lin, S. Sheu, C. Kuo, P. Tseng, M. Chang, K. Su, C. Lin, K. Tsai, S. Lee, S. Liu, Y. Chen, H. Lee, C. Hsu, F. T. Chen, T. Ku, M. Tsai, and M. Kao. A nonvolatile

look-up table using ReRAM for reconfigurable logic. In *2014 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pages 133–136, November 2014. doi: 10.1109/ASSCC.2014.7008878.

[51] Jayasimha Atulasimha and Supriyo Bandyopadhyay. *Nanomagnetic and Spintronic Devices for Energy-Efficient Memory and Computing*. John Wiley & Sons, January 2016. ISBN 978-1-118-86925-3. Google-Books-ID: DQx6CwAAQBAJ.

[52] Gefei Wang. *Conception et développement de nouveaux circuits logiques basés sur des spin transistor à effet de champ*. thesis, Paris Saclay, February 2019. URL `http://www.theses.fr/2019SACLS056`.

[53] Anis Feki. Solutions of subthreshold SRAM in ultra-wide-voltage range in advanced CMOS technologies for biomedical and wireless sensor applications. page 152.

[54] Sophiane Senni, Lionel Torres, Pascal Benoit, Abdoulaye Gamatie, and Gilles Sassatelli. Normally-Off Computing and Checkpoint/Rollback for Fast, Low-Power, and Reliable Devices. *IEEE Magnetics Letters*, 8:1–5, 2017. ISSN 1949-307X, 1949-3088. doi: 10.1109/LMAG.2017.2712780. URL `http://ieeexplore.ieee.org/document/7942072/`.

[55] Mengting Zhao. Approximate Computing et Conception d'Opérateurs Arithmétiques Approximatifs. page 29.

[56] Virtex-5 Family Overview (DS100). page 15, 2015.

[57] Bernard Dieny, Ronald B. Goldfarb, and Kyung-Jin Lee. *Introduction to Magnetic Random-Access Memory*. John Wiley & Sons, December 2016. ISBN 978-1-119-00974-0. Google-Books-ID: 3BGcDQAAQBAJ.

[58] Erya Deng. Design and development of low-power and reliable logic circuits based on spin-transfer torque magnetic tunnel junctions. page 215.

[59] Olivier Goncalves. Conception sur mesure d'un FPGA durci aux radiations à base de mémoires magnétiques. page 180.

[60] S. Senni, T. Delobelle, O. Coi, P. Peneau, L. Torres, A. Gamatie, P. Benoit, and G. Sassatelli. Embedded systems to high performance computing using STT-MRAM. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 536–541, March 2017. doi: 10.23919/DATE.2017.7927046.

[61] Lionel Torres, Raphael Martins Brum, Luis Vitorio Cargnini, and Gilles Sassatelli. Trends on the application of emerging nonvolatile memory to processors and programmable devices. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pages 101–104, Beijing, May 2013. IEEE. ISBN 978-1-4673-5762-3 978-1-4673-5760-9 978-1-4673-5761-6. doi: 10.1109/ISCAS.2013.6571792. URL http://ieeexplore.ieee.org/document/6571792/.

[62] SecretBlaze | ADAC, . URL http://www.lirmm.fr/ADAC/?page_id=462.

[63] Programmable logic device, June 2019. URL https://en.wikipedia.org/w/index.php?title=Programmable_logic_device&oldid=902641037. Page Version ID: 902641037.

[64] Xilinx, June 2019. URL https://fr.wikipedia.org/w/index.php?title=Xilinx&oldid=159862602. Page Version ID: 159862602.

[65] http://www.xilinx.com.

[66] opalkelly.com. Products Archive. URL https://opalkelly.com/products/.

[67] Z. Almohaimeed and M. Sima. Look-Up tables with multiple inputs for secured-by-design FPGAs. In *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1–4, October 2016. doi: 10.1109/MWSCAS.2016.7870159.

[68] D. Kumar, P. Kumar, and M. Pattanaik. Performance Analysis of 90nm Look Up Table (LUT) for Low Power Application. In *2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, pages 404–407, September 2010. doi: 10.1109/DSD.2010.72.

[69] Chen-Chang Zhu, Xue-Gong Zhou, Hao Zhou, Yue-Er Shan, Yan-Feng Xu, and Kai Hu. Performance evaluation of input sharing LUT architectures in FPGA. In *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pages 710–712, October 2016. doi: 10.1109/ICSICT.2016.7999019.

[70] D. Suzuki, M. Natsui, S. Ikeda, T. Endoh, H. Ohno, and T. Hanyu. Design of a variation-resilient single-ended non-volatile six-input lookup table circuit with a redundant-magnetic tunnel junction-based active load for smart Internet-of-things applications. *Electronics Letters*, 53(7):456–458, March 2017. ISSN 0013-5194, 1350-911X. doi: 10.1049/el.2016.4233. URL https://digital-library.theiet.org/content/journals/10.1049/el.2016.4233.

[71] S. M. Trimberger. Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology. *Proceedings of the IEEE*, 103(3):318–331, March 2015. ISSN 0018-9219. doi: 10.1109/JPROC.2015.2392104.

[72] Using Nonvolatile Static RAMs - Application Note - Maxim, . URL https://www.maximintegrated.com/en/app-notes/index.mvp/id/540.

[73] What is 3d NAND flash memory? What is its competitive advantage?, . URL https://www.elinfor.com/knowledge/what-is-3d-nand-flash-memory-what-is-its-competitive-advantage-p-10920.

[74] Yahya Lakys, Weisheng ZHAO, Jacques-Olivier Klein, and Claude Chappert. Hardening Techniques for MRAM-Based Nonvolatile Latches and Logic. *IEEE Transactions on Nuclear Science*, 59:1136–1141, August 2012. doi: 10.1109/TNS.2012.2195677.

[75] T. Taylor and G.A. Maston. Standard test interface language (STIL) a new language for patterns and waveforms. In *Proceedings International Test Conference 1996. Test and Design Validity*, pages 565–570, Washington, DC, USA, 1996. Int. Test Conference. ISBN 978-0-7803-3541-7. doi: 10.1109/TEST.1996.557091. URL http://ieeexplore.ieee.org/document/557091/.

[76] Richard Dorrance, Juan G. Alzate, Sergiy S. Cherepov, Pramey Upadhyaya, Ilya N. Krivorotov, Jordan A. Katine, Juergen Langer, Kang L. Wang, Pedram Khalili Amiri, and Dejan Markovic. Diode-MTJ Crossbar Memory Cell Using Voltage-Induced Unipolar Switching for High-Density MRAM. *IEEE Electron Device Letters*, 34(6):753–755, June 2013. ISSN 0741-3106, 1558-0563. doi: 10.1109/LED.2013.2255096. URL http://ieeexplore.ieee.org/document/6513288/.

[77] url http://www.biu-montpellier.fr/florabium/jsp/nnt.jsp?nnt=2012MON20046, .

[78] Richard Herveille. Wishbone Specification. page 140.

[79] Remi Busseuil, Lyonel Barthe, Gabriel Marchesan Almeida, Luciano Ost, Florent Bruguier, Gilles Sassatelli, Pascal Benoit, Michel Robert, and Lionel Torres. Open-Scale: A Scalable, Open-Source NOC-based MPSoC for Design Space Exploration. In *2011 International Conference on Reconfigurable Computing and FPGAs*, pages 357–362, Cancun, Mexico, November 2011. IEEE. ISBN 978-0-7695-4551-6 978-1-4577-1734-5. doi: 10.1109/ReConFig.2011.66. URL http://ieeexplore.ieee.org/document/6128603/.

[80] GREAT - A H2020 ICT project at SPINTEC, June 2016. URL http://www.spintec.fr/great-a-h2020-fet-project/.

[81] Data Encryption Standard, July 2019. URL https://en.wikipedia.org/w/index.php?title=Data_Encryption_Standard&oldid=905592598. Page Version ID: 905592598.

[82] Design Vision, . URL https://www.utdallas.edu/~akshay.sridharan/index_files/Page6328.htm.

Résumé

Après de nombreuses études au cours des dernières décennies, les technologies émergentes de mémoires non volatiles décollent enfin dans le marché des semi-conducteurs. Elles ont comme objectif principal de prendre le relais des mémoires flash et DRAM qui touchent à leurs limites en termes de densité, de miniaturisation, de consommation ou d'amélioration de la vitesse. Parmi les technologies émergentes, la mémoire MRAM passe de simple « candidat potentiel » il y a quelques années à des mémoires fabriquées par de grandes industries, aujourd'hui disponibles sur le marché, suscitant un fort intérêt général dans le monde industriel de la microélectronique. Ses atouts permettent d'intégrer cette mémoire dans des flots de conception full custom et numérique afin d'améliorer certaines performances soit au niveau cellule élémentaire soit au niveau architecture. C'est pourquoi nous proposons dans une première partie la conception de circuits hybrides CMOS/magnétique de type LUT (Look Up Table) en technologie STT-MRAM (Spin Transfer Torque) ayant pour but de réaliser un démonstrateur et de le tester par la suite. La conception full custom de A à Z de LUT innovantes a été mise en œuvre. Nous proposons dans la deuxième partie la conception d'une mémoire embarquée en technologie SOT (Spin Orbit Torque), pour laquelle un brevet d'invention a été déposé. Enfin, dans la dernière partie, ce type de mémoire SOT-MRAM ainsi que d'autres de type STT-MRAM ont été intégrées dans un processeur afin d'évaluer les éventuels intérêts de ces technologies magnétiques STT et SOT dans ce type de circuit largement répandus.

Mots clés: MRAM: Magnetic Random Access Memory, technologie non volatile, LUT hybride, Spin Transfer Torque, Spin Orbit Torque, processeur volatil.