Conception sur mesure d'un FPGA durci aux radiations à base de mémoires magnétiques

Olivier Goncalves Gonçalves

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **NANO ELECTRONIQUE ET NANO TECHNOLOGIES**

Arrêté ministériel : 7 août 2006

Présentée par

« Olivier / GONÇALVES »

Thèse dirigée par **« Bernard DIENY »** et
Co-encadré **par « Guillaume PRENAT »**

préparée au sein du **SPINTEC – CEA/CNRS/UJF**
dans **l'École Doctorale EEATS : Electronique, Electrotechnique, Automatique et Traitement du Signal**

Conception sur mesure d'un FPGA durci aux radiations à base de mémoires magnétiques

Thèse soutenue publiquement le **« 19 Juin 2013 »**,
devant le jury composé de :

Mme Lorena ANGHEL
Professeur à Grenoble INP, Présidente du jury
Mr Lionel TORRES
Professeur à l'université de Montpellier II, Rapporteur
Mr Jacques Olivier KLEIN
Professeur à l'université Paris Sud, Rapporteur
Mr Bertrand GRANADO
Professeur à l'UPMC et à l'ENSEA, Examinateur
Mr Christian GAMRAT
Ingénieur chercheur au CEA/LIST, Examinateur
Mr Richard FOURNEL
Directeur R&D à STMicroelectronics, Examinateur
Mr Bernard DIENY
Ingénieur chercheur au CEA à Spintec, Directeur de thèse
Mr Guillaume PRENAT
Ingénieur chercheur au CEA à Spintec, Encadrant de thèse

Université Joseph Fourier / Université Pierre Mendès France / Université Stendhal / Université de Savoie / Grenoble INP

I. TABLE DES MATIERES

II. TABLE DES FIGURES

III. RESUME

Le but de la thèse a été de montrer que les cellules mémoires MRAM présentent de nombreux avantages pour une utilisation en tant que mémoire de configuration pour les architectures reconfigurables et en particulier les FPGAs (Field Programmable Gate Arrays). Ce type de composant est programmable et permet de concevoir un circuit numérique simplement en programmant des cellules mémoires qui définissent sa fonctionnalité. Un FPGA est principalement constitué de cellules mémoires. C'est pourquoi elles déterminent en grande partie ses caractéristiques comme sa surface ou sa consommation et influencent ses performances comme sa rapidité. Les mémoires MRAM sont composées de Jonctions Tunnel Magnétiques (JTMs) qui stockent l'information sous la forme d'une aimantation. Une JTM est composée de trois couches : deux couches de matériaux ferromagnétiques séparées par une couche isolante. Une des deux couches ferromagnétiques a une aimantation fixée dans un certaine direction (couche de référence) tandis que l'autre peut voir son aimantation changer dans deux directions (couche de stockage). Ainsi, la propagation des électrons est changée suivant que les deux aimantations sont parallèles ou antiparallèles c'est-à-dire que la résistance électrique de la jonction change suivant l'orientation relative des aimantations. Elle est faible lorsque les aimantations sont parallèles et forte lorsqu'elles sont antiparallèles. L'écriture d'une JTM consiste donc à changer l'orientation de l'aimantation de la couche de stockage tandis que la lecture consiste à déterminer si l'on a une forte ou une faible résistance.

Les atouts de la JTM font d'elle une bonne candidate pour être une mémoire dite universelle, bien que des efforts de recherche restent à accomplir. Cependant, elle a de nombreux avantages comme la non-volatilité, la résistance aux radiations, la rapidité et la faible consommation à l'écriture comparée à la mémoire Flash. En effet, la consommation d'une mémoire Flash NOR embarquée 90nm est estimée à 100pJ/bit tandis que pour une MRAM en technologie STT elle est de 2,5 pJ/bit [34]. Notons que dans le cas de la thèse, la technologie TAS a été utilisée car elle était disponible contrairement à la STT. Elle a une consommation de l'ordre de 300pJ/bit mais peut être grandement amélioré grâce au partage de la ligne de champ d'écriture. Grâce à ces avantages, on peut déjà l'utiliser dans certaines applications et en particulier dans le domaine du spatial. En effet, l'utilisation dans ce domaine permet de tirer parti de tous les avantages de la JTM en raison du fait qu'elle est intrinsèquement immune aux radiations et non-volatile. Elle permet donc de réaliser un FPGA résistant aux radiations et avec une basse consommation.

Le travail de la thèse s'est donc déroulé sur trois ans. La première année a d'abord été dédiée à l'état de l'art afin d'apprendre le fonctionnement des JTMs, l'architecture des FPGAs, les techniques de durcissement aux radiations et de basse consommation ainsi que le fonctionnement des outils utilisés en microélectronique. Au bout de la première année, un nouveau concept d'architecture de FPGA a été proposé. Les deuxième et troisième années ont été dédiées à la réalisation de cette innovation avec la recherche de la meilleure structure de circuit et la réalisation d'un circuit de base d'un FPGA ainsi que la conception puis la fabrication d'un démonstrateur. Le démonstrateur a été testé avec succès et a permis de prouver le concept. La nouvelle architecture de circuit de FPGA a permis de montrer que l'utilisation des mémoires MRAM comme mémoire de configuration de FPGA était avantageuse et en particulier pour les technologies futures.

IV. ABSTRACT

The aim of the thesis was to show that MRAM memory has many advantages for use as a configuration memory for reconfigurable architectures and especially Field Programmable Gate-Arrays (FPGAs). This type of component is programmable and allows designing a digital circuit simply by programming memory cells that define its functionality. An FPGA is thus mainly composed of memory cells. That is why they largely determine its characteristics as its surface or power consumption and affect its performance as its speed. MRAM memories are composed of Magnetic Tunnel Junctions (JTMs) which store information in the form of a magnetization. A JTM is composed of three layers: two layers of ferromagnetic material separated by an insulating layer. One of the two ferromagnetic layers has a magnetization pinned in a fixed direction (reference layer) while the other one can have its magnetization switched between two directions (storage layer). Thus, the propagation of the electrons is changed depending on whether the two magnetizations are parallel or antiparallel that is to say that the electrical resistance of the junction changes according to the orientation of the magnetizations. It is low when the magnetizations are parallel and high when antiparallel. Writing a JTM consists in changing the orientation of the magnetization of the storage layer while reading consists in determining if the resistance is high or low.

The advantages of the JTM make it a good candidate to be used as a universal memory although research efforts are still needed. However, it has many advantages such as non-volatility, radiation resistance, speed and low power writing consumption compared to the flash memory. Indeed, the consumption of an embedded 90nm NOR Flash is estimated to 100pJ/bit while 2.5 pJ / bit for a STT MRAM technology in [34]. Note that in the case of this work, the TAS technology was used because it was available unlike the STT. It has a consumption of about 300pJ/bit but can be greatly improved through the sharing of writing field line. With these advantages, we may already use it in some applications and in particular in the field of space. Indeed, its use in this area allows taking advantage of all of the benefits of JTM due to the fact that it is intrinsically immune to radiation and non-volatile. It therefore enables to make a radiation hardened and low power FPGA with new functionalities.

The work of this thesis is held over three years. The first year was dedicated to the state of the art in order to learn the mechanisms of JTMs, the architecture of FPGAs, radiation hardening and low power consumption techniques as well as the operation of the tools used in microelectronics. After the first year, a new FPGA architecture concept was proposed. The second and third years were devoted to the realization of this innovation with the search for the best circuit structure and the realization of an elementary component of a FPGA and the design and manufacture of a demonstrator. The demonstrator has been successfully tested and proved the new concept. The new circuit architecture of FPGA has shown that the use of MRAM cells as configuration memories for FPGAs was particularly advantageous for future technologies.

V. REMERCIEMENTS

J'aimerais remercier tout d'abord Guillaume Prenat qui m'a permis de faire la thèse et qui m'a aidé quotidiennement en me donnant des conseils et en me guidant. Je le remercie également de m'avoir initié au Triathlon et j'espère pouvoir être un jour meilleur que lui.

Je remercie également Alain Schuhl, Bernard Dieny et Jean-Pierre Nozières de m'avoir accueilli à Spintec et je remercie les membres du jury d'avoir accepté notre invitation.

Merci à toute l'équipe du laboratoire Spintec pour la bonne ambiance quotidienne : Rachel et Catherine pour leur aide, les doctorants et post-doctorants pour les sorties, Eric et Guillaume pour les repas distrayants à la cantine. Merci à Gregory Di Pendina pour ses conseils lors de la conception du démonstrateur et sa bonne humeur. Je souhaite également bonne chance à l'équipe design de Spintec : Guillaume, Wei, Greg et Christophe qui vient d'arriver. Je souhaite bonne chance à Virgile Javerliac et ses collègues pour la startup qu'ils sont en train de mettre en route.

Je remercie également Crocus technologie pour nous avoir donné l'occasion de fabriquer un démonstrateur et merci à l'équipe de Grenoble de nous avoir accueillis lors de la conception du circuit. Merci en particulier à Jeremy Herault et Lucien Lombard pour leur aide sur le run Crocus.

Merci aux nageurs de Spintec pour leur bonne humeur : Greg, Gilles, Emilie, Cécile et Guillaume (encore lui !).

Finalement, je tiens à remercier ma famille et en particulier mes parents pour leur soutient pendant les trois ans et demi.

VI. INTRODUCTION GENERALE

VI.1 CONTEXTE

Les FPGAs (Field-Programmable Gate Arrays) sont des architectures reconfigurables qui permettent de concevoir des circuits numériques simplement en programmant des cellules mémoires qui déterminent leur fonctionnalité. Les FPGAs sont couramment utilisés pour le prototypage ou les produits de moyenne ou petite série car ils évitent le recourt à la conception complexe et couteuse d'un composant spécifique (ASIC). A cause du coût de plus en plus élevé de la conception et la fabrication d'un ASIC, les FPGAs sont utilisés pour des volumes de produits de plus en plus grands. Il existe, par exemple, des FPGAs d'entrée de gamme comme ceux de Lattice Semiconductor de la famille iCE40 destinés aux applications mobiles. Le succès des FPGAs est non seulement lié au coût élevé des ASICs mais également le haut degré d'intégration qui permet d'implémenter des fonctions complexes en rivalisant par exemple avec les DSPs. Ce haut degré d'intégration est permis grâce aux technologies avancées. En effet, les fabricants de FPGA tentent toujours de rester dans les technologies les plus avancées afin d'augmenter la densité et la rapidité des FPGAs tout en diminuant leur consommation. Ceci est très bien illustré par les startups Achronix et Tabula qui vont faire fondre leurs FPGAs par Intel dans le procédé de fabrication 22 nm.

Cependant, les nœuds technologiques avancés font face à de nombreux défis et notamment les difficultés liées au procédé de fabrication qui augmente la dispersion des caractéristiques, l'augmentation des courants de fuite qui augmente la consommation au repos des circuits, l'augmentation de la densité d'énergie des circuits qui engendre des températures élevées et menace la durée de vie des transistors et une sensibilité aux radiations accrue. Dans le domaine des circuits, les mémoires, qui constituent la majorité de la surface d'un FPGA, sont donc sujettes à ces problèmes. Pour les résoudre, une des solutions consiste à faire appel à de nouvelles technologies de mémoire et en particulier les mémoires MRAMs. Les mémoires MRAMs sont constituées de Jonctions Tunnel Magnétiques (JTMs). Elles stockent l'information sous la forme d'une aimantation. Les JTMs sont typiquement de forme sphérique ou elliptique et sont constituées d'un empilement de trois couches. Deux couches ferromagnétiques séparées par une couche isolante. Une des deux couches ferromagnétiques a une aimantation fixée dans un certaine direction (couche de référence) tandis que l'autre peut voir la sienne changer dans deux directions (couche de stockage). Ainsi, la propagation des électrons est changée suivant que les deux aimantations sont parallèles ou antiparallèles c'est-à-dire que la résistance électrique de la jonction change suivant l'orientation des aimantations. Elle est faible lorsque les aimantations sont parallèles et forte lorsqu'elles sont antiparallèles. L'écriture d'une JTM consiste donc à changer l'orientation de l'aimantation de la couche de stockage tandis que la lecture consiste à déterminer si l'on a une forte ou une faible résistance.

Une cellule mémoire MRAM présente de nombreux avantages comparée aux cellules mémoires existantes dans les technologies classiques telles les cellules SRAM, DRAM ou Flash. Elle est non-volatile, a une bonne rapidité, une faible consommation à l'écriture (de l'ordre de la pJ/bit pour les MRAMs en technologies STT [34]), une endurance quasiment illimitée et est immune aux radiations. Sa non-volatilité permet de

concevoir des circuits basse consommation afin de limiter la consommation d'énergie. Sa vitesse d'écriture est de l'ordre de la dizaine de nanoseconde et sa consommation à l'écriture est faible comparée à celle des cellules Flash. En effet, la consommation d'une mémoire Flash NOR embarquée 90nm est estimée à 100pJ/bit tandis que pour une MRAM en technologie STT et de 65nm, elle est de 2,5 pJ/bit [34]. Les technologies 90 et 65 nm étant proche, on peut affirmer que la Flash NOR consomme plus que la STT. Notons que dans le cas de la thèse, la technologie TAS 130nm a été utilisée car elle était disponible contrairement à la STT. Elle a une consommation de l'ordre de 300pJ/bit mais peut être grandement amélioré grâce au partage de la ligne de champ d'écriture. Son endurance quasiment illimitée est un grand avantage par rapport à la plupart des mémoires non-volatiles comme la flash ou même d'autres nouvelles technologies de mémoire telles que les PCRAMS, les Redox RAMs ou les FeRAMs qui ont des endurances limitées. Ensuite, grâce à leur immunité intrinsèque aux radiations, elles permettent de fiabiliser les systèmes dans lesquelles elles sont intégrées. Un autre atout est le fait que des mémoires MRAMs sont déjà commercialisées par la société Everspin (spin-off de Freescale). Elle est la seule à commercialiser des MRAMs mais d'autres startups sont sur le point d'en commercialiser comme Crocus-Technology, une spin-off du laboratoire Spintec du CEA. La mémoire MRAM a donc déjà des débouchés dans l'industrie ce qui prouve l'intérêt grandissant pour cette nouvelle technologie. Grâce à tous ces avantages, la MRAM est une des candidates pour être la mémoire universelle et remplacer ainsi les mémoires d'aujourd'hui. Cependant des efforts restent à accomplir notamment en termes de vitesse et de consommation à l'écriture pour pouvoir concurrencer en particulier la SRAM. D'autre part, le procédé de fabrication nécessite encore du développement car la technologie n'est pas encore mature. Une nouvelle technologie de JTM, encore en phase de recherche, pourrait rivaliser avec la SRAM grâce à sa vitesse d'écriture de quelques centaines de picosecondes, c'est la JTM dite perpendiculaire.

Bien que les MRAMs ne soient vendues que sous forme de bloc mémoire, des recherches sont en cours pour les intégrer dans des circuits logiques, afin de tirer partie notamment leur non-volatilité pour faire diminuer la consommation statique des circuits lorsqu'ils sont inutilisés, répondant ainsi à la problématique de l'augmentation des courants de fuite dans les technologies avancées. Il est également possible d'utiliser le fait que les JTMs sont immunes aux radiations afin d'augmenter la fiabilité du circuit. Les JTMs peuvent être intégrées dans des circuits logiques reconfigurables, comme par exemple les FPGAs, et en particulier remplacer les cellules mémoires de configuration SRAM associée à la Flash par des circuits à bases de JTMs. Ainsi, différentes architectures de Look-Up Tables (LUTs), le circuit de base d'un FPGA, permettant d'accomplir une fonction numérique simple, ont été conçues avec des JTMs. Hassoun et Black [5] ont été parmi les premiers à proposer un circuit permettant de faire des calculs grâce à des composants magnétiques, les vannes de spin. En modifiant les données stockées dans les vannes de spin, on pouvait changer la fonction implémentée. Ensuite, des architectures ont été proposées afin de rendre les cellules mémoires SRAM des FPGAs non-volatiles. La plus classique est la cellule de Black et Dass [8] qui consiste à associer une SRAM et deux JTMs dont les états sont complémentaires pour rendre la SRAM non-volatile. Une variante de cette architecture permet de rendre la cellule SRAM tolérante aux radiations comme proposé dans [10]. D'autres structures de LUT ont été proposées dans la littérature comme dans [12], où une cellule de configuration est simplement composée de deux JTMs dont les états sont complémentaires. La lecture se fait grâce au passage d'un courant à travers les jonctions et en comparant la différence de tension grâce à la différence des résistances des JTMs. Le premier FPGA entier a été

proposé dans le cadre d'un projet ANR, le projet SPIN, avec pour but la fabrication d'un capteur à base de vannes de spin et d'un FPGA MRAM complet. Le FPGA MRAM a été conçu conjointement par le LIRMM et la startup MENTA qui ont converti leur FPGA embarqué SRAM en un FPGA MRAM.

La thèse a été effectuée au laboratoire Spintec au CEA de Grenoble. Elle s'est déroulée dans l'équipe conception du laboratoire dirigée par Guillaume Prenat, également co-encadrant de la thèse. Le laboratoire Spintec effectue des recherches sur la spintronique. C'est au laboratoire Spintec qu'une nouvelle technologie de JTM a été inventée et développée : la JTM à écriture assistée thermique appelée TAS pour Thermally Assisted Switching. La problématique consistant à réaliser un FPGA à partir de cellules mémoires MRAM s'est posée dès l'année 2003 avec la thèse de Virgile Javerliac. Il a mis en place les outils nécessaires à la conception d'un circuit hybride CMOS/Magnétique et en particulier le développement du modèle de JTM nécessaire pour les simulations de circuit contenant des JTMs. Le développement du modèle a ensuite été continué par Mourad El Baraji, Guillaume Prenat, Wei Guo et Abdelilah Mejdoubi. C'est grâce à ce modèle, et donc à leur travail, que la conception des circuits présentés dans cette thèse a été rendue possible. Un autre sujet a été développé par Grégory Di Pendina dans le cadre d'une thèse, qui s'est déroulée en même temps que la mienne. Elle a eu pour résultat un flot de conception permettant de développer des circuits numériques en technologie hybride CMOS/magnétique et une architecture innovante de bascule. Tous ces sujets de recherches prouvent l'intérêt grandissant pour la mémoire MRAM et c'est dans ce contexte que s'est déroulée la thèse.

VI.2 OBJECTIFS

Le but de la thèse est de montrer l'intérêt des JTMs en tant qu'éléments de mémoire de configuration d'un FPGA destiné au domaine du spatial. Plus précisément, le but était de trouver une application dans laquelle l'utilisation de la JTM amenait un gain élevé en termes de densité, de consommation, de vitesse ou de fiabilité et de trouver une architecture adaptée afin de tirer avantage de tous les atouts des JTMs. Le sujet initial visait une application dans le domaine du spatial et cela s'est confirmé à la suite de la recherche bibliographique. En effet, les JTMs étant immunes aux radiations, c'est le principal atout qui fait défaut aux autres technologies de mémoire et notamment SRAM et Flash. C'est donc dans ce domaine que les JTMs peuvent se différencier.

Un des autres objectifs de la thèse était de réaliser un démonstrateur pour avoir un impact plus fort. Cependant, l'accès à une technologie CMOS/Magnétique n'était pas chose aisée, compte tenu du manque de maturité du procédé. Des démonstrateurs avaient déjà été réalisés dans le cadre de projets de recherche ANR, mais n'ont pas abouti à des circuits fonctionnels. . De plus, la thèse ne s'inscrivait pas dans un projet, donc il était plus difficile de mettre en place une telle technologie. L'un des choix à faire lors de la thèse a donc été de savoir dans quelle technologie concevoir les circuits. Le choix s'est alors porté sur la technologie CMOS 130 nm de ST avec des JTMs en technologie TAS car elle présentait le plus de chance d'aboutir à un démonstrateur étant donné que c'est dans cette technologie que les démonstrateurs précédents avaient été réalisés. Finalement, une opportunité s'est présentée grâce à Crocus-Technology, la startup lancée par Spintec, qui nous a permis de placer deux circuits simples une de leurs tranches de test.

Le but principal de la thèse était donc la conception d'un FPGA entier à base de JTMs avec une architecture spécifique permettant de tirer avantage de tous les atouts de la technologie magnétique. Cela impliquait de concevoir une tuile, c'est-à-dire un circuit contenant quelques LUTs et des interconnexions programmables constituant ainsi un « motif » élémentaire. Plusieurs tuiles sont ensuite connectées directement les unes aux autres afin de former une matrice de N x M tuiles, formant ainsi un FPGA entier.

VI.3 PLAN DU MANUSCRIT

Le manuscrit est divisé en quatre grandes parties. La première est dédiée à l'état de l'art qui comprend, dans le premier chapitre, la description d'un FPGA en général en se plaçant du point de vue du concepteur de FPGA, c'est-à-dire que sont décrites les différentes architectures possibles avec leurs avantages et inconvénients en termes de surface, rapidité et consommation. On décrit notamment comment choisir la taille d'une LUT ou le nombre d'interconnexions. Ensuite, les différentes mémoires de configuration, utilisées dans les FPGAs commerciaux, sont décrites comme les mémoires SRAMs, Flash et antifuse. Les deux chapitres suivants décrivent des techniques pour diminuer la consommation d'énergie des circuits et pour durcir les circuits aux radiations. On enchaine ensuite sur les mémoires MRAM et la description des JTMs utilisées comme nouvelles cellules mémoires de configuration dans le troisième chapitre. Le quatrième chapitre décrit les FPGAs ou LUTs à base de JTMs décrits dans la littérature ainsi que des FPGAs basés sur d'autres nouvelles technologies de mémoire comme les FeRAMs et PCRAMs. Cette partie consacrée à l'état de l'art forme plus de la moitié du manuscrit en raison des différents domaines impliqués dans la conception d'un FPGA à base de MRAM. De plus les techniques existantes de réduction de la consommation et de durcissement aux radiations doivent être connues afin de comprendre les avantages des MRAMs.

La deuxième grande partie comprend la description théorique de la nouvelle architecture de FPGA à base de JTMs. On présente d'abord la structure des LUTs et des interconnexions. Ensuite, on présente la structure des circuits de configuration qui forment la principale innovation. La fin de ce chapitre est dédiée à des considérations sur la fiabilité et les avantages de la nouvelle architecture de circuit de configuration.

La troisième partie décrit la tuile qui a été implémentée durant la thèse dans la technologie 130 nm de ST et dont les circuits ont été simulés. Les différents circuits de base sont décrits ainsi que des données sur leur densité et leur consommation. La fin de ce chapitre résume les caractéristiques de la nouvelle architecture du FPGA ainsi qu'une évaluation de ses avantages en fonction des technologies de circuit utilisées pour le concevoir.

La quatrième partie est dédiée à la description du démonstrateur et des résultats des tests obtenus. Les résultats en termes de consommation, densité et surtout fonctionnalité sont présentés.

La conclusion et les perspectives terminent le manuscrit et permettent de faire un bilan de la nouvelle architecture ainsi que les performances que l'on peut en attendre grâce à l'évolution des techniques de fabrication des JTMs et la maturation de leur technologie.

VII. GLOSSAIRE

FPGA (Field Programmable gate-Array) : composant numérique programmable permettant de concevoir tout type de circuit numérique (additionneur, multiplieur, processeur, …) simplement en programmant des cellules mémoires permettant de définir la fonctionnalité du circuit.

LUT (Look-Up table) : c'est le composant élémentaire d'un FPGA. Ce circuit permet d'implémenter une fonction numérique simple (fonction NAND, NOR, XOR, …) dans un FPGA. L'assemblage de plusieurs LUTs permet de réaliser une fonction complexe comme un additionneur par exemple.

JTM (Jonction Tunnel Magnétique) : cellule mémoire permettant de stocker l'information sous forme d'une aimantation. C'est un composant de taille nanométrique composé de trois couches : deux couches de matériaux ferromagnétiques séparées par une couche isolante. Une des deux couches ferromagnétiques a une aimantation fixée dans un certaine direction (couche de référence) tandis que l'autre peut voir son aimantation changer dans deux directions (couche de stockage). Ainsi sa résistance électrique change suivant l'orientation des aimantations. Elle est faible lorsque les aimantations sont parallèles et forte lorsqu'elles sont antiparallèles.

MRAM (Magnetic Random Access Memory) : nouvelle technologie de mémoire magnétique. Elle est composée de JTMs comme cellules de base.

LET (Linear Energy Transfer) : désigne la quantité d'énergie déposée par une particule en traversant un composant. Son unité est le cm^2Mev/mg et dépend du matériau traversé.

ASIC (Application-Specific Integrated Circuit) : circuit intégré spécialisé. Les fonctionnalités implémentées dans ce type de composant sont spécifiques à une application et faites sur-mesure. Concevoir un ASIC nécessite de réaliser un circuit et donc un jeu de masques spécifique.

TAS (Thermally Assisted Switching) : technique d'écriture de JTM assistée thermiquement. C'est un nouveau mode d'écriture de mémoire MRAM développée au laboratoire Spintec.

STT (Spin Transfert Torque) : nouvelle technologie de mémoire MRAM dont l'écriture se fait par transfert de couple par spin.

FIMS (Field Induced Magnetic Switching) : méthode d'écriture par l'application d'un champ magnétique extérieur.

Wafer : tranche de silicium sur laquelle sont réalisés des circuits intégrés.

ANR (Agence Nationale de la Recherche) : agence française de financement de la recherche. Des projets de recherche comme SPIN ou CILOMAG ont été financés par l'ANR.

CMOS (Complementary Metal Oxide Semiconductor) : technologie de fabrication de circuits intégrés.

Bitstream : dans le cadre de cette thèse, le bitstream désigne le fichier stockant la configuration complète d'un FPGA.

TMR (Triple Modular Redundancy) : dans un contexte de fiabilisation de circuits numériques, la TMR désigne la technique de durcissement aux radiations consistant à dupliquer un circuit trois fois et faire un vote majoritaire à leurs sorties afin de masquer une éventuelle erreur dans l'un des circuits.

TMR (Tunneling Magneto-Resistance) : dans le contexte des JTMs et des nanotechnologies magnétiques en générales, le ratio de TMR permet d'indiquer la différence de résistance entre l'état parallèle et antiparallèle exprimée en un pourcentage de résistance nominale.

RA : paramètre de la JTM qui est le produit de la surface par sa résistance et qui permet de chiffrer la résistance de la barrière tunnel indépendamment de sa surface. Il est exprimé en $\Omega.\mu m^2$.

SAF (Synthetic Anti-Ferromagnetic) : c'est un antiferromagnétique synthétique dont la structure est composée de deux couches ferromagnétiques avec des aimantations opposées séparées par une couche de couplage. Permet notamment de stabiliser l'aimantation d'une couche de JTM.

Crocus-Technology : startup créée par le laboratoire Spintec du CEA qui vise à commercialiser la technologie TAS.

Everspin : spin-off de la société Freescale, commercialisant des mémoires MRAM à base de JTM en technologie FIMS et STT.

Rad : unité de mesure de la dose de radiation absorbée par un matériau. Elle dépend du matériau.

I. ETAT DE L'ART

VIII. Les FPGAs

Un FPGA (Field-Programmable Gate Array) est un composant électronique programmable. Il permet de synthétiser n'importe quel circuit numérique (multiplieur, additionneur, processeur, …), pourvu que le FPGA soit assez complexe pour le contenir. Un FPGA est composé de centaines de cellules logiques réalisant chacune une fonction logique élémentaire programmable. Ces cellules sont organisées en une matrice et reliées entre elles par un vaste réseau d'interconnexions programmables. Une fonction logique complexe est implémentée en programmant les fonctions élémentaires de chaque cellule logique et en les reliant entre elles en programmant chaque interconnexion. Elle communique avec les composants électroniques extérieurs via des plots d'entrée/sortie également programmables.

Cette flexibilité est le principal avantage des FPGAs sur les ASICs. Elle permet de réduire les coûts de fabrication d'un circuit : le coût de développement d'un ASIC peut être de plusieurs millions d'euros tandis qu'un FPGA nécessitera, au plus, quelques milliers d'euros. De plus, le temps nécessaire pour développer le circuit sera plus court pour un FPGA. Cependant cette flexibilité se paie en termes de surface, de consommation et de vitesse. En effet, pour une même fonction logique, un FPGA nécessite entre 20 et 35 fois plus de surface que la même fonction implémentée sur un ASIC [1]. Elle est également 3 à 4 fois moins rapide et consomme environ 10 fois plus qu'un ASIC [1]. Un FPGA est plus cher à l'unité qu'un ASIC en raison de sa surface plus grande. C'est pour ces raisons que les FPGAs sont utilisés pour le prototypage ou les productions en faibles ou moyens volumes tandis que les ASICs doivent être produits à des millions d'exemplaires pour être rentables.

La première partie est une introduction aux FPGAs. Les parties suivantes décrivent plus précisément chaque constituant d'un FPGA.

VIII.1 Architectures FPGA

Un FPGA est constitué principalement de quatre composants :

- **les blocs logiques configurables**
- **les cellules mémoires**
- **le réseau d'interconnexions programmables**
- **les blocs d'entrée/sortie**

Les blocs logiques programmables réalisent une fonction logique élémentaire (exemple : une fonction NAND à 3 entrées). En assemblant plusieurs de ces fonctions élémentaires, on obtient une fonction complexe. On peut, par exemple, réaliser un additionneur, un multiplieur ou un processeur. Le constituant principal d'un bloc logique est la LUT (Look-Up Table). Elle est constituée de plusieurs entrées (k entrées) et d'une sortie et permet de réaliser n'importe quelle fonction logique à k entrées et une sortie.

Chaque LUT sera programmée de façon à réaliser la sous-fonction qui lui a été attribuée. L'architecture des blocs logiques programmables doit être choisie de façon à pouvoir réaliser une fonction complexe avec le minimum de surface (donc avec un minimum de blocs logiques) et le plus rapidement possible. La fonction réalisée par chaque bloc logique est programmable, c'est-à-dire que l'on va écrire la configuration

du circuit dans des cellules mémoires. Ces cellules mémoires vont alors activer différents transistors pour accomplir la fonction désirée. Le choix des cellules mémoires est important car il détermine en partie les performances et les caractéristiques du FPGA. Les blocs logiques sont assemblés entre eux grâce à un vaste réseau d'interconnexions programmables. La configuration de ce réseau est également importante. En effet, le réseau doit être assez complexe pour pouvoir implémenter n'importe quelle fonction avec un minimum de surface et il ne doit pas utiliser une surface trop grande afin de limiter la consommation et la perte de rapidité. Il y a donc un compromis à faire entre flexibilité (surface minimum) et performances (rapidité et consommation). Puis pour communiquer avec les composants extérieurs, le FPGA est relié à des blocs d'entrée/sortie programmables. Pour une flexibilité maximum, un bloc d'entrée/sortie doit pouvoir être configuré en entrée, en sortie ou en trois-états et accepter tous les standards d'entrées/sorties (CMOS, TTL, LVDS, …).

La programmation du FPGA consiste à écrire dans les cellules mémoires de configuration. Pour déterminer la valeur à stocker dans chaque cellule mémoire, le programmeur est aidé d'un logiciel de développement. A partir d'un fichier texte écrit dans un langage de description de circuits numériques ou d'un schéma, le logiciel effectue la synthèse du circuit à implémenter. L'étape de synthèse consiste à convertir les fichiers textes (décrivant le circuit) en un fichier RTL (Register Transfer Level) (qui décrit le circuit au niveau porte logique). Par exemple, considérons la fonction logique de la Figure 1.

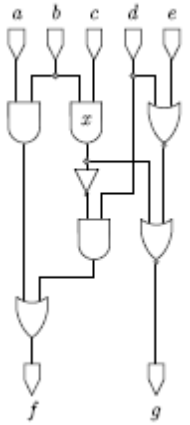


Figure 1 : fonction logique représentée par les portes logiques [11]

Elle possède 5 entrées et 2 sorties. Dans un ASIC, elle aurait été implémentée en assemblant des portes logiques élémentaires (comme la NAND ou un inverseur). Dans un FPGA, on réalise cette fonction en assemblant plusieurs LUTs. Prenons l'exemple d'une LUT à 4 entrées. On ne peut pas réaliser ce circuit avec une seule LUT car il possède 5 entrées et 2 sorties tandis que la LUT n'a que 4 entrées et une sortie. On implémente ce circuit en divisant le circuit en sous-groupes chacun pouvant être implémenté par une LUT comme dans la Figure 2. Cette étape est appelé « mapping ». On obtient alors la Figure 3.

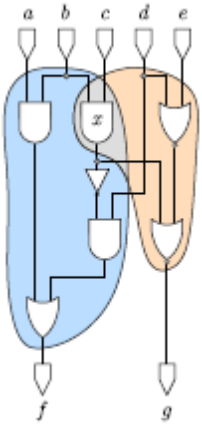


Figure 2 : découpage en sous-groupes [11]

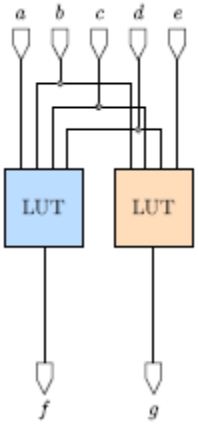


Figure 3 : circuit qui sera implémenté dans le FPGA pour réaliser la fonction [11]

L'étape suivante est le placement-routage. Cette étape a pour objet de générer le fichier de configuration qui enregistre la valeur (0 ou 1) de chaque cellule mémoire. Elle se déroule en deux temps. D'abord, on place chaque porte logique du circuit dans le FPGA cible. On détermine donc la configuration de chaque bloc logique (donc des LUTs). Ensuite, on fait le routage, c'est-à-dire que l'on relie les blocs logiques entre eux grâce aux interconnexions programmables. Au final on obtient la configuration de chaque cellule mémoire de configuration : le fichier généré est appelé le bitstream. Pour finir, on charge la configuration dans le FPGA à l'aide d'un programmateur vendu par le fabriquant du FPGA. L'algorithme de placement-routage est également important car il doit optimiser l'espace et la rapidité du circuit implémenté dans le FPGA.

Les sous-parties qui suivent, décrivent chacun de ces quatre composants en détails. La dernière sous-partie décrit les composants que l'on peut trouver dans les FPGAs modernes.

VIII.2 Bloc logique configurable

Un bloc logique élémentaire (CLB pour configurable Logic Block) permet d'implémenter une fonction logique simple faisant partie d'un circuit réalisant une fonction plus complexe. Il est principalement composé d'une LUT (Look-Up Table). C'est ce composant qui permet d'implémenter une fonction élémentaire. Une LUT est caractérisée par le nombre d'entrées k : il correspond au nombre d'entrées de la

fonction élémentaire. Par exemple, une LUT-2 (donc k = 2) possède 2 entrées et permet de programmer n'importe quelle fonction logique possédant 2 entrées et une sortie (comme une porte OU).

Concrètement une LUT est composée d'un multiplexeur possédant 2^k entrées avec k bits de sélection et une sortie. Les entrées de la fonction sont les k bits de sélection, la sortie de la fonction est la sortie du multiplexeur et les différentes valeurs de la fonction sont placées en entrée du multiplexeur.

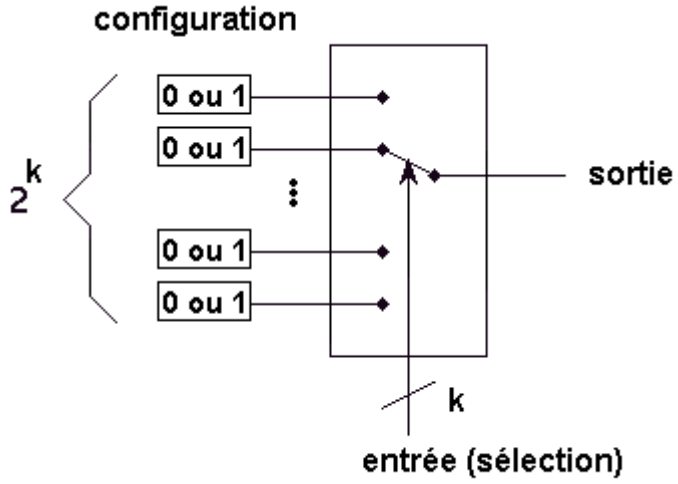
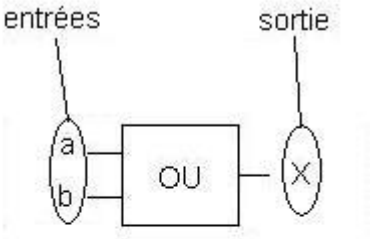


Figure 4 : LUT à k entrées

Prenons l'exemple d'une LUT à 2 entrées réalisant la fonction OU. Dans une fonction OU, la sortie est à 1 si au moins une de ses entrées est à 1. Donc la sortie est à 0 si et seulement si ses deux entrées sont à 0. On obtient donc la table de vérité ci-dessous (Figure 5). Le schéma de la LUT réalisant la fonction OU est présenté en Figure 6.



b	a	X
0	0	0
0	1	1
1	0	1
1	1	1

Figure 5 : table de vérité Porte OU

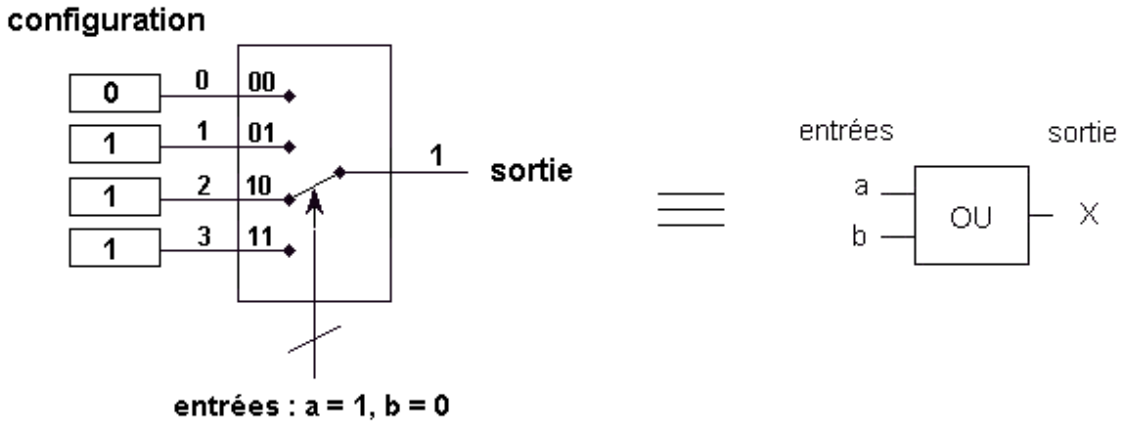


Figure 6 : programmation d'une porte OU à l'aide d'une LUT-2

Comme le montre le schéma, des cellules mémoires sont placées aux entrées du multiplexeur. Ce sont ces cellules mémoires qui sont programmées pour réaliser n'importe quelle fonction à deux entrées. Lors de la programmation, on recopie la table de vérité (table de vérité précédente) dans les cellules mémoires. Ensuite, lors de la phase de fonctionnement, on applique l'entrée de la fonction sur les bits de sélection. Ces bits vont alors sélectionner l'entrée qui va être dirigée vers la sortie. Dans notre exemple, on place en entrée de la fonction la valeur « 10 ». Le multiplexeur va donc véhiculer en sortie la valeur qui est sur la ligne n°2. Dans notre cas, la sortie sera donc « 1 ».

Le principal choix à faire lors de la conception d'un bloc logique est la taille k de la LUT. Pour cela, il faut prendre en compte deux contraintes :

Plus k augmente, moins il faudra de blocs logiques pour faire une fonction complexe. Donc le design prendra de moins en moins de surface (jusqu'à un certain point expliqué ultérieurement). De plus, le chemin critique sera diminué car il y aura moins de blocs logiques sur le chemin critique.

Plus k augmente, plus le nombre de cellules mémoires augmente (de façon exponentielle). Cela impliquera un nombre de lignes de plus en plus élevé pour interconnecter toutes les entrées des blocs logiques. De plus, le délai de la LUT augmente ce qui diminue la rapidité du bloc logique. Finalement, plus le bloc logique est grand plus il consomme.

Il faut donc faire un compromis entre surface, rapidité et consommation. Concernant la consommation, elle dépend du nombre de LUT implémentées et du nombre d'interconnexions utilisées ainsi que de leur longueur. De plus, la consommation dynamique dépend également de l'application. L'estimation de la consommation est donc complexe et ne sera pas développée dans le cadre de cette thèse.

Etudions donc le compromis en surface. Il faut prendre en compte deux effets qui interviennent dans la surface d'un bloc logique. D'une part, plus k est grand, plus le nombre de blocs logiques pour implémenter un circuit sera réduit. Mais d'autre part, plus k est grand, plus le nombre de cellules mémoires sera élevé (augmente exponentiellement : 2^k). De plus, il faudra alors complexifier le réseau d'interconnexions pour pouvoir connecter toutes ses entrées entre elles. Ce problème a déjà été étudié dans la littérature, notamment dans [9]. La figure suivante représente la surface d'un bloc logique en fonction du nombre d'entrée k d'une LUT. La surface est évaluée par rapport à la surface d'un transistor de taille minimum. La taille d'un transistor est normalisée par rapport à la taille d'un transistor minimum. De cette façon, la

comparaison est indépendante du nœud technologique. La surface affichée comprend la surface d'un bloc logique et la surface du réseau d'interconnexions nécessaires pour router un bloc logique. La deuxième courbe montre le nombre moyen de blocs logiques (cluster) nécessaires pour implémenter 28 circuits de test. Chaque bloc logique contient une seule LUT. Pour obtenir la surface totale moyenne d'un circuit, il suffit de multiplier les deux courbes précédentes. On obtient donc la Figure 8.

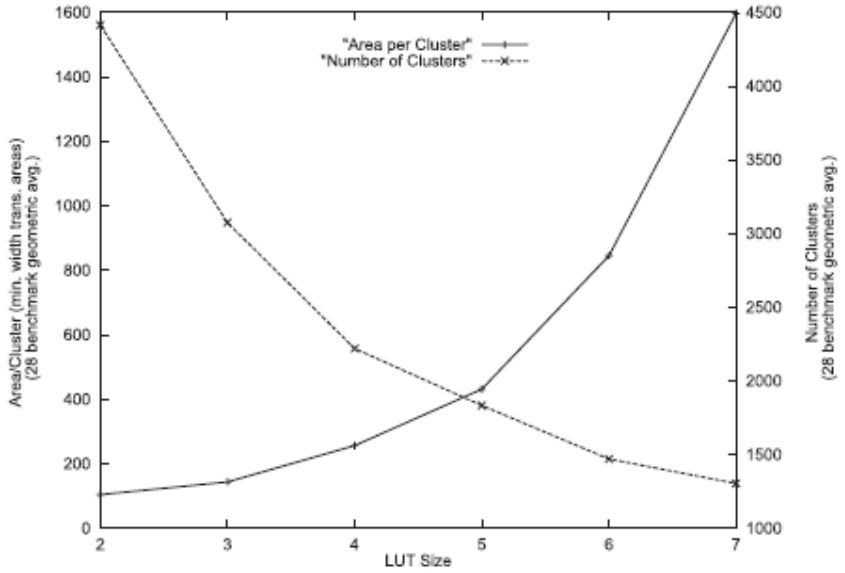


Figure 7 : taille d'un bloc logique et nombre de blocs logiques en fonction de la taille d'une LUT [9]

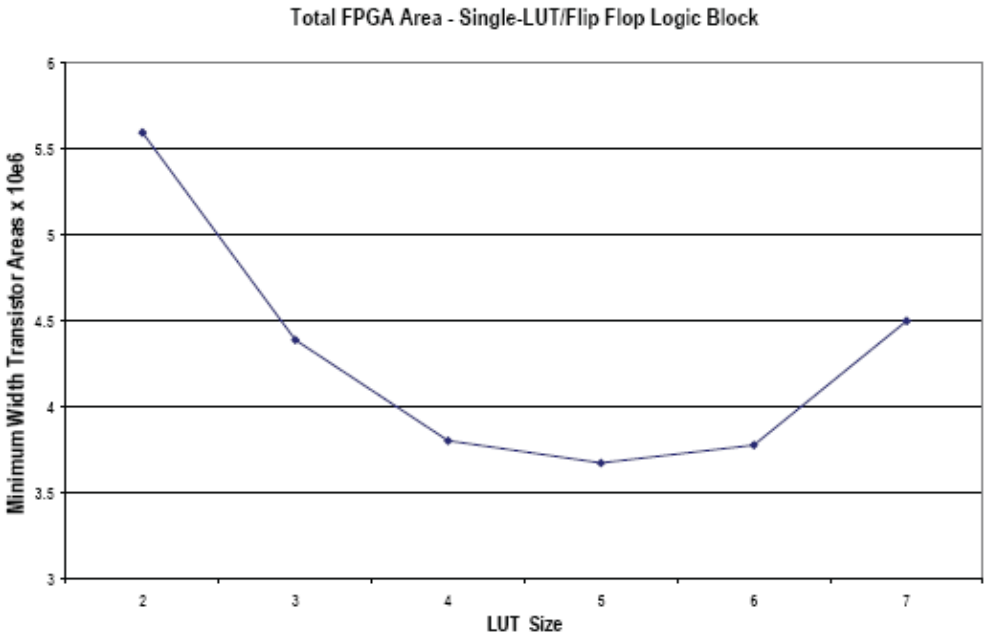


Figure 8 : surface totale moyenne d'un design en fonction de la taille d'une LUT [9]

Dans cette étude, on voit que la taille optimale d'une LUT est de 5. Dans les FPGAs modernes, les blocs logiques contiennent plusieurs LUTs qui partagent les mêmes entrées. Elles sont assemblées d'une façon plus ou moins complexe par des interconnexions internes pour former un bloc logique optimal. L'augmentation du

nombre de clusters engendre une plus faible augmentation de la surface (quadratiquement au lieu d'exponentiellement [5]). Ceci est dû au fait que les entrées sont partagées par plusieurs LUTs. Il y a donc moins d'entrées et donc le réseau de routage est moins complexe (par rapport à un bloc contenant n LUT où chaque LUT possède ses propres entrées).

Pour finir, étudions le compromis en rapidité. Il faut prendre en compte deux effets. D'une part, plus la taille k d'une LUT augmente, plus le nombre de blocs logiques nécessaires pour implémenter un circuit diminue. Le chemin critique demande donc moins de blocs logiques donc le délai du chemin critique est diminué. D'autre part, plus k augmente, plus la surface du bloc logique augmente. Le délai d'un bloc logique est donc plus grand. Ce problème a également été étudié dans [9]. La figure suivante montre le nombre moyen de blocs logiques sur le chemin critique (moyenne sur 28 circuits tests) et le délai d'un bloc logique en fonction de la taille k d'une LUT.

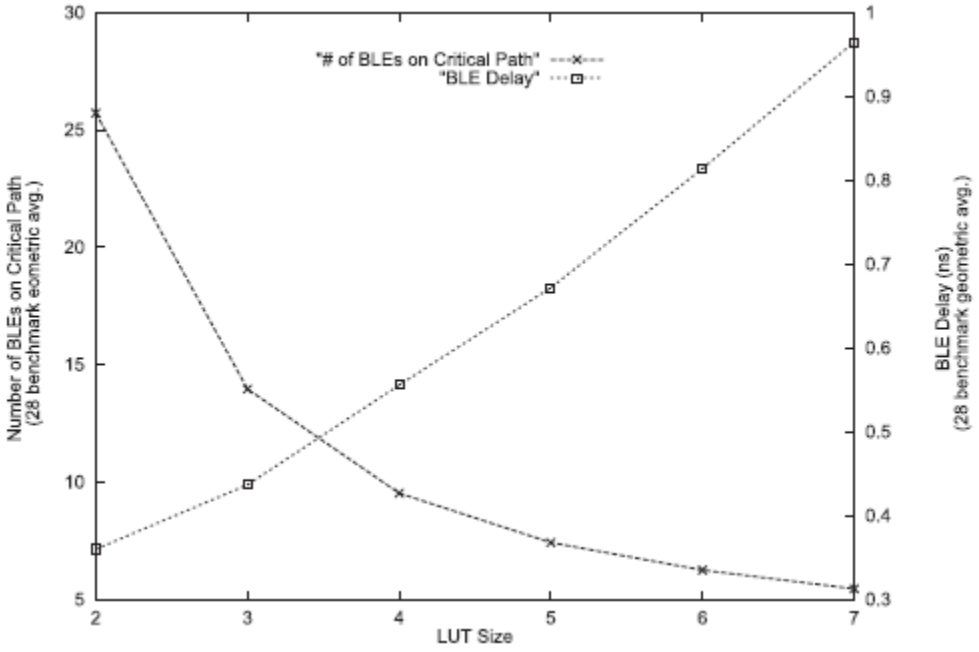


Figure 9 : nombre de blocs sur le chemin critique et délais par bloc en fonction de la taille k d'une LUT [9]

Le délai du chemin critique est obtenu en multipliant ces deux courbes. On obtient alors la figure suivante. L'étude du chemin critique a été faite pour des blocs contenant plusieurs LUTs (de 1 à 10 LUTs par bloc logique).

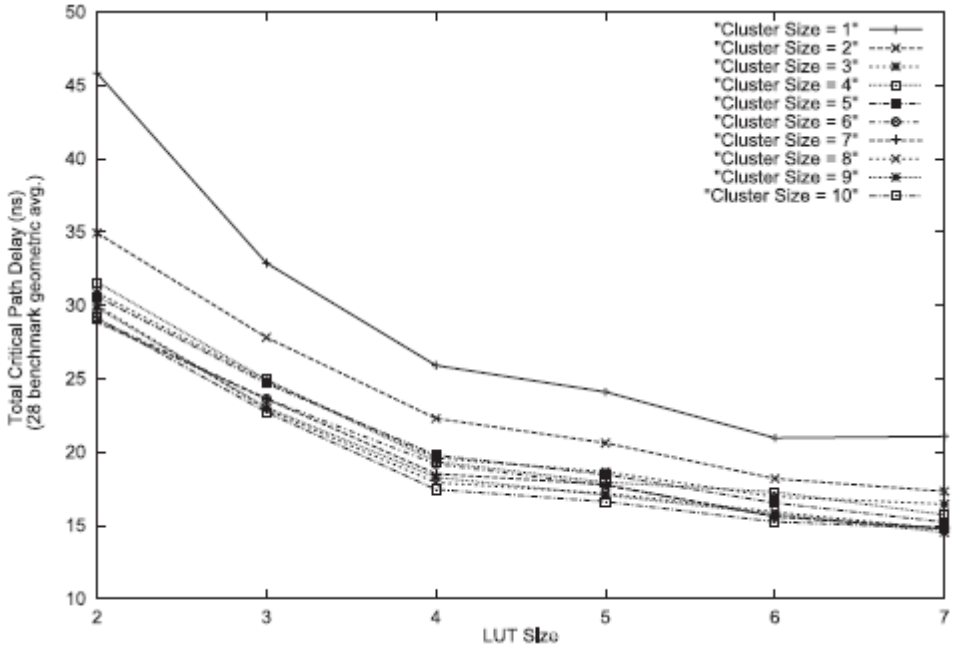


Figure 10 : délai du chemin critique en fonction de la taille k d'une LUT [9]

On voit que le chemin critique est de plus en plus faible. Le gain en rapidité est significatif jusqu'à une taille k de 6 et pour un nombre de LUTs de 3.

Les blocs logiques et les interconnexions sont programmables. La configuration du circuit est stockée dans des cellules mémoires. Elles sont présentes dans les LUTs et déterminent les fonctions réalisées par les différentes LUTs. Elles sont présentes dans le réseau d'interconnexions où elles activent/désactivent des transistors pour établir des connexions entre deux lignes. Le choix de la technologie est important car il détermine en partie les performances et les caractéristiques du FPGA (surface, rapidité, consommation). La partie suivante décrit donc les principales technologies de cellules mémoire utilisées dans les FPGAs.

Dans le cadre de cette thèse, la tuile conçue a été réalisée avec des LUTs à quatre entrées et quatre LUTs par tuile. C'est le bon compromis entre performance et complexité. En effet, une LUT avec 5 entrées aurait été plus longue à concevoir car le nombre de JTMs nécessaires aurait augmenté et le nombre d'interconnexions également.

VIII.3 Technologies de mémoire

Les FPGAs étant principalement constitués de cellules mémoire, il est important de connaitre les caractéristiques de ces mémoires pour comprendre les caractéristiques finales du FPGA. Il existe différentes catégories de FPGAs en fonction de la technologie de la mémoire de configuration. Trois types de mémoires sont utilisés dans les FPGAs du commerce : antifusible, FLASH et SRAM. Les paragraphes suivants décrivent donc ces mémoires.

La technologie la plus utilisée est la cellule SRAM. Elle est notamment utilisée dans les FPGAs de Xilinx, Altera et Lattice. La Figure 11 suivante montre le schéma d'une cellule SRAM.

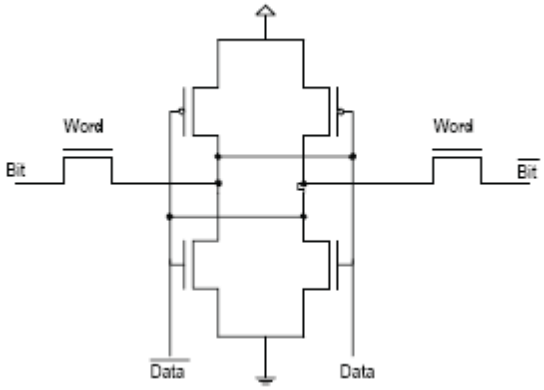


Figure 11 : cellule SRAM [5]

Elle est composée de six transistors ce qui est une taille élevée comparée aux autres technologies de mémoire. Mais le procédé de fabrication est le procédé standard CMOS. Ces mémoires peuvent donc bénéficier de la réduction de taille des transistors et ainsi être plus rapide, avoir une consommation dynamique plus faible (mais des courants de fuite et donc une consommation statique plus élevés). La cellule SRAM a également l'avantage de pouvoir être lue et écrite un nombre infini de fois.

En plus de la taille de la cellule, la SRAM a l'inconvénient d'être volatile, c'est-à-dire qu'elle perd sa donnée lorsqu'elle est hors tension. Il faut donc ajouter une mémoire annexe (une mémoire FLASH) pour stoker la configuration et la charger lors de chaque mise sous tension du FPGA. Le fait de charger la configuration à chaque mise sous tension est un risque pour la confidentialité des données de configuration. Les données peuvent être interceptées durant le transfert. Cependant, les FPGAs modernes possèdent des circuits de cryptage des données, ce qui réduit le risque.

La mémoire FLASH est une autre technologie de mémoire utilisée principalement dans les FPGAs de chez Actel. Une cellule mémoire Flash est principalement composée d'un transistor avec une grille flottante, dont l'état « chargé » ou « non-chargé » modifie la tension de seuil. Cette valeur de tension de seuil détermine la valeur stockée dans la cellule. Dans le cas des FPGAs de Actel, la cellule est composée d'un seul transistor pour la programmation et d'un transistor utilisé comme interrupteur.

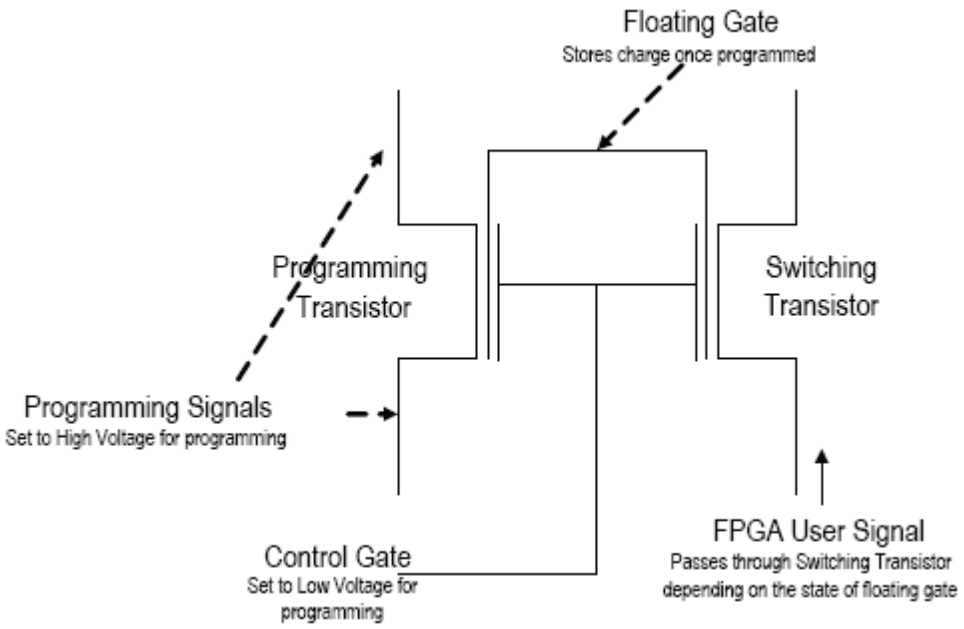


Figure 12 : cellule Flash connectée à son transistor d'interconnexion dans un FPGA flash d'Actel [2]

Elle nécessite donc moins de surface que la cellule SRAM. Il faut ajouter un circuit pour la programmation. Mais le surcoût en surface reste plus bas que la SRAM car plusieurs cellules FLASH peuvent partager le circuit de programmation pour être programmées ensemble. Cette technologie a en plus l'avantage d'être non-volatile. Le circuit est donc fonctionnel dès la mise sous tension du FPGA et ne nécessite pas de mémoire annexe. Cependant la FLASH a un nombre de cycles de programmation limité de l'ordre de quelques 100 000 cycles. Les FPGAs Flash de chez Actel ne peuvent pas être reprogrammée plus de 500 à 1000 fois [2]. Ce faible nombre de cycle est dû au fait que les FPGAs Flash de Actel sont destinés à des applications, entre autres, dans le domaine du spatial et nécessitent donc une fiabilité élevée des cellules mémoire Flash au détriment de leur endurance. De plus, les FPGAs Flash nécessitent un procédé CMOS qui inclut un procédé FLASH en option.

La dernière technologie de mémoire utilisée dans les FPGAs est la technologie antifusible. Elle est présente dans les FPGAs de chez Actel. Une cellule antifusible ne représente aucune surface de silicium. En effet, elle est composée d'une couche de matériau isolant (comme de l'oxyde de silicium) déposée entre deux couches de métal. Elle permet d'établir une connexion entre deux couches de métal. La programmation se fait en appliquant un fort courant dans la cellule, ce qui rend la cellule passante (résistance entre 20 et 100 ohms [3]). Pour connecter deux lignes de métal, il faut donc appliquer un fort courant dans la cellule. Pour les laisser déconnectées, il suffit de ne pas appliquer de courant. Cette technologie nécessite donc un circuit supplémentaire pour générer les forts courants lors de la programmation. Mais en combinant plusieurs cellules mémoires, le surcoût en silicium est limité. L'un des principaux inconvénients est le fait que les cellules ne peuvent être programmées qu'une seule fois. De plus, on ne peut pas programmer le FPGA sur son circuit imprimé. Il faut un équipement spécial pour le programmer. L'autre inconvénient est le process CMOS non-standard qui doit prendre en compte la technologie antifusible. Le procédé n'est pas très développé et il a été prouvé que l'on ne pouvait pas améliorer le procédé en dessous de la technologie 150

nm [5], cependant, des mémoires résistives émergeantes pourront à terme remplacer les cellules antifuse au delà du nœud 150 nm.

VIII.4 Réseau d'interconnexions

Les blocs logiques élémentaires sont reliés entre eux grâce à un réseau d'interconnexions complexe. Le réseau d'interconnexions est programmable c'est-à-dire que pour relier une ligne de métal à une autre, on active un transistor qui fonctionne comme un interrupteur. Le transistor est activé grâce à une cellule mémoire : si la cellule mémoire a été programmée à 1 alors le transistor est passant (lignes de métal reliées) et si la mémoire est à 0, alors le transistor est bloqué (lignes de métal déconnectées). Pour véhiculer les signaux, on peut également utiliser un jeu de multiplexeurs et de démultiplexeurs. La programmation se fait en configurant les bits de sélection de chaque multiplexeur/démultiplexeur pour véhiculer les signaux sur les lignes désirées. Le réseau d'interconnexions est important car il constitue une grande partie de l'espace occupé dans le FPGA : entre 50 et 80 % selon [1]. Par conséquent, entre 60 et 80 % de la consommation dynamique est due au réseau d'interconnexions selon [7].

Le choix de la structure du réseau est importante car elle détermine en grande partie les caractéristiques et les performances du FPGA : surface, consommation et rapidité. En effet, plus une ligne est longue plus sa capacité parasite et sa résistance sont grandes. Par conséquent, pour un même courant de charge, le temps de transition est plus grand donc le circuit est plus lent. Ou bien, pour une même rapidité, on doit véhiculer un courant plus fort, en ajoutant des drivers de lignes, et donc consommer plus. Ensuite, plus on connecte de signaux à une ligne (pour plus de flexibilité), plus il faudra de transistors pour les interconnecter et donc plus la capacité parasite de la ligne augmente. Il y a donc un compromis à faire entre la flexibilité du circuit, la rapidité et la consommation. Plus la flexibilité du circuit sera grande, plus la surface sera grande, plus le circuit consommera et moins il sera rapide

Il existe plusieurs types de structures de réseau d'interconnexions [6]:

- **îlot logique (island style)**
- **logique en ligne (row-based)**
- **mer de porte logique (sea-of-gate)**
- **hiérarchique (hierarchical)**
- **structure unidimensionnelle (one-dimensional structure)**

Décrivons ces structures.

VIII.4.1 Îlot logique

Ce type de structure est le plus répandu. On la trouve dans la plupart des FPGAs modernes [5].

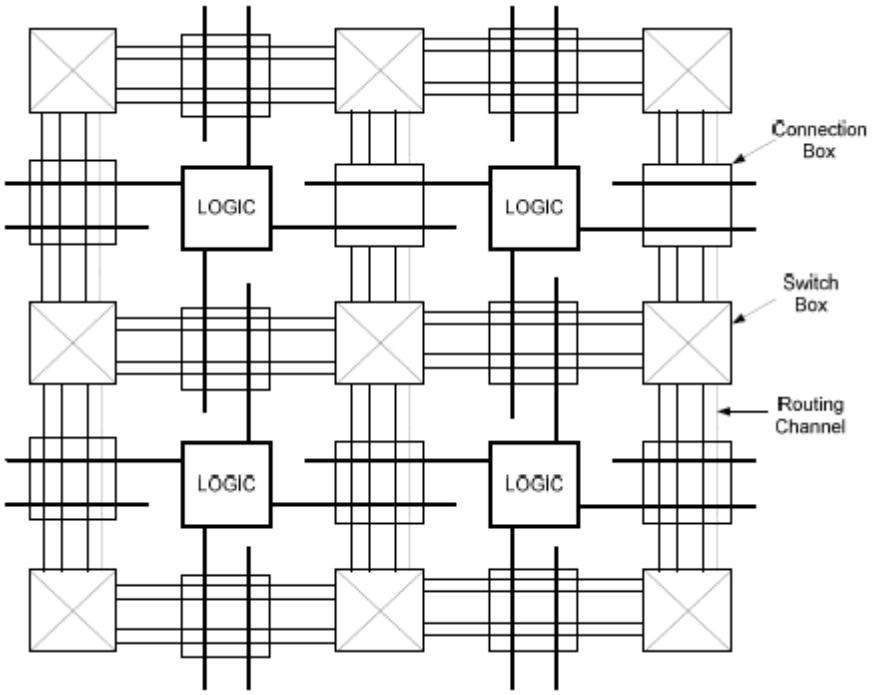


Figure 13 : îlot logique [6]

Dans cette structure, les blocs logiques sont entourés de connexions (horizontales et verticales). Les entrées et sorties des blocs logiques sont reliées aux lignes par des connexions programmables. Une entrée ou une sortie peut être reliée à plusieurs lignes. On définit alors les rapports $F_{c,in}$ (respectivement $F_{c,out}$) comme étant le nombre de lignes connectées à une entrée (respectivement à une sortie) sur le nombre total de lignes sur une longueur (horizontale ou verticale)[5]. Il existe des lignes courtes qui relient un seul bloc logique tandis que les lignes longues en relient plusieurs. Le nombre de lignes et leur longueur sont des paramètres importants qui déterminent directement les performances du FPGA. En effet, plus il y a de lignes, plus le FPGA est flexible mais plus il consomme et est lent. Ensuite, plus les lignes sont longues, plus la résistance et la capacité parasite de la ligne sont grandes et donc plus le circuit est lent. Pour étudier ce problème et trouver un bon compromis, des études ont été faites à partir de circuits test représentant, le plus fidèlement possible, ce qui se fait comme circuit cible dans les FPGAs (multiplieurs, filtres numériques, séquenceurs, interfaces de communication, …). Les FPGAs sont conçus de manière à pouvoir implémenter toutes les fonctions possibles. Le nombre de connexions nécessaires pour un circuit peut être estimé grâce à la loi de Rent. Il s'agit d'une relation empirique liant le nombre d'entrées/sorties (P) au nombre de portes logiques (G) d'un circuit : $P = KG^B$ [5] où B est un paramètre appelé exposant de Rent et K est une constante. L'exposant de Rent B permet de déterminer la répartition des longueurs des lignes pour un circuit spécifique grâce à la relation suivante : $f_L = L^{2B-3}$ [5], où f_L est la fraction de ligne de longueur L. Dans la pratique, B varie entre 0.5 et 0.75 pour la plupart des circuits. Les FPGAs sont conçus de façon à pouvoir implémenter toutes les fonctions possibles (pour les plus grandes valeurs de B). Par conséquent, dans la plupart des cas, des lignes sont inutilisées.

Les lignes sont reliées entre elles grâce à des blocs de routage. Ils se situent aux intersections des lignes horizontales et verticales. Leur architecture détermine la flexibilité et les performances du FPGA. La problématique est de relier les lignes d'un côté, aux lignes de tous les autres côtés. L'une des principales caractéristiques est le nombre de connexions possibles d'une ligne vers les autres, noté F_s. Ce nombre donne

une estimation de la flexibilité du bloc de routage. Par exemple, dans la Figure 14, $F_s = 3$ pour les deux types de blocs de routage. Pour un même côté, chaque ligne appartient à un domaine. On retrouve les mêmes domaines pour chaque côté. Dans notre exemple, il y a 4 domaines. Dans les deux cas, une ligne peut être connectée à 3 autres lignes situées sur les 3 autres côtés (ce qui veut dire que $F_s = 3$). Dans le premier cas (disjoint), les domaines sont disjoints, c'est-à-dire qu'une ligne ne peut être connectée qu'aux lignes situées dans le même domaine. La connexion entre deux domaines se fait grâce aux blocs logiques. Dans le deuxième cas (Wilton), une ligne peut être connectée à 3 autres lignes (car $F_s = 3$) dont deux d'entres elles sont situées dans des domaines différents. Dans ce cas, le routage est plus facile car il existe plusieurs chemins possibles pour établir une connexion entre deux blocs logiques. Il existe d'autres architectures de blocs de routage. Dans tous les cas, il faut faire un compromis entre flexibilité, surface du bloc, rapidité et consommation.

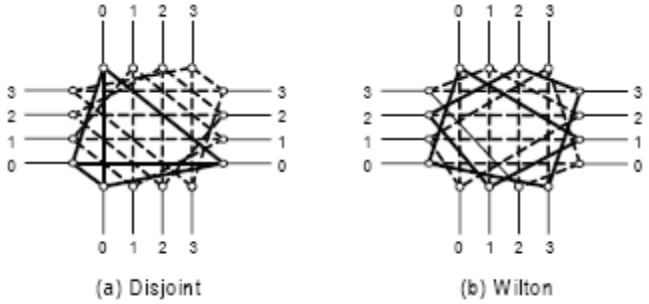


Figure 14 : architectures de routage [5]

Décrivons maintenant comment les lignes sont interconnectées. Il existe plusieurs circuits pour interconnecter des lignes : des pass-transistors, les buffers trois-états et les multiplexeurs. On les appelle des commutateurs car ils véhiculent des signaux d'une ligne sur une autre. Ils ont chacun leurs avantages et leurs inconvénients c'est pourquoi on rencontre chacun de ces dispositifs à différents niveaux d'un FPGA. Encore une fois, le choix et le dimensionnement de ces composants détermine les caractéristiques et les performances du FPGA. En effet, la taille des transistors influe sur la rapidité et la consommation des commutateurs. La résistance à l'état ON des transistors influe sur la rapidité du circuit. De plus, la largeur et l'espacement entre les lignes de métal influe sur la capacité parasite de la ligne. Les multiplexeurs nécessitent plus de transistors que les deux autres types de commutateurs. Les pass-transistors nécessitent peu de place et sont plus rapides pour les lignes courtes qui contiennent peu de commutateurs. A l'inverse, les buffers sont plus rapides lorsqu'ils sont utilisés dans des lignes contenant beaucoup de commutateurs. C'est pour cela que les buffers et les pass-transistors sont tous les deux utilisés. Il a été montré [8] qu'un réseau avec 50% de pass-transistors et 50% de buffers était plus performant (pour une même surface) que le même réseau avec un seul type de commutateur.

VIII.4.2 Logique en ligne

Dans cette structure (Figure 15), les blocs logiques sont organisés en ligne. Ils sont séparés par des lignes horizontales qui véhiculent les signaux. Des lignes verticales chevauchent les blocs logiques pour connecter les lignes horizontales entre elles. Le nombre de lignes horizontales et leur longueur doivent être optimisés en fonction de la

surface, de la consommation et la rapidité voulues. Pour cela, la longueur de chaque ligne et le nombre de transistors de routage doivent être déterminés de façon à limiter la résistance et la capacité parasite des interconnexions.

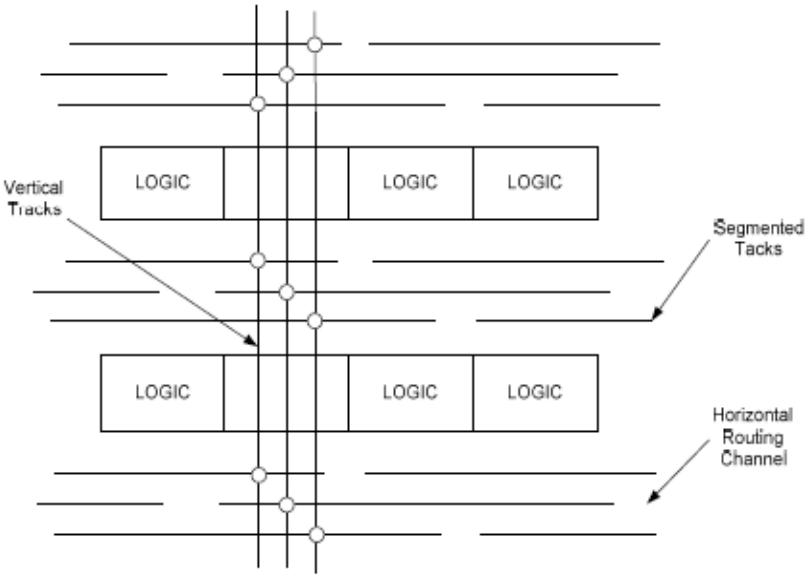


Figure 15 : logique en ligne [6]

VIII.4.3 Mer de portes logiques

La Figure 16 montre un schéma de cette structure. Dans cette structure, le réseau d'interconnexions prend très peu de place. En effet, les blocs logiques sont organisés en matrice. Chaque bloc est relié uniquement à ses voisins (à 4, 6 ou 8 voisins). Ceci limite le nombre et la longueur des interconnexions donc elles sont rapides. Dans la pratique, les performances de cette structure sont limitées à cause des parasites qui se propagent d'un bloc logique à l'autre.

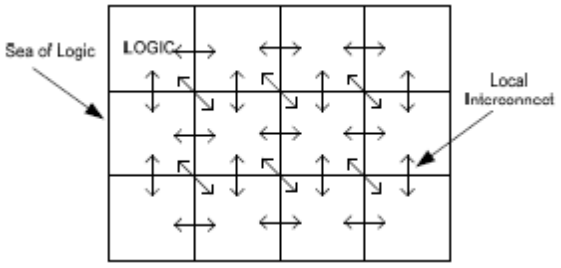


Figure 16 : mer de portes logiques [6]

VIII.4.4 Architecture hiérarchique

Dans cette structure (Figure 17), les blocs logiques sont organisés en groupes. Les blocs à l'intérieur d'un groupe sont interconnectés par des liaisons courtes et forment le niveau 0 d'interconnexions. Le niveau 1 est formé par des liaisons plus longues qui relient plusieurs groupes entre eux. C'est pourquoi cette structure est appelée hiérarchique.

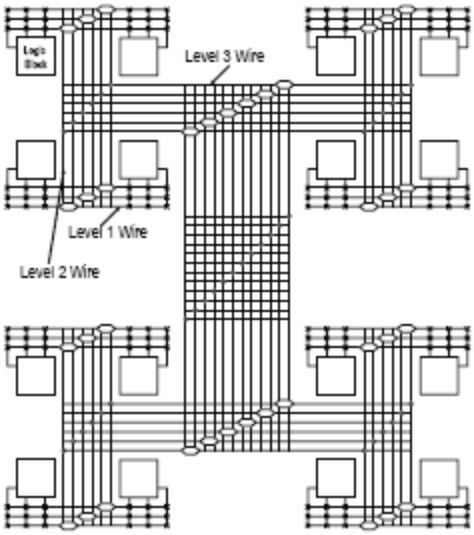


Figure 17 : structure hiérarchique [5]

Elle peut être avantageusement utilisée si l'algorithme de placement-routage est performant et si le circuit est bien dimensionné. En effet, si le circuit requiert un nombre de blocs logiques supérieur à la taille d'un groupe, par exemple, alors de longues connexions vont être utilisées pour des blocs qui devraient être voisins. Ceci va allonger les temps de propagations et augmenter la consommation du circuit. Le nombre de niveaux hiérarchiques et le nombre de blocs logiques par groupe doivent donc être bien choisis de façon à pouvoir implémenter la plupart des fonctions numériques de manière efficace. Ce type d'architecture est utilisé dans certains FPGA de chez Altera.

VIII.4.5 Structure unidimensionnelle

Dans ce cas, les blocs logiques et les lignes sont sur plusieurs colonnes. Cette structure (Figure 18) est plus simple que les structures 2D décrites précédemment. L'algorithme de routage est donc plus simple car il y a moins de possibilités de routage à tester. Cependant, cette structure est moins flexible que les structures 2D. Donc dans certains cas, la structure 2D pourrait être plus avantageuse notamment dans le cas où le circuit aurait une surface trop grande. Il faudrait alors interconnecter un grand nombre de lignes et utiliser des blocs logiques dédiés seulement au routage.

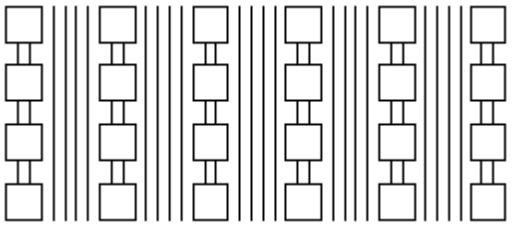


Figure 18 : structure unidimensionnelle [6]

VIII.5 Bloc d'entrée/sortie

Un bloc d'entrée/sortie est l'interface entre les composants extérieurs et la logique interne du FPGA appelé « cœur logique » ('logic core' qui désigne l'ensemble des blocs logiques et du réseau d'interconnexions). Ainsi, lorsqu'il est configuré en entrée, s'il détecte un 0 ou un 1 logique en entrée, il doit véhiculer un 0 ou un 1 logique vers la logique interne du FPGA. Lorsqu'il est configuré en sortie, il doit véhiculer le signal du FPGA vers les composants extérieurs. Un bloc d'entrée/sortie doit donc adapter les niveaux logiques et les tensions d'alimentation des circuits à l'intérieur et à l'extérieur du composant. Un FPGA étant par nature très flexible, il doit pouvoir s'interfacer avec tous les standards d'entrée/sortie. Il existe un grand nombre de standards. Chacun a ses propres niveaux logiques : le niveau logique 1 peut correspondre à des tensions de 1.2, 2.5, 3.3 V, etc. Chaque standard possède sa propre structure : simple ou différentielle (deux signaux dans un état opposé pour plus de fiabilité). Certains standards nécessitent une tension de référence pour comparer le signal d'entrée et déterminer le niveau logique. De plus, les standards sont plus ou moins rapides et supportent des courants différents. Concevoir un bloc d'entrée/sortie pouvant supporter tous ces standards est donc très difficile. Certains standards sont très peu compatibles, c'est pourquoi la taille des blocs d'entrée/sortie est grande. Par exemple, les blocs d'entrée/sortie des FPGAs Altera Stratix 1S20 et Cyclone 1C20 occupent 43% et 30% de la surface du FPGA [5]. La surface relative des entrées/sorties est plus grande dans les FPGAs simples que dans les FPGAs complexes.

L'idéal serait que tous les blocs d'entrée/sortie soient équivalents et que l'on puisse configurer chaque bloc dans n'importe quel standard indépendamment des autres blocs d'entrée/sortie. Mais la surface consommée serait exorbitante d'autant plus qu'il y a de plus en plus d'entrées/sorties dans les FPGAs et il y a un grand nombre de standards. C'est pourquoi les FPGAs modernes possèdent des banques d'entrées/sorties. Chaque banque possède sa propre tension d'alimentation et sa propre tension de référence qui sont partagées par ses blocs d'entrée/sortie. Ainsi, une seule banque ne peut pas supporter plusieurs standards simultanément. Mais différentes banques peuvent implémenter des standards différents car elles ont des tensions d'alimentation et de référence indépendantes. On peut même mettre hors tension les banques inutilisées pour diminuer la consommation.

VIII.6 Les FPGAs modernes

Dans les sections précédentes, on a vu que le principal atout des FPGAs par rapport aux ASICs est la flexibilité et le faible coût d'investissement. Ainsi, les FPGAs sont très intéressants pour les produits vendus en faibles volumes. Tandis que pour les forts volumes, les ASICs sont plus économiques. Cependant, les ASICs possèdent encore beaucoup d'avantages par rapport aux FPGAs. En effet, les ASICs sont plus rapides, consomment moins et ont une surface (et donc un coût) plus faible que les FPGAs pour un même circuit. Le fossé qui existe entre les ASICs et les FPGAs est immense [1]:

- **un FPGA demande 20 à 35 fois plus de surface qu'un ASIC**
- **un FPGA est 3 à 4 fois moins rapide qu'un ASIC**
- **un FPGA consomme (consommation dynamique) 10 fois plus qu'un ASIC**

De nombreuses recherches existent pour réduire ces écarts de performance. L'une des techniques est d'introduire dans le FPGA des circuits logiques spécifiques c'est-à-dire non programmables. Ce sont des circuits gravés dans le silicium qui accomplissent une fonction précise. On peut choisir de les utiliser ou pas mais on ne peut pas les supprimer. On parle de « hard logic » lorsqu'un bloc n'est pas programmable, contrairement aux blocs logiques programmables, appelés « soft logic » car leur fonctionnement peut être déterminé simplement en programmant des cellules mémoire. Les circuits « hard » implémentés dans les FPGAs sont des circuits très courants dans les systèmes numériques. Le premier circuit introduit dans un FPGA est la bascule. C'est un circuit très utilisé, on en trouve donc dans chaque bloc logique. Il peut être utilisé ou pas. Ensuite, des blocs de mémoire RAM et des multiplicateurs sont apparus. Les multiplicateurs sont très utilisés dans le traitement de signal. Enfin, on peut trouver également des processeurs complets. Par exemple, les FPGAs Virtex 5 de Xilinx possèdent un processeur PowerPC 440. L'utilisation de ces blocs « hard » évite donc d'utiliser des blocs logiques programmables. Cependant, si certains blocs ne sont pas utilisés, de la surface silicium est gâchée. L'amélioration des performances dues à ces blocs « hard » dépend fortement du circuit utilisé. Par exemple, un circuit tel un contrôleur UART (port série d'un PC) est constitué principalement de séquenceurs. Dans le FPGA, il ne sera implémenté qu'avec des blocs logiques programmables. Les blocs mémoires, les processeurs et les multiplieurs seront donc inutiles. Tandis qu'un filtre FIR (filtre à réponse impulsionnelle finie) sera plus performant avec des multiplieurs et des blocs de mémoires. Dans la pratique, le gain en performance dû aux blocs « hard » est limité [1]. La rapidité est légèrement améliorée avec l'utilisation de blocs « hard ». Il y a principalement deux raisons :

- **Le bloc « hard » peut diminuer seulement une petite partie du chemin critique. Les autres parties du chemin critiques peuvent rester dominante donc le gain en rapidité est limité.**
- **Un bloc « hard » peut améliorer plusieurs chemins critiques mais certains restent inchangés. Donc la rapidité n'est pas significativement améliorée.**

Le gain est donc principalement un gain en surface. Le gain en vitesse dépend fortement du circuit implémenté. Le gain en consommation est faible. Il faut noter également que le nombre de LUTs et d'interconnexions est fixe donc une partie du FPGA sera inutilisée ce qui n'a pas été compté dans les mesures faites dans l'étude [1]. Pour que l'utilisation des blocs « hard » soit optimale, les fabricants de FPGAs ont créé plusieurs familles de FPGAs répondant aux besoins de différents marchés (télécommunication, automobile, industrie). Par exemple, un FPGA destiné à faire du traitement de signal aura un grand nombre de multiplieurs.

Pour conclure, l'écart de performance existant entre les FPGAs et les ASICs, où les ASICs sont plus rapides, consomment moins et ont une plus faible surface, est très important. Il est difficile de diminuer cet écart en raison du besoin de flexibilité des FPGAs. Il faut faire des compromis entre flexibilité, consommation, surface et rapidité du FPGA. Une solution pourrait être la reconfiguration dynamique.

VIII.7 Placement-routage

Cette partie décrit les différentes étapes à suivre pour concevoir et programmer un circuit dans un FPGA. D'abords, il faut décrire le circuit que l'on veut implémenter dans le FPGA. On peut soit le concevoir graphiquement en faisant le schéma du circuit soit le décrire grâce à un langage de description de matériel (HDL pour Hardware

Description Language). La conception graphique devient fastidieuse lorsque l'on doit concevoir un circuit complexe. C'est pourquoi, les HDLs sont le plus souvent utilisés. Les plus populaires sont le VHDL (VHSIC Hardware Description Language; VHSIC: Very-High-speed Integrated Circuit) et le Verilog. Concrètement, on écrit le programme de description dans un fichier texte. Voici un exemple de fichier écrit en VHDL qui décrit un additionneur 1 bit (Figure 19).

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity full_add1 is
  port (
    a, b, cin : in  std_logic;
    s, cout   : out std_logic;
    );
end entity;

architecture arc of full_add1 is

  signal resultat : unsigned(1 downto 0);

begin

  resultat <= ('0' & a) + ('0' & b) + ('0' & cin);
  s        <= resultat(0);
  cout     <= resultat(1);

end arc;
```

Figure 19 : fichier VHDL décrivant un additionneur 1 bit [12]

Il existe des mots clés dans les langages pour placer des composants dans le circuit et les relier entre eux. On peut également déclarer des composants et les décrire. On peut décrire un composant de deux façons :

- **Structurelle : en spécifiant les connexions entre composants**
- **Comportementale : en spécifiant le comportement à l'aide de formules ou d'un programme d'action (à ne pas confondre avec un programme d'ordinateur)**

Pour gagner du temps, on peut utiliser des composants existants précompilés et optimisés en surface et rapidité. Ils sont le plus souvent vendus par les vendeurs de FPGAs ou des entreprises spécialisées. Ce sont des IPs (Intellectual Properties).

Une fois le schéma décrit, on compile le circuit. On appelle cette étape la synthèse. On va traduire les fichiers textes (écrit avec un HDL) en un schéma au niveau portes logiques (Figure 20).

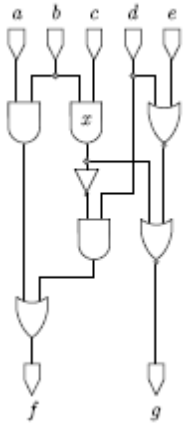


Figure 20 : fonction logique représentée par les portes logiques [11]

Le résultat est stocké dans un fichier RTL. Ensuite, on vérifie que le schéma généré fonctionne selon les spécifications de départ grâce à un logiciel de simulation. On fait seulement une vérification fonctionnelle indépendamment du FPGA cible. Si la simulation est concluante alors on passe à l'étape suivante sinon, on modifie les fichiers textes. Ensuite, intervient l'étape du « mapping », qui consiste à convertir les portes logiques en un schéma de LUTs et d'interconnexions. On regroupe donc les portes logiques dans une LUT (Figure 22).

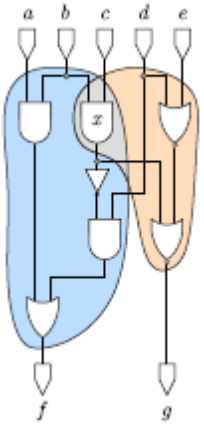


Figure 21 : découpage du circuit pour implémenter les blocs logiques [11]

Cette étape est importante car l'algorithme doit utiliser le moins de LUTs possible pour diminuer la consommation et également diminuer la longueur du chemin critique pour que le circuit soit le plus rapide possible. Après le mapping, on a donc un schéma de LUTs (Figure 22).

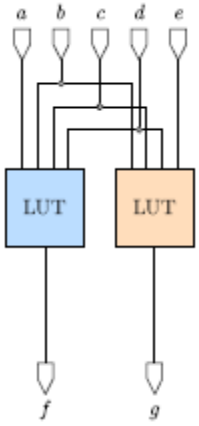


Figure 22 : on connecte les LUT entre elles pour former le circuit final [11]

L'étape suivante est le placement-routage. Elle se déroule en deux étapes :

- **le placement : on regroupe et on place chaque porte logique du schéma dans les blocs logiques du FPGA utilisé. On détermine donc l'état de chaque cellule mémoire (mémoire de configuration) contenues dans les LUTs**
- **Le routage : on connecte les blocs logiques entre eux grâce au réseau d'interconnexions. On détermine donc l'état de chaque cellule mémoire du réseau d'interconnexions du FPGA utilisé. Ces cellules mémoire activent/désactivent des transistors qui établirons une connexion entre deux lignes de métal.**

Une fois ces étapes accomplies, on connait l'état de chaque bit de configuration. On génère donc le fichier contenant la configuration du FPGA : c'est le bitstream. A ce stade, on peut vérifier si le circuit fonctionne bien et estimer ses performances grâce à une nouvelle simulation. On tient compte des temps de propagation des blocs logiques. C'est donc une simulation réaliste. Si elle est concluante, alors on peut passer à la dernière étape. Sinon, on doit recommencer le processus en modifiant l'architecture du circuit pour qu'il réponde aux spécifications.

La dernière étape consiste à configurer le FPGA, c'est-à-dire charger le bitstream dans la mémoire de configuration. On le fait à l'aide d'un programmateur vendu par le fabricant du FPGA. Il se branche à partir d'un port d'ordinateur (port USB par exemple). L'autre extrémité se branche sur des ports d'entrée/sortie spécialisés d'un FPGA ou d'une mémoire Flash (pour les FPGAs SRAM) appelé port JTAG. Par l'intermédiaire de ce port, les données de configuration sont envoyées en série sur une entrée reliée à un registre à décalage qui passe par chaque bit de configuration du FPGA. Les données écrites sont récupérées grâce à une sortie. Ceci permet de vérifier que les données de configuration ont bien été enregistrées. Une fois l'étape de configuration terminée, le FPGA est opérationnel. Pour le cas des FPGAs de type SRAM, on enregistre la configuration dans une mémoire FLASH. Ensuite, à chaque mise sous tension du circuit, les données de configuration sont transférées de la mémoire FLASH vers le FPGA.

IX. Architectures reconfigurables dynamiquement

Le chapitre précédent sur les FPGAs a décrit des FPGAs dont la configuration est statique, c'est-à-dire que la configuration du FPGA reste inchangée. On peut cependant imaginer changer sa configuration durant son fonctionnement afin d'optimiser la consommation, la surface et la rapidité du circuit, ce qui permettrait de se rapprocher des performances d'un ASIC. Cette méthode est la reconfiguration dynamique et n'est actuellement possible que dans les FPGAs SRAM.

La reconfiguration dynamique consiste à reconfigurer un FPGA durant son fonctionnement. Elle permet de n'utiliser que les circuits dont on a besoin à un instant donné. Prenons l'exemple d'un codeur/décodeur d'images JPEG. A un instant donné, on instancie dans le FPGA seulement la partie qui sert au fonctionnement. Ainsi, si l'on souhaite coder une image en JPEG, alors seulement la partie codeur sera implémentée dans le FPGA. La partie inutilisée, le décodeur, restera en mémoire. A l'inverse, lorsque l'on aura besoins de décoder une image JPEG, alors on chargera dans le FPGA la configuration du circuit permettant le décodage de l'image. De cette façon on économise de la surface sur le FPGA. On peut implémenter un système complet sur une surface plus faible ce qui diminue sa consommation. De plus, le circuit étant moins complexe, à un instant donné, il est plus rapide car on peut diminuer le chemin critique. La reconfiguration dynamique permet donc de se rapprocher des performances des ASICs. Mais pour tirer pleinement avantage de cette méthode, un système complet doit être optimisé en temps et en surface. En effet, cette méthode pose une nouvelle problématique : quelles parties du système peuvent être implémentées séparément et à quels instants elles le seront.

Cette problématique est semblable à celle de la mémoire virtuelle. En effet, dans le cas de la mémoire virtuelle, on souhaite faire fonctionner plusieurs programmes en même temps dans une mémoire principale restreinte (exemple : mémoire RAM d'un PC). Pour ce faire, on garde en mémoire principale les programmes qui sont très utilisés. Les programmes peu utilisés ou en attente sont stockés dans la mémoire de masse (exemple : disque dur). Lorsqu'un programme est rappelé ou lorsque la période d'attente est terminée, alors le programme est transféré de la mémoire de masse vers la mémoire principale. Donc la problématique est de savoir quel programme doit être chargé dans la mémoire principale et à quel moment ils sont mis en mémoire de masse. Il y a donc une analogie entre mémoire virtuelle et reconfiguration dynamique. C'est pourquoi la reconfiguration dynamique est basée sur le concept de matériel virtuel [6] : Pour implémenter un système sur une surface restreinte d'un FPGA, on place sa configuration complète dans une mémoire annexe et on implémente dans le FPGA seulement les parties utilisées.

Il faut noter qu'un FPGA contient forcément une partie qui n'est pas reconfigurable dynamiquement. En effet, la configuration des entrées/sorties ne change pas souvent.

Dans cette partie, la reconfiguration dynamique sera abordée dans le cadre des FPGAs mais elle intervient plus généralement dans les architectures reconfigurables de tous types (processeurs reconfigurables, …). Comme expliqué précédemment, on peut faire un parallèle entre reconfiguration et mémoire virtuelle, j'utiliserai donc parfois le vocabulaire de la programmation. Regardons maintenant quels sont les différents types de reconfiguration.

IX.1 Types de reconfiguration

IX.1.1 Multi-contexte

Dans une reconfiguration multi-contexte, un bit de configuration est composé de plusieurs cellules (N bits) mémoires. A un instant donné, seul un bit est utilisé pour configurer l'élément mémoire. Pour changer de configuration, il suffit de changer de bit. Ce changement de configuration peut être très rapide : de l'ordre de la nanoseconde. Ceci évite les temps de latence dus au chargement de la configuration dans la mémoire (peut être de l'ordre de la milliseconde ou plus).

IX.1.2 Reconfiguration partielle

La reconfiguration partielle permet de configurer une partie du FPGA pendant que les autres sont toujours en fonctionnement. Cela évite de reconfigurer l'ensemble du FPGA. Cette fonctionnalité est disponible dans les FPGAs modernes comme les FPGAs Virtex de Xilinx. Certains systèmes peuvent nécessiter une mise à jour partielle. Grâce à la reconfiguration d'une petite partie de ces systèmes, une grande quantité de données de configuration est économisée et la rapidité de la mise à jour est améliorée. Cependant, pour cela, il faut spécifier l'adresse du bloc à configurer. Si le nombre de blocs est trop grand, l'économie en données de configuration pourrait être très limitée. Il faut donc faire un compromis entre la flexibilité (grand nombre de blocs) et l'économie en donnée de configuration (limiter le nombre d'adresses donc de blocs).

IX.1.3 Reconfiguration en pipeline [6]

Cette catégorie de reconfiguration est destinée aux applications pouvant être implémentées sous forme de pipeline. La reconfiguration se propage d'étage en étage. Chaque étage est configuré en une seule fois ce qui nécessite beaucoup de données en même temps. C'est le principal inconvénient de cette méthode.

IX.2 Catégories de reconfigurations dynamiques [6]

La reconfiguration dynamique permet d'améliorer les performances d'un système implémenté sur un FPGA. Elle peut se faire de trois façons différentes au niveau système suivant les performances requises. Ces catégories de reconfiguration sont décrites dans les sous-parties suivantes.

IX.2.1 Algorithmique

Dans ce cas, le but de la reconfiguration algorithmique est de reconfigurer un système qui garde la même fonctionnalité mais a des performances, une précision, une consommation ou ressources différentes. Par exemple, prenons le cas d'un système (un satellite) destiné à faire des multiplications sur des données qui arrivent de manière continue. Si l'on souhaite de la performance on dupliquera le multiplieur N fois pour faire des calculs parallèles afin d'augmenter la rapidité du système de N fois. Mais si les

conditions changent de telle façon que la fiabilité devienne un critère important (satellite soumis à de soudaines fortes radiations), on peut changer la « politique » de reconfiguration : on duplique le multiplieur trois fois pour faire le même calcul trois fois et on fait un vote majoritaire pour plus de fiabilité.

IX.2.2 Architecturale

Dans ce cas, la topologie du circuit est modifiée. Elle peut être nécessaire lorsque les ressources matérielles sont limitées. Par exemple, admettons qu'un module A était en train de fonctionner et qu'un module B de plus haute priorité (module de gestion d'une panne par exemple) est appelé soudainement. Alors on peut diminuer la surface occupée par le module A pour laisser le module B s'exécuter. Lorsque B a fini, on retrouve la situation initiale. On retrouve l'analogie avec l'exécution de tâche sur un OS temps réel où l'on peut interrompre des tâches non prioritaires pour exécuter des tâches importantes comme la gestion de la sécurité.

IX.2.3 Fonctionnelle

Dans ce cas, on souhaite faire tourner plusieurs circuits sur un même FPGA limité en surface. Puisque la surface est limitée et que l'on ne peut pas implémenter tous les circuits en même temps, on va alterner l'exécution des différents circuits assez rapidement pour que l'exécution semble parallèle. L'ordonnancement des circuits doit être optimisé pour utiliser au mieux les ressources du FPGA. Une fois de plus, on peut faire l'analogie avec un OS multitâches. En effet, pour exécuter plusieurs tâches « en même temps » on exécute une tâche durant une certaine durée, puis on passe à une autre tâche, et ainsi de suite pour toutes les tâches. Dans un OS multitâche on commute entre les tâches toutes 10 ms environs.

IX.3 Méthodes pour accélérer la reconfiguration

La reconfiguration dynamique présente de nombreux avantages. Cependant, pour qu'elle soit rapide et efficace, il faut limiter la taille des données de configuration, améliorer la vitesse de transfert des données et limiter le plus possible la taille du circuit de gestion de la reconfiguration. Il existe plusieurs méthodes pour améliorer la reconfiguration. Les principales sont décrites dans les sous-parties suivantes.

IX.3.1 Pré-chargement

Le pré-chargement consiste à charger la configuration à l'avance pour diminuer le temps latence entre deux changements de configuration. C'est particulièrement avantageux dans les systèmes avec multi-contextes ou partiellement reconfigurables. Par exemple, on reconfigure un contexte pendant que l'autre fonctionne puis lorsque ce dernier a fini sa fonction, on commute entre les deux contextes. Le changement de contexte est très rapide. La principale difficulté est de déterminer suffisamment à l'avance quelle configuration devra être implémentée.

IX.3.2 Compression

Pour diminuer la taille des données de configuration, une méthode consiste à compresser les données. De cette façon, les données prennent peu de place en mémoire et sont transférées plus rapidement dans le FPGA. Le principal inconvénient étant le temps et la circuiterie supplémentaires nécessaires à la décompression des données.

IX.3.3 Portabilité et défragmentation

Dans les systèmes partiellement reconfigurables, il peut se produire des conflits lorsque deux modules à implémenter se chevauchent. C'est le même problème qui apparait dans l'occupation d'une mémoire par plusieurs programmes. Dans ce cas les programmes sont fragmentés. Mais dans un FPGA les modules ne peuvent pas être fragmentés car cela nécessiterait une architecture FPGA beaucoup trop volumineuse. Chaque système doit donc pouvoir changer de place dans le FPGA sans que sa fonction change. Dans le cas d'un programme on dit qu'il doit être portable. Un FPGA qui aurait cette fonctionnalité doit donc pouvoir déterminer où implémenter chaque système sans qu'ils se chevauchent. De plus chaque système doit être « portable ».

IX.3.4 Configuration cache

Cette méthode concerne les systèmes multi-contextes. Le principe est le même que celui des mémoires caches dans les processeurs. Elle consiste à garder la configuration dans le FPGA pour diminuer le nombre d'accès mémoire et les temps de latence dus au transfert des données de configuration. Dans ce cas, la mémoire cache des processeurs est équivalente aux contextes dans les FPGAs. Comme dans les processeurs, la principale difficulté est de déterminer quelles configurations restent dans les différents contextes et quels contextes sont remplacés par une autre configuration.

X. Conception basse consommation

X.1 Power gating[15]

Le power gating est une méthode qui consiste à couper l'alimentation d'un module lorsqu'il est inutilisé. Cette méthode permet de réduire fortement la consommation statique. Pour cela, on introduit un transistor à très faible courant de fuite qui fait office d'interrupteur. On peut également implémenter un contrôleur d'alimentation pour activer ou non le transistor en fonction du fonctionnement. La Figure 23 montre une porte NAND que l'on peut éteindre et rallumer grâce au transistor de contrôle de l'alimentation (en rouge).

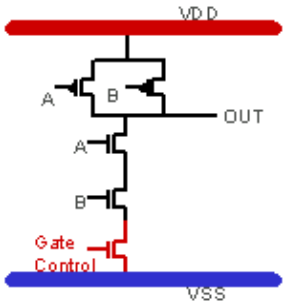


Figure 23 : Power gating appliqué à une porte NAND [15]

Pour mettre en œuvre cette méthode de manière avantageuse, il faut prendre en compte certaines contraintes. Tout d'abord, il faut dimensionner le transistor de manière à ne pas pénaliser la rapidité du circuit. Il doit donc être relativement gros par rapport aux autres transistors du circuit. Ensuite, il ne doit pas être trop pénalisant en termes de place. Il y a un compromis à trouver entre la taille du circuit et le nombre de circuits à contrôler. En effet, si l'on veut contrôler plusieurs circuits simples, on pourra gérer la consommation de manière très flexible, mais la place prise par le contrôle de l'alimentation (transistor ON/OFF + gestion de la consommation) sera énorme. Il faut faire également attention aux parasites générés sur l'alimentation à cause de la mise sous tension du circuit. Un fort appel en courant peut entraîner une baisse de la tension d'alimentation et éventuellement générer des erreurs dans le système notamment au niveau des bascules. Pour limiter ce problème, on peut allumer les sous-circuits les uns après les autres pour limiter l'appel en courant. Pour finir, une certaine quantité d'énergie est nécessaire pour activer les transistors d'alimentation. Il faut donc veiller à ce que la mise hors tension du sous-circuit soit suffisamment longue pour que le gain en consommation soit significatif. Il faut noter que les blocs éteints ont des sorties dont l'état est indéterminé. Ceci peut poser problème pour les blocs sous tension dont les entrées sont reliées aux sorties « éteintes ». Il faut donc placer entre les entrées et les sorties des portes ET ou OU comme le montre la Figure 24. Cette porte met la sortie à une valeur fixe lorsque le bloc est éteint et reproduit fidèlement la sortie lorsqu'il est en fonctionnement.

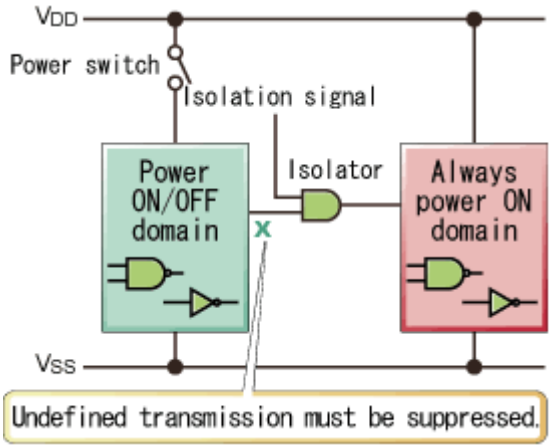


Figure 24 : porte permettant d'isoler le circuit hors tension du circuit sous tension [14]

X.2 Clock gating

Le clock gating consiste à arrêter l'horloge d'un registre dont on sait qu'il ne va pas changer d'état durant un certain temps. Par exemple, lorsqu'un programme exécute une boucle *while,* qui attend l'apparition d'un évènement comme l'appui d'une touche sur un clavier, le temps d'attente peut être long (quelques secondes), donc une mise en veille de la machine à état et de l'horloge peut limiter la consommation. Il faut savoir que le réseau de distribution de l'horloge consomme beaucoup car il a une forte capacité parasite, en raison de la longueur des pistes d'horloge.

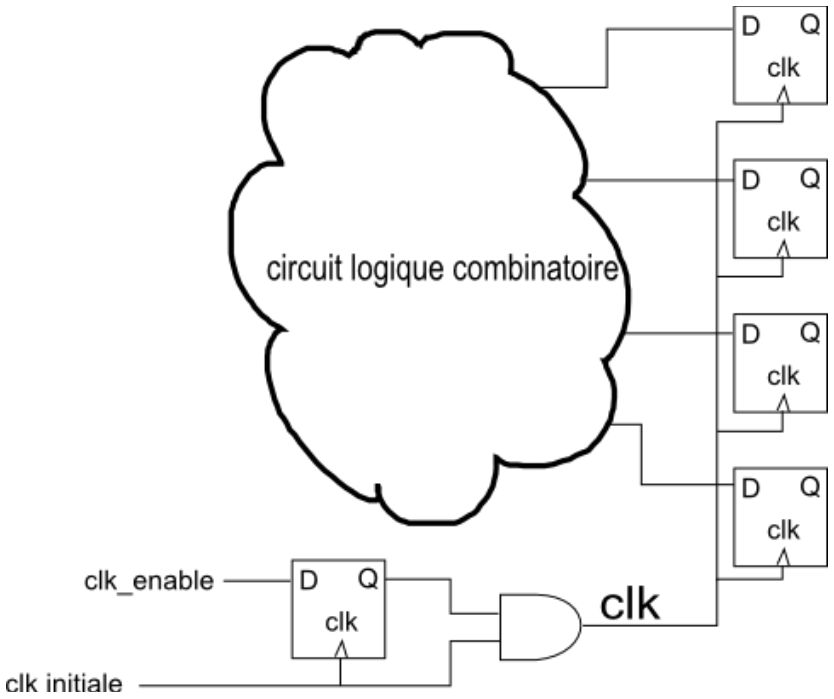


Figure 25 : exemple de circuit logique avec "clock gating"

X.3 DTMOS (Dynamic Threshold MOS)

Un DTMOS est un transistor MOS dont on change la tension de seuil de manière dynamique. Pour cela, on polarise le substrat. On a différentes configurations possibles listées dans la figure 24.

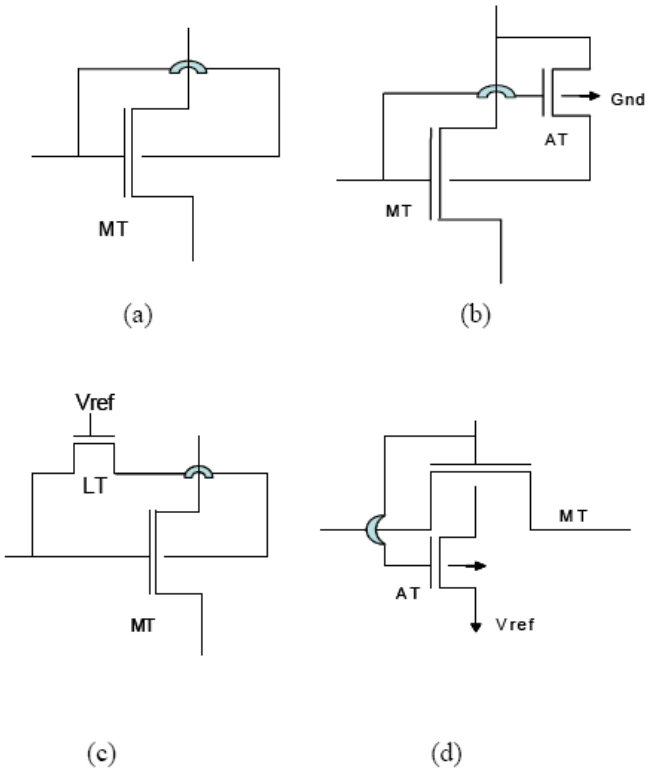


Figure 26 : configuration possibles d'un DTMOS [16]

Pour expliquer son fonctionnement, prenons le cas d'un transistor MOS canal N. Le principe est le même pour chaque configuration. Il suffit de relier la grille et le substrat. Lorsque le transistor est à l'état ON, la tension de seuil est basse donc la résistance à l'état ON du transistor est basse. Le circuit est donc plus rapide. Quand le transistor est à l'état OFF, la tension de seuil est élevée, les courants de fuite sont donc plus faible. On a donc un transistor plus rapide et qui consomme moins à l'état statique. L'inconvénient est que les substrats des transistors doivent être isolés les uns des autres ce qui nécessite des procédés de fabrication spécifiques comme la technologie Silicon On Insulator (SOI) ou triple caissons. De plus un transistor sera plus gros. La configuration la plus simple de DTMOS est le schéma (a). La grille est directement reliée au substrat. Dans ce cas, il faut veiller à ne pas dépasser la tension de seuil des diodes substrat-source et substrat-drain. Pour cela, on peut ramener la tension d'alimentation à 0.6 V ce qui diminue le courant à l'état ON et donc la rapidité. C'est pourquoi il existe d'autres configurations possibles. Dans la configuration (b), le drain et la grille des deux transistors sont connectés. Mais il faut ajouter un transistor ce qui prend de la place. Dans la configuration (c), la grille et le substrat sont reliés par un transistor. Sa grille est reliée à la tension d'alimentation. De cette façon, le substrat est polarisé à (V_{dd} - V_{th}), ce qui abaisse la tension en dessous du seuil des diodes. L'inconvénient est que la grille du transistor supplémentaire est constamment à V_{dd}, donc il y a plus de courant de fuite par la grille (par effet tunnel). Dans la configuration (d), seule la grille est partagée par les

deux transistors, ce qui limite les courants de fuite par rapport à l'architecture (c) et on peut partager le transistor supplémentaire entre plusieurs transistors qui partagent le même signal (comme dans le cas d'un multiplexeur). Cette architecture permet de limiter le surcoût en silicium.

X.4 Multi-V_{th} design

Il existe des technologies dans lesquelles il est possible d'implémenter des transistors ayant des tensions de seuil différentes (basse et haute tension de seuil). Ainsi on peut optimiser la consommation statique en mettant des transistors à faible seuil sur le chemin critique pour que son délai soit faible et des transistors à seul élevé sur les autres parties du design où la faible rapidité des transistors n'est pas critique.

X.5 Multiple tension d'alimentation

On peut utiliser des tensions d'alimentation différentes pour des blocs dont les fréquences de fonctionnement sont différentes. Pour les blocs fonctionnant à faibles fréquences, on baisse la tension d'alimentation de façon à diminuer leur fréquence maximale de fonctionnement. En effet, lorsque l'on baisse la tension d'alimentation, le courant à l'état ON des transistors diminue, ce qui augmente le temps de monté du signal. Cependant le gain en consommation dynamique est appréciable car l'énergie consommée lors de la commutation d'un transistor est fortement diminuée. Cette énergie peut être donnée par la relation suivante :

$$E_d = \frac{1}{2} C_g V_{dd}^2$$

Où C_g est la capacité de grille et V_{dd} la tension d'alimentation. Lorsque l'on diminue la tension d'alimentation, l'énergie diminue fortement car elle dépend du carré de cette tension. C'est pourquoi la diminution de la tension d'alimentation peut être une bonne solution pour diminuer la consommation dynamique.

Pour mettre en œuvre cette solution, il faut cependant ajouter des circuits de mise à niveau des tensions logiques entre les blocs dont les tensions d'alimentation sont différentes, comme le montre la Figure 27.

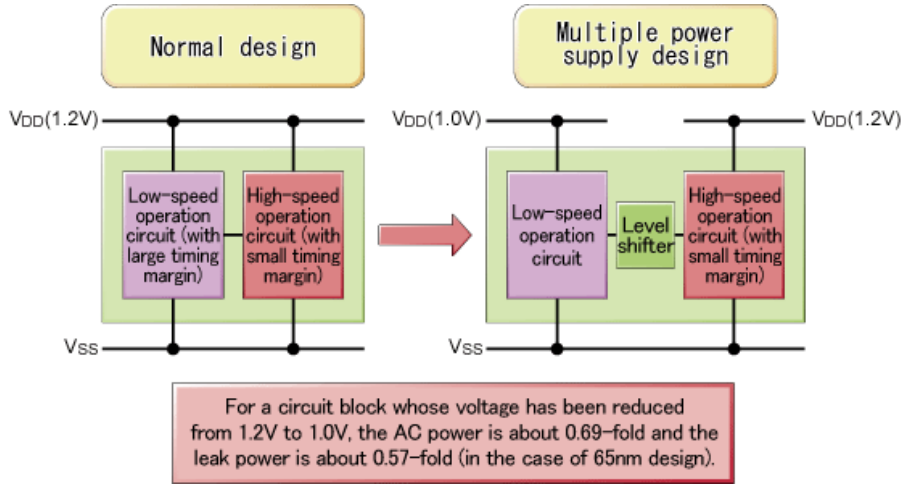


Figure 27 : mise à niveau des niveaux de tensions logiques [14]

X.6 Dynamic voltage and frequency scaling (DVFS)

Cette technique consiste à changer la tension d'alimentation de façon dynamique, c'est-à-dire pendant le fonctionnement du circuit, la tension d'un circuit en fonction de son utilisation. Par exemple, si un circuit nécessite de la rapidité, sa tension d'alimentation sera augmentée pour augmenter sa vitesse. A l'inverse, si la fonction est en mode basse consommation, la tension sera baissée. Lorsque l'on modifie la tension d'alimentation d'un circuit la fréquence maximale de fonctionnement change. C'est pourquoi la fréquence de fonctionnement est également changée de façon dynamique, en même temps que la tension. La diminution de la fréquence permet également de diminuer la température dissipée et permet donc de limiter l'utilisation de ventilateur pour évacuer la chaleur. Cela permet également d'utiliser un boitier en plastique, dont le coût est peu élevé, au lieu d'un boitier en métal. De plus, les températures élevées accélérant le vieillissement, la fiabilité du circuit sera améliorée.

XI. Les effets des radiations sur les circuits CMOS

XI.1 Source de radiation

Dans l'espace, il existe principalement deux sources de radiation : le rayonnement cosmique et le vent solaire. Le rayonnement cosmique a pour origine des événements spatiaux violents comme les supernovae au cours desquels de grandes quantités de particules très énergétiques sont éjectées. Ce rayonnement est constitué de proton (92%), de particules alpha (6%) et de noyaux d'atomes plus lourds (2%). Le flux de particule est d'environ 36000 particules/cm².h, contre 14 particules/cm².h à la surface de la terre [19].

Les vents solaires ont pour origines les éruptions à la surface du soleil. Le flux de particules dépend de l'activité du soleil. Ce cycle est de 11 ans environ et est lié à l'apparition de taches sombres en surface. Plus il y a de tâches plus il y a d'éruptions solaires. Ces particules sont moins énergétiques que les particules cosmiques.

Lorsque les particules arrivent sur terre, elles interagissent avec les atomes de l'atmosphère en produisant une trainée de particules secondaires constituée de neutron, de muons et de pions. Les pions et les muons ont une durée de vie très courte, c'est pourquoi on les retrouve principalement à haute altitude. Les neutrons, par contre, ont une durée de vie de 11 minutes environ, on en retrouve donc en faible quantité au niveau de la mer. Que ce soit au niveau de la mer ou dans l'espace on retrouve également un rayonnement alpha dû au boitier. Détaillons maintenant les effets de ces radiations sur les circuits CMOS.

XI.2 Total ionizing dose (TID) effects

Cet effet traduit l'impact des radiations sur le vieillissement du composant. C'est donc l'accumulation de radiation qui provoque cet effet. Il a pour conséquence de modifier les caractéristiques des transistors comme la tension de seuil et donc l'augmentation des courants de fuite. A terme, le transistor est inutilisable.

Lorsqu'une particule pénètre dans le silicium, des paires électrons trous sont créées sur son trajet. Dans les matériaux conducteurs, les trous et les électrons se recombinent rapidement grâce à leur forte mobilité dans ces matériaux. Dans les isolants (oxydes de grille, séparation des couches de métal), les électrons et les trous sont moins mobiles donc une partie des trous (99%) va se recombiner avec les électrons tandis que les autres vont être piégés dans l'isolant. Ces trous piégés vont faciliter la conduction à travers l'isolant donc les tensions de seuil des transistors NMOS et PMOS vont être modifiées et les courants de fuite à travers la grille, dans le canal et entre les transistors vont augmenter au fur et à mesure.

Cet effet est de moins en moins préoccupant lorsque l'on diminue la taille des transistors car les épaisseurs d'oxydes sont plus fines et donc les trous et les électrons se recombinent plus facilement.

L'unité de dose de radiation absorbée par un circuit est le rad. C'est une ancienne unité, elle a été remplacée par le gray (Gy) dans le Système international. Mais c'est le rad que l'on retrouve le plus souvent dans la littérature. Donnons maintenant quelques

ordres de grandeur. A la surface de la terre, un circuit CMOS sera soumis à environ 25 rad/an. Dans l'espace, un satellite géostationnaire est soumis à environ 30 krad/an [18]. De manière générale, un composant électronique destiné au spatial résiste jusqu'à quelques centaines de krad : 300 krad pour la plupart des applications [21]. Les composants utilisés dans les expériences de physiques de haute énergie doivent supporter plusieurs Megarads [18].

XI.3 Déplacement d'atomes

Un deuxième effet est le déplacement d'atomes du réseau cristallin. En effet, les particules incidentes ont une énergie très élevée. Il peut donc apparaître des lacunes lors de l'impact des particules sur les atomes de silicium. Les caractéristiques des transistors peuvent donc changer. Les transistors bipolaires sont les plus impactés par ce phénomène. Cependant, les transistors MOS y sont très peu sensibles car la conduction se fait à l'interface oxyde de grille – substrat. Ils sont plus influencés par le vieillissement décrit précédemment. Cet effet ne sera donc pas développé.

XI.4 Single Event Effects (SEE) [19]

Les SEEs sont des effets transitoires. Lorsqu'une radiation arrive dans le silicium, une trainée de paires électron – trou est créé. Le champ électrique, présent dans les jonctions PN ou dans l'oxyde de grille, peut empêcher les électrons et les trous de se recombiner. Les électrons peuvent alors se propager dans les nœuds du circuit électrique et faire commuter un transistor. Une telle perturbation a une durée de l'ordre de la centaine picosecondes environ. Il peut alors apparaître une erreur transitoire. En effet, dans un circuit logique combinatoire, une perturbation peut générer une erreur qui peut être cachée si elle n'apparait pas sur un front d'horloge. Par contre, si un SEE affecte une cellule mémoire, comme les cellules SRAM, alors l'état de la cellule peut basculer et l'information est modifiée. Dans ce cas, l'erreur est permanente, c'est un Single Event Upset (SEU). Il faut alors un circuit pour détecter cette erreur et éventuellement la corriger. La plupart des SEUs n'affectent qu'une cellule mémoire, seulement un faible pourcentage des SEUs (entre 1 et 5%) correspondent à une SEU qui affecte deux bits. Ce pourcentage risque d'augmenter au fur et à mesure de la diminution de la taille des transistors. En effet, si une particule a un angle d'incidence très faible, proche de la surface, la trajectoire de la particule va passer par plusieurs transistors. Cet effet sera donc plus probable si les transistors ont une faible surface car ils sont plus sensibles que les transistors en technologie mature.

La quantité d'énergie déposée par une particule en traversant un composant est appelé LET (pour Linear Energy Transfer). Son unité est le cm^2Mev/mg et dépend du matériau traversé. Il existe une valeur particulière de LET, le LET critique qui permet d'indiquer la fiabilité d'un circuit. Sa valeur est déterminée à partir d'une courbe expérimentale montrant la section efficace en fonction du LET. Pour cela, le composant à tester est soumis à un jet d'ions. Pour faire varier le LET, on utilise différents ions à différentes énergies. On relève ensuite le nombre d'erreurs N mesuré durant l'exposition du composant aux particules. On peut alors calculer la section efficace avec la formule suivante :

$Xs = N/(F.t.\cos\theta)$

Où θ est l'angle du flux de particules avec la normale, F est le flux de particules et t est le temps d'exposition.

On obtient alors une des courbes présentées dans la Figure 28 où l'on voit les courbes d'un composant durci et non durci. Par définition, le LET critique est la valeur de LET qui est à 10% de la valeur maximale du LET (LET_{asympt}).

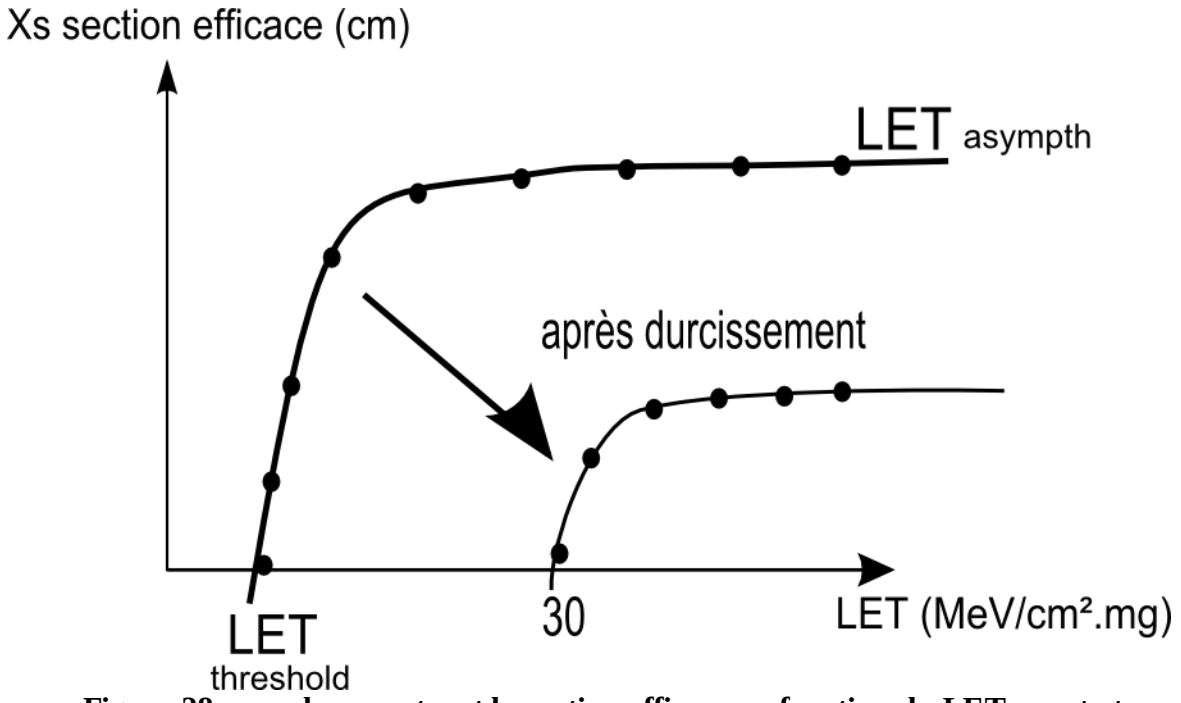


Figure 28 : courbes montrant la section efficace en fonction du LET avant et après durcissement

Pour qu'une radiation puisse faire basculer un transistor, il faut que la charge apportée dépasse un certain seuil : c'est la charge critique (*critical charge* Q_{crit}). On peut la déterminer par simulation en injectant une impulsion de courant et en augmentant son amplitude jusqu'à faire basculer une cellule SRAM par exemple. Lorsque la cellule bascule, on peut déterminer la charge grâce au courant. Il existe une formule proposée par Hazucha et Svensson qui permet de déterminer le taux d'erreurs transitoires (*Soft Error Rate* SER) d'un circuit [19]:

$SER\ circuit = Cst \times Flux \times Surface \times e^{-Qcrit/Qcoll}$

Cst est une constante dépendant du nœud technologique et du circuit, Flux est le flux de particules (neutron), Surface est la surface du circuit sensible aux radiations et Q_{coll} est l'efficacité de la capture de charge (c'est-à-dire le rapport entre la charge capturée et la charge totale déposée par la radiation). Donc, plus un transistor a une faible taille plus il sera sensible à une radiation incidente. Mais plus il est petit, plus la probabilité de collision est faible. Globalement, il semble que ces deux effets se compensent. On observe un SER relativement constant d'un nœud technologique à l'autre, pour un même circuit. Mais pour un circuit de plus en plus dense, le SER augmente.

Dans un circuit, les points les plus sensibles aux radiations sont ceux en haute impédance. En effet, lorsque le transistor est bloqué, le drain sera à V_{dd} et le substrat à 0V. Le champ électrique dans la jonction PN sera donc maximal, la charge accumulée sera donc élevée ce qui risque de perturber le circuit. Tandis que si un transistor NMOS est fermé, la tension entre drain et source sera nulle donc le champ électrique dans la jonction PN entre Drain et substrat sera nul. Les électrons et les trous pourront donc se recombiner plus facilement, la charge accumulée sera donc faible donc il est peu probable que le circuit soit perturbé.

Un autre effet indésirable des radiations est le Single Event Latchup (SEL). Il intervient lorsque deux transistors NMOS et PMOS sont côte à côte. Ils forment une liaison PN-PN parasite qui correspond à un thyristor. Une radiation peut activer ce thyristor c'est-à-dire le rendre passant. Il court-circuite alors V_{dd} et gnd. Le courant est important il y a donc un risque de destruction des transistors ou bien de générer des erreurs en changeant des données.

Pour connaître l'immunité des circuits numériques face aux radiations, on utilise couramment le FIT (Failure in time). C'est une unité de mesure. Un FIT correspond à une erreur pour un milliard d'heure. Par exemple [19], prenons un circuit contenant 10^9 transistors. Chaque transistor a un taux d'erreur de 0.00001 FIT. Alors le taux d'erreur d'un circuit complet est de : 10^9 x 0.00001 = 10^4 FIT. Si maintenant un système contient 100 exemplaires de ce circuit, alors le taux d'erreur de ce système est de 100 x 10^4 = 10^6 FIT. Ceci correspond à une erreur en 40 jours environ. A titre d'exemple, un FPGA Virtex 5 de Xilinx, de technologie 65 nm, a un taux d'erreur moyen de 151 FIT pour 1 Mb de mémoire de configuration [20].

XII. Méthodes de durcissement des circuits face aux radiations

XII.1 Technologie SOI

Pour limiter le vieillissement dû aux radiations, la technologie Silicon On Isulator (SOI) est souvent utilisée. Dans cette technologie, le wafer est constitué d'une couche de matériaux isolant ($Si0_2$) couvert d'une fine couche de Silicium. En dessinant une tranchée d'isolation autour d'un transistor, on peut l'isoler de ses voisins pour éviter les courants de fuite entre transistors. Cela permet également d'éliminer le risque de SEL.

L'utilisation de cette technologie permet également de faire des circuits moins sensibles aux SEEs [19]. La Figure 29 compare un transistor NMOS en technologie classique et en SOI.

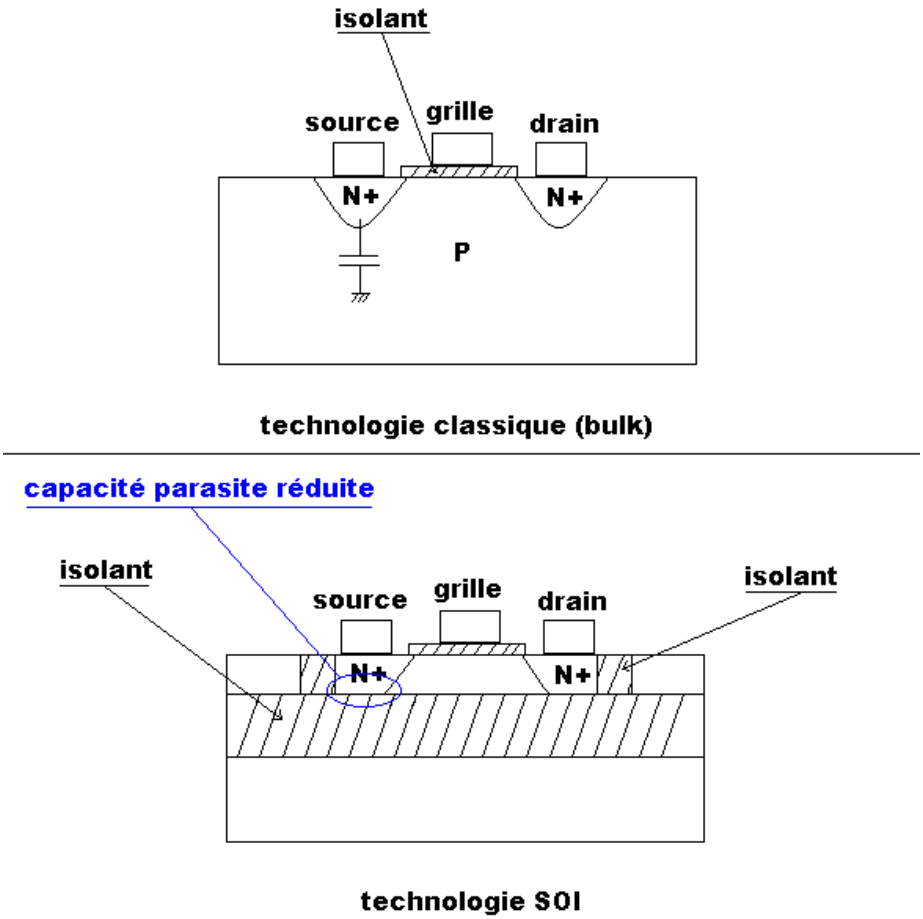


Figure 29 : comparaison technologies Silicium massif (bulk) et SOI

Avec la technologie SOI le volume des zones sensibles est réduit ce qui réduit les risques d'erreurs. De plus, la rapidité du circuit est améliorée grâce à l'effet body flottant et le fait que les capacités parasites drain-substrat et source-substrat sont réduites [19].

XII.2 Enclosed layout transistor (ELT)

Un ELT est un transistor MOS dont la forme est particulière. Elle est présentée dans la Figure 30.

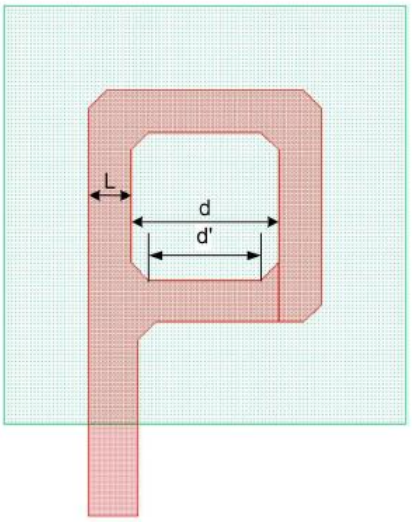


Figure 30 : layout d'un transistor ELT [18]

Cette structure de transistor est utilisée pour éliminer les courants de fuite à l'extérieur du transistor. Il permet à un circuit de résister à des doses de radiation de plusieurs Megarads. L'inconvénient est qu'il a un coût en surface plus élevé que pour un transistor classique. Cependant, les circuits dans des technologies submicroniques vieillissent moins rapidement faces aux radiations. Donc ce type de transistor sera moins utilisé dans les technologies avancées.

XII.3 Redondance spatiale

Une des méthodes les plus classiques de durcissement aux radiations est la TMR (pour Triple Modular Redundancy). Elle consiste à implémenter trois fois le même circuit et faire un vote majoritaire en sortie. De cette façon, si une erreur survient dans un des modules et que les deux autres modules sont corrects, la sortie du vote majoritaire sera exacte. Cependant, si une erreur survient dans deux modules, la sortie sera fausse. Pour qu'une telle situation ne se produise pas, il faut que les modules soient les plus indépendants possibles. Ainsi, les points sensibles, au niveau logique, doivent être suffisamment éloignés les uns des autres pour qu'une particule n'induise pas d'erreur dans deux circuits. Ensuite, par exemple, si la même alimentation est partagée entre les trois modules et qu'une perturbation sur la tension d'alimentation arrive, elle risque de générer des erreurs dans les registres des trois modules. L'idéal serait donc une alimentation séparée pour chaque module.

On distingue plusieurs types de structures en TMR suivant le niveau de fiabilité requis. Ainsi au niveau logique, on peut décider de protéger seulement les registres car ce sont les circuits les plus sensibles aux radiations (Figure 31 : registres en TMR).

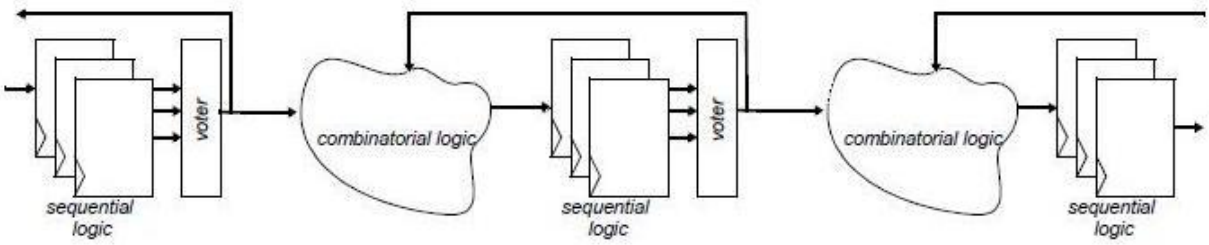


Figure 31 : registres en TMR [23]

Pour plus de fiabilité, on peut protéger également les circuits combinatoires (Figure 32 :). Rappelons que les erreurs générées dans les circuits combinatoires n'auront des conséquences que si l'erreur arrive au moment du front montant de l'horloge des registres. En effet, les sorties des circuits combinatoires ne sont prises en compte qu'au moment où elles sont stockées dans le registre. En dehors de ce bref instant, il n'y aura que des « pics » logiques (*glitches* en anglais) qui n'auront aucune conséquence sur le résultat des calculs. C'est pourquoi les circuits combinatoires sont moins sensibles aux radiations que les registres : les circuits combinatoires sont sensibles sur une courte durée tandis que le contenu des registres est sensible constamment. Notons également qu'une boucle de rétroaction est présente. Les données en sorties des blocs de vote des registres sont véhiculées vers le circuit combinatoire précédent pour corriger les éventuelles erreurs et ainsi éviter que l'erreur ne se propage vers les étages suivants. Pour accroitre encore la fiabilité, il est possible de tripler également les circuits de vote majoritaire (Figure 34). La Figure 34 montre un schéma possible de module de vote majoritaire.

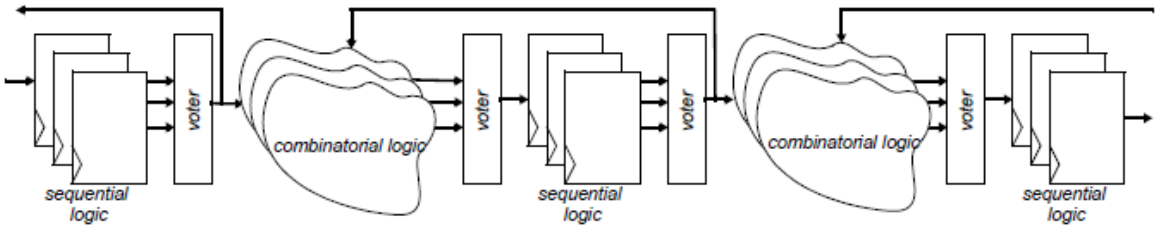


Figure 32 : TMR pour les circuits combinatoires et registres [23]

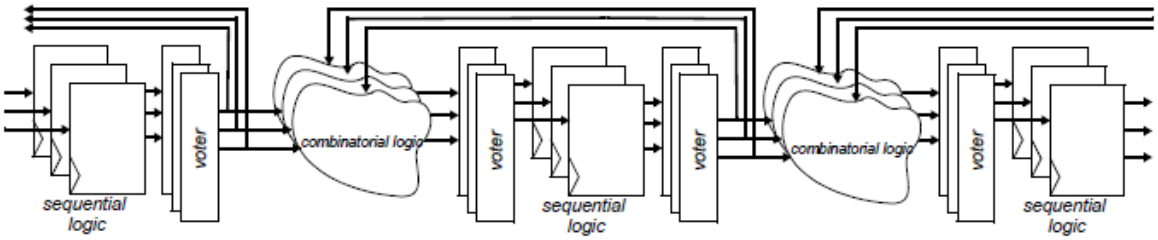


Figure 33 : TMR pour les circuits combinatoires, registres et vote majoritaire [23]

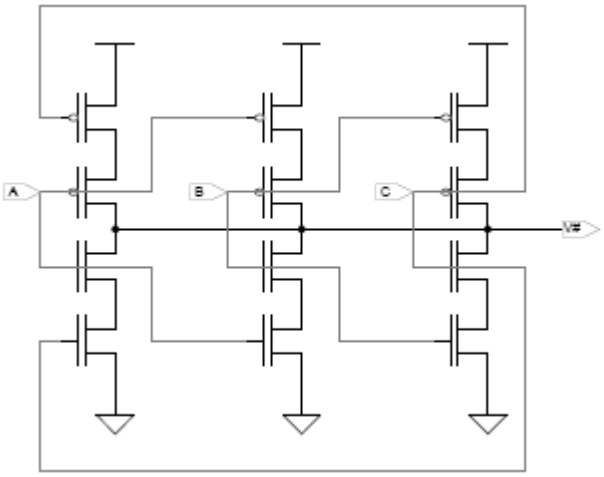


Figure 34 : schéma d'un module de vote majoritaire [18]

La TMR peut également être implémentée au niveau des composants. Au lieu d'implémenter la TMR dans un même boitier, trois boitiers seront utilisés et un autre composant sera utilisé pour le vote majoritaire. Cela permet d'augmenter la fiabilité car les composants sont indépendants.

La TMR a coût en surface silicium très élevé (3,2 fois environ) par rapport à un circuit classique et les performances sont diminuées de 10 % environ. De plus la consommation est d'autant plus grande que la surface est élevée.

XII.4 Redondance temporelle

La redondance temporelle consiste à protéger un circuit combinatoire. Dans ce cas, on retrouve un module de vote majoritaire. Les trois signaux présents en entrée du vote sont le signal à protéger mais retardés avec chacun un retard différent. Ainsi, le premier signal n'est pas retardé, le second est retardé de N ns et le troisième est retardé de 2N ns. De cette façon, si une particule génère une impulsion dans le circuit combinatoire (comme dans l'exemple de la Figure 35), et que les délais ont bien été choisis, alors l'erreur n'apparaitra que dans un seul signal à la fois arrivant sur le vote majoritaire. L'impulsion sera donc masquée.

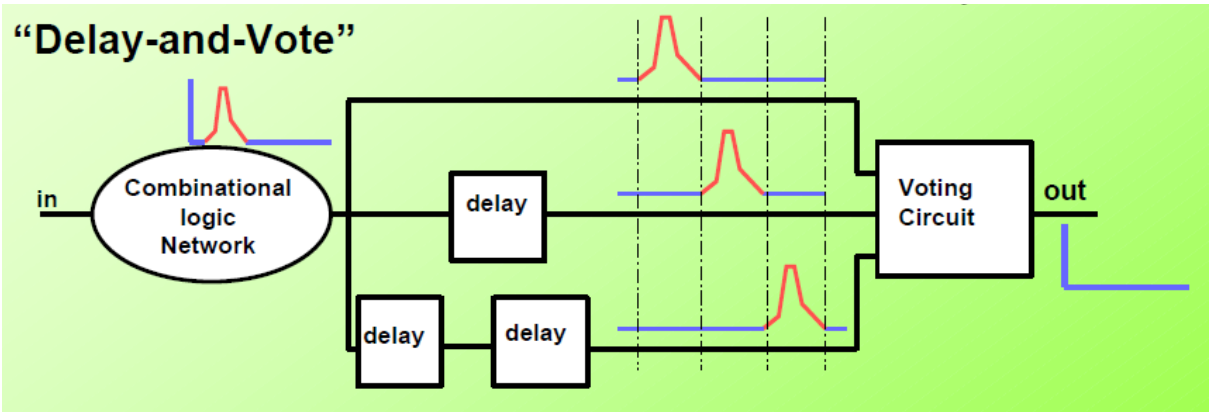


Figure 35 : redondance temporelle [24]

Pour que les impulsions n'arrivent pas en même temps sur le vote majoritaire, le délai N doit être au moins égal à la durée maximale d'une impulsion. De plus, il ne doit pas être trop élevé pour ne pas pénaliser inutilement la rapidité du circuit. On peut donc choisir N de façon à être égal à la durée maximale d'une impulsion générée par l'impact d'une particule. Pour que le vote s'effectue correctement, le signal logique doit être stable durant la phase de vote, ce qui pénalise la rapidité.

XII.5 Dual Modular Redundancy (DMR)

La DMR est une variante de la TMR où il n'y a que deux modules redondants. Dans un fonctionnement normal, les deux modules doivent donner le même résultat. Lorsqu'ils sont en désaccord, alors il n'est pas possible de déterminer directement quel module est correct. Il existe alors plusieurs techniques permettant de déterminer le résultat correct lorsqu'une erreur survient. Par exemple, chacun des modules peut être doté d'un circuit de détection d'erreur. Lorsqu'un désaccord se produira, les signaux d'erreur des deux modules seront vérifiés et celui ne présentant pas d'erreur sera considéré comme correct. Le module en erreur sera alors corrigé par le vrai résultat. Ces modules peuvent par exemple être des blocs mémoires avec un code de détection d'erreur comme un bit de parité.

Il existe une variante qui consiste à utiliser un vote majoritaire dont la sortie est branchée sur une des trois entrées [28]. Les deux autres entrées étant branchées sur deux circuits redondants (Figure 36).

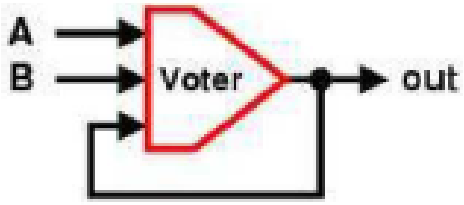


Figure 36 : vote majoritaire avec retour sur l'entrée (*self-voting majority circuit*) [28]

Lorsque les entrées A et B sont en même temps dans l'état '1' alors la sortie est à '1'. Quand A et B sont en même temps à '0' alors la sortie est à '0'. Lorsque A et B sont différents alors la sortie reste dans le même état. Ce type de circuit est appelé C-élément ou bien « guard gates » [29] et est utilisé dans les circuits durcis aux radiations. Comme pour la TMR, si une erreur intervient sur deux signaux, l'erreur ne sera pas masquée. Cependant, le coût en surface est moins important que pour la TMR (33% de moins) et avec une protection comparable aux radiations [28].

XII.6 Augmentation de la capacité des nœuds

Pour réduire la sensibilité aux radiations, on peut augmenter la capacité des nœuds de la cellule mémoire pour augmenter Q_{crit}. Pour cela, on peut par exemple augmenter la taille des transistors pour augmenter la capacité parasite des nœuds ou bien ajouter des capacités aux nœuds sensibles [19]. La Figure 37 montre une structure de capacité

possible. Elle est formée des deux transistors N et PMOS dont les sources et drains sont connectés ensembles (respectivement à la masse et à vdd) tandis que leurs grilles sont connectées ensembles au nœud sensible. La Figure 38 présente une cellule SRAM avec des capacités à ses deux nœuds sensibles. Cette ajout augmente donc Q_{crit} mais augmente également le temps d'écriture et la consommation lors de l'écriture.

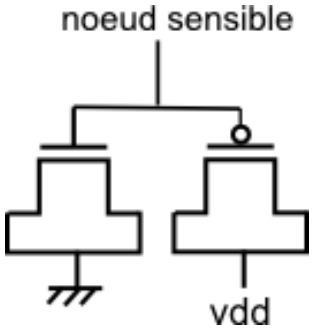


Figure 37 : capacité faite avec les capacités parasites des transistors NMOS et PMOS

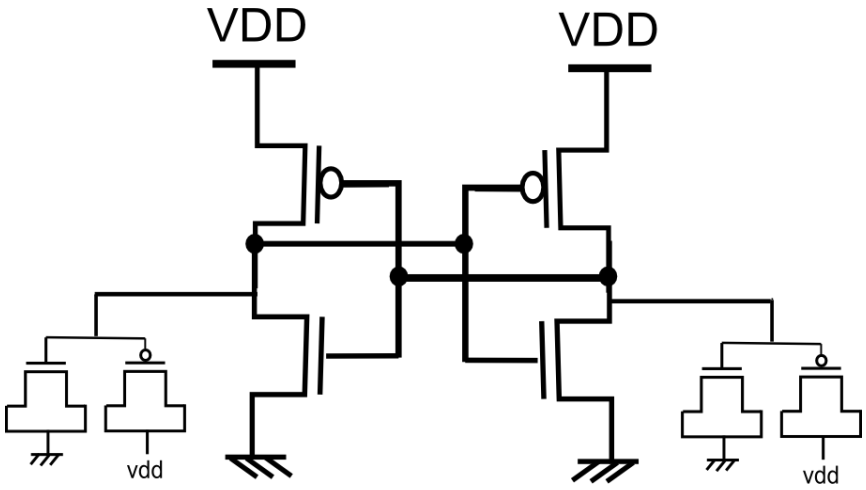


Figure 38 : cellule SRAM avec des capacités à ses deux nœuds sensibles

XII.7 Cellules mémoires durcies

Les circuits les plus sensibles dans un système sont les cellules mémoire car une erreur conduit à une perte d'information. Elle peut être permanente si elle n'a pas été stockée dans une mémoire annexe immune aux radiations. Les cellules mémoires SRAM sont parmi les plus sensibles. Pour rendre une cellule SRAM immune aux radiations, on ajoute des transistors qui permettent de retrouver l'information après l'apparition d'une erreur. L'information est redondante. En effet, au lieu de deux nœuds dans une cellule SRAM classique, il y a quatre nœuds dans une mémoire DICE présentée dans la Figure 39.

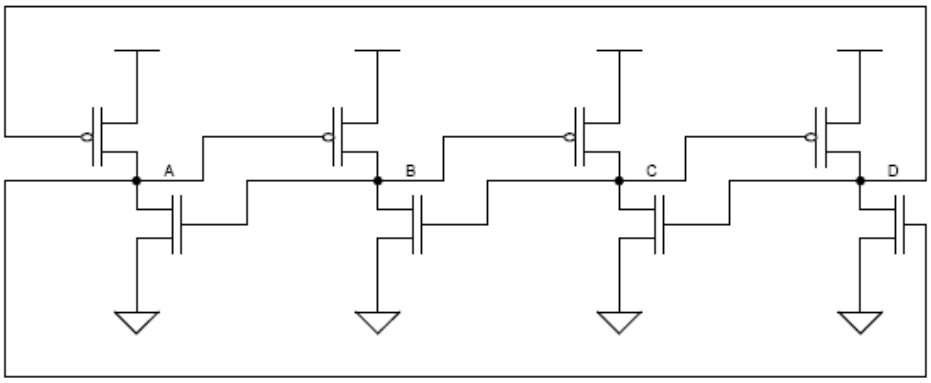


Figure 39 : cellule DICE [18]

Comme le montre la figure, la grille de chaque PMOS est connectée à son nœud mémoire gauche et la grille de chaque NMOS est connectée à son nœud mémoire de droite. Les 0 logiques se propagent vers la droite car un PMOS est activé par un 0 et les 1 logiques se propagent vers la gauche car les NMOS sont activés par un 1. De cette façon, si une erreur survient sur un nœud, l'erreur ne se propage que sur le nœud suivant (gauche ou droite suivant le transistor impacté). Les deux autres nœuds ne sont pas touchés.

Prenons un exemple. Admettons que les nœuds A, B, C et D soient dans la configuration (1, 0, 1, 0) et qu'une particule touche la grille du PMOS connecté au nœud B. Alors la nouvelle configuration est (1, 1, 1, 0). Le 1 sur le nœud B va activer le NMOS du nœud A. Le nœud A va donc perdre son information puisque Vdd et gnd seront court-circuités. Les nœuds C et D, vont être en haute impédance mais ne vont pas perdre leur information car les capacités de ces nœuds n'auront pas le temps de se décharger (les temps de transitions sont rapides). Lorsque la perturbation cesse, le nœud B retrouve son information grâce au nœud C, de même l'information de A est restaurée grâce au nœud D.

Cette structure permet de protéger partiellement la cellule des radiations. En effet, si une radiation touche deux nœuds mémoires alors l'information est changée. Comme souvent, l'inconvénient de cette méthode est une surface silicium plus élevée et donc une plus faible rapidité et une plus forte consommation. Il existe une variante de la DICE : la SERT (Single-Event Resistant Topology). On ajoute un transistor NMOS sur chaque nœud pour éviter l'état de court-circuit et donc diminuer la consommation.

XII.8 Code correcteurs d'erreurs (ECC)

Les codes correcteurs d'erreurs sont très utilisés dans les télécommunications pour coder un canal soumis à du bruit parasite qui peut générer des erreurs de communication. Au niveau des circuits, ils sont utilisés dans les mémoires pour coder les mots ou bien dans les bus de données. Dans tous les cas, il s'agit de fiabiliser la transmission ou le stockage d'information. Le fait de coder un mot consiste à convertir un mot de K bits en un autre mot de N bits (avec N supérieur à K). Dans le cas où les mots code sont séparables, le mot codé a deux parties distinctes : la donnée brute qui est le mot initial de K bits et un ou plusieurs bits de redondance permettant de déterminer si le mot présente une erreur. Dans le cas séparable, il n'y aura pas de phase de décodage car le mot initial est transmis sans modification, il y aura seulement une phase

de détection et éventuellement de correction d'erreur grâce aux bits de redondance. Dans le cas non séparable, le mot initial ne sera pas récupérable directement, il faudra faire des calculs pour le retrouver et déterminer s'il y a eu une ou plusieurs erreurs de transmission. Les codes séparables sont donc plus simples à implémenter, demandent moins de ressources et ont donc moins d'impact en termes de rapidité, de consommation et de surface.

Un code séparable très utilisé est le code de Hamming SECDED (Single Error Correction Double Error Detection). Dans ce cas, les bits de redondance permettent de détecter deux erreurs et en corriger une. Pour corriger plus d'erreur, on peut augmenter le nombre de bits redondants et généraliser : si l'on veut corriger N erreurs alors on doit avoir 2N+1 bits de redondance (on pourra détecter 2N erreurs). Cependant, pour corriger plus d'erreurs, un nombre plus important de bits redondants est nécessaire et des circuits de codage/décodage plus complexe. La surface du circuit et sa consommation augmentent donc et des performances diminuent.

XII.9 Mémoire

Le durcissement d'une mémoire peut également se faire à plusieurs niveaux comme décrit précédemment. Ainsi on peut la durcir au niveau du procédé de fabrication en utilisant un substrat SOI, ou bien en choisissant une technologie mémoire résistante aux radiations. Au niveau du circuit on peut choisir d'implémenter une mémoire avec des cellules durcies au prix de l'augmentation de la surface. La redondance spatiale est également utilisée comme la TMR ou bien les codes correcteurs d'erreurs et en particulier le code de Hamming SECDED (Single Error Correction Double Error Detection). Une autre technique spécifique aux mémoires est appelée scrubbing [32]. Elle consiste à balayer périodiquement toutes les adresses de la mémoire en lisant chaque mot stocké et en vérifiant si une erreur s'est produite à l'aide d'un code correcteur d'erreur comme le code de Hamming. Si une erreur est détectée, elle est corrigée grâce aux bits de redondance. Cette méthode permet d'éviter l'accumulation d'erreurs durant le fonctionnement de la mémoire. Le scrubbing est également utilisé dans les FPGAs et sera donc décrite plus en détail par la suite.

XII.10 Logiciel

Un système peut être durci aux radiations au niveau matériel mais aussi logiciel. Un FPGA peut implémenter un processeur qui exécute un programme contenant du code destiné à détecter et éventuellement corriger une erreur. L'avantage du durcissement au niveau du logiciel est le coût moindre en performance et consommation. Le durcissement du programme consiste à de la redondance. Elle peut se traduire par l'ajout de code : des instructions sont exécutées plusieurs fois. C'est notamment le cas dans la technique de SWIFT (pour Software Implemented Fault Tolerance) [26] où les instructions sont dupliquées puis comparées pour détecter les éventuelles erreurs. Ensuite, il peut se produire des erreurs dans les branchements. Une technique a été proposée dans [27] qui consiste à associer une signature à chaque bloc de code.

XII.11 Durcissement sur un FPGA

Le durcissement d'un circuit implémenté sur un FPGA peut se faire à plusieurs niveaux :

- **choix de la technologie de mémoire de configuration**
- **structure du circuit : on peut utiliser les mêmes techniques de durcissement décrit précédemment au niveau circuit comme le codage, redondance spatiale**
- **algorithme de placement-routage**
- **reconfigurer le FPGA périodiquement pour éviter l'accumulation d'erreur dans la mémoire de configuration : c'est appelé scrubbing**

Détaillons chacune de ces techniques.

XII.11.1 Technologie mémoire

Le choix de la technologie mémoire est une des plus importantes car elle détermine en grande partie la fiabilité du circuit aux radiations. Ainsi, les cellules mémoire Flash de configuration résistent bien aux SEUs. Cependant, elles supportent une faible dose de radiation. A l'inverse, les cellules SRAM supportent une forte dose de radiations (plusieurs centaines de krad [22]) mais sont très sensibles aux SEUs. Donc les FPGAs à mémoires Flash peuvent être utilisés dans les applications à faible dose de radiation (inférieures à 30 krad) mais demandant de la fiabilité comme dans l'aéronautique ou le spatial pour des missions courtes. Quant aux FPGAs à mémoires antifusibles, ils sont totalement résistants aux SEUs mais ne peuvent pas être reprogrammés [21]. Pour utiliser les FPGAs SRAM, il faudra appliquer des méthodes de durcissement au niveau circuit pour les rendre résistants aux SEUs. Dans la pratique, on peut combiner FPGA SRAM et antifuse pour tirer avantage de leurs avantages respectifs. 3 FPGAs SRAM peuvent être utilisés en TMR et un FPGA antifuse implémente le module de vote majoritaire.

XII.11.2 Structure du circuit

Il est possible d'appliquer les méthodes décrites précédemment de durcissement au niveau de la structure du circuit. Ainsi, la redondance spatiale peut être utilisée comme la TMR. Dans ce cas, le module de vote majoritaire peut être implémenté dans un FPGA antifusible insensible aux SEUs car le vote majoritaire doit également être fiable. En effet, s'il était implémenté dans un FPGA SRAM, une seule erreur de configuration au niveau du vote majoritaire pourrait provoquer une erreur à sa sortie. Ensuite, le circuit en TMR peut être implémenté à plusieurs niveaux : les trois modules peuvent être implémentés dans un seul FPGA SRAM (Figure 40) ou bien dans trois FPGAs SRAM (Figure 41).

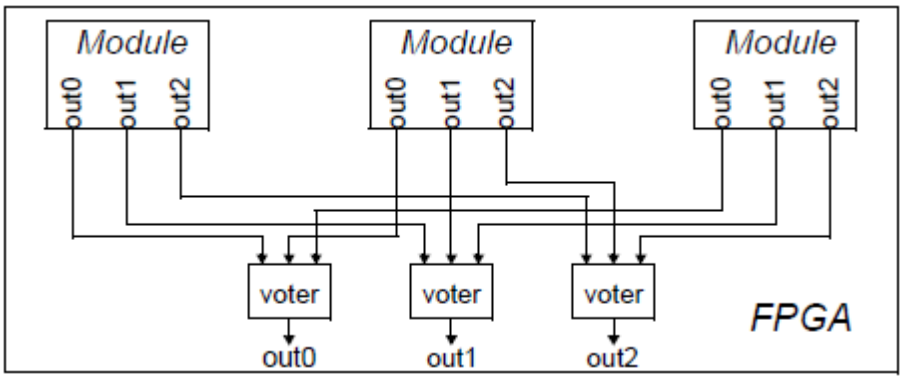


Figure 40 : TMR totalement implémentée dans un FPGA [22]

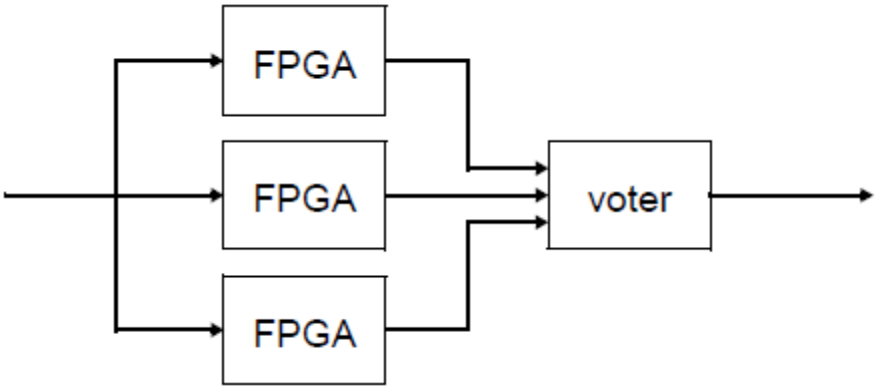


Figure 41 : TMR implémentée dans plusieurs FPGAs [22]

XII.11.3 Algorithme de placement-routage

Le durcissement doit non seulement intervenir au niveau matériel mais aussi au niveau logiciel et notamment à l'algorithme qui implémente le circuit dans le FPGA : l'algorithme de placement-routage. Comme expliqué précédemment, la principale méthode de durcissement d'un circuit dans un FPGA est l'utilisation de la TMR. Pour que la TMR soit efficace, il faut que les trois circuits redondants soient indépendants pour qu'une erreur dans l'un n'affecte pas un autre et génère ainsi de multiples erreurs. Ce cas de figure peut arriver lorsqu'une particule génère une erreur dans le réseau d'interconnexion et de routage. Les signaux de deux modules redondants peuvent, par exemple, être court-circuités et donc générer une erreur dans les deux modules ce qui aboutirait à une erreur en sortie du vote majoritaire (le vote majoritaire ne corrige qu'une seule erreur). Pour limiter ce risque, un algorithme dédié au placement-routage de circuits durcis aux radiations doit être utilisé. Par exemple, l'algorithme RoRA [25] permet de durcir un circuit aux radiations. Il part d'un circuit de base. La TMR est générée automatiquement. Ensuite le circuit est placé et routé de façon à ce que les circuits redondants soient indépendants et ne puissent pas interagir entre eux lorsqu'une particule impact le circuit.

XII.11.4 Scrubbing

Les FPGAs, par rapport à un circuit numérique classique (ASIC, processeur), sont plus sensibles aux effets des radiations. Ceci est dû au fait que la fonctionnalité du FPGA est stockée dans des cellules mémoire qui sont sensibles aux radiations. C'est pourquoi, en plus de tous les types d'effet transitoires décrit précédemment, les FPGAs sont sensibles aux SEUs de configuration. Pour corriger ces erreurs une technique spécifique aux FPGAs à base de SRAM existe : la technique de scrubbing. Elle consiste à reprogrammer la mémoire de configuration de façon périodique afin de corriger les éventuelles SEUs de configuration, exactement comme pour une mémoire car un FPGA est équivalent à une mémoire. Cependant, elle ne permet pas de corriger les SEUs survenues dans les bascules de l'utilisateur. Il faut donc également durcir la partie utilisateur pour les protéger des autres effets des radiations présents dans n'importe quel type de circuit numérique. Ainsi, elle peut être utilisée en combinaison avec la TMR. Notons que le scrubbing ne diminue pas la sensibilité aux radiations mais évite l'accumulation d'erreur.
Il existe deux types de scrubbing [30]:

- **blind scrubbing (Figure 42) : consiste à réécrire la configuration de façon périodique dans le FPGA à partir d'une mémoire non-volatile externe. Ce processus est géré par un circuit de contrôle externe qui lit les données de configuration en mémoire externe puis les écrits dans le FPGA afin de corriger les erreurs de configuration. Pour que le durcissement soit efficace, il faut que le circuit de control et la mémoire externe soient également durcis aux radiations.**

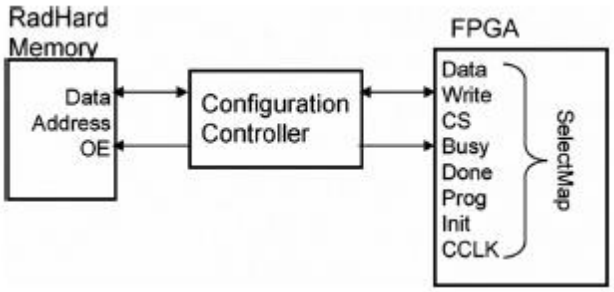


Figure 42 : schéma d'un FPGA avec le contrôleur de scrubbing et la mémoire externe [30]

- **Readback scrubbing (Figure 43) : consiste à lire la configuration du FPGA, détecter si une erreur s'est produite puis corriger les éventuelles erreurs. Pour cela, il y a plusieurs méthodes :**
  - **Vérification puis correction avec copie de référence (*Readback and repair with golden copy*) : la configuration du FPGA est comparée avec une configuration de référence stockée dans une mémoire annexe. Si une erreur est détectée, elle est corrigée grâce à la copie en mémoire. Cela nécessite donc une mémoire externe également durcie aux radiations**
  - **Vérification avec ECC puis correction avec copie de référence (*Readback with ECC and repair with golden copy*) : les données de configuration sont préalablement codées avec, par exemple un code de Hamming. Lors du scrubbing, la configuration est lue puis les bits de codage sont vérifiés pour détecter les erreurs. Ensuite, si une**

erreur est détectée, elle est corrigée grâce à la copie. Cela nécessite donc une mémoire externe également durcie aux radiations

- **Vérification avec ECC puis correction avec ECC (*Readback and repair with ECC)* : les données de configuration sont préalablement codées, par exemple avec un code de Hamming, permettant la correction d'une erreur. Lors du scrubbing, la configuration est lue puis les bits de codage sont vérifiés pour détecter les erreurs. Ensuite, si une erreur est détectée, elle est corrigée grâce au code. C'est la méthode qui demande le moins de composants mais l'inconvénient est que si des erreurs multiples se sont produites dans un mot de configuration, elles ne pourront pas être corrigées.**

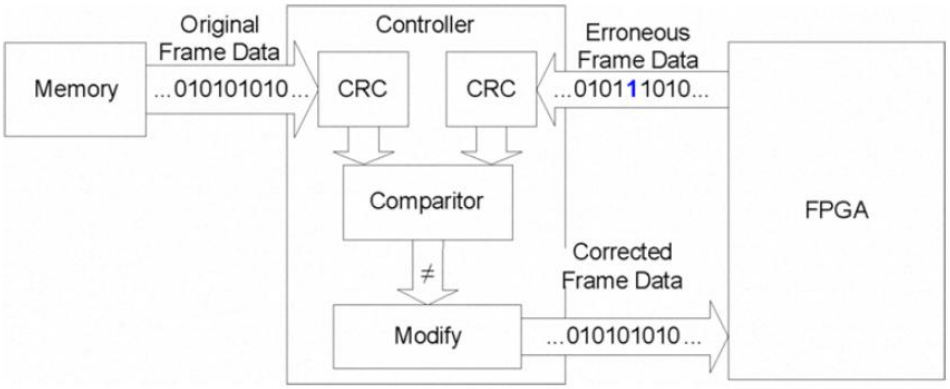


Figure 43 : Readback scrubbing [30]

Dans les deux cas, le scrubbing n'interfère pas avec le fonctionnement du FPGA. Le circuit de l'utilisateur fonctionnement normalement. On peut alors définir deux grandeurs qui caractérisent la phase de scrubbing :

- **la durée de la phase de scrubbing : elle dépend principalement de la taille du FPGA, de la méthode de scrubbing et de la rapidité du circuit de configuration. En effet, la taille du FPGA, c'est-à-dire le nombre de cellules mémoires de configuration, est le premier paramètre car plus il y a de cellules mémoires, plus longue sera leur lecture. Ensuite, la méthode de configuration impliquera un temps plus ou moins long suivant le type de mémoire externe utilisée ou l'utilisation d'un code correcteur d'erreur. Ensuite, la rapidité du circuit de configuration influe également sur la durée de scrubbing. En effet, la configuration du FPGA peut se faire via un port série comme le port JTAG ou bien en parallèle comme avec le port SelectMap [30]**

- **la fréquence de scrubbing (*scrub rate*) : c'est le nombre de phase de scrubbing par unité de temps. Elle est déterminée par le taux de SEU, c'est-à-dire le nombre d'erreur de configuration par unité de temps. Dans la pratique, on choisit une fréquence de scrubbing 10 fois supérieur au taux d'erreur attendu [31] pendant le fonctionnement. Par exemple, avec un taux d'erreur de 1 erreur de configuration par heure, on choisit d'effectuer une phase de scrubbing toutes les 6 minutes [31]. Notons que plus la fréquence de scrubbing est élevée plus le FPGA SRAM se rapproche de la fiabilité d'un FPGA antifuse du point de vue de l'immunité aux SEUs de configuration**

La technique de scrubbing est une méthode efficace lorsqu'elle est utilisée dans un circuit durci avec de la TMR car elle évite l'accumulation d'erreur [33]. En effet, elle permet de corriger une erreur apparue dans un des circuits redondants et limite donc le risque d'erreurs multiples.

XII.12 REFERENCES

[1] I. Kuon and J. Rose: Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203–215, 2007.

[2] Actel Corporation: ProASIC3 flash family FPGAs. *Actel Corporation*, October 2005.

[3] C.-C. Shih, R. Lambertson, F. Hawley, F. Issaq, J. McCollum, E. H. H. Sakurai, H. Yuasa, H. Honda, T. Yamaoka, T. Wada, and C. Hu: Characterization and modelling of a highly reliable metal- to-metal antifuse for high-performance and high-density field-programmable gate arrays. *Proceedings of the 1997 IEEE International Reliability Physics Symposium*, pp. 25–33, 1997.

[4] Actel Corporation: Axcelerator family FPGAs. *Actel Corporation*, May 2005.

[5] Ian Kuon, Russell Tessier, Jonathan Rose: FPGA Architecture: Survey and Challenges. *Foundations and Trends in Electronic Design Automation*, v.2 n.2, p.135-253, February 2008

[6] Architectures and Methodologies for Dynamic Reconfigurable Logic (ADMREL) - Information Societies Technology (IST) Program: Survey of existing fine grain reconfigurable hardware platforms. November 2002. v2.0.

[7] Mingjie Lin, Abbas El Gamal: A Low-Power Field-Programmable Gate Array Routing Fabric. *IEEE transactions on very large scale integration (VLSI) systems*, vol. 17, n°10, October 2009

[8] V. Betz and J. Rose: FPGA routing architecture: Segmentation and buffering to optimize speed and density. in *Proceeding: ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pp. 140–149, February 1999.

[9] E. Ahmed: The Effect of Logic Block Granularity on Deep-Submicron FPGA Performance and Density. Master's thesis, University of Toronto, Department of Electrical and Computer Engineering, 2001.

[10] F. Li, Y. Lin, L. He, D. Chen, and J. Cong: Power modeling and characteristics of field programmable gate arrays. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 11, pp. 1712–1724, November 2005.

[11] A. Ling, D. P. Singh, and S. D. Brown: FPGA technology mapping: A study of optimality. in *Proc. Des. Autom. Conf.*, 2005, pp. 427–432.

[12] http://comelec.enst.fr/hdl/vhdl_exemples.html

[13] http://www.netrino.com/Embedded-Systems/How-To/Reconfigurable-Computing

[14] http://www.fujitsu.com/global/services/microelectronics/technical/lowpower/

[15]
http://www.powermanagementdesignline.com/howto/181500691;jsessionid=NNNDVN1KQOFCUQSNDLPCKHSCJUNN2JVN

[16] Kureshi A.K. and Mohd. Hasan: DTMOS Based Low Power High Speed Interconnects for FPGA. *JOURNAL OF COMPUTERS*, VOL. 4, NO. 10, OCTOBER 2009

[17] ECSS: ECSS-E-ST-50-12C. 31 July 2008

[18] Sandro BONACINI PhD thesis. Development of Single-Event Upset hardened programmable logic devices in deep submicron CMOS. Institut National Polytechnique de Grenoble, 16th November 2007.

[19] Shubu Mukherjee: Architecture Design For Soft Errors. *Elsevier*, February 2008

[20] A. Lesea: Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits WP286 (v1.0.1). *Xilinx*, May 2009

[21] Ramin Roosta. A Comparison of Radiation-Hard and Radiation-Tolerant FPGAs for Space Applications. *NASA Electronic Parts and Packaging Program*, 30 December 2004.

[22] Xilinx: Radiation-Hardened, Space-Grade Virtex-5QV Family Overview. *Xilinx*, 8 Mars, 2012

[23] Sandi Habinc: Functional Triple Modular Redundancy (FTMR). *Gaisler Research*, December 2002

[24] M. P. Baze, J. C. Killens, R. A. Paup, W. P. Snapp: SEU Hardening Techniques for Retargetable, Scalable, Sub-Micron Digital Circuits and Libraries. Presentation, *Boeing Space and Communications*, 2002

[25] L. Sterpone, M. Reorda, M. Violante : RoRA: a reliabilityoriented place and route algorithm for SRAM-based FPGAs. *Research in Microelectronics and Electronics*, 2005, Volume 1, 2005. p.173-176.

[26] G. A. Reis et al: SWIFT: Software Implemented Fault Tolerance. *In Proc. of the Int'l Symp. on Code Generation and Optimization*, p. 243-254, Mar. 2005.

[27] N. Oh, P. P. Shirvani, and E. J. McCluskey: Control-flow checking by software signatures. Volume 51, pages 111– 122, March 2002.

[28] J. Teifel: Self-voting dual-modular-redundancy circuits for single event transient mitigation. *IEEE Trans. Nucl. Sci.*, vol. 55, no. 6, pp.3435 -3439 2008

[29] A. Balasubramanian, B. L. Bhuva, J. D. Black, and L. W. Massengill: RHBD techniques for mitigating effects of single-event hits using guard-gates. *IEEE Trans. Nucl. Sci.*, vol. 52, no. 6, pp. 2531–2535, Dec. 2005.

[30] J. Heiner , B. Sellers , M. Wirthlin and J. Kalb: FPGA partial reconfiguration via configuration scrubbing. *11th Int. Workshop, Field-Programmable Logic and Applications and Lecture Notes in Computer Sci.*, 2009

[31] C. Carmichael, M. Caffrey, and A. Salazar: Correcting single event upsets through Virtex partial configuration. *Xilinx Corporation*, Tech. Rep. XAPP216 v1.0, June 2000.

[32] A. Saleh, 1. Serrano, and J. Patel: Reliability of scrubbing recovery techniques for memory systems. *Reliability, IEEE Transactions on*, vol. 39, no. 1, pp. 114-122, Apr 1990.

[33] M. Berg: The NASA goddard space flight center radiation effects and analysis group Virtex 4 scrubber. in *Xilinx Radiation Test Consortium (XRTC) Meeting, 2007*.

[34] INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS 2011 EDITION EMERGING RESEARCH DEVICES, 2011

[35] Cargnini, L.V., L.Torres, L. Sassatelli, G. :Improving the Reliability of a FPGA Using Fault-Tolerance Mechanism Based on Magnetic Memory (MRAM), 2010 International Conference on Reconfigurable Computing and FPGAs (ReConFig), 13-15 Dec. 2010

XIII. Les MRAMs

La MRAM (Magnetic Random Access Memory) est une nouvelle technologie de mémoire. Elle combine les avantages de nombreuses mémoires actuelles. Ainsi, elles ont la non-volatilité de la mémoire Flash, une densité proche de la DRAM, une vitesse d'écriture beaucoup plus rapide et plus économe en énergie que la mémoire Flash (de l'ordre de la pJ/bit pour les STT MRAMs contre la centaine de pJ/bit pour la Flash [34]) et sont résistantes aux radiations. Elles concurrencent d'autres nouvelles technologies de mémoire comme la ReRAM (resistive RAM), FeRAM (Ferroelectric RAM), CBRAM (Conductive Bridge RAM), PCRAM (Phase Change RAM), memristor ou les Redox RAMs. Cependant, la MRAM est déjà commercialisée par Everspin depuis 2006 [1] et fait partie des nouvelles technologies de mémoire prometteuses comme l'indique un rapport de l'ITRS [2]. De nouvelles Startups, fabricants de MRAMs, devraient bientôt commercialiser des mémoires standalone comme Crocus-Technology qui a été créée en 2006 par le laboratoire Spintec et qui va commercialiser une technologie de MRAM développée par le laboratoire.

La MRAM est le fruit de recherches menées dans le domaine de la spintronique qui ont déjà abouti à de grandes avancées dans le domaine du stockage d'information. Une des plus importantes est l'invention de la vanne de spin par Bernard Dieny et al. [3] utilisée dans les disques durs et qui a permis d'accroitre significativement la quantité d'information stockée. La spintronique vise à développer des moyens de stockage de l'information en utilisant une des propriétés de l'électron, le spin. C'est le spin des électrons d'un matériau qui détermine ses propriétés magnétiques. Ainsi, les dispositifs inventés dans le domaine de la spintronique stockent l'information sous forme d'une aimantation contrairement à des circuits électroniques tels les cellules SRAM ou Flash qui utilisent la charge de l'électron pour stocker l'information.

XIII.1 Spintronique

Le nom de cette science vient du spin de l'électron qui est une de ses propriétés au même titre que sa masse ou sa charge. Le spin d'un électron peut avoir deux états : spin « up » ou spin « down ». Il est communément décrit comme la rotation de l'électron sur lui-même : la rotation dans un sens correspond au spin up et dans l'autre sens, le spin down. Cependant cette description est fausse car on ne peut pas décrire rigoureusement cette propriété quantique en faisant appel à la mécanique classique.

C'est le spin des électrons d'un matériau qui détermine ses propriétés magnétiques. Ainsi, dans un matériau où les moments magnétiques sont désordonnés, la somme de ces moments est nulle et donc le matériau n'est pas aimanté. C'est un matériau paramagnétique (Figure 44a) comme le cuivre ou l'aluminium. Dans un matériau ferromagnétique (Figure 44b) comme le fer ou le nickel, les moments magnétiques sont tous alignés dans la même direction ce qui se traduit par une aimantation. Dans un matériau anti-ferromagnétique (Figure 44c), les moments magnétiques sont parallèles mais dans des directions opposées à leur voisin. Les moments sont donc localement orientés dans la même direction mais avec des sens opposés ce qui se traduit par une aimantation nulle à l'échelle macroscopique.

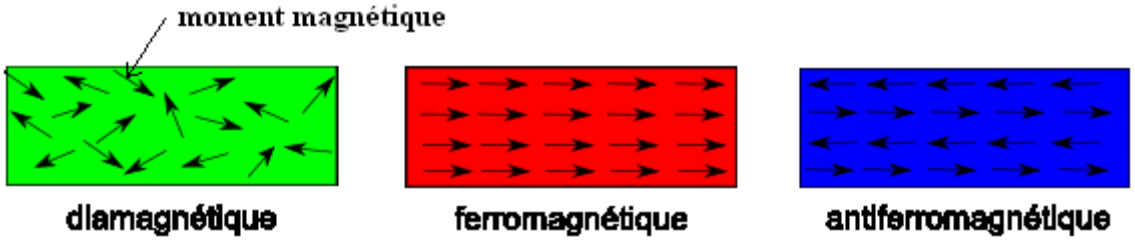


Figure 44 : classification des matériaux en fonction de l'orientation de leur moment magnétique [16]

Décrivons maintenant le trajet des électrons dans un matériau ferromagnétique ayant une certaine aimantation. Leur comportement est différent suivant que le spin des électrons majoritaires a la même direction que l'aimantation du matériau ou pas. En effet, s'ils ont la même direction, les électrons de spin majoritaire traverseront le matériau sans résistance tandis que les électrons de spin minoritaire seront réfléchis, ralentis ou bloqués. La résistance de l'ensemble sera alors faible. Dans le cas contraire, si le spin des électrons majoritaires est dans la direction opposée à celle de l'aimantation, alors les électrons majoritaires seront ralentis, bloqués ou bien réfléchis tandis que les électrons minoritaires traverseront sans résistance. La résistance sera plus forte. Ce phénomène est appelé magnéto-résistance et est à la base de la spintronique. Il a d'abord été mentionné par Mott [17] puis démontré expérimentalement et théoriquement [18][19] à la fin des années 60 quand la différence de résistance n'était que de quelques pourcents.

Une des principales découvertes de la spintronique est la découverte de la magnétorésistance géante (GMR pour Giant Magneto-Resistance) en 1988 par Albert Fert et Peter Grunberg, de façon indépendante, et qui leur a valu le prix Nobel de physique en 2007. Le phénomène de la GMR intervient dans les couches minces. La Figure 45 montre deux schémas illustrant la GMR. Deux couches minces ferromagnétiques sont isolées par une couche non ferromagnétique (comme du cuivre). L'effet de la GMR est un changement de résistance des couches minces lorsque l'on applique un champ magnétique extérieur pour faire changer l'orientation relative des aimantations des deux couches ferromagnétique. Dans l'exemple n°1, les aimantations des deux couches ferromagnétiques sont parallèles. Les électrons de spin « up » se propageront donc à travers les couches sans résistances tandis que les électrons de spin « down » se propageront mal. Dans le cas où les aimantations des couches sont antiparallèles, les électrons de spin « up » se propageront bien dans la première couche mais mal dans la deuxième et le contraire dans le cas des électrons de spin « down ». On aboutit donc, dans le premier cas, à une résistance plus faible que dans le second.

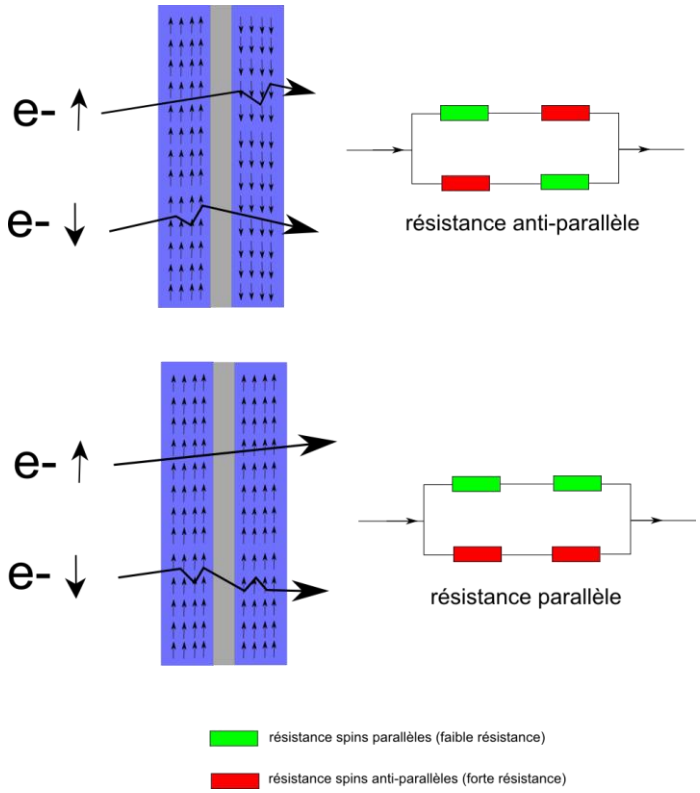


Figure 45 : description de la GMR dans des couches minces

Une conséquence de la découverte de la GMR a été l'invention de la vanne de spin. Dans ce cas, le dispositif est constitué de plusieurs couches minces. Le plus simple est 3 couches minces : deux couches ferromagnétiques séparées par une couche non-ferromagnétique conductrice. Une des deux couches ferromagnétiques a une aimantation fixe. C'est la couche de référence. L'autre couche ferromagnétique a une aimantation libre, c'est la couche de stockage, on peut donc l'orienter dans les deux directions, parallèle ou antiparallèle à celle de la couche de référence, avec un comportement hystérétique. On change donc l'orientation de la couche de stockage en fonction de la donnée à écrire. Pour lire l'information, il suffit de faire passer un courant à travers l'empilement et déterminer si l'on a une faible ou une forte résistance. Cette invention a permis d'accroitre fortement la densité des disques durs actuels et est la principale application industrielle de la spintronique.

Les phénomènes physiques mis en œuvre dans les nanotechnologies décrites dans ce chapitre peuvent être décrits par l'équation de Landau-Lifshitz-Gilbert (équation LLG Équation 1).

$$\frac{\partial M}{\partial t} = \underbrace{-\gamma M \times H_{eff}}_{\text{Precession}} \underbrace{-\frac{\alpha\gamma}{M_s} M \times \left(M \times H_{eff}\right)}_{damping} \underbrace{-\gamma a M \times \left(M_P \times M\right) - \gamma b \left(M \times M_P\right)}_{STT}$$

Équation 1 : équation de Landau-Lifshitz-Gilbert

Elle décrit la dynamique de l'aimantation M d'un matériau en prenant en compte le champ magnétique effectif Heff qui comprend le champ magnétique extérieur, ainsi que les champs internes influençant l'aimantation (c'est-à-dire le champ d'anisotropie, le champ de magnétisation …). P est la polarisation en spin de la couche de référence, α est le coefficient d'amortissement de Gilbert, γ est le coefficient gyromagnétique. On peut voir qu'il y a trois principaux termes dans l'équation. Le premier correspond au

mouvement de précession de l'aimantation M autour du champ effectif Heff. Le second correspond à l'amortissement qui s'oppose au mouvement et qui maintient l'aimantation dans son état final. Le troisième terme correspond au couple de transfert de spin (Spin transfer torque, STT). Ce terme permet de changer l'aimantation ou non en fonction des caractéristiques de la densité de courant de façon interne, c'est-à-dire sans appliquer de champ magnétique extérieur. Pour cela, ce terme doit être prépondérant devant le terme d'amortissement. Dans le terme de STT, deux effets interviennent, à savoir respectivement le couple planaire et le couple perpendiculaire avec les coefficients correspondants a et b. On remarque que si les termes d'amortissement et de STT se compensent alors l'aimantation oscille de façon permanente à une fréquence de l'ordre du GHz. Ce phénomène fait l'objet d'étude pour une utilisation dans le domaine des oscillateurs haute fréquence.

XIII.2 La JTM

Les MRAMs sont principalement composées de Jonctions Tunnel Magnétiques (JTM pour Magnetic Tunel Junction, ou JTM). Une JTM est composée de trois couches. Deux couches ferromagnétiques séparées par une couche isolante. Le fonctionnement est similaire à celui de la vanne de spin. Une des couches ferromagnétiques, appelée couche de référence, a une aimantation fixe tandis que l'autre couche, la couche de stockage, peut voir son aimantation orientée dans les deux sens, parallèle ou antiparallèle à celle de la couche de référence, avec un comportement hystérétique. Ainsi, lorsque les aimantations des deux couches sont dans l'état parallèle, la résistance de la JTM est faible tandis que dans l'état antiparallèle, la résistance est forte. Le courant passe à travers la jonction par effet tunnel. L'effet physique qui intervient dans le cas de la JTM est la magnétorésistance tunnel (TMR pour Tunneling Magneto-Resistance). La Figure 46 représente un cycle d'hystérésis. Il montre qu'il est nécessaire d'appliquer un champ magnétique suffisamment important pour passer de l'état parallèle à l'état antiparallèle. Ce champ est appelé champ coercitif et doit être orienté dans un sens ou dans l'autre en fonction de la donnée à écrire. Lorsque que le champ est en dessous de cette valeur (en valeur absolue), l'état de la jonction reste stable, la donnée est donc bien stockée de façon permanente.

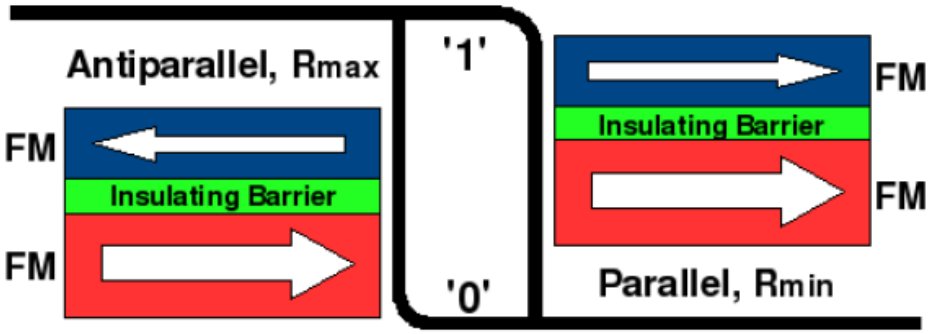


Figure 46 : fonctionnement d'une JTM

La stabilité de la jonction est la capacité à maintenir l'aimantation de la couche de stockage dans la même direction durant une longue période (idéalement indéfiniment). La stabilité est liée à l'anisotropie qui tend à maintenir l'aimantation dans une direction.

En effet, lorsque le champ magnétique extérieur est nul, l'aimantation d'un matériau ferromagnétique s'aligne spontanément parallèlement à une direction. Il existe deux types d'anisotropie : anisotropie magnétocristalline et anisotropie de forme. Dans le cas de l'anisotropie magnétocristalline, le moment magnétique a tendance à s'aligner sur un axe, l'axe dit facile, dont la direction dépend de la structure cristalline du matériau. L'anisotropie de forme est due à la forme de la jonction tunnel. A champ nul, l'aimantation du matériau est parallèle à l'axe facile qui est la position dans laquelle l'énergie d'anisotropie est minimale. C'est pourquoi, dans le cas d'une JTM en forme d'ellipse par exemple, le moment magnétique s'aligne dans la direction du plus grand axe et reste dans le plan des couches. Lorsqu'un champ magnétique extérieur est appliqué, l'anisotropie s'oppose au changement d'aimantation car elle tend à l'orienter dans la direction de l'axe facile. L'anisotropie permet donc une bonne stabilité de l'aimantation de la couche de stockage et permet en particulier d'éviter les changements spontanés d'aimantation dus aux fluctuations thermiques. Cependant, il est nécessaire d'appliquer un champ magnétique extérieur élevé pour contrer l'anisotropie, ce qui augmente l'énergie nécessaire à l'écriture de la jonction. Il y a donc un compromis à trouver entre stabilité de la donnée et énergie d'écriture de la jonction. En pratique, les mémoires non-volatiles ont une stabilité, autrement appelée durée de rétention, de 10 ans.

Un des paramètres les plus importants pour une JTM est le ratio de TMR qui permet d'indiquer la différence de résistance entre l'état parallèle et antiparallèle exprimée en un pourcentage. Il est définit comme le rapport suivant :

$$TMR = (Rap - Rp) / Rp$$

Où Rap est la résistance dans l'état antiparallèle et Rp la résistance dans l'état parallèle. La TMR dépend principalement des matériaux utilisés dans la jonction pour faire la couche isolante. Par exemple, pour une JTM avec une barrière tunnel en alumine, on peut atteindre une TMR de 70 % [6] et même 500% avec une barrière en MgO [7]. Dans notre cas, les JTMs utilisées pendant la thèse avait une TMR de 120%.

Un autre paramètre important de la JTM est son RA qui est le produit de la surface par sa résistance et qui permet de chiffrer la résistance de la barrière tunnel indépendamment de sa surface. Il est exprimé en Ω.µm². Dans notre cas, les JTMs avaient un RA de 25 Ω.µm². Avec un diamètre de 130nm pour les JTMs, la résistance dans l'état parallèle est de 1,9 kΩ. Avec une TMR de 120 %, on a une résistance dans l'état antiparallèle de 4,2 kΩ.

La forme et les dimensions de la JTM sont également des paramètres car ils permettent d'estimer la durée de rétention, comme expliqué précédemment, et l'énergie nécessaire à l'écriture, c'est-à-dire pour modifier l'orientation de l'aimantation de la couche de stockage. Par exemple, dans le cas d'une mémoire de processeur embarqué, on peut avoir besoin d'une mémoire MRAM dont la durée de rétention est longue mais avec peu de cycles d'écritures. Dans ce que cas, on peut choisir d'avoir une JTM de forme elliptique. Au contraire, si l'on conçoit une mémoire cache avec des MRAMs, on peut choisir de se contenter d'une durée de rétention courte, par exemple quelques mois, afin de diminuer l'énergie d'écriture. Dans ce cas, la forme de la JTM se rapprochera d'un disque.

XIII.3 Lecture

La lecture de l'information consiste donc à déterminer si la résistance de la jonction est forte ou faible. Il existe plusieurs techniques décrites dans la littérature. Décrivons les principaux circuits utilisés.

La principale méthode de lecture utilisée dans les mémoires MRAM consiste à comparer le courant passant à travers la JTM à un courant moyen de référence. Le courant de référence est le courant moyen entre courant dans l'état parallèle et antiparallèle traversant une JTM. Le courant moyen est typiquement généré en mettant en parallèle des JTMs. Il doit y avoir autant de JTMs dans l'état parallèle qu'antiparallèle. Le nombre de JTMs utilisé doit être suffisamment grand pour masquer les dispersions des caractéristiques des JTMs. La Figure 47 montre les cellules de références et les JTMs à lire. Le courant de référence est comparé au courant traversant la JTM à lire avec un amplificateur différentiel.

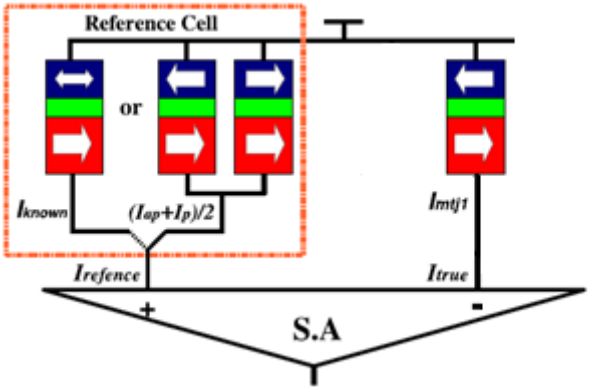


Figure 47 : technique de lecture par référence

L'inconvénient de cette méthode est la sensibilité aux défauts de fabrication qui peuvent mener à une confusion des états parallèle et antiparallèle. En effet, une JTM pourrait, par exemple, avoir une résistance dans l'état parallèle relativement élevée. Le circuit de lecture pourrait alors la confondre avec une résistance antiparallèle.

Une autre technique de lecture consiste à associer deux JTMs pour constituer un bit de donnée (Figure 48). Les JTMs sont dans un état complémentaire. L'une est dans l'état parallèle tandis que l'autre est dans l'état antiparallèle. Une des JTMs contient donc la donnée brute tandis que l'autre contient le bit complémentaire. Pour lire, on ajoute donc un amplificateur différentiel qui détecte la différence de courant, comme décrit précédemment, et détermine ainsi la donnée stockée. L'avantage de cette méthode est une meilleure fiabilité de lecture par rapport aux dispersions globales des caractéristiques : les deux JTMs étant proches physiquement, les variations de procédé ont de fortes chances d'être faibles. De plus, l'approche différentielle double le signal disponible pour la lecture. Cependant, il faut deux JTMs pour représenter un bit d'information. On double donc le nombre de JTMs nécessaires. Il y a donc une perte en densité.

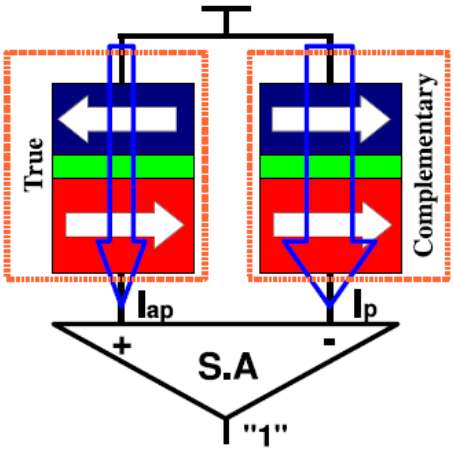


Figure 48 : twin cell

Une autre technique de lecture, appelée « self reference », est plus compliquée que les méthodes précédentes. Il y a plusieurs étapes. Premièrement, un courant est appliqué à travers la JTM à lire. Le résultat de cette lecture, sous forme de tension, est stocké de façon temporaire. Ensuite, la JTM est écrite avec un état connu, par exemple parallèle, puis un courant est appliqué afin d'avoir une image sous forme de tension de l'état de la JTM. Ensuite, la JTM est écrite avec l'état opposé, donc antiparallèle, afin de déterminer de la même façon la tension associée à cette état. Pour finir, la première tension est comparée avec la tension moyenne des deux dernières tensions pour déterminer la donnée originale. La donnée initiale est ensuite réécrite. Cette méthode ne demande pas de cellule de référence donc elle est plus dense. Mais le procédé de lecture est long, le circuit de lecture est complexe, et la consommation à la lecture est élevée. Cependant, ce procédé de lecture est plus fiable que la première technique car il permet de réduire les erreurs de lecture dues aux variations du procédé de fabrication, la résistance de la jonction étant comparée à ses propres résistances parallèle et antiparallèle. Mais la complexité de cette technique la rend difficilement utilisable. Notons que Crocus-Technlology a mis au point un nouveau type de jonction tunnel permettant de simplifier la mise en œuvre de cette technique. Dans ce cas, la couche de référence de la JTM n'a pas d'aimantation réellement fixe. On doit générer un champ magnétique externe, sous la couche de référence, qui va donner un sens à l'aimantation de la couche de référence. En faisant varier le sens de l'aimantation de la couche de référence, on peut déterminer l'état de la couche de stockage en observant la variation de courant. Si le courant diminue, alors on est passé de l'état parallèle à l'état antiparallèle et le contraire si le courant augmente. Cette méthode de lecture est très fiable par rapport aux dispersions des caractéristiques et elle ne nécessite pas de changer l'état de la jonction pour la lire. De plus, cette méthode simplifie la technique de lecture self-reference car il n'est pas besoin de lire et écrire plusieurs fois les données puis de les stocker. Cette technologie est donc plus rapide, plus simple, plus économe en énergie et est donc commercialisable.

La méthode de lecture adoptée dépend donc en partie des variations de procédé de fabrication mais également de la TMR qui doit être suffisamment élevée pour se protéger des différents bruits parasites durant la lecture. Afin de différencier l'état parallèle de l'état antiparallèle correctement, il faut, en pratique, une TMR de plus de 70%. En dessous de 70%, on peut utiliser des twin cells pour une lecture plus fiable.

XIII.4 Ecriture par champ

L'écriture d'une JTM consiste à changer l'orientation de l'aimantation de la couche de stockage. Il existe deux types d'écriture :

- **En appliquant un champ magnétique extérieur**
- **En appliquant un courant à travers la JTM**

Les paragraphes suivants décrivent les principales technologies de JTM. Certaines sont ou vont bientôt être commercialisées, d'autres sont en phase de recherche. Les JTMs à écriture par champ sont d'abords présentées car ce sont les premières générations de JTMs et donc les plus matures. Ensuite, sont présentées les JTMs à écriture par courant interne qui sont encore en phase de développement ou de recherche mais constituent l'avenir des MRAMs.

XIII.4.1 FIMS

La première génération de JTMs a été la FIMS (Field Induced Magnetic Switching). La Figure 49 montre le schéma de principe d'écriture d'une JTM en technologie FIMS. Elle consiste à générer deux champs magnétiques perpendiculaires pour changer la direction de l'aimantation de la couche de stockage. Ces champs magnétiques sont générés en faisant passer un courant suffisamment fort dans deux lignes de métal perpendiculaires. Grâce à la loi de Biot-Savart, on peut déterminer la valeur du champ magnétique généré. La somme vectorielle des deux champs magnétiques donne un champ magnétique suffisamment grand pour changer la direction de l'aimantation de la couche de stockage. Dans une matrice de JTMs en FIMS, les lignes de champ sont organisées en lignes verticales et horizontales. De cette façon, seule la JTM à l'intersection des deux lignes où l'on applique un courant sera écrite.

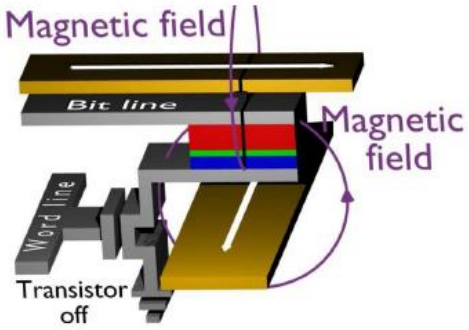


Figure 49 : FIMS JTM

Cependant, des problèmes d'écriture existent car il est possible qu'une JTM soit écrite alors qu'une seule de ces lignes de champ génère un champ. Ce problème augmente avec la densité des mémoires. Pour résoudre ce problème de sélectivité, Savtchenko [8] de chez Freescale, a proposé une solution consistant en une nouvelle structure de JTM, présentée dans la Figure 50. Dans ce cas, la couche de stockage est remplacée par une structure à trois couches. Deux couches ferromagnétiques avec des aimantations opposées avec une couche de couplage. Cette structure est appelé SAF

pour Synthetic Anti-Ferromagnetic. De plus, l'axe le plus long de la JTM est incliné de 45° avec la ligne de champ horizontale (Figure 51).

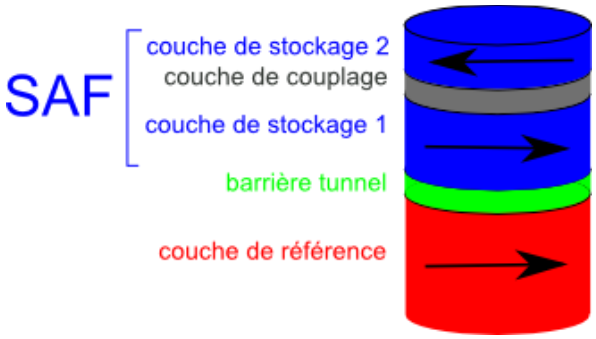


Figure 50 : Toggle JTM

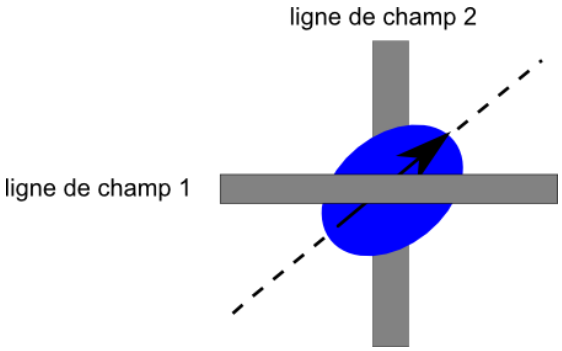


Figure 51 : lignes d'écriture par champ

Cette structure est associée à une technique d'écriture spécifique, utilisant un champ magnétique tournant, appelée Toggle. Elle consiste à appliquer la même séquence de courant pour faire basculer l'aimantation dans un sens ou dans l'autre. Pour écrire, il faut donc d'abord lire la donnée stockée pour ensuite la comparer à la donnée à écrire. Si les deux valeurs sont différentes, alors on applique la séquence de courant pour faire basculer l'aimantation. Lorsqu'elles sont identiques, il n'est pas nécessaire d'appliquer la séquence. Le fait d'appliquer une séquence d'impulsions de courant pour générer un champ tournant permet de s'affranchir des problèmes de sélectivité. Cependant, le courant d'écriture reste très élevé. Pour l'instant, cette technologie de JTM est la seule à être commercialisée. C'est maintenant Everspin, spin-off de Freescale, qui la vend. L'inconvénient de cette technologie est sa capacité à être miniaturisée, car la barrière d'énergie diminue avec la taille de la jonction [9] alors que l'énergie d'écriture diminue peu. Cette technologie n'est donc pas réellement scalable. En dessous de 100 nm, la rétention n'est plus assurée à cause des fluctuations thermiques.

XIII.4.2 TAS

La technologie TAS (Thermally Assisted Switching) a été inventée au laboratoire Spintec et va être commercialisée par la startup Crocus Technologie, une spin-off du laboratoire. De plus, c'est cette technologie qui a été utilisée dans le cadre de cette thèse. Elle va donc être décrite plus en détail. Elle consiste à chauffer la cellule pour changer plus facilement son aimantation. Les différentes couches d'une TAS sont présentées dans le Figure 52.

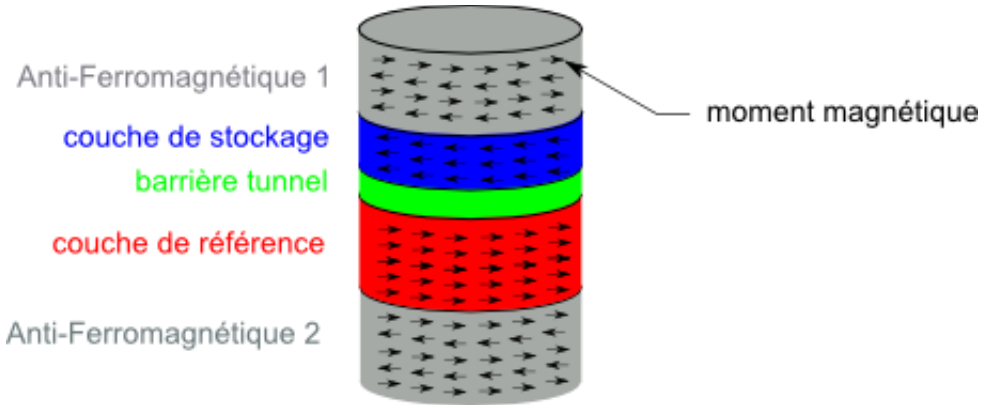


Figure 52 : TAS JTM

On retrouve l'empilement des deux couches ferromagnétiques séparées par un matériau isolant, la barrière tunnel. La JTM a une forme qui approche celle d'un disque afin de modifier l'orientation de l'aimantation de la couche de stockage avec peu d'énergie. Afin de garantir la stabilité, une couche de matériau antiferromagnétique a été couplée à chaque couche ferromagnétique. En effet, la couche antiferromagnétique a une aimantation globalement nulle mais localement les moments magnétiques adoptent la même direction que la couche voisine. On parle d'énergie d'échange permettant de stabiliser l'aimantation à l'interface. C'est donc la couche antiferromagnétique qui maintient la stabilité des deux couches ferromagnétiques.

Pour écrire la JTM, la cellule va être chauffée jusqu'à dépasser une température dite de blocage de la couche antiferromagnétique couplée à la couche de stockage qui se situe entre 150 et 250°C. C'est alors que les moments magnétiques de la couche AFM1 sont désordonnés et ne maintiennent plus la stabilité de la couche de stockage. On peut donc changer l'orientation de son aimantation en appliquant un champ magnétique extérieur relativement faible comparé à celui nécessaire pour la technologie FIMS. Le chauffage de la JTM est assuré en faisant passer un courant à travers la jonction. Afin que le chauffage se fasse le plus efficacement possible, une barrière thermique peut être ajoutée. Elle consiste en un matériau qui a une faible conductivité thermique, pour maintenir la chaleur dans la JTM et une résistance relativement faible. Notons que la température de blocage de l'antiferromagnétique associé à la couche de référence est suffisamment élevée pour ne pas la dépasser (plus de 300°C). Le champ magnétique extérieur est maintenu jusqu'à ce que la température de la cellule descende en dessous de la température de blocage (Figure 53). Lors de ce refroidissement, la couche antiferromagnétique va retrouver son ordre, mais en s'alignant, à l'interface, avec la couche de stockage qui a elle-même été retournée. A la fin de la phase d'écriture, l'énergie d'échange va donc maintenir l'aimantation de la couche de stockage dans sa nouvelle orientation. Cette technique d'écriture permet de combiner les aspects rétention et consommation à l'écriture, puisqu'à température ambiante, le champ d'échange assure une stabilité importante tandis qu'au-delà de la température de blocage, celui-ci disparait et la forme circulaire de la jonction assure une consommation à l'écriture relativement faible.

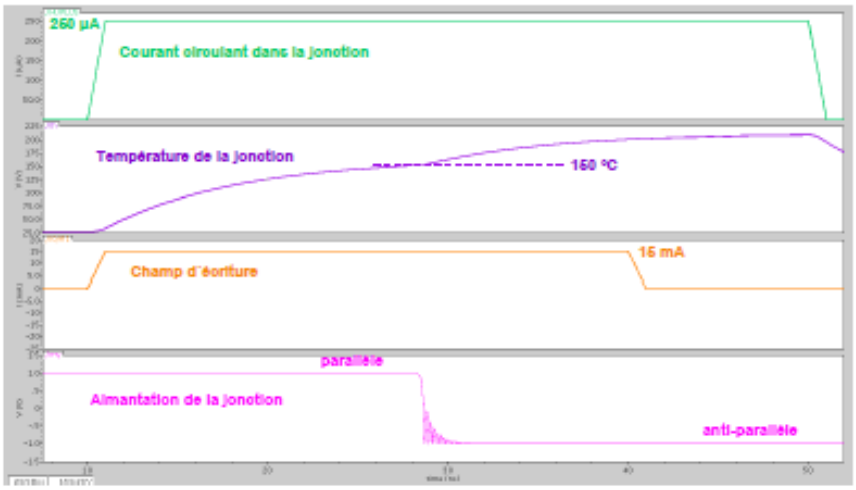


Figure 53 : cycle d'écriture d'un JTM en technologie TAS [16]

Par exemple, dans le cas de cette thèse, les cellules TAS utilisées avaient une température de blocage de 170°C environ. Elles pouvaient atteindre cette température en 10 ns environ et la phase de refroidissement durait 20 ns. Le cycle d'écriture est donc de 30 ns. L'écriture est donc très rapide comparée à celle d'une cellule de mémoire Flash. Ensuite la TAS a une très bonne stabilité car elle est assurée par l'antiferromagnétique et non par l'anisotropie. De ce fait, l'énergie nécessaire à l'écriture est plus faible pour la technologie FIMS. L'autre avantage par rapport à la FIMS est la grande sélectivité. En effet, seules les cellules chauffées sont écrites. Le champ magnétique extérieur ne peut pas écrire les cellules « froides ». Enfin, la TAS peut être miniaturisée car le courant de chauffage diminue avec la surface de la cellule.

Le courant de chauffage est unidirectionnel ce qui simplifie le circuit électronique de commande. Quant au champ magnétique extérieur, il est généré en faisant passer un courant dans un sens ou dans l'autre, en fonction de la donnée à écrire, à travers une piste de métal passant sous la jonction. Dans notre cas, le courant de chauffage était de l'ordre de quelques centaines de microampères pour une cellule TAS de 130 nm et un courant de champ de 15 mA environ. La consommation à l'écriture de la TAS, en technologie 130 nm, est fortement pénalisée par le courant nécessaire pour générer le champ magnétique d'écriture. Notons, cependant, que l'on peut partager la ligne de champ entre plusieurs JTMs afin de les écrire en même temps et ainsi diminuer l'énergie d'écriture par bit. Ceci nécessite une écriture en deux phases : dans la première phase, le champ magnétique nécessaire pour écrire un « 1 » logique par exemple est généré, et les cellules devant être écrites à « 1 » sont chauffées. Dans la seconde phase, le champ magnétique nécessaire pour écrire un « 0 » est généré et les autres cellules sont chauffées. On peut remarquer cependant qu'une TAS consomme trop de courant pour pouvoir être utilisée dans les applications de logiques pures où elle devrait être régulièrement écrite. Elle peut néanmoins être utilisée pour stocker un bit de façon non-volatile, pour rendre une SRAM non-volatile par exemple.

XIII.5 Ecriture par courant polarisé en spin

XIII.5.1 STT planaire

Les paragraphes précédents portaient sur l'écriture par champ magnétique externe. Cependant, il est possible d'utiliser un courant polarisé en spin pour changer l'orientation de la couche de stockage de la jonction. L'écriture consiste à faire passer un courant à travers la jonction, dans un sens ou dans l'autre. Il n'y a donc pas de champ magnétique extérieur. Ce courant permet d'accroitre ou diminuer l'amortissement lorsque l'aimantation de la couche de stockage oscille dans le plan. Lorsque la densité de courant est suffisamment élevée, c'est-à-dire qu'elle dépasse la densité critique, l'aimantation de la couche de stockage bascule. Ce courant critique peut être déterminé par l'équation :

$$Jc = \frac{2e\alpha Mt(Hk + H + 2\pi Ms)}{\hbar\eta}$$

Où η est l'efficacité du transfert de couple, e la charge de l'électron, h la constante réduite de Plank, α le coefficient d'amortissement, M l'aimantation de saturation de la couche de stockage, t son épaisseur, H le champ appliqué, Hk le champ d'anisotropie (en particulier l'anisotropie de forme dans le plan) et $2\pi M_s$ le terme de démagnétisation hors du plan. C'est ce terme qui est dominant et permet de dire que la densité de courant critique dépend peu de l'anisotropie de forme. On peut donc appliquer à la cellule une forme elliptique allongée, dont le rapport des axes peut aller jusqu'à 2,5, sans augmenter excessivement la densité de courant critique qui peut varier de 5.10^6 à 10^7A/cm². La forme allongée, permet d'augmenter la stabilité thermique de la cellule et donc la durée de rétention.

Comme l'écriture se fait seulement grâce à un courant à travers la jonction, il n'est pas nécessaire d'appliquer un champ magnétique extérieur et donc il n'y a pas de ligne de champ. Cela simplifie les circuits à base de MTJ en technologie STT. En effet, une MTJ est associée à un transistor comme le montre la Figure 54. Cependant, étant donné qu'il est nécessaire de faire passer un courant à travers la jonction pour la lire, il faut limiter le courant de lecture. Il doit être suffisamment bas pour ne pas écrire la cellule. Ensuite, la tension ne doit pas être trop élevée pour ne pas détruire la barrière tunnel par claquage. La conception d'un circuit à base de MTJs STT consiste donc à déterminer les différentes tensions en fonction du mode de fonctionnement lecture/écriture.

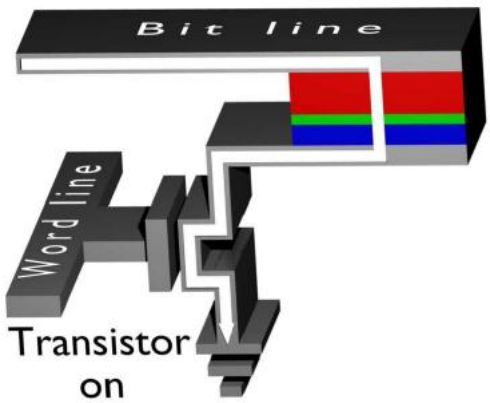


Figure 54 : CIMS JTM

XIII.5.2 MRAM Perpendiculaire

Une autre technologie d'écriture par courant polarisé en spin a été mise au point afin de réduire le courant critique, la MTJ perpendiculaire. La structure est présentée dans la Figure 55. Dans ce cas, la densité de courant est donnée par l'équation :

$$j_{WR\,out-of-plane} = \left(\frac{2e}{\hbar}\right)\frac{\alpha\, t_F}{P}\left(2K_{eff}\right)$$

Où Keff représente le coefficient d'anisotropie qui est dominé par la contribution d'interface qui permet de maintenir l'aimantation hors du plan. Le coefficient de stabilité thermique est donné par $\Delta = K_{eff}V/k_BT$. On remarque alors que le courant d'écriture est directement proportionnel à la stabilité thermique. Ceci est un avantage par rapport aux jonctions planaires dans lesquelles la valeur du courant d'écriture est essentiellement dominée par le champ démagnétisant, qui ne contribue pas à la stabilité thermique de la jonction. Cela signifie que l'on peut obtenir des courants d'écriture beaucoup plus faibles pour la même stabilité thermique. Dans la pratique, le coefficient d'amortissement est plus élevé qu'en planaire à cause de l'aimantation perpendiculaire ce qui limite l'avantage de cette technique.
Pour contrecarrer cet effet, le laboratoire Spintec a utilisé la forte anisotropie de l'interface métal/oxyde pour avoir une aimantation perpendiculaire. Cette anisotropie associée à un faible couplage spin-orbite permet d'obtenir un faible coefficient d'amortissement. Le courant d'écriture obtenu est plus faible qu'en planaire avec des exemples entre $8{,}10^5$ et $2{,}10^6$A/cm ² et une TMR de l'ordre de 100%. Les progrès techniques permettront d'améliorer encore ses caractéristiques et envisager une commercialisation.

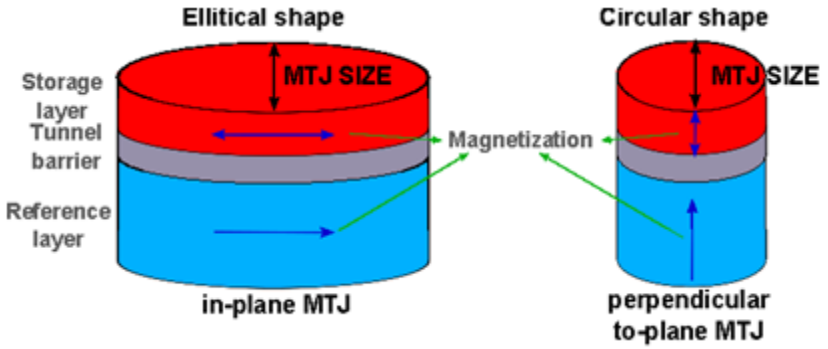


Figure 55 : comparaison entre JTM planaire et perpendiculaire

XIII.6 ETAPE DE CONCEPTION SUR MESURE D'UN CIRCUIT ELECTRONIQUE

La conception d'un circuit numérique à base de JTMs fait partie du domaine de la conception dite « full custom » ou sur mesure. En effet, il faut concevoir le circuit jusqu'au niveau transistor afin d'optimiser son fonctionnement. De plus, il n'existe pas de cellule standard numérique à base de JTM. La conception des circuits figurant dans cette thèse s'est donc faite de manière « full custom » c'est-à-dire que l'on conçoit le circuit en assemblant et en dimensionnant les transistors, puis l'on réalise le layout « à la main ». Décrivons donc les différentes étapes de ce niveau de conception.

Les étapes de la conception sont résumées dans le schéma de la Figure 56. On part d'abord du cahier des charges qui indique quelles sont les performances et caractéristiques que l'on attend du circuit. Ensuite, on conçoit le circuit en assemblant les transistors. Le circuit doit être optimisé de façon à ce qu'il réponde aux exigences du cahier des charges. Pour cela, on dimensionne les transistors. On peut faire des analyses paramétriques afin de déterminer la taille idéale d'un transistor pour qu'il réponde aux contraintes. On fait alors des simulations électriques afin de vérifier le comportement analogique du circuit (que celui-ci soit destiné à une application analogique ou numérique, comme c'est le cas ici). Elles se font sur la base de modèles qui prédisent le comportement des composants. On peut faire des simulations en prenant en compte les dispersions des caractéristiques dues aux variations de procédé (simulation de Monté Carlo). On peut également simuler le circuit à différentes températures. Ainsi, on vérifie la fiabilité du circuit par rapport aux erreurs dues au procédé de fabrication et aux conditions de fonctionnement. Si les simulations ne donnent pas les résultats attendus, il faut modifier le circuit pour corriger les défauts.

Une fois que l'on obtient les bons résultats en simulation, on peut passer à l'étape du layout. Cette étape consiste à dessiner les transistors et les pistes de métal du circuit couche par couche. Ce layout sera utilisé au final pour la fabrication des masques utilisés durant le processus de fabrication. Ensuite, on vérifie que les règles de dessin imposées par le fabriquant ont bien été respectées c'est-à-dire que les espacements et les largeurs des lignes, par exemple, ont bien été respectés. C'est l'étape de DRC pour Design Rule Checking. Ensuite, il y a l'extraction qui consiste à extraire les dispositifs du layout ainsi que les connexions pour déterminer le schéma qui a été dessiné dans le layout. Ensuite, on compare le schéma extrait à celui qui a été conçu à l'étape précédente. C'est l'étape du LVS pour Layout Versus Schematic. Cela permet de vérifier que le layout est fidèle au schéma que l'on a conçu précédemment. Ces deux étapes sont importantes pour ne pas faire d'erreur de conception et pour que le circuit ait de bonnes chances de fonctionner. Ensuite a lieu l'extraction des éléments parasites c'est-à-dire les capacités, les résistances et les inductances parasites du circuit en fonction de la géométrie du circuit qui est maintenant connue et vérifiée. Pour finir, on fait une simulation post-layout. C'est une simulation qui prend en compte les parasites. C'est donc une simulation plus fidèle de la réalité qui permet de déterminer si le circuit fonctionnera une fois fabriqué. Si les simulations post-layout sont concluantes alors la conception du circuit est terminée sinon, il faut recommencer le processus afin de faire fonctionner le circuit.

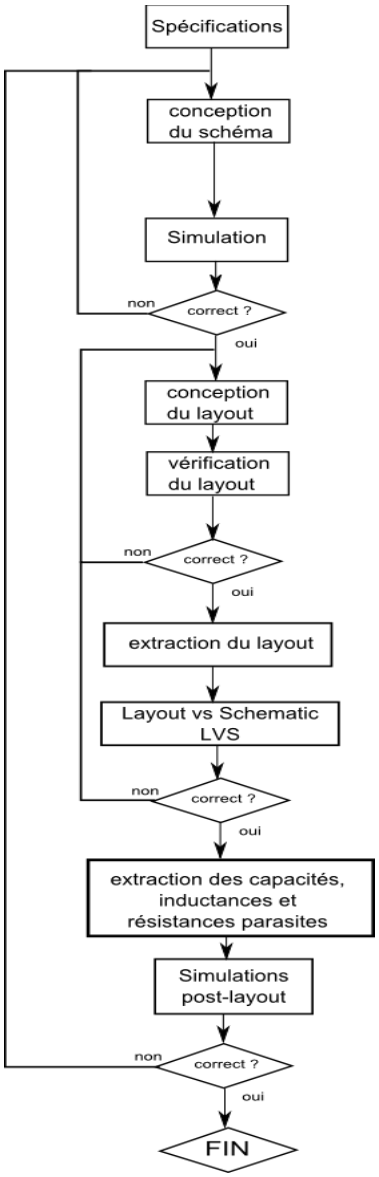


Figure 56 : étapes de la conception d'un circuit "full custom" [13]

Toutes les informations nécessaires aux simulations (modèles), au dessin du layout et à l'extraction sont regroupées dans un kit de conception. Chaque fondeur, c'est le nom donné aux fabricants de circuits intégrés, a ses propres kits pour chacune des technologies qu'il propose.

Durant la thèse, toutes ces étapes ont pu être accomplies. Un circuit complexe a été conçu en s'arrêtant à la simulation post-layout car il n'était pas possible de le faire fabriquer. Ensuite, un démonstrateur implémentant un circuit plus simple a été conçu puis fabriqué en respectant toutes les étapes. La conception d'un circuit hybride CMOS/magnétique n'est possible qu'avec les modèles de simulation du fonctionnement d'une JTM ainsi que le layout et ses données d'extraction. Il faut donc un design kit magnétique. Les modèles ont été développés par le laboratoire Spintec et ont été utilisés lors de cette thèse. Ils ont été intégré dans un kit de conception magnétique qui a été greffé à un kit de conception CMOS pour donner un kit hybride CMOS/magnétique. Le chapitre suivant décrit le modèle ainsi que la technologie hybride CMOS/magnétique.

XIII.7 KIT DE CONCEPTION MAGNETIQUE

Un kit de conception regroupe les informations d'une technologie permettant de simuler le comportement des composants (transistor, résistance, diode, …) disponibles dans cette technologie. Il comprend également les données et paramètres relatifs à la technologie ainsi que les règles de dessin. Un kit magnétique a été conçu au CMP, notamment par Gregory Di Pendina, et reprend toutes ces informations. C'est ce kit qui a été utilisé dans la conception des circuits présentés dans la partie « Simulations ». Décrivons donc le modèle utilisé lors de la simulation des JTMs et le procédé hybride CMOS/magnétique.

XIII.7.1 Description du modèle

Les simulateurs électriques imposent un formalisme particulier aux modèles qui doivent être compatibles avec les dispositifs électroniques standards. Il est donc nécessaire de développer un modèle électrique équivalent des composants électroniques nouveau utilisés en simulation, comme la JTM, à partir de composants standards comme par exemple des résistances ou des condensateurs. Le modèle des JTMs a été développé ces dernières années au laboratoire Spintec successivement par Virgile Javerliac, Mourad El Baraji, Guillaume Prenat, Wei Guo et Abdelilah Mejdoubi. Au final, plusieurs technologies de JTM ont été modélisées : FIMS, TAS et STT. Décrivons le principe général de la modélisation. Un modèle permet de prévoir le comportement électrique d'un composant en fonction des entrées que l'on lui envoie. Dans le cas, de la JTM, on peut prévoir son comportement grâce aux équations du magnétisme et notamment l'équation LLG décrite précédemment. Cependant la résolution de ces équations est très complexe voire impossible car elle mènerait à des temps de simulation extrêmement longs. Le schéma électrique est conçu de façon à ce que les données de sortie soient les même qu'avec des calculs avec les formules du magnétisme. Le schéma équivalent n'a donc aucune signification physique, c'est seulement un moyen d'optimiser le fonctionnement du simulateur.

Le modèle de JTM a été développé pour le moteur de simulation SPECTRE avec l'interface CMI (Compiled-Model Interface) en langage C fourni par Cadence Design Systems. La CMI est constituée d'un ensemble de fonctions et de structures de données utilisées pour décrire le modèle. Une fois compilé, le modèle est une « boite noire » dont on ne considère que les courants et tensions à chacun de ses nœuds. On sait, grâce à la loi de Kirchhoff, que la somme de ces courants est nulle. Connaissant le schéma équivalent électrique, on peut établir une relation entre les courants et tensions. Grâce à ces relations, on en déduit une matrice qui va être utilisée par le simulateur pour ses calculs. Le système est alors mis sous forme matricielle, plus précisément une Jacobienne. La CMI a été choisie en raison de sa flexibilité et sa programmation bas niveau qui permet d'écrire des modèles sophistiqués tout en ayant de bonnes performances en termes de vitesse de simulation. Elle est utilisée pour les modèles complexes.

La Figure 57 montre le schéma d'un circuit simple à base de JTM et d'un transistor de sélection. La JTM est en technologie TAS. On retrouve donc les connexions reliant la JTM au circuit CMOS. Ce sont les terminaux bl0 et bl1. On représente également dans le symbole les terminaux correspondant à la ligne de champ, fl0 et fl1. Bien que ce ne soit qu'un simple fil, il est nécessaire de le représenter et de l'orienter car de son orientation dépend le sens de l'aimantation de la couche de stockage. Le terminal

th indique la température de la jonction. Il est uniquement utilisé en simulation lors de la conception du circuit pour débugger. Le terminal my est également utilisé uniquement lors de la simulation et indique la composante suivant l'axe facile. Les terminaux th et my n'existent donc pas physiquement mais sont utilisés pour vérifier que la température de blocage a bien été dépassée et que la cellule a bien changé d'état.

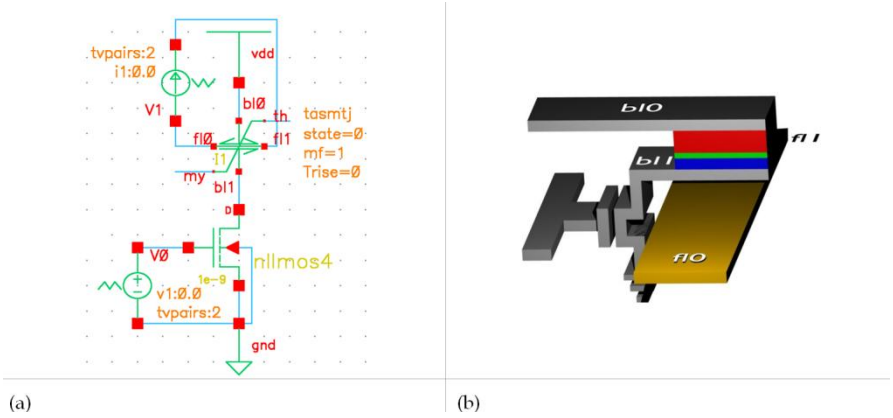


Figure 57 : schéma d'un circuit à base de JTM en technologie TAS

Ce schéma est valable aussi bien pour la TAS que pour la technologie STT. Dans ce cas, la ligne de champ n'est pas utilisée. Le modèle décrit le comportement statique et dynamique de la JTM en étant le plus fidèle à la réalité. Il prend en compte notamment l'effet de la température qui est important pour la technologie TAS, ainsi que les fluctuations thermiques pour la STT et le phénomène de transport en fonction des aimantations des différentes couches. La Figure 58 montre les courbes de simulation d'un circuit à base de JTM en technologie TAS dans lequel on écrit une donnée. La courbe (a) représente le courant à travers la jonction TAS c'est-à-dire le courant de chauffage. La température est représentée par la courbe (c). On voit bien que la température augmente à l'instant où l'on active le courant de chauffage et diminue lorsqu'il est arrêté. La courbe (b) représente le courant de génération du champ magnétique d'écriture. La courbe (d) représente la tension aux bornes de la JTM et la courbe (e) sa résistance. On passe de l'état parallèle, c'est-à-dire faible résistance, à l'état antiparallèle, où la résistance est plus forte. L'instant où le changement d'aimantation intervient se produit lorsque le champ magnétique extérieur est appliqué (courbe b) et que la température a dépassé la température de blocage (courbe c). On observe alors un changement de l'aimantation grâce à des oscillations durant un régime transitoire qui correspond à de faibles oscillations de l'aimantation.

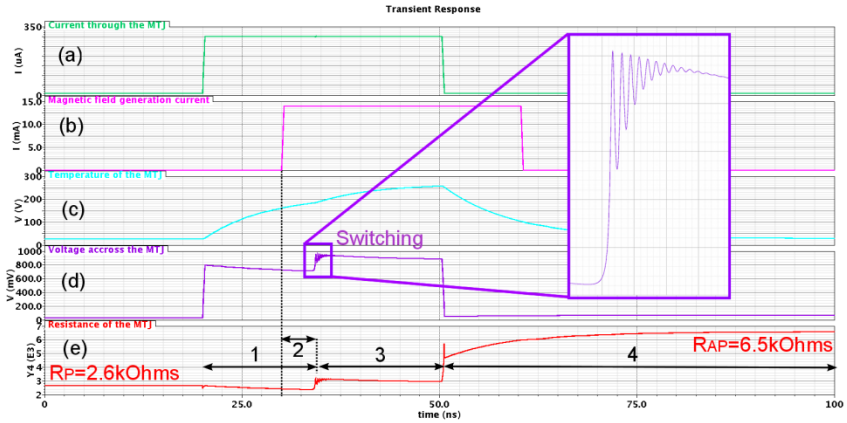


Figure 58 : simulation d'une JTM TAS

Les faibles pentes de la tension aux bornes de la JTM avant et après le changement d'aimantation sont dues à la température qui modifie la conduction à travers la barrière tunnel. On retrouve bien que la résistance de la JTM est plus faible dans l'état parallèle qu'antiparallèle.

Description du process CMOS/magnétique

Concernant le layout, on dispose dans la librairie d'une cellule que l'on peut instancier, appelée pcell, représentant le layout de la JTM. Une pCell est un layout élémentaire d'un dispositif, dont on peut choisir les paramètres de taille et dont la géométrie s'adapte automatiquement pour respecter les règles de dessin. La Figure 59 montre le layout de la pcell d'une JTM associée à la pcell de son transistor de sélection.

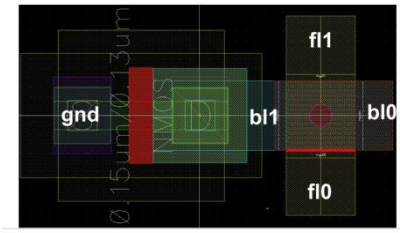


Figure 59 : layout de la pcell d'une JTM et du transistor associé en technologie TAS

Cette figure est le layout le plus simple d'une MRAM. On peut voir la JTM, sur le côté droit où la cellule est le point rouge. Les pistes de métal relient la cellule au transistor. La grille du transistor (polysilicium) est en rouge.

Comme le montre la Figure 60, un circuit hybride CMOS/magnétique est constitué d'abord des couches CMOS, c'est-à-dire les transistors et les niveaux de métaux et de via. Au-dessus du dernier niveau de métal on trouve les couches magnétiques.

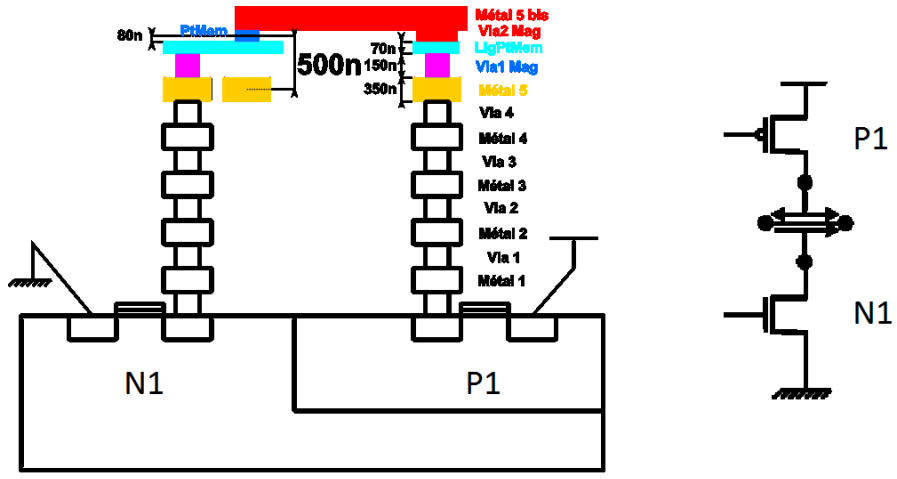


Figure 60 : section transversale d'un circuit hybride CMOS/magnétique

Les couches magnétiques sont donc constituées des couches suivantes :

- Métal 5 (en jaune) : c'est le métal supérieur de la technologie CMOS. Il permet de connecter les terminaux de la JTM au circuit CMOS. Dans le cas de la TAS, c'est avec ce métal que l'on réalise la ligne de champ
- Via1Mag (en mauve) : ce via connecte les terminaux de la JTM au métal 5

- **LigPtMem (en bleu clair) : c'est le niveau de métal qui se connecte au terminal bas de la JTM. Il fait également le lien entre le via1Mag et le via2Mag**
- **Via2Mag : ce via fait le lien entre le métal LigPtMeM et le Métal5 bis**
- **Métal5bis (en rouge) : c'est le niveau de métal qui est connecté au terminal haut de la JTM**
- **PtMem : c'est la JTM. Elle comprend toutes les couches qui constituent la JTM**

Notons qu'il n'est pas possible d'appliquer une rotation au layout d'une JTM, comme cela est traditionnellement réalisé par les outils de placement/routage dans le but d'optimiser le layout en surface. En effet, durant la fabrication, le wafer est soumis à un recuit sous champ magnétique de façon à imposer une aimantation à la couche de référence. Les couches de référence des JTMs du wafer ont donc toutes la même direction. Par conséquent, si une JTM est tournée, le champ magnétique appliqué pour écrire la JTM ne sera pas bien orienté. Les JTMs d'un circuit doivent donc toutes être orientées dans la même direction et il n'est possible que de les translater. Dans notre cas, la ligne de champ est horizontale pour être parallèle à l'aimantation de la couche de référence.

XIII.8 REFERENCES

[1]David Lammers. EE Times. http://eetimes.com/electronics-news/4062196/MRAM-debut-cues-memory-transition (Page consultée en 2010)

[2]http://www.itrs.net/Links/2010ITRS/2010Update/ToPost/ERD_ERM_2010FINALReportMemoryAssessment_ITRS.pdf. (Page consultée en 2010)

[3]Dieny, B., V.S. Speriosu, S.S.P. Parkin, B.A. Gurney, D.R. Wilhoit, and D. Mauri : Giant magnetoresistive in soft ferromagnetic multilayers. *Phys. Rev.* B, 43(1):1297–1300, Jan 1991.

[4] Baibich, M.N., J.M. Broto, A. Fert, F.N. Van Dau, F. Petroff, P. Etienne, G. Creuzet, A. Friederich, and J. Chazelas. Giant magnetoresistance of (001)fe/(001)cr magnetic superlattices. *Phys. Rev.* Lett., 61(21):2472–2475, Nov 1988.

[5] Binasch, G., P. Grünberg, F. Saurenbach, andW. Zinn: Enhanced magnetoresistance in layered magnetic structures with antiferromagnetic interlayer exchange. *Phys. Rev.* B, 39(7):4828–4830, Mar 1989.

[6] Wang, D., C. Nordman, J.M. Daughton, Z. Qian, J. Fink, D. Wang, C. Nordman, J.M. Daughton, Z. Qian, and J. Fink: 70% TMR at Room Temperature for SDT Sandwich JunctionsWith CoFeB as Free and Reference Layers. *IEEE Transactions onMagnetics*, 40:2269–2271, July 2004

[7] Lee, Y.M., J. Hayakawa, S. Ikeda, F. Matsukura, and H. Ohno: Effect of electrode composition on the tunnel magnetoresistance of pseudo-spin-valvemagnetic tunnel junction with a mgo tunnel barrier. *Applied Physics Letters*, 90(21):212507, 2007.

[8] Savtchenko, L., B.N. Engel, N.D. Rizzo, M. DeHerrera, and J. Janesky: Method of writing to scalable magnetoresistance random access memory element. U.S. Patent 6,545,906 B1, April 2003.

[9] Prejbeanu, I.L., M. Kerekes, R.C. Sousa, H. Sibuet, O. Redon, B. Dieny, and J.P. Nozieres: Thermally assisted mram. *Journal of Physics: Condensed Matter*, 19(16):165218, 2007.

[10] Sousa, R. and I. Prejbeanu: Non-volatile magnetic random access memories (mram). *Comptes Rendus Physique*, 6:1013–1021, nov 2005.

[11] J. Slonczewski: Current-driven excitation of magnetic multilayers. *Journal of Magnetism and Magnetic Materials*, 159:L1–L7, jun 1996.

[12] L. Berger: Emission of spin waves by a magnetic multilayer traversed by a current. *Phys. Rev. B*, 54(13):9353–9358, Oct 1996.

[13] JAVERLIAC Virgile. Développement d'un modèle compact de la jonction tunnel magnétique de première génération et son intégration dans la réalisation d'architectures logiques reprogrammables hybrides magnétique-cmos. Thèse de doctorat en micro nanoélectronique. L'Institut Polytechnique de Grenoble, 2006 , 205p.

[14] GUO Wei. Compact Modeling of Magnetic Tunnel Junctions and Design of Hybrid CMOS/Magnetic Integrated Circuits. Thèse de doctorat en micro nanoélectronique. L'Institut Polytechnique de Grenoble, 2010, 203p.

[15] J. Slonczewski: Current-driven excitation of magnetic multilayers. *Journal of Magnetism and Magnetic Materials,* 159:L1–L7, jun 1996.

[16] DI PENDINA Gregory. Conception innovante et Développement d'outils de conception d'ASIC pour Technologie Hybride CMOS / Magnétique. Thèse de doctorat en nanoélectronique et nanotechnologies. Université de Grenoble, 2012, 191p.

[17] N. Mott. volume A 153, 1936.

[18] A. Fert and I. A. Campbell. Phys. Rev. Lett. 21, 1968.

[19] B. Loegel and F. Gautier. J. Phys. Chem. Sol. 32, (2723), 1971.

XIV. Etat de l'art des FPGAs à base de mémoires MRAM

La première et la plus évidente des applications des JTMs est la mémoire MRAM standalone. Une mémoire MRAM est déjà commercialisée par Everspin. On peut toutefois utiliser ces mémoires dans d'autres types d'applications et en particulier dans les circuits logiques. C'est le sujet qui est étudié dans l'équipe de Spintec, dirigée par Guillaume Prenat, où la thèse s'est déroulée. Les principaux atouts de la JTM que l'on peut exploiter dans les circuits logiques sont sa non-volatilité pour faire des circuits consommant peu d'énergie associée à une grande vitesse d'écriture, de l'ordre de quelques nanosecondes. On peut notamment utiliser les JTMs dans des bascules pour les rendre non-volatiles. Ainsi, on peut sauvegarder localement et rapidement les registres d'un processeur avant de couper l'alimentation [8]. Son état sera restauré instantanément lorsqu'il sera remis sous tension. Cela permet de concevoir des circuits logiques avec une consommation plus faible qu'en utilisant des cellules Flash, gourmandes et lentes à l'écriture, ou des RAMs à très faible courant de fuite et de plus, distantes sur la puce. On peut également concevoir des circuits combinatoires embarquant des JTMs et donc des capacités de mémorisation. C'est le concept de « logic in memory » ([4]) où des cellules mémoire non-volatiles sont distribuées dans le circuit logique de façon à concevoir des circuits basse consommation ou à offrir de nouvelles fonctionnalités. [1] présente par exemple une Unité Arithmétique et Logique (ALU pour Arithmetic Logic Unit) avec une faible consommation et une faible surface. Ces avantages sont possibles car le procédé de fabrication de la MRAM est compatible avec celui des circuits CMOS standards et permet donc de concevoir des circuits avec de la mémoire non-volatile embarquée.

Le travail présenté dans cette thèse a pour but de montrer l'intérêt de l'utilisation des JTMs dans la logique reconfigurable. Les paragraphes suivants vont donc présenter les principaux circuits logiques programmables à base de JTMs développés dans la littérature.

XIV.1 Hassoun et Black

Les premiers à s'être intéressés à la logique reconfigurable à base de composants magnétiques sont Hassoun et Black dans [5]. La Figure 61 montre le schéma de principe.

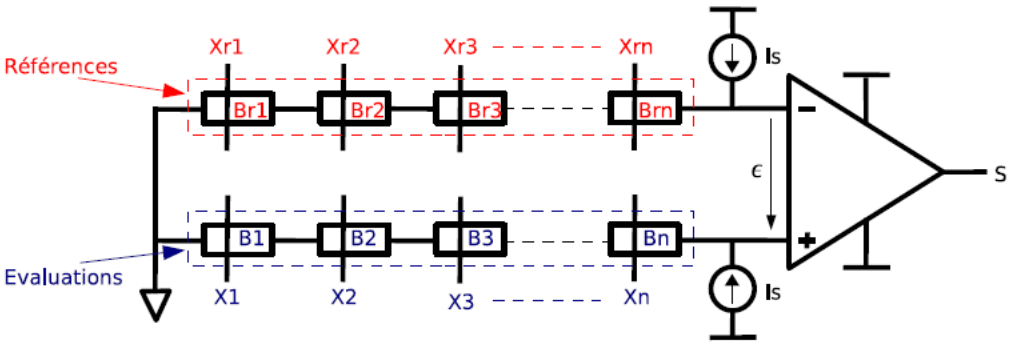


Figure 61 : circuit logique reconfigurable à N entrées [5]

Le circuit est constitué de vannes de spin avec deux couches libres. Notons qu'un circuit similaire a ensuite été conçu avec des JTMs [6]. Il y a deux parties dans ce circuit : une partie référence et une partie évaluation. Les entrées sont représentées par Xrn et Xn où l'information est codée suivant le sens du courant. La programmation de la fonction à accomplir consiste à imposer une orientation de l'aimantation de la couche dure des vannes de spin. Les générateurs de courant Is imposent un courant dans celles-ci. Leur résistance dépendant de l'orientation des couches dures et du sens du courant dans les entrées, les résistances totales des lignes de référence et d'évaluation seront différentes. On a la formule suivante :

$$\varepsilon = (k - l).\Delta R.Is + Voffset$$

Où k est le nombre de résistances dans l'état antiparallèle de la partie évaluation, l le nombre de résistances dans l'état antiparallèle de la partie référence, Voffset est la tension d'offset du comparateur et $\Delta R = Rap - Rp$.

Si ε est négatif, la sortie est à 0 et à 1 s'il est à positif. La fonction calculée est déterminée en jouant sur l'orientation des aimantations des couches dures, la résistance de la partie référence et la tension d'offset du comparateur. Cependant, toutes les fonctions ne peuvent pas être programmées. Plus il y a d'entrées, moins il y a de fonctions programmables. Ainsi, pour 4 entrées, on peut programmer 95% des fonctions possibles, le maximum étant 216, c'est-à-dire 65536 fonctions.

XIV.2 Black et Das

Le circuit proposé par W.C. Black et B. Das [8] est montré dans la Figure 62. Le principe est d'associer une cellule SRAM et deux JTMs dont les états sont complémentaires pour constituer une cellule SRAM non-volatile. Les JTMs stockent la donnée à sauvegarder de façon permanente et la cellule SRAM lit la donnée pour la restaurer. On transfert donc la donnée de la JTM sur la SRAM. Notons que durant le fonctionnement du circuit, les données de la SRAM et des JTMs peuvent être différentes. En fonctionnement normal, la cellule peut agir comme une simple SRAM car les JTMs n'influent pas sur les phases d'écriture et de lecture de la SRAM. Ensuite, le bit stocké dans la SRAM peut être sauvegardé de façon non-volatile dans les JTMs ou inversement, le bit stocké dans les JTMs peut être transféré à la SRAM. Il est possible d'utiliser cette fonctionnalité dans une Look-Up Table avec deux contextes où un contexte est stocké dans la SRAM et l'autre dans les JTMs comme dans [9].

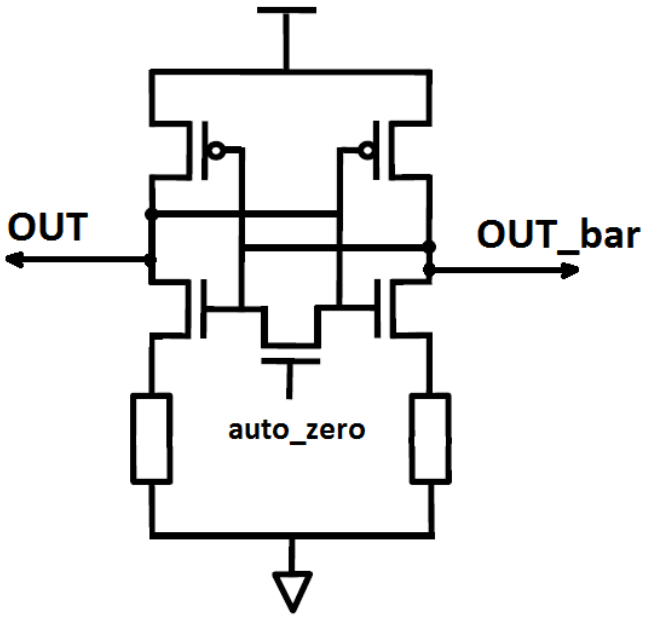


Figure 62 : circuit proposé par W.C. Black et B. Das

Ainsi, pour lire le bit stocké dans les JTMs, il faut d'abord rendre passant le transistor auto_zero. Cela a pour effet d'amener le circuit dans un état métastable, les sorties OUT et OUT_bar ont donc une tension intermédiaire entre 0 et Vdd, différentes car la résistance des deux branches est différente à cause de la présence des JTMs dans un état magnétique opposé. Le latch sert alors de sense amplifier. Ensuite, l'auto_zéro est désactivé. Les résistances des JTMs étant différentes, le circuit va alors basculer dans un état stable qui dépend de l'état des JTMs. C'est un circuit simple qui permet de rendre une SRAM non-volatile. Des transistors supplémentaires doivent être ajoutés pour écrire les JTMs et dépend donc de la technologie de JTM, cependant ces transistors n'influencent pas la lecture des données. Ce circuit peut constituer une mémoire de configuration dans les LUTs ou les interconnexions. L'inconvénient de cette méthode est principalement le fait que l'on perd en densité car la SRAM est composée de 6 transistors et des transistors supplémentaires sont nécessaires pour écrire et sélectionner les JTMs.

XIV.3 Mémoire durcie aux radiations

Une variante du circuit décrit précédemment est la cellule mémoire décrite dans [10] qui consiste à placer les JTMs, non pas connectées à la masse mais sur le chemin de rétroaction du latch (voir la Figure 63).

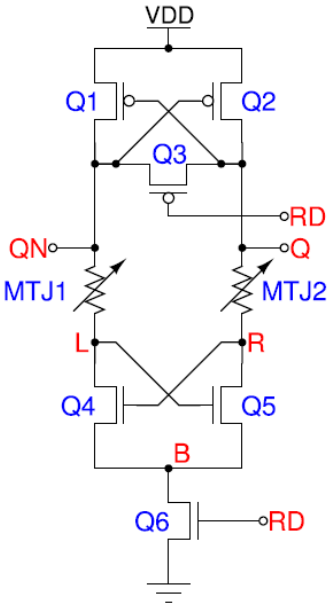


Figure 63 : mémoire de configuration durcie [10]

Le fonctionnement de cette cellule est basé sur le même principe que la cellule de Black et Das. On ferme le transistor Q3 pour équilibrer le circuit. Après l'avoir ouvert, il tend vers un état stable déterminé par l'état des JTMs.

Ce circuit est une adaptation d'une technique de durcissement aux radiations [11] d'une SRAM consistant à placer des résistances suffisamment élevées pour former un filtre passe-bas et ainsi éliminer les pics de courant dus au passage d'une particule énergétique. Cela permet de rendre une SRAM résistante aux radiations avec un faible coût en surface. Dans ce circuit, les résistances ont été remplacées par des JTMs car ce sont des résistances. Cela permet donc d'accroitre la résistance de la cellule aux erreurs transitoires. A ce circuit, il faut ajouter les transistors de sélection et d'écriture. Le circuit présenté dans [9] était à base de FIMS. Le circuit d'écriture est donc indépendant du circuit de lecture ce qui simplifie le schéma du circuit.

L'inconvénient de ce circuit est la perte en densité. De plus, pour les particules les plus énergétiques, le circuit pourrait malgré tout générer une erreur.

XIV.4 LUT avec logique en mode courant

Décrivons maintenant une table de correspondance à base de JTMs [12]. Autrement appelée LUT pour Look-Up Table. La Figure 64 montre le schéma de cette LUT. Les cellules mémoires de configuration sont simplement constituées de deux JTMs dont les états sont complémentaires. Le multiplexeur va sélectionner le couple de JTMs qui va être lu par le sense amplifier (SA).

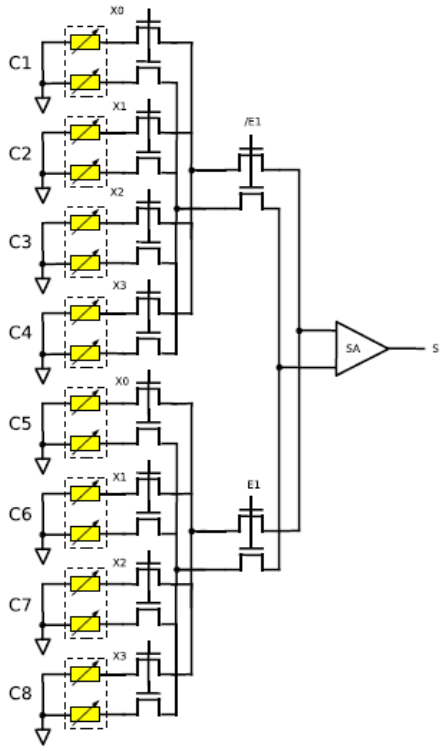


Figure 64 : LUT-3 à base de JTMs en complémentaire [12]

Le sense amplifier est décrit dans la Figure 65. C'est un amplificateur de lecture en mode courant. La lecture consiste à imposer un courant et comparer les tensions aux bornes des deux jonctions. La tension de polarisation doit être choisie de façon à ce que la variation de tension entre les états parallèles et antiparallèles soit maximale. En effet, il faut savoir que plus la tension de polarisation est élevée plus la TMR diminue. La TMR est donc maximale lorsque la tension est nulle. Mais une tension nulle ne permet pas de déterminer la donnée stockée dans les JTMs. Il faut donc trouver le bon compromis entre la variation de courant entre les deux états parallèle et antiparallèle et la valeur absolue du signal pour que la mesure des résistances soit optimale. Dans ce cas, la polarisation était de 300 mV. Cette tension de polarisation est imposée par les transistors N0 et N1 ainsi que les courants I0 et I1. Des miroirs de courant constitués par les transistors P0, P2 et P1, P3 forment des charges actives. Un buffer constitué d'une chaine d'inverseurs permet de mettre en forme le signal afin que le niveau de tension en sortie soit compatible avec les niveaux de tension logique et que l'impédance de sortie soit faible afin d'avoir une sortance élevée.

Les avantages de cette architecture sont la forte densité et la rapidité. En effet, les cellules mémoires de configuration sont seulement constituées des JTMs. Hors les JTMs peuvent être placées au-dessus des transistors. Elles n'induisent donc pas de surface supplémentaire. Seuls les transistors d'écriture prennent de la place mais leur taille dépend de la technologie de JTM et des dimensions des cellules.

Les inconvénients sont d'abord la consommation car le fonctionnement de la LUT nécessite d'appliquer un courant en permanence à travers les JTMs sélectionnées ce qui augmente fortement la consommation statique du circuit. Le fait d'appliquer un courant en permanence à travers la JTM pourrait poser des problèmes de fiabilité car une tension est appliquée sur l'oxyde ce qui pourrait mener à un claquage de la jonction. Pour y remédier, le circuit de lecture est optimisé de façon à limiter la tension de polarisation des jonctions. Un autre inconvénient est le fait que cette technique ne peut

pas être appliquée pour former le circuit de configuration des interconnexions d'un FPGA car elle est constamment active contrairement aux JTMs des LUTs.

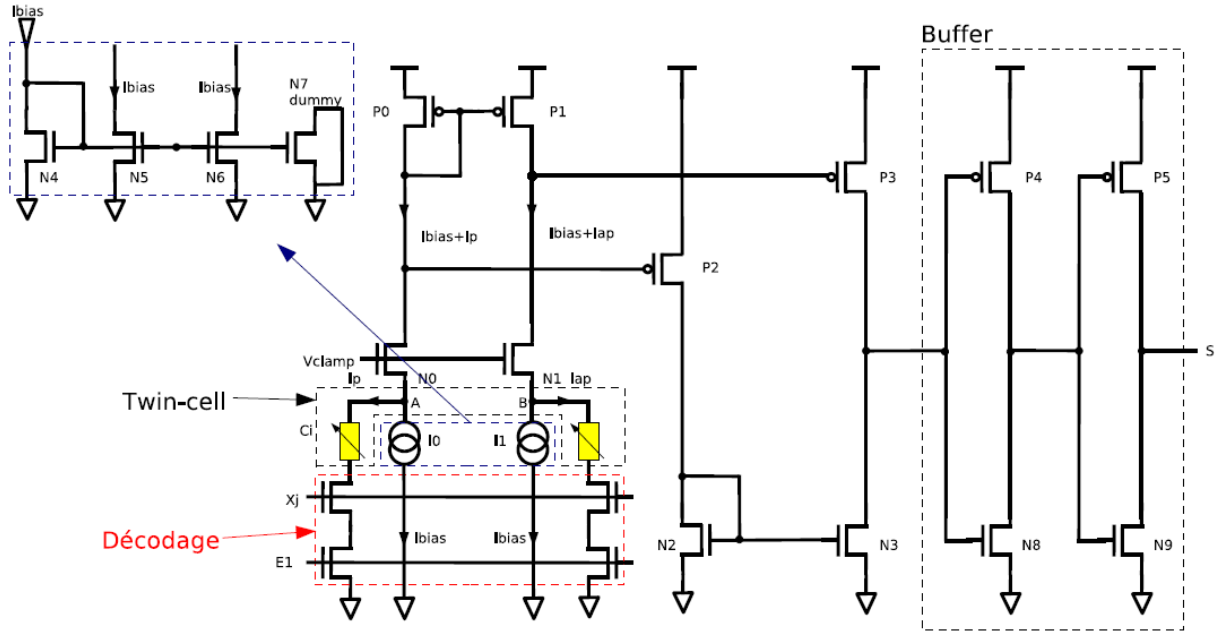


Figure 65 : amplificateur de lecture en mode courant [12]

XIV.5 Logique dynamique en mode courant

Décrivons maintenant la Look-Up Table développée à l'université de Tohoku [7]. Le principe repose sur l'utilisation de la logique dynamique en mode courant. La Figure 66 montre le schéma de principe d'une LUT à deux entrées.

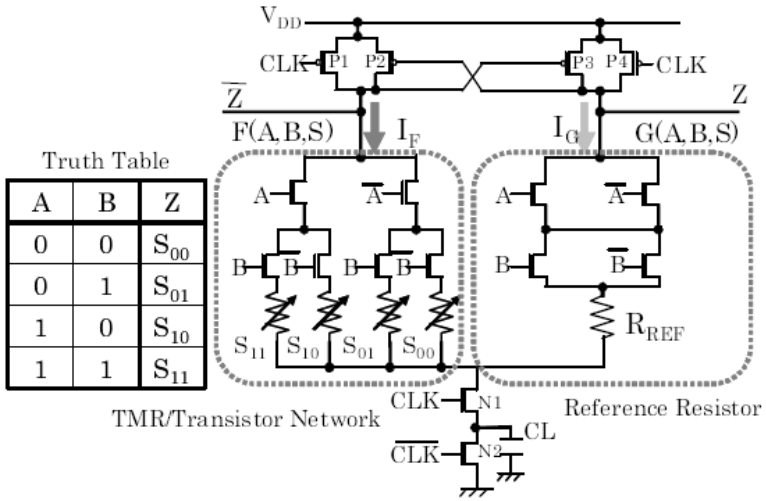


Figure 66 : LUT à 2 entrées à base de JTMs [7]

Notons qu'il y a deux parties dans ce circuit. Le réseau de transistors et de JTMs puis le circuit de référence qui a une résistance Rref qui doit être comprise entre les résistances dans l'état parallèle et antiparallèle. Décrivons le fonctionnement de ce circuit. Il y a d'abord la phase de pré-charge, quand CLK est à 0. P1 et P4 sont ON pour

charger les nœuds Z et Z_bar à Vdd. En même temps, N2 est ouvert pour que CL se décharge. N1 est OFF pour qu'il n'y ait pas de chemin de conduction directe entre Vdd et gnd. Ensuite vient la phase d'évaluation, lorsque CLK est à 1. P1, P4 et N2 sont alors OFF. N1 est ON pour créer un courant entre les nœuds de sortie Z, Z_bar et CL. Les courants If et Ig sont différents et fonction de la JTM sélectionnée par les entrées A et B. L'un des deux nœuds Z ou Z_bar va donc charger plus rapidement CL que l'autre. Ceci est accéléré par les transistors P2 et P3 qui vont maintenir le résultat de l'évaluation. Ils se comportent donc comme un comparateur. La donnée en sortie dépend de l'état des JTMs. On peut donc programmer n'importe quelle fonction logique à deux entrées simplement en programmant les JTMs. Les JTMs utilisées pour ce circuit sont des CIMSs. L'écriture se fait en faisant passer un courant dans un sens ou dans l'autre dans les JTMs. Il y a donc un générateur de courant (non représenté). Ensuite les JTMs à écrire sont sélectionnées par un transistor de sélection connecté à un côté de la JTM puis un transistor autorisant l'écriture est connecté à l'autre comme le montre la Figure 67.

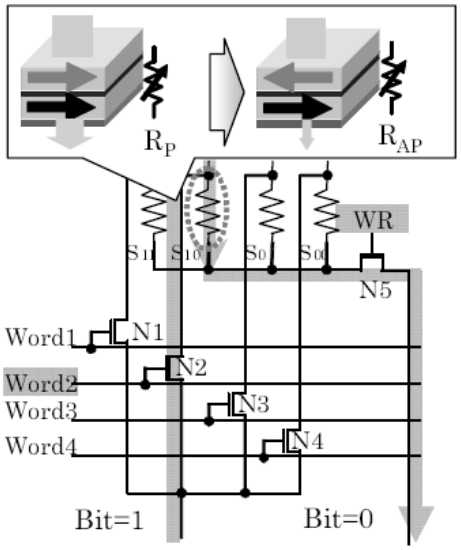


Figure 67 : circuit d'écriture des JTMs [7]

Les avantages de cette architecture sont la rapidité car les circuits dynamiques sont rapides. Ensuite, le circuit est dense car les JTMs ne demandent que très peu de surface. Finalement, il est non-volatile grâce à l'utilisation des JTMs. La consommation statique est donc négligeable. Cependant, les inconvénients sont d'abord le fait qu'il y a une fréquence d'horloge minimale pour que le circuit dynamique fonctionne. De plus, la conception d'un circuit dynamique, de surcroit à base de JTM, est très complexe. En effet, il faut insérer des buffers entre chaque étage des LUTs et la conception de l'arbre d'horloge est très délicate. Enfin, cette technique ne permet de concevoir que des LUTs, les interconnexions programmables ne peuvent pas être configurées de cette façon. Hors, le circuit de configuration des interconnexions constituent plus de la moitié du circuit de configuration d'un FPGA.

XIV.6 FPGA MRAM du LIRMM

Décrivons maintenant le projet le plus ambitieux sur la fabrication d'un FPGA complet à base de mémoire MRAM [9] (Figure 68). Il a été conçu dans le cadre d'un projet ANR, le projet SPIN, qui a pour but la fabrication d'un capteur à base de vannes de spin et d'un FPGA MRAM complet. Le FPGA MRAM a été conçu conjointement par le LIRMM et la startup MENTA qui ont converti leur FPGA embarqué SRAM en un FPGA MRAM. La technologie de mémoire MRAM utilisée est la TAS.

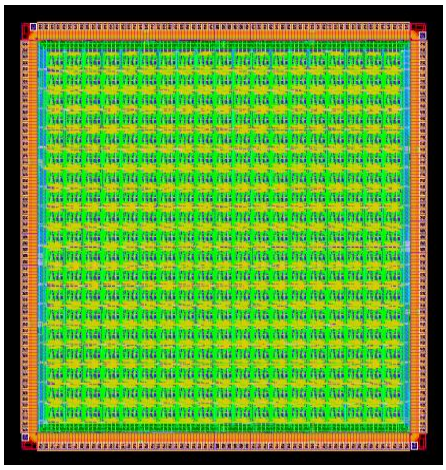


Figure 68 : FPGA MRAM [9]

Les cellules mémoires de configuration ont l'architecture proposée par Black et Das décrite précédemment adapté à la mémoire TAS (Figure 69). On retrouve la structure de la cellule SRAM avec le transistor permettant de faire un auto-zero. Des transistors d'accès permettent d'isoler la cellule SRAM lorsque l'on ne lit pas les JTMs. Les JTMs sont connectées à des transistors de chauffage qui sont utilisés lors de la phase de d'écriture. Un générateur de courant (non représenté) permet de générer le champ magnétique qui écrit les JTMs.

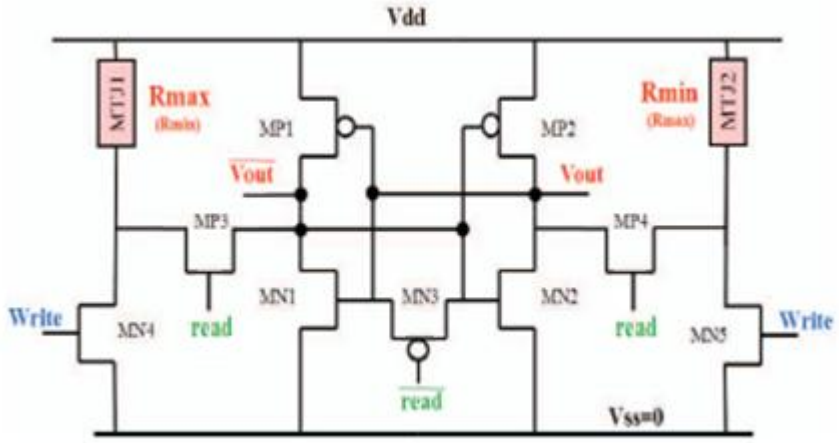


Figure 69 : mémoire de configuration utilisée dans le FPGA MRAM du projet SPIN [9]

Cette cellule a ensuite été intégrée dans le FPGA complet à partir de l'architecture d'un FPGA embarqué existant, eFPGA de MENTA, en l'adaptant à la MRAM, en particulier le circuit d'écriture pour la programmation du FPGA. Par exemple, la ligne de champ et le générateur de courant ont été partagés par plusieurs JTMs. Une tuile complète est représentée dans la Figure 71. Une tuile est un circuit élémentaire répétable, comme un motif, afin de faire un FPGA entier en assemblant les tuiles les

unes à côté des autres comme une matrice. Le FPGA complet est finalement compatible avec les outils de développement de MENTA pour lesquels la présence des MRAMs est transparente. Ainsi, il possible de programmer et tester le FPGA comme un FPGA SRAM classique de MENTA. Le démonstrateur est en cours de test.

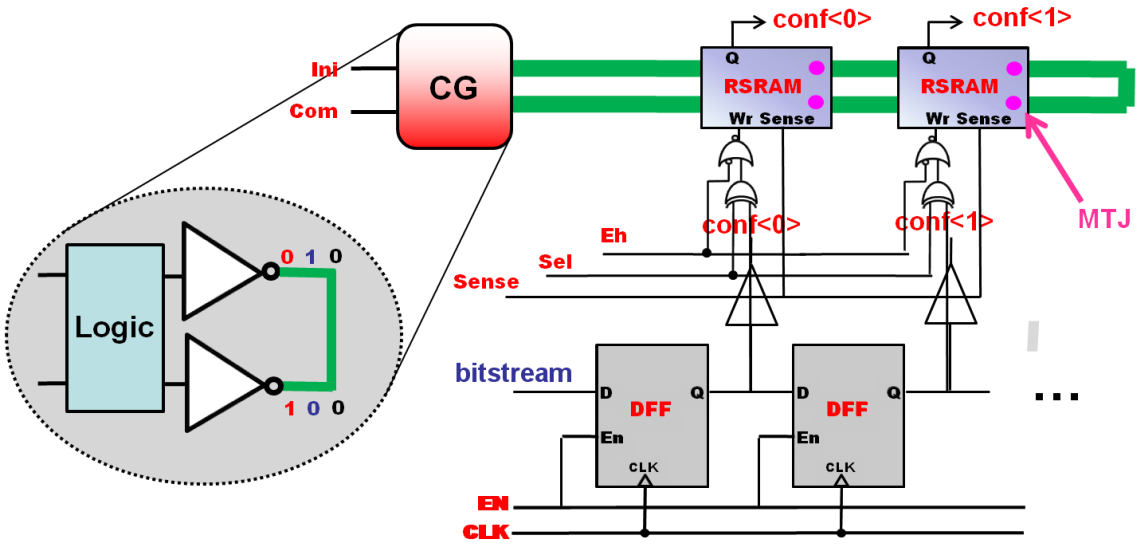


Figure 70 : circuit de configuration du FPGA MRAM

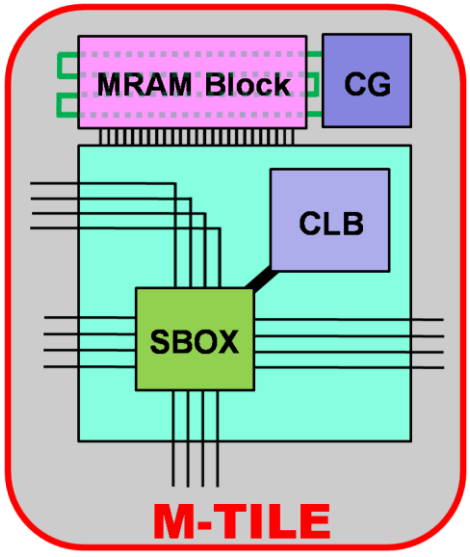


Figure 71 : schéma d'une Tuile à base de MRAM [9]

XIV.7 FPGA à base de nouvelles technologies de mémoires

Les MRAMs étant des mémoires résistives comme la plupart des nouvelles technologies de mémoires (sauf les mémoires ferroélectriques qui sont capacitives), on peut adapter les circuits décrit précédemment à d'autres technologies de mémoire comme les PCRAMs ou les CBRAMs. Décrivons maintenant deux autres circuits de FPGA à base de deux autres nouvelles technologies.

XIV.7.1 Ferroelectric DPGA

Un FPGA a été conçu à base de mémoire Ferroélectrique par Fujitsu [11]. Le FPGA conçu est dynamiquement reconfigurable et possède 8 contextes. Il est destiné à des applications dans le domaine de la sécurité. Dans ce cas, l'application implémentée est un circuit de cryptage et décryptage de données. En effet, grâce au fait que la mémoire de configuration est interne et non externe comme pour les FPGAs SRAM, la configuration du circuit ne peut pas faire l'objet de « reverse engineering ». Les mémoires de configuration ont été faites avec des cellules SRAM auxquelles ont été ajoutées quatre cellules Ferroélectriques.

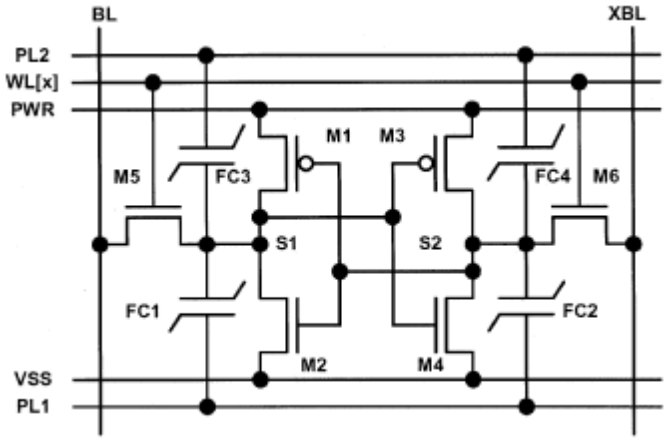


Figure 72 : cellule de configuration ferroélectrique [11]

Son fonctionnement est similaire à celui de la cellule de Black et Das. Le démonstrateur a été conçu avec la technologie CMOS 0,35 µm.

XIV.7.2 PCRAM pour FPGA

Pour finir, notons qu'une cellule mémoire de configuration à base de PCRAMs a été proposée dans [14]. Elle consiste à associer deux PCRAMs en série afin de constituer un diviseur de tension (Figure 73). Les deux PCRAMs sont dans des états complémentaires.

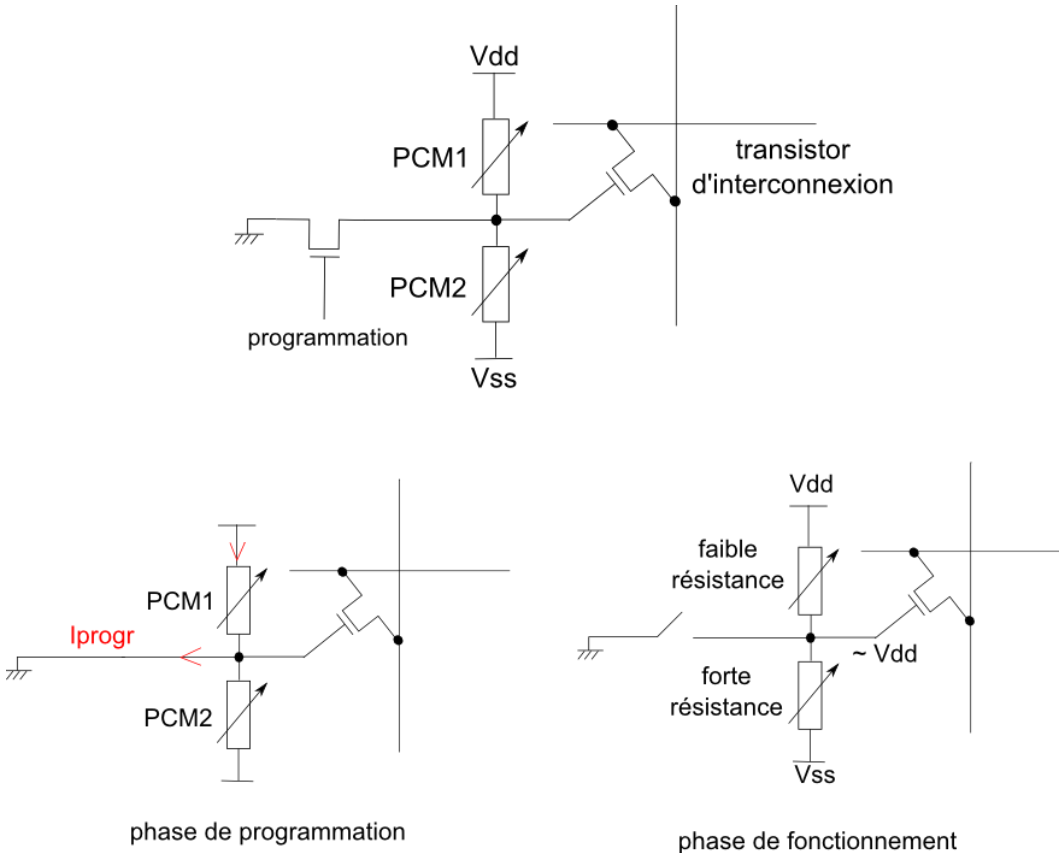


Figure 73 : cellule mémoire de configuration à base de PCRAM [14]

Ainsi lorsque la PCRAM 1 est dans un état bas, c'est-à-dire que la résistance est très faible de l'ordre des kΩ, et que la PCRAM 2 est dans un état haut, c'est-à-dire que la résistance est forte de l'ordre du MΩ, alors la tension du nœud de sortie est proche de Vdd. Ainsi, lorsqu'un transistor NMOS d'interconnexion est connecté à cette mémoire de configuration, il devient passant car la tension à sa grille est supérieure à sa tension de seuil. Lorsque les PCRAMs changent d'état, la tension en sortie est proche de Vss et donc le transistor d'interconnexion est bloqué. Un transistor est ajouté à la cellule de configuration pour programmer les PCRAMs (non représenté). L'avantage de cette architecture est la densité. Mais l'inconvénient est la consommation statique qui peut être de l'ordre du µA. Pour un FPGA contenant des millions d'éléments mémoire de configuration, la consommation statique serait élevée. Notons que l'on peut adapter ce circuit à la MRAM mais la consommation statique serait beaucoup plus élevée car les résistances aussi bien à l'état parallèle que antiparallèle sont relativement faibles, de l'ordre de quelques kΩ pour des JTMs d'une taille de 130 nm.

XIV.8 REFERENCES

[1] DI PENDINA Gregory. Conception innovante et Développement d'outils de conception d'ASIC pour Technologie Hybride CMOS / Magnétique. Thèse de doctorat en nanoélectronique et nanotechnologies. Université de Grenoble, 2012, 191p.

[2] A. Mochizuki, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 6, pp. 1408–1415, 2005

[3] GUO Wei. Compact Modeling of Magnetic Tunnel Junctions and Design of Hybrid CMOS/Magnetic Integrated Circuits. Thèse de doctorat en micro nanoélectronique. L'Institut Polytechnique de Grenoble, 2010, 203p.

[4] S. Matsunaga, J. Hayakawa, S. Ikeda, K. Miura, T. Endoh, H. Ohno, T. Hanyu. MTJ-Based Nonvolatile Logic-in-Memory Circuit: Future Prospects and Issues, *In Proceedings of Design, Automation and Test in Europe Conference (DATE'09)*, 2009.

[5] M. M. Hassoun, W. C. Black Jr., E.K.F Lee, R.L Geiger, and A. Hurst Jr. Field programmable logic gates uisng GMR devices. *IEEE Transactions on Magnetics*, 33(5) :3307–3309, Septembre 1997.

[6] R. Richter, L. Bar, J. Wecker, and G. Reiss. Nonvolatile field programmable spin-logic for reconfigurable computing. *Applied Physics Letters*, 80(7) :1291, Fevrier 2002.

[7] Daisuke Suzuki, Tetsuo Endoh et Takahiro Hanyu. TMR Logic based LUT for quickly wake-up FPGA. *Tohoku University*, 2008

[8] W. C. Black Jr. and B. Das. Programmable logic using giant-magnetoresistance and spin dependent tunneling devices. *Journal of Applied Physics*, 87(9) :6674, Mai 2000.

[9] Y. Guillemenet, L. Torres, G. Sassatelli. Non-volatile run-time field-programmable gate arrays structures using thermally assisted switching magnetic random access memories. *Computers & Digital Techniques*, Mai 2010

[10] K. J. Hass, G. W. Donohoe, Y.-K. Hong. SEU-Resistant Magnetic Flip Flops. *12th NASA Symposium on VLSI Design*, Oct. 4-5, 2005

[11] F. W. Sexton, W. T. Corbett, R. K. Treece, K. J. Hass, K. L. Hughes, C. L. Axness, G. L. Hash, M. R. Shaneyfelt, and T. F. Wunsch. SEU simulation and testing of resistor-hardened D-latches in the SA3300 microprocessor. *IEEE Trans. Nucl. Sci.*, vol. 38, no. 6, pp. 1521–1528, Dec. 1991.

[12] JAVERLIAC Virgile. Développement d'un modèle compact de la jonction tunnel magnétique de première génération et son intégration dans la réalisation d'architectures logiques reprogrammables hybrides magnétique-cmos. Thèse de doctorat en micro nanoélectronique. L'Institut Polytechnique de Grenoble, 2006 , 205p.

[13] Shoichi Masui, Tsuzumi Ninomiya, Michiya Oura, Wataru Yokozeki, Kenji Mukaida, and Shoichiro Kawashima. A Ferroelectric Memory-Based Secure Dynamically Programmable Gate Array. *IEEE Journal of solid-state circuits*, vol. 38, n° 5, Mai 2003

[14] P.-E. Gaillardon, M.H. Ben-Jamaa, M. Reyboz, G.B. Beneventi, F. Clermidy, L. Perniola, I. O'Connor. Phase-change-memory-based storage elements for configurable logic Field-Programmable Technology, 8-10 Dec. 2010

II. ARCHITECTURE INNOVANTE

XV. DESCRIPTION du FPGA MRAM

L'objectif de la thèse est de concevoir un FPGA à base de cellules MRAM afin de démontrer ses avantages par rapport aux autres types de mémoire utilisés dans les FPGA actuels (antifuse, Flash et SRAM). Après la bibliographie établie en première année de thèse, il était clair que l'avantage principal de la MRAM était son immunité face aux radiations et la possibilité d'éteindre le FPGA sans perdre la configuration, ce qui permet notamment de faire des circuits basse consommation ou d'améliorer la fiabilité. Un domaine dans lequel les contraintes de consommation et de fiabilité sont capitales est le spatial. C'est pourquoi le FPGA conçu dans le cadre de cette thèse a une architecture adaptée au spatial. Du point de vue du circuit utilisateur - c'est-à-dire la structure de la tuile comme le nombre de LUT, d'interconnexion et leur organisation - l'architecture du FPGA est classique. C'est un FPGA de type « îlot logique » composé de tuiles regroupant quatre LUTs à 4 entrées, chaque LUT étant associée à une bascule. C'est la partie configuration qui est la plus importante ici, puisqu'elle contient le circuit à base de cellules MRAM. La particularité étant que l'on peut mettre le circuit de configuration hors tension grâce à une entrée « ON/OFF » afin notamment d'économiser de l'énergie. Dans cette partie on décrira donc le FPGA en commençant par l'architecture d'une tuile. On poursuivra par l'architecture du réseau d'interconnexion. Puis, on finira par le circuit de configuration contenant les cellules mémoire MRAM et qui a donc été l'objet de recherche durant la thèse.

XV.1 DESCRIPTION d'une tuile

Le FPGA est composé de tuiles regroupées entre elles par les interconnexions. On peut voir une tuile comme un motif que l'on va répéter plusieurs fois pour former un FPGA entier. Elle contient quatre LUTs à 4 entrées. Le choix d'une LUT à quatre entrées est motivé par le fait que c'est le bon compromis entre surface et rapidité comme démontré dans [9]. Chaque LUT est associée à une bascule pour pouvoir faire des circuits séquentiels comme des registres. On ajoute un multiplexeur pour choisir d'utiliser ou non la bascule. Dans un circuit numérique, les bascules sont les parties les plus sensibles car la donnée peut à tout moment être changée à cause d'une radiation. Dans le FPGA, chaque bascule est donc durcie aux radiations avec la structure DICE décrite dans l'état de l'art. On a le schéma de la Figure 74.

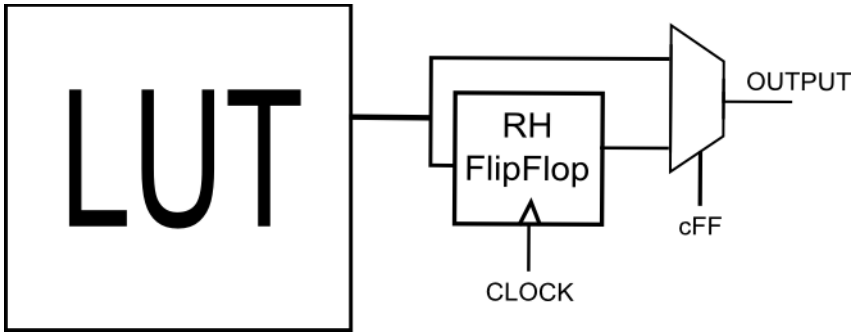


Figure 74 : schéma d'une LUT et de sa bascule durcie aux radiations

Une tuile est composée de quatre LUTs. Dans une tuile, les bascules et les LUTs ont la même horloge, le même signal de reset et le même signal ON/OFF. Ceci est fait pour simplifier le circuit de ces signaux globaux. De même, les signaux de données sont partagés pour simplifier le réseau d'interconnexion local ce qui permet de gagner en surface et rapidité. Il a été montré de façon empirique [9] que le nombre d'entrées optimal I pour une tuile est donné par la formule suivante :

$$I = K/2*(N + 1)$$

Où K est le nombre d'entrées d'une LUT et N est le nombre de LUTs dans une tuile. Dans notre cas : K = 4 et N = 4. Donc la tuile devrait posséder 10 entrées. Ici, la tuile est composé de 12 entrées : trois entrées pour chaque côté pour que la tuile soit symétrique et pour faciliter le routage. Il y a également quatre entrées supplémentaires de rétroaction c'est-à-dire que les sorties des LUTs sont reliées aux entrées dans le cas où on a besoin de cascader plusieurs LUTs. Chaque LUT peut être connectée à chacune des entrées (on dit « fully connected ») grâce à une matrice de connexion locale. Cela permet de faciliter le placement-routage du FPGA car l'algorithme aura un plus grand nombre de connexions possibles. Le schéma de la Figure 75 montre une tuile.

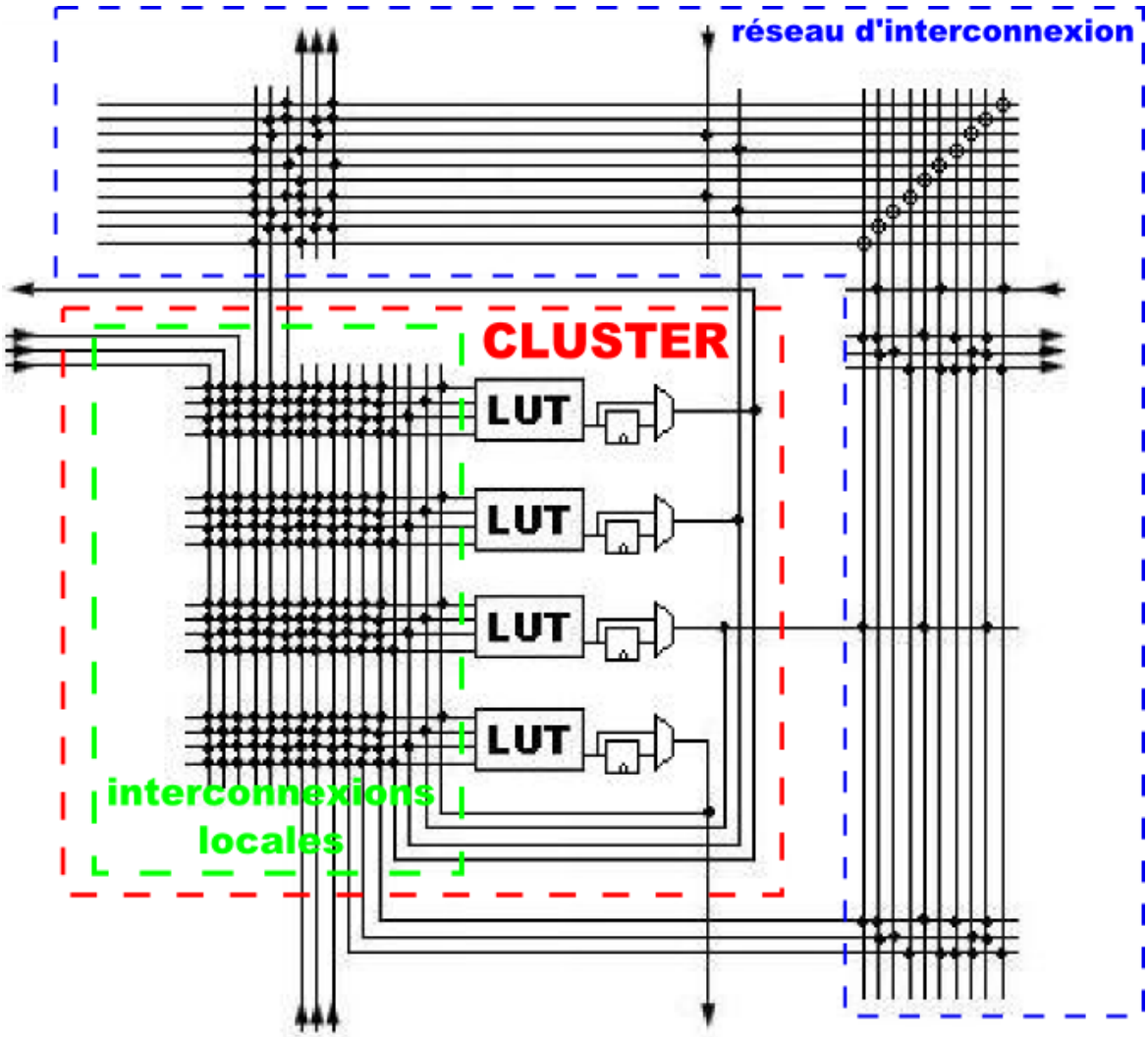


Figure 75 : schéma de la tuile réalisée

XV.2 DESCRIPTION du réseau d'interconnexions

Le réseau d'interconnexions est composé de 10 lignes bidirectionnelles horizontales et verticales. L'architecture de routage entre ces différentes lignes est de type disjoint. Les entrées de la tuile sont reliées à ce réseau grâce à trois lignes sur chaque côté. Chaque sortie d'une LUT est reliée à un côté.

L'architecture du FPGA est de type ilot logique. De cette façon, la tuile et les interconnexions sont faites de façon modulaire. Ceci permet de concevoir un FPGA complet relativement facilement en connectant plusieurs tuiles entre elles. La Figure 76 présente un assemblage de tuiles formant un FPGA complet (Figure 77).

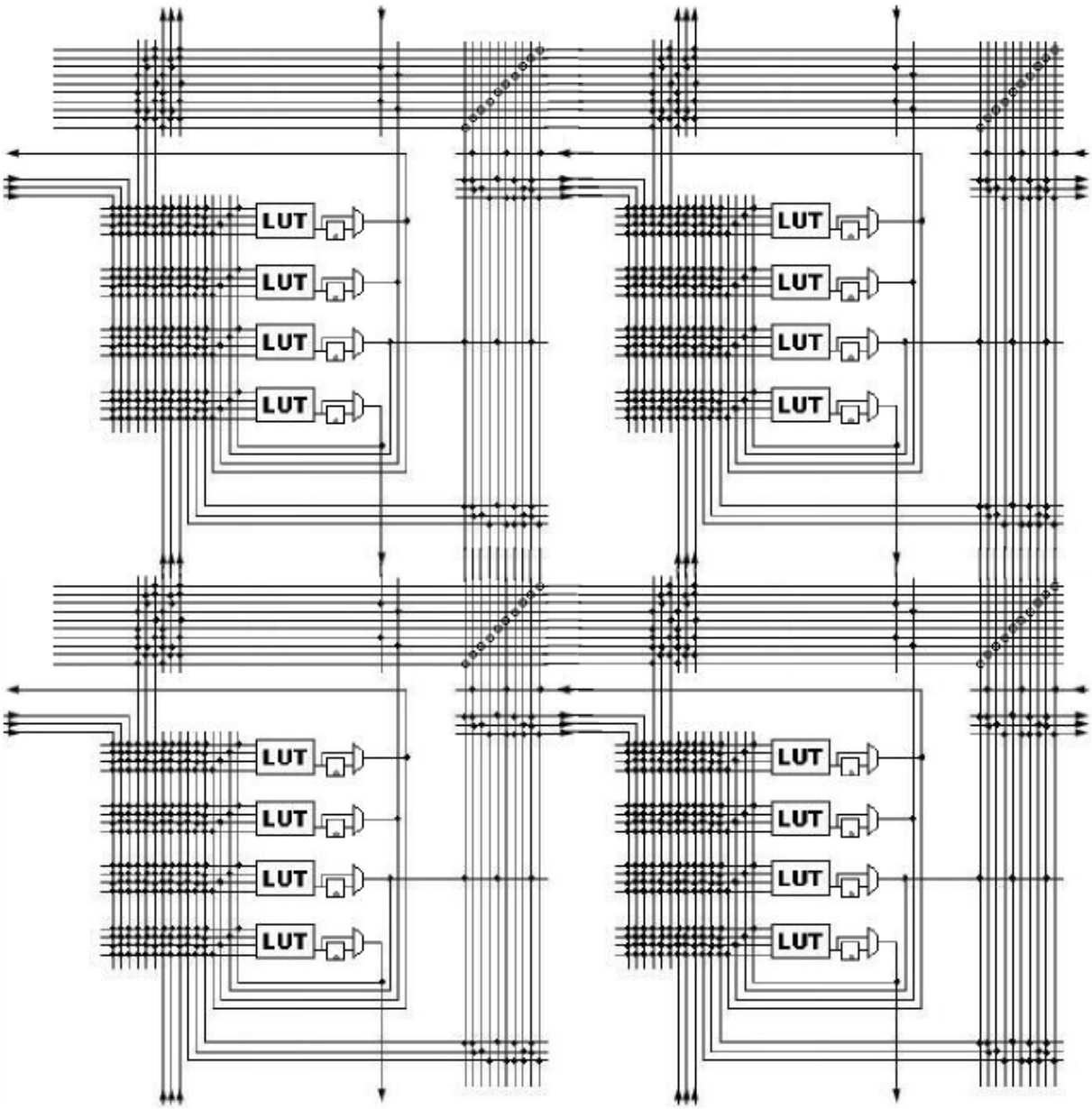


Figure 76 : assemblage des tuiles

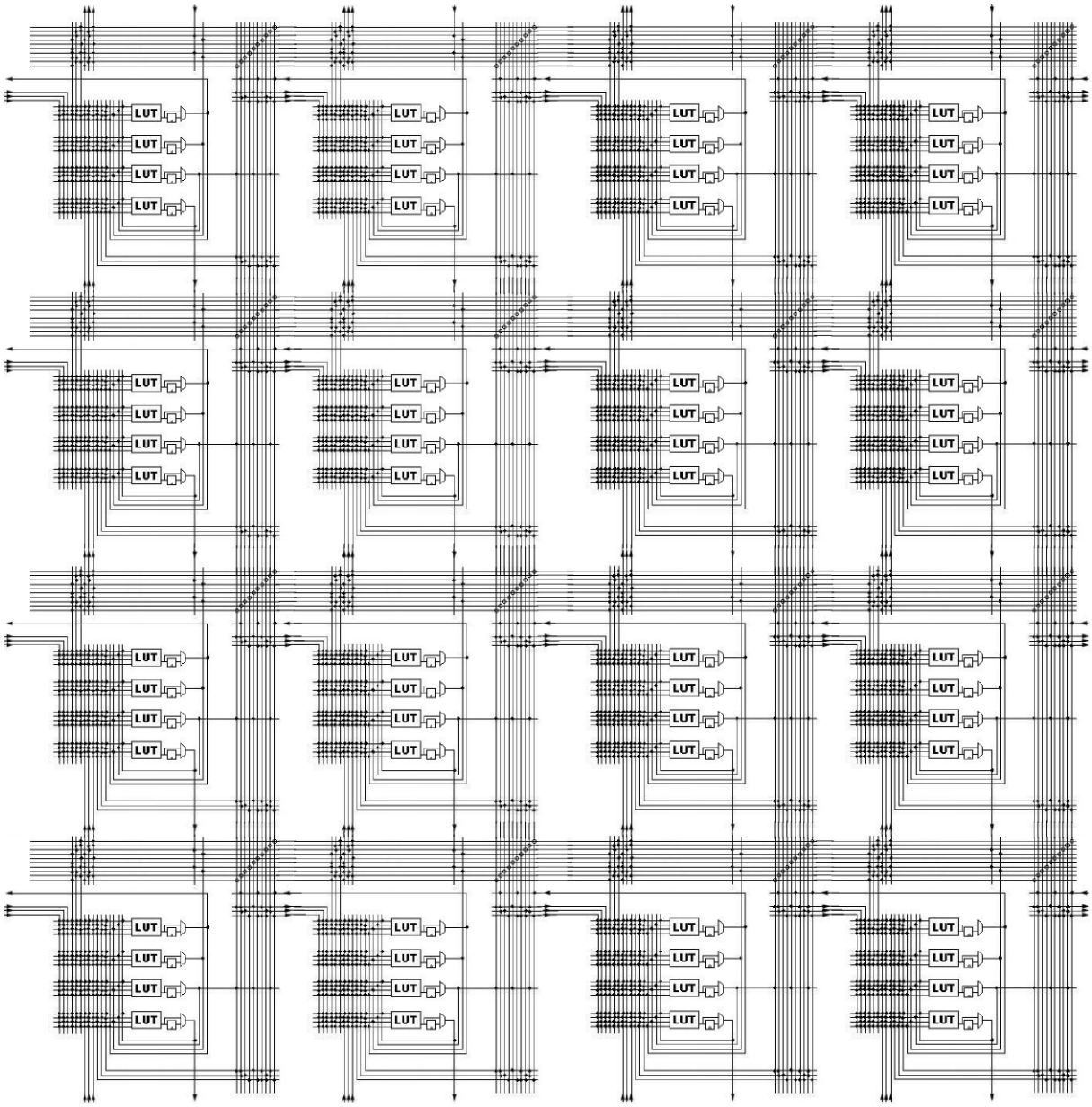


Figure 77 : FPGA basique avec 16 tuiles

Bien que la structure soit modulaire, il faut cependant intercaler des répétiteurs afin de pouvoir router les longs signaux c'est-à-dire ceux qui parcourent plusieurs blocs de routages successifs ou qui sont connectés à un nombre de LUTs élevé. Ces répétiteurs sont des buffers que l'on peut utiliser ou non en programmant une mémoire de configuration. Ces répétiteurs n'ont pas pu être implémentés dans la Tuile conçue durant la thèse par manque de temps.

XV.3 DESCRIPTION DU CIRCUIT DE CONFIGURATION

Le circuit de configuration est la valeur ajoutée de la thèse puisqu'il est fait à base de cellules mémoires de type MRAM. Le circuit a été conçu de façon à tirer parti de tous les avantages de la MRAM à savoir : la densité, la non-volatilité et l'immunité aux radiations. Décrivons d'abord le schéma de configuration de la LUT (Figure 78).

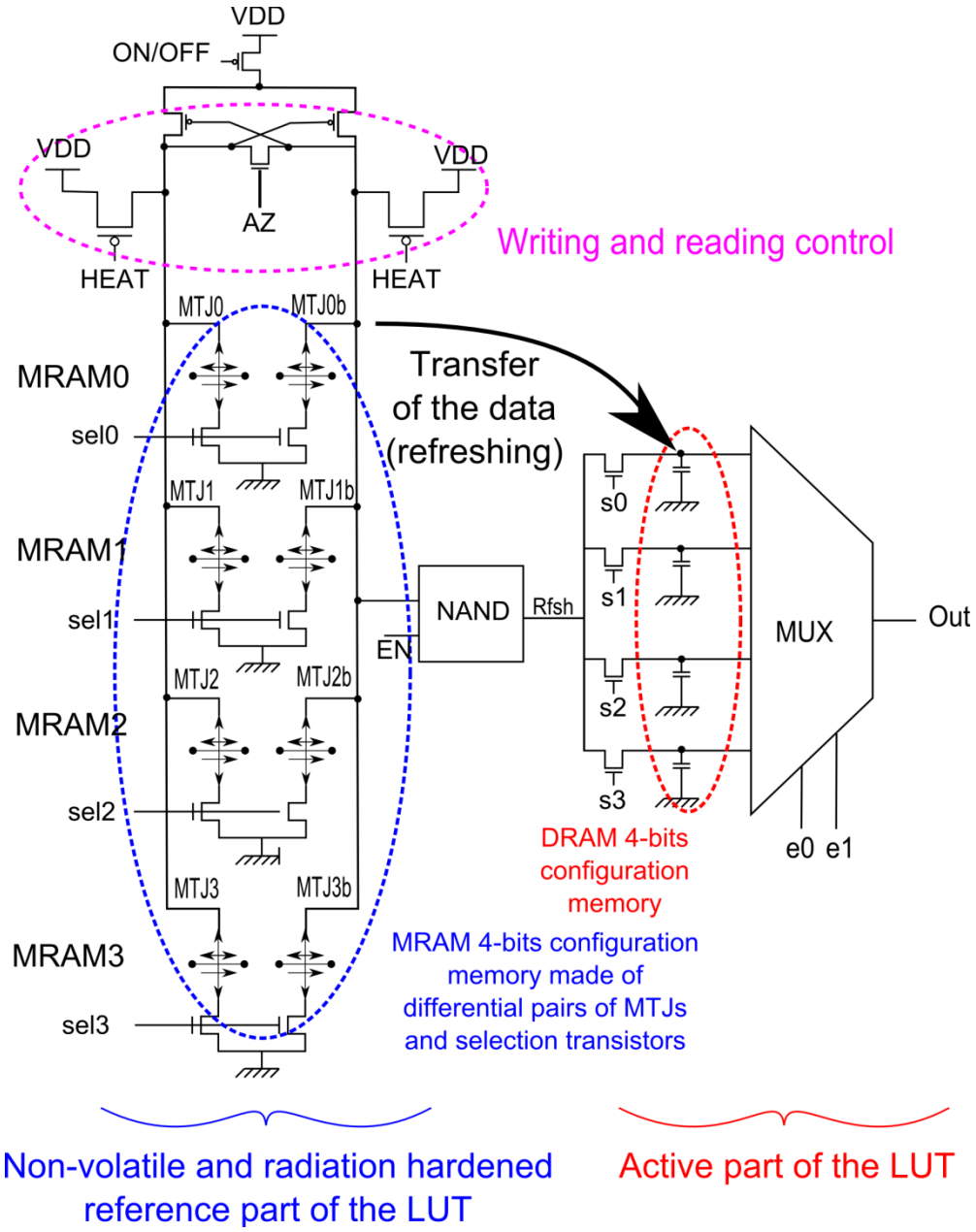


Figure 78 : schéma d'une LUT

La LUT présentée combine cellules mémoires MRAM et DRAM pour tirer avantage de leurs atouts respectifs. Comme le montre la figure, on peut diviser le circuit en 2 parties : le circuit de référence qui contient la mémoire de configuration MRAM, en technologie TAS, qui est donc non-volatile et durcie aux radiations, et la partie active qui comprend le multiplexeur avec à ses entrées des capacités qui stockent la configuration pendant que le FPGA est en fonctionnement. Une cellule mémoire MRAM est composée de deux JTMs dont les états sont complémentaires et de leur transistor de sélection. Régulièrement, pendant le fonctionnement, la configuration des cellules MRAM est lue puis transférée aux capacités. Pour cela, la première cellule mémoire MRAM est sélectionnée grâce à son transistor de sélection. Ensuite, la donnée est lue grâce au module de lecture (le module d'écriture, constitué des transistors de chauffage Heat et le générateur de courant, est coupé). Puis la capacité correspondante est sélectionnée grâce à son transistor de sélection. La porte NAND met en forme la donnée

lue puis rafraichit la capacité. Ensuite, le processus recommence en sélectionnant séquentiellement les autres cellules MRAMs jusqu'à avoir rafraichi toutes les capacités.

Il faut noter que la tension présente sur la capacité DRAM est de $V_{dd} - V_{th}$ lorsque la cellule est configurée à « 1 » en raison du transistor de sélection. On peut remédier à ce problème en appliquant une tension Vsel supérieure à V_{dd} sur la grille du transistor d'interconnexion. Notons que cela n'affecte pas la fiabilité du transistor car cette tension est appliquée uniquement lorsqu'il est sélectionné lors du rafraichissement ce qui est peu fréquent. Cela pose cependant des problèmes d'implémentation puisque deux tensions sont nécessaires. On peut donc implémenter un multiplexeur numérique pour régénérer le signal.

La programmation ne fait intervenir que le bloc de mémoire MRAM. Premièrement, la première cellule MRAM est sélectionnée. Ensuite, les transistors de chauffage Heat sont activés. Ils restent activés jusqu'à ce que la température de blocage soit dépassée. A ce moment là, le chauffage s'arrête (transistors Heat désactivés). Le générateur de courant est activé pour générer le champ magnétique extérieur qui va changer la donnée écrite en mémoire. C'est le sens du courant qui détermine la donnée écrite. Lorsque la température est redescendue en dessous de la température de blocage, le générateur de courant est alors éteint et le processus d'écriture est terminé. On passe ensuite à la cellule MRAM suivante jusqu'à avoir écrit toutes les cellules du bloc mémoire. Pendant le fonctionnement, lorsque la LUT n'est ni en mode programmation ni en rafraichissement, le bloc mémoire est éteint pour limiter sa consommation. Il faut noter également que la phase de rafraichissement est transparente, c'est-à-dire que le fonctionnement de la LUT n'est pas affecté par le rafraichissement. Ceci est indispensable pour le bon fonctionnement du circuit implémenté dans le FPGA car dans le cas contraire, il faudrait interrompre son fonctionnement ce qui compliquerait sa conception et limiterait fortement ses performances.

L'exemple de la Figure 78 est une LUT à deux entrées. Cependant, on peut généraliser à une LUT à N entrées. Dans ce cas, il y a 2^N cellules mémoires de configuration organisées de la façon suivante : 2^N capacités avec leur transistor de sélection, un multiplexeur qui aura donc 2^N canaux, 2^N couples complémentaires de JTMs et donc 2×2^N JTMs. Les JTMs sont complémentaires pour que la lecture soit plus fiable mais on peut utiliser une seule JTM par mémoire de configuration pour économiser en surface. Pour cela, il faut changer le module de lecture et d'écriture pour qu'il détermine la résistance de la JTM à lire. Ces méthodes ont été décrites dans l'état de l'art. Pour concevoir les capacités, on peut utiliser soit des cellules DRAMs soit utiliser la capacité d'un transistor. Dans le cadre de cette thèse, la deuxième méthode a été utilisée car elle est plus simple à mettre en œuvre notamment parce qu'elle ne nécessite pas de masque supplémentaire. Il faut noter également que les capacités de configuration de la LUT doivent être connectées à un inverseur. Ceci est dû au fait que si les capacités étaient directement connectées au multiplexeur, elles se déchargeraient rapidement au fur et à mesure des calculs du multiplexeur. Cela ajoute des transistors, mais c'est indispensable et le surcoût en surface est limité car les mémoires de configuration des LUTs représentent une faible part de la surface de la tuile. La majeure partie étant les interconnexions qui ne nécessitent pas d'inverseurs supplémentaires.

Maintenant, décrivons le circuit de configuration du réseau d'interconnexions locales. Le principe de fonctionnement est le même que celui de la LUT. A chaque transistor permettant de relier deux fils, on connecte une capacité et un transistor de sélection (Figure 79). Le bloc mémoire MRAM adjacent stocke la configuration du bloc d'interconnexion. Pour écrire dans chaque capacité, on va donc sélectionner chaque

capacité séquentiellement ainsi que la JTM associée et rafraichir la donnée comme décrit précédemment.

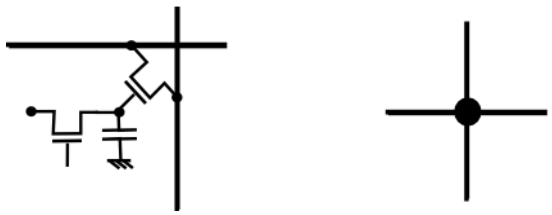


Figure 79 : interconnexion programmable de deux lignes et symbole correspondant

Ce type de configuration permet d'avoir une faible surface en raison du faible nombre de transistors. On retrouve cette structure de mémoire de configuration pour les blocs d'interconnexion et également de routage. La Figure 80 montre comment le bloc d'interconnexions s'interface avec le bloc de mémoire MRAM.

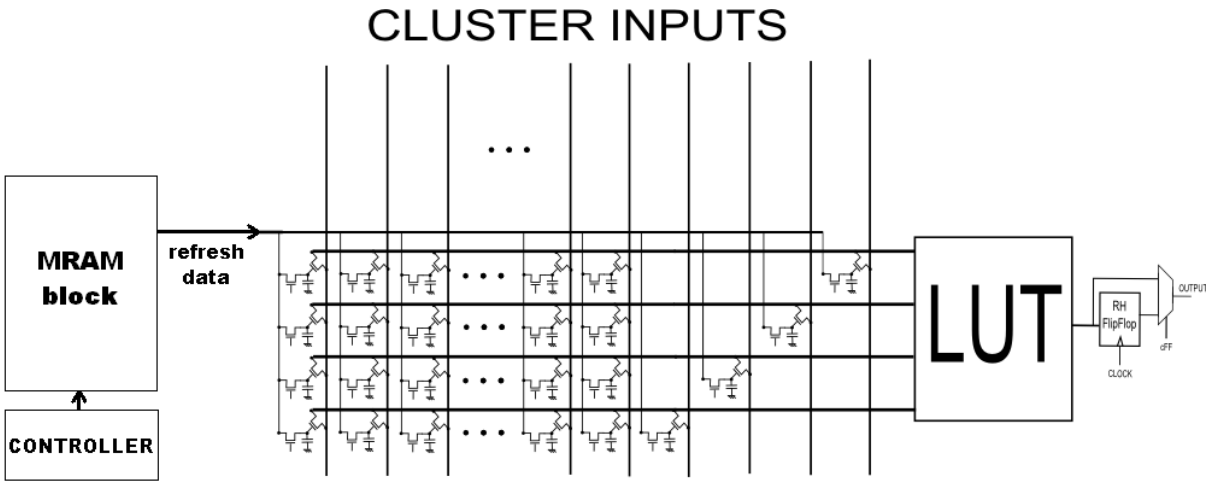


Figure 80 : schéma d'une LUT et de son réseau d'interconnexion programmable

Le contrôleur sélectionne chaque cellule MRAM et la capacité correspondante et transfert la donnée de la cellule MRAM vers la capacité. Concernant le bloc de routage, on a la Figure 81.

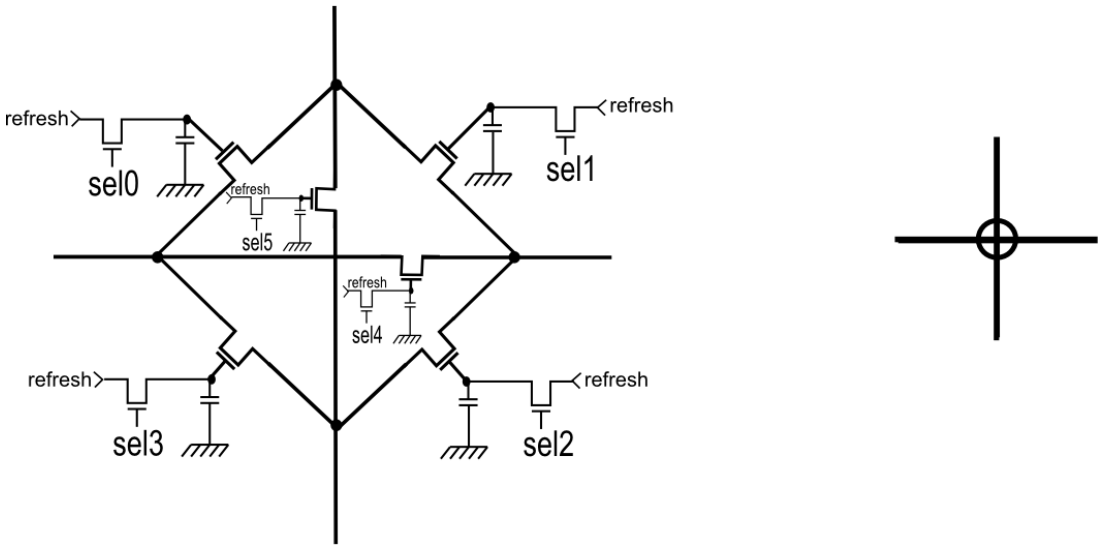


Figure 81 : nœud de routage programmable avec capacité et symbole correspondant

De la même façon, on peut connecter les quatre fils entre eux suivant différentes configurations en fonction de l'activation des différents transistors. Le signal commun, refresh, est la donnée qui doit être écrite à un instant donné dans la capacité sélectionnée. Le bloc de routage, formé de ces points de routage, est très dense en raison du faible nombre de transistors. Les transistors de sélection doivent avoir un très faible courant de fuite pour limiter le nombre de périodes de rafraichissement.

Si l'on assemble les blocs de routage, d'interconnexion et les LUTs, on obtient une tuile (Figure 82).

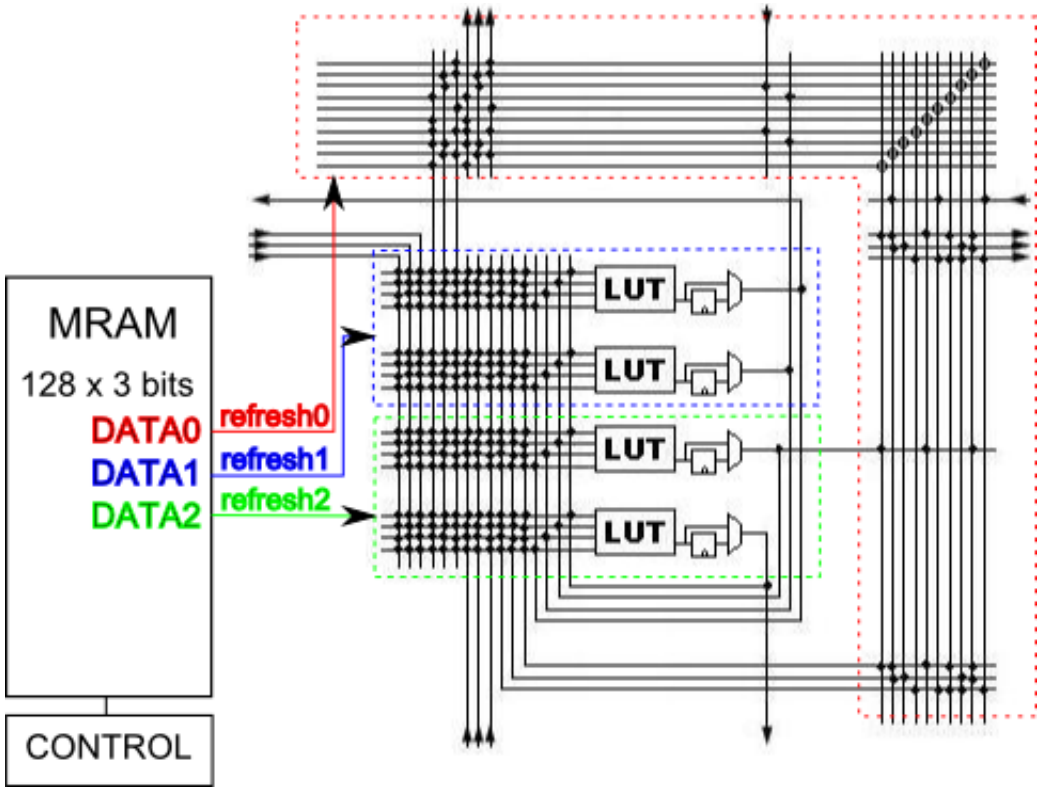


Figure 82 : schéma d'une Tuile

Un bloc mémoire MRAM adjacent à la tuile permet d'écrire la configuration de chaque partie de la tuile. On peut faire varier la taille du bloc mémoire en fonction du nombre de bits de configuration de la tuile.

La programmation du FPGA consiste à écrire chaque bit de configuration dans les blocs mémoire MRAM. On met le FPGA en mode programmation grâce à un signal commun à toutes les tuiles du FPGA : le bit progr. La Figure 83 montre la méthode de programmation.

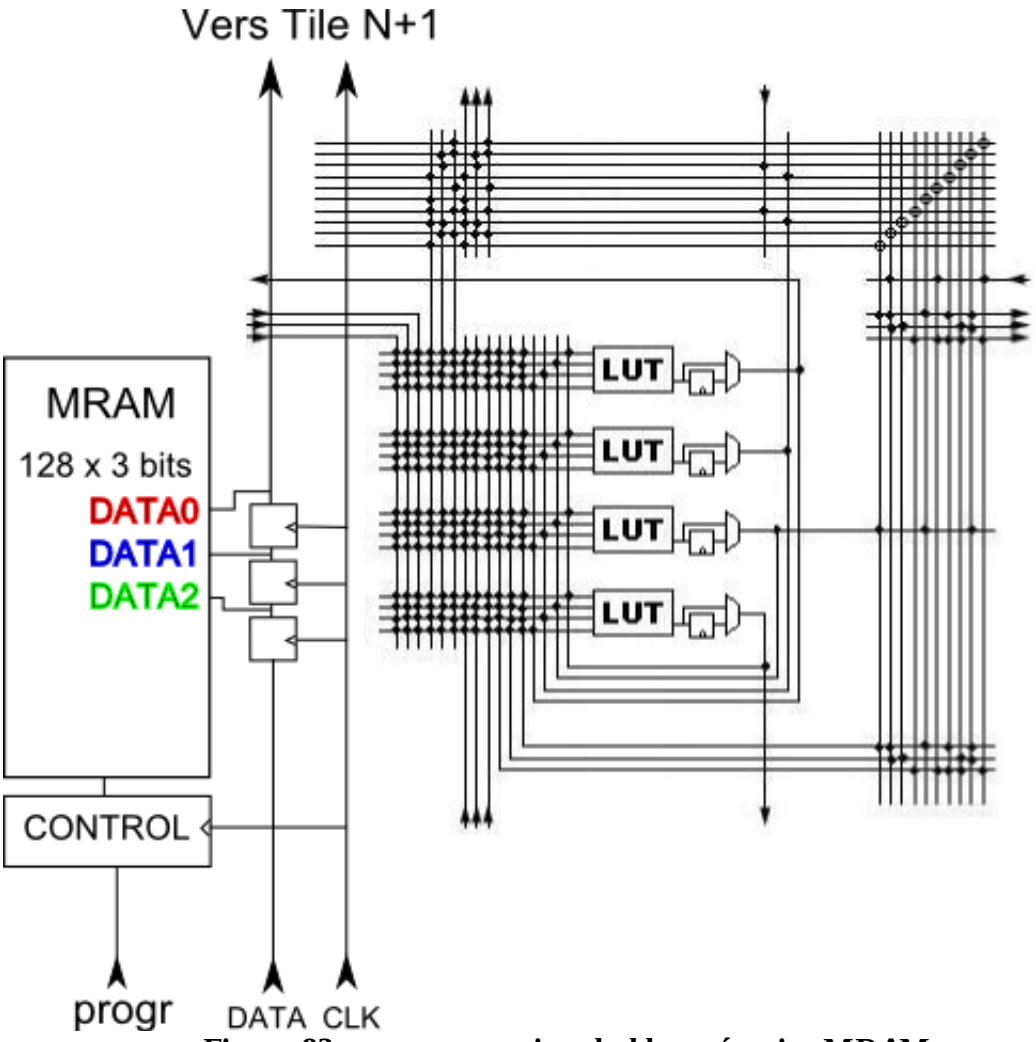


Figure 83 : programmation du bloc mémoire MRAM

Dans ce cas, les blocs mémoire sont écrits comme une mémoire SRAM classique. Le circuit de contrôle permet de sélectionner chaque bit individuellement les uns à la suite des autres et de mettre le bloc mémoire en mode écriture. Les données sont véhiculées dans le bloc mémoire grâce à des bascules. Lors de la phase de programmation, toutes ces bascules sont connectées de façon à former un long registre à décalage. Chaque bascule est connectée à une entrée d'écriture du bloc mémoire adjacent. Le bloc mémoire est organisé en 165 adresses et 3 bits de données. La séquence de programmation du FPGA est la suivante :

- le contrôleur de chaque tuile du FPGA sélectionne l'adresse 0 en mode écriture
- les données à écrire sont véhiculées dans le registre à décalage
- lorsque les bits sont présents dans toutes les bascules, le contrôleur écrit les 3 bits à l'adresse 0 en même temps

- l'opération se répète jusqu'à la dernière adresse. Dans notre cas, jusqu'à l'adresse 165

Il n'a pas été possible de faire un FPGA entier donc cette technique de programmation n'a pas pu être implémentée. Seuls les blocs de mémoire MRAM et les circuits de control de lecture et d'écriture ont pu l'être.

XV.4 Description du bloc mémoire

Le bloc de mémoire MRAM peut avoir plusieurs architectures possibles suivant les contraintes et les compromis à faire. Tout d'abord, on doit déterminer la structure de la mémoire, c'est-à-dire le nombre de bits total (déterminé par le nombre de mémoires de configuration de la tuile) puis la taille d'un mot en mémoire, c'est-à-dire le nombre de bits que l'on peut écrire ou lire à un instant donné. Dans l'exemple précédent, il y avait en tout 165 x 3 bits et un mot comptait donc 3 bits. On pourra alors déterminer le temps nécessaire au rafraichissement c'est-à-dire le nombre de cycles pour lire toute les cellules mémoire. Dans l'exemple, il y a 165 adresses possibles donc il faudra compter 165 cycles pour rafraichir la configuration de la tuile. Ensuite, on doit déterminer le nombre de JTMs par bit de configuration. Dans l'exemple de la Figure 78, il y a deux JTMs par bit de configuration pour des raisons de robustesse par rapport au procédé de fabrication et au circuit de lecture. On peut également utiliser une seule JTM par bit pour économiser du silicium mais il faudrait alors un module de lecture plus complexe et donc plus grand que celui de l'exemple. Il y a donc un compromis à trouver entre surface, fiabilité et rapidité du rafraichissement. Notons également que pour limiter le surcoût en surface des modules de lecture, d'écritures et de control, il faut veiller à ce que leur surface soit négligeable par rapport à la matrice de JTMs.

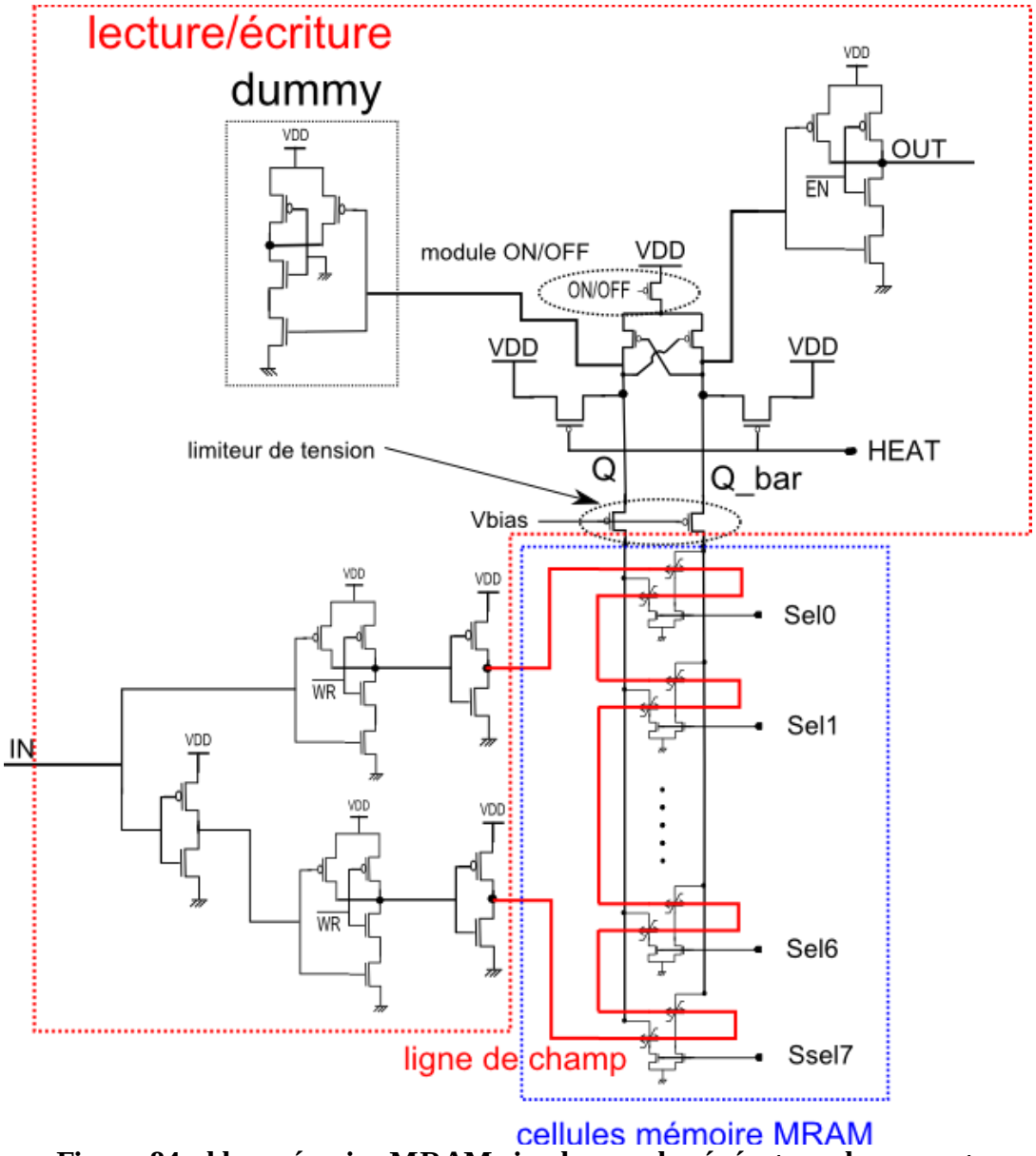


Figure 84 : bloc mémoire MRAM simple avec le générateur de courant

Pour limiter la tension aux bornes des JTMs, on peut ajouter des transistors permettant de limiter la tension en déterminant la tension Vbias. La limitation en tension permet d'améliorer la fiabilité du circuit en ne dépassant pas la tension de claquage des JTMs. Afin d'améliorer encore la fiabilité de lecture, on peut également rendre le circuit symétrique en connectant une porte NAND permettant la mise en forme signal de lecture, sur Q et Q_bar.

XV.5 Fiabilité

L'utilisation de JTM n'immunise pas un circuit contre les erreurs transitoires dues à des particules énergétiques. En effet, si une JTM est bien immune aux radiations, les circuits CMOS qui l'entourent y sont sensibles. Il est donc nécessaire de les protéger. Pour cela, on peut s'appuyer sur les JTMs en tant que données de référence afin de corriger les éventuelles erreurs. Cependant, il faut que les données écrites dans les JTMs

ne contiennent pas d'erreur. Dans le cas où l'on écrit les données de configuration dans la mémoire MRAM, on peut vérifier que les données ont bien été écrites en lisant chaque JTM plusieurs fois. Ce test est long mais ce n'est pas contraignant dans le cadre du développement d'un circuit pour le spatial car une fois écrites, les données de configuration ne seront que rarement réécrites. Ce test de fiabilité peut donc être fait sans que cela affecte les performances du circuit. Concernant la fiabilité par rapport à la phase de lecture, des erreurs peuvent intervenir à plusieurs niveaux. Une radiation peut impacter le circuit de lecture pendant une lecture et générer alors un mauvais résultat. Ceci peut être évité en durcissant les circuits de lecture afin de diminuer leur sensibilité. Ensuite, des erreurs de lecture peuvent survenir à cause des bruits parasites divers. Une marge de lecture doit être prise. On peut également utiliser des twins cells pour accroitre la fiabilité de la lecture. On peut également avoir un claquage de la jonction tunnel durant la lecture. Ceci peut être évité en limitant la tension aux bornes des JTMs durant la lecture grâce à des limiteurs de tension.

Au niveau de la jonction tunnel, on peut avoir plusieurs causes de défaillances. Il y a, comme dans tout circuit, une mortalité infantile due à des barrières tunnel claquées. Le vieillissement de cette barrière dans le temps est également source de défaillance. Ensuite, il y a des dispersions de la résistance dues:

- **au contrôle du diamètre des jonctions**
- **aux redépots**

Ensuite, des problèmes d'écriture peuvent apparaitre dus à l'apparition de vortex, au piégeage de paroi ou autres raisons micromagnétiques, surtout pour les grosses tailles de jonction. Les problèmes de rétention peuvent apparaitre à cause de l'activation thermique. De plus, une donnée peut être erronée à cause d'un champ extérieur parasite au moment de l'écriture.

Dans les paragraphes qui suivent, on considère donc que les données écrites dans la mémoire MRAM de configuration du FPGA sont correctes. Ils décrivent plusieurs structures possibles, à partir de l'architecture de circuit décrite précédemment, afin de fiabiliser le FPGA grâce aux mémoires MRAMs.

XV.5.1 Scrubbing

Le circuit innovant décrit dans ce chapitre permet de s'appuyer sur les mémoires de référence que l'on considère immune aux radiations, les cellules MRAMs, c'est-à-dire que leurs données ne changent pas lorsqu'une particule les touche. Comme mentionné précédemment, le contenu des MRAMs de configuration ne change pas durant le fonctionnement. Leurs données sont donc correctes et le resteront. Le rafraichissement des cellules DRAM peut être utilisé comme méthode de scrubbing afin de corriger périodiquement leurs éventuelles erreurs. Cette méthode ne diminue pas la sensibilité du circuit aux particules mais évite l'accumulation d'erreur. Il faut remarquer cependant que si la fiabilité repose uniquement sur la technique de scrubbing, elle sera peu efficace. En effet, si une particule provoque une erreur dans une cellule DRAM de configuration, alors cette erreur mènera rapidement à une erreur au niveau système avant qu'un rafraichissement ne corrige l'erreur. C'est également le cas avec une période de rafraichissement très courte. La technique du scrubbing n'est efficace que combinée avec une autre technique de durcissement et en particulier la redondance spatiale. Le circuit à implémenter dans le FPGA est ainsi instancié trois fois et connecté à un module de vote majoritaire. Lorsqu'une erreur de configuration survient dans un des circuits redondants, alors elle est masquée par les autres circuits qui ont peu de chance d'être également affectés. C'est pourquoi, d'une manière générale, la technique du scrubbing

est utilisée principalement en combinaison avec la technique de la TMR. L'architecture de circuit de configuration décrite ici est très avantageuse quand elle est utilisée avec une technique de redondance spatiale. Le paragraphe suivant montre comment l'utiliser avec de la TMR.

XV.5.2 TMR

Grâce au fait que les JTMs sont immunes aux radiations, on peut les utiliser comme référence des données. Dans le cas où l'on utilise une technique de redondance spatiale, il n'est alors pas nécessaire de dupliquer les JTMs. Il est alors possible de mettre en commun les JTMs entre les circuits redondants. Cela signifie que le bloc de mémoires MRAM peut être commun aux trois circuits redondants. On ne duplique que les circuits CMOS c'est-à-dire que les LUTs, les interconnexions et les cellules DRAM de configurations correspondantes sont dupliquées tandis que les cellules mémoires MRAM ne le sont pas. Ainsi, lors de la phase de rafraichissement, le bloc de MRAM rafraichi chaque circuit redondant successivement (Figure 85). Grâce à ce partage du bloc de mémoire entre plusieurs circuits, la surface est économisée. On économise en effet deux blocs de mémoires MRAM. Le FPGA devient plus complexe mais la complexité serait la même si on appliquait une technique de TMR classique. De plus, les données lues par le bloc de mémoires MRAM pour rafraichir les DRAMs ne nécessitent que quelques fils pour être véhiculées. Donc cela n'ajoute pas de complexité au circuit de configuration. Le fait que le FPGA soit redondant par construction permet de rendre indépendant les trois circuits redondants ce qui augmente la fiabilité du système par rapport à un circuit implémenté dans un FPGA commercial. En effet, dans les FPGAs existants, les circuits sont implémentés sur la même matrice de LUTs et d'interconnexions. Si une erreur survient sur une LUT, alors elle sera masquée grâce aux deux autres circuits. Cependant, si elle frappe une interconnexion, deux fils de circuits redondants différents peuvent être court-circuités et ainsi provoquer une erreur dans deux circuits redondants. L'erreur ne sera donc pas masquée par la redondance et pourra provoquer une erreur au niveau système. Des algorithmes de placement-routage peuvent être utilisés pour rendre les trois circuits indépendants mais cela ajoute de la complexité. La structure décrite ici complexifie la matrice de FPGA, mais simplifie le placement-routage et diminue la surface occupée grâce au partage du bloc de mémoire MRAM. De plus, la fiabilisation du circuit est transparente pour le concepteur du circuit à implémenter puisque la redondance est déjà présente par construction.

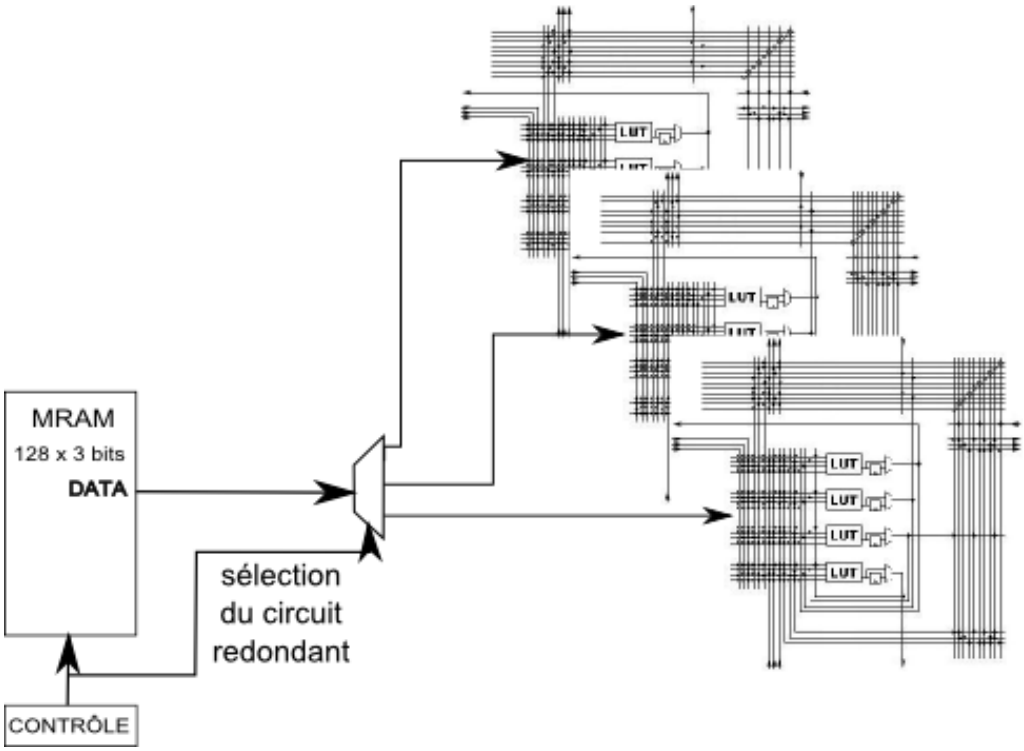


Figure 85 : TMR adaptée à l'architecture innovante

XV.5.3 Redondance temporelle

L'inconvénient principal de la technique de la TMR est le fait que la surface soit plus que triplée et le que la consommation soit également triplée. La redondance temporelle permet de fiabiliser un circuit avec une augmentation de surface beaucoup plus faible mais au prix d'une diminution de la rapidité. On peut encore l'adapter dans notre cas en s'appuyant sur la fiabilité des JTMs. Le principe consiste à faire les calculs une fois durant une période de rafraichissement, puis faire les calculs une deuxième fois après une phase de rafraichissement (Figure 86). Les données sont alors comparées et les calculs dont les résultats sont différents sont calculés une troisième fois. Les éventuelles erreurs apparues sont corrigées après la période de rafraichissement. Il est peu probable qu'une erreur apparaisse deux fois de suite sur le même calcul. C'est donc une bonne méthode pour rendre un système, un processeur par exemple, fiable en économisant en surface. Le seul surcout en surface est la mémoire stockant les données. La quantité de mémoire utilisée est dimensionnée en estimant la quantité de données produites durant une période de rafraichissement. Cette quantité est ensuite doublée afin de pouvoir sauvegarder les données produites durant les deux premières périodes de calculs redondants. On remarque que plus une période de rafraichissement est courte, plus la quantité de mémoire de stockage nécessaire sera faible. Donc en augmentant la fréquence de rafraichissement, on diminue la quantité de mémoire nécessaire mais on augmente également la consommation. Il y a donc un compromis à trouver entre fiabilité et consommation d'énergie. Cela augmente également le temps de calcul car il faut calculer les données au moins deux fois. Si des données ont le même résultat deux fois de suite, alors il n'est pas nécessaire de faire un troisième calcul. Notons également que les erreurs sont rares ce qui implique qu'il y aura très peu de troisièmes calculs à faire. On peut donc en déduire qu'il faudra deux fois plus de mémoires de données et

que les calculs seront deux fois moins rapides. L'application pour laquelle cette méthode est la plus adaptée est le cas où un processeur est implémenté dans le FPGA.

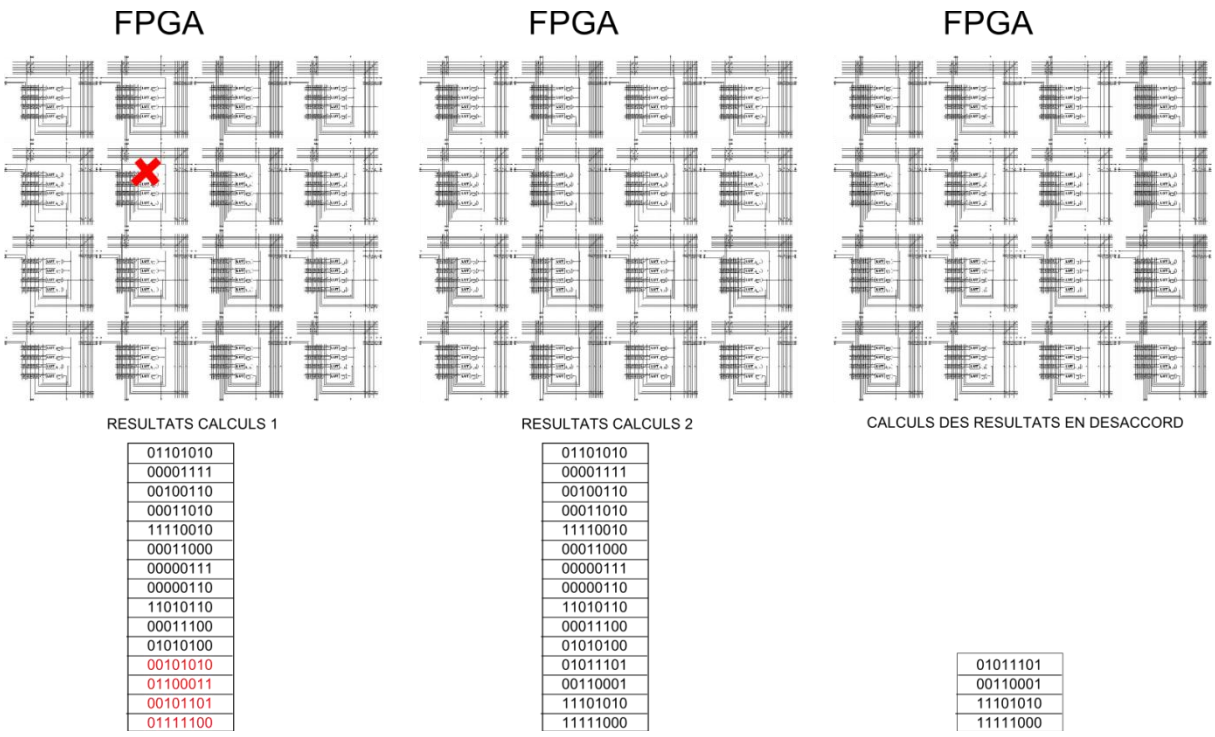


Figure 86 : redondance temporelle appliquée à la nouvelle architecture

XV.6 Optimisation de l'utilisation d'un FPGA

Dans un FPGA commercial, les ressources matérielles sont fixes. Une LUT inutilisée consommera donc de l'énergie. Elle sera incluse dans la consommation statique et sa présence augmente la surface du FPGA et donc son prix. De plus, les interconnexions inutilisées occupent également de la surface et augmentent la capacité parasite des lignes et donc la consommation dynamique sans aucun bénéfice pour les performances du circuit implémenté dans le FPGA. En pratique un FPGA doit être utilisé au maximum à 70% de ses ressources. Au-delà, l'algorithme de placement-routage serait moins efficace pour optimiser la surface et la rapidité du circuit ce qui conduirait, par exemple, à des chemins de données très longs et donc très lents. Les performances du circuit seraient donc diminuées. La solution serait de trouver un FPGA plus complexe mais son prix serait plus élevé.

L'architecture proposée durant la thèse et décrite précédemment est très flexible et permet de mieux optimiser l'utilisation du FPGA ainsi que ses caractéristiques. Il a été imaginé plusieurs méthodes permettant d'optimiser l'utilisation et les caractéristiques du FPGA telles la consommation ou la fiabilité :

- **Utilisation des blocs de mémoires MRAM inutilisées comme blocs de mémoire de données non-volatile et durcies aux radiations**
- **Utilisation des blocs de mémoires DRAM inutilisées en blocs de mémoire de données volatiles**
- **Utilisation des blocs de mémoires MRAM et DRAM inutilisées en blocs de mémoire de données hybride**

- **Mise hors tension des blocs mémoires inutilisés pour économiser de l'énergie**
- **Utilisation d'un bloc de mémoires MRAM inutilisé à la place d'un bloc de MRAM défectueux pour améliorer le rendement de fabrication du FPGA**

La tuile conçue lors de la thèse n'a pas été implémentée avec ses fonctionnalités car elles ne peuvent être utilisées que dans un FPGA entier. Elles n'ont donc pas été conçues afin de ne pas perdre de temps pour des fonctionnalités qui ne pourraient pas être testées. Cependant, les adaptations nécessaires pour les rendre possibles sont mineures. Il est donc envisageable, à l'avenir, de les incorporer à la tuile existante. Les considérations suivantes sont donc théoriques et restent à prouver mais on peut raisonnablement les envisager.

XV.6.1 Mémoire MRAM comme bloc de mémoire de donnée

Afin d'optimiser l'utilisation d'un FPGA, il est possible de transformer un bloc de mémoire MRAM de configuration en bloc de mémoire MRAM de donnée (Figure 87). En effet, c'est possible grâce au fait que le bloc de mémoire MRAM est indépendant des LUTs et interconnexions. On peut donc utiliser simplement ce bloc pour un autre usage sans perturber les autres circuits. On peut utiliser cette fonctionnalité dans le cas où le circuit implémenté dans le FPGA est un processeur qui va stocker son programme dans un bloc de mémoire MRAM. Cela améliore la résistance du système aux radiations et rend non-volatile la mémoire du processeur. Il peut alors utiliser des techniques de réduction de la consommation comme le power gating pour économiser son énergie. La consommation de cette mémoire en mode lecture est comparable à celle d'une mémoire RAM classique. S'agissant d'une mémoire stockant le programme, elle n'est que rarement écrite ce qui nous permet de négliger la consommation du mode d'écriture. Les données sont écrites dans des registres ou des mémoires RAM et les données importantes ou sensibles peuvent être écrites dans la mémoire MRAM. La capacité à utiliser les mémoires de configuration comme des mémoires de donnée est déjà présent dans les FPGAs SRAM et est appelée mémoire distribuée. On peut donc l'adapter à cette architecture.

Cette fonctionnalité peut être ajoutée en permettant la connexion des interconnexions programmables aux entrées/sorties de lecture et d'écriture du bloc de mémoire MRAM. Un bit de configuration permet également de connecter ces entrées/sorties aux interconnexions. Pour ce faire, le circuit de contrôle du rafraichissement du bloc de MRAM devra être adapté pour se connecter au réseau d'interconnexion. On peut également envisager de l'adapter pour transformer le bloc en mémoire FIFO car il contient un compteur, or un compteur est nécessaire dans une FIFO pour incrémenter les adresses de lecture et d'écriture. Il faut cependant veiller à ce que ces ajouts ne détériorent pas la densité du FPGA en utilisant trop de surface. Les changements doivent être mineurs. Pour minimiser leur impact, il est possible de les partager entre plusieurs blocs de mémoire MRAM.

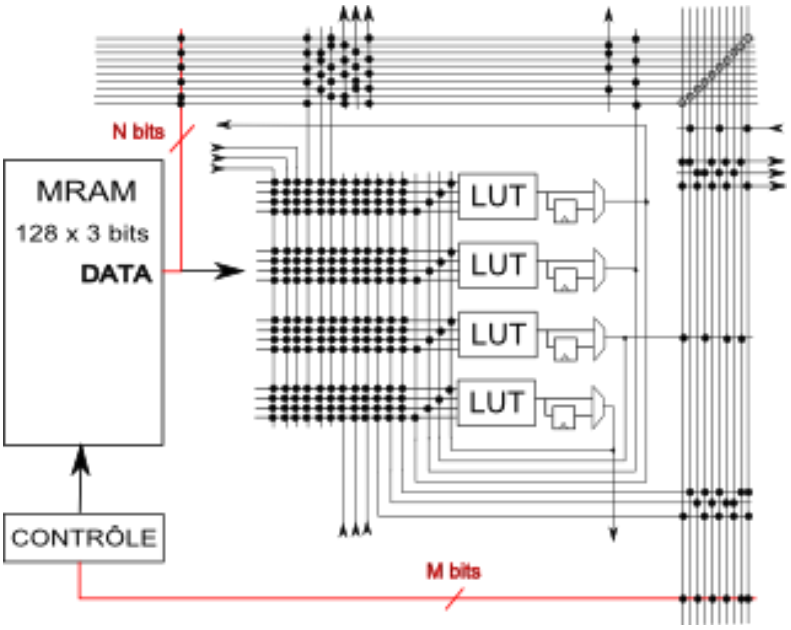


Figure 87 : bloc de mémoires MRAM utilisé comme mémoire de données

XV.6.2 Mémoire DRAM comme mémoire de données volatiles

Il est également possible d'utiliser les mémoires DRAM ou pseudo DRAM des LUTs pour une autre utilisation que de la mémoire de configuration. Elles peuvent être converties en mémoire de données volatiles (Figure 88). Cependant, cette transformation est moins évidente que pour le cas du bloc de mémoires MRAM. Pour cela, Il faut un circuit permettant d'écrire des DRAMs, un autre permettant de les lire, puis une connexion vers les interconnexions programmables. Pour l'écriture, on peut utiliser le circuit de contrôle du rafraichissement qui peut, entre autres, sélectionner la cellule mémoire DRAM à écrire. Une connexion programmable entre le circuit d'écriture des DRAMs et le réseau d'interconnexion est nécessaire pour pouvoir véhiculer les données à écrire. Pour cela, le circuit de contrôle du rafraichissement peut être connecté au réseau d'interconnexions comme dans le cas du bloc de MRAM. Ensuite, pour la lecture, elle peut être implémentée via le multiplexeur de la LUT dont les entrées sélectionnent la donnée à lire et la dirige vers la sortie de lecture. A partir de là, le signal peut être routé dans le FPGA grâce au réseau d'interconnexions programmables comme pour l'utilisation classique d'une LUT. Étant donné que l'on a des circuits de lecture et d'écriture indépendants, la mémoire implémentée est une DRAM à double ports. Notons que l'on peut convertir uniquement les cellules DRAM de configuration des LUTs et non des interconnexions car dans ce cas, il faudrait ajouter beaucoup d'éléments nouveaux ce qui réduirait considérablement la densité du FPGA.

L'implémentation de ce type de mémoire est rendu difficile car il faudrait prendre en compte également le rafraichissement des données ce qui serait complexe. Le concepteur du circuit serait obligé de l'implémenter dans son circuit car il n'est pas possible de l'implémenter simplement en programmant quelques bits de configuration. En effet, cela ajouterait de la complexité qui serait utilisée rarement et diminuerait la densité du FPGA. On peut donc imaginer une mémoire où les données sont lues et écrites de façon hybrides c'est-à-dire que les données sont écrites dans la partie MRAM et lues rapidement sur la partie DRAM. C'est décrit dans le paragraphe qui suit.

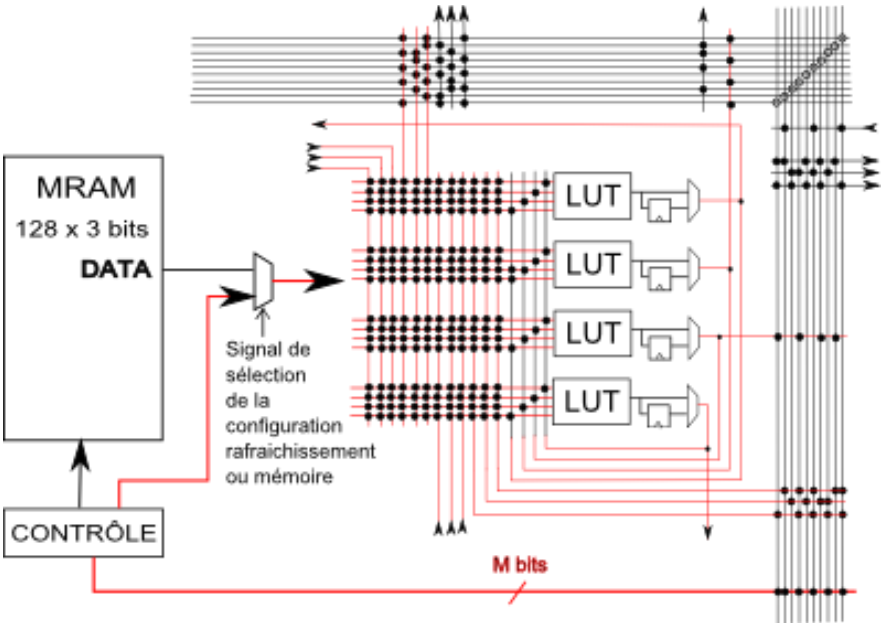


Figure 88 : cellules mémoire DRAM de configuration utilisées comme cellules mémoires de données

XV.6.3 mémoires MRAM et DRAM utilisées comme mémoire de données hybride

Les blocs de mémoire MRAM et de DRAM peuvent être combinés pour former une mémoire non-volatile (Figure 89). On peut ainsi écrire les données dans les cellules mémoires MRAM et les lire à partir des cellules DRAM. Pour cela, le circuit de contrôle doit être adapté pour pouvoir se connecter au réseau d'interconnexions lorsque c'est nécessaire. Ainsi, les données à écrire sont véhiculées à travers le FPGA jusqu'au bloc de mémoire MRAM grâce au réseau d'interconnexion où l'on connecte également les entrées des données d'écriture. La lecture se fait sur les cellules mémoire DRAM des LUTs. Les cellules DRAM des interconnexions ne peuvent pas être utilisées. La lecture se réalise grâce au multiplexeur de la LUT qui véhicule les données en sortie et finalement sur le réseau d'interconnexions grâce à des interconnexions programmables classiques (il n'est pas nécessaire d'ajouter de la logique pour rendre cela possible). Les données des cellules DRAMs sont actualisées grâce au rafraichissement à partir des cellules MRAMs.

L'avantage de cette structure est la rapidité de lecture car on lit directement les données grâce aux LUTs et la non-volatilité des données. Cependant, l'écriture se fait sur la partie MRAM, donc elle consomme de l'énergie et les données peuvent être lues uniquement après un rafraichissement des DRAMs ce qui implique que les données ne sont disponibles qu'après un certain temps.

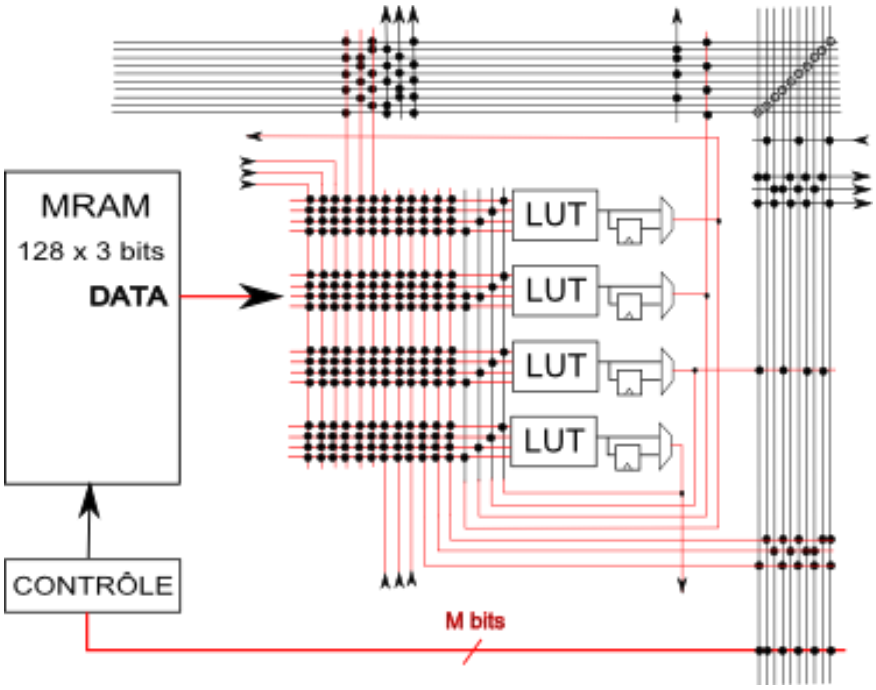


Figure 89 : cellules MRAMs et DRAMs utilisées comme mémoires de données

XV.6.4 Mise hors tension des blocs mémoires inutilisés et power gating

Il est possible de couper l'alimentation du bloc de mémoire MRAM afin d'économiser de l'énergie en dehors des périodes de rafraichissement (Figure 90). Il est également possible d'adapter cette fonctionnalité pour que le circuit implémenté dans le FPGA puisse couper l'alimentation des sous-circuits inutilisés de façon dynamique. Il serait alors possible d'implémenter des techniques de power gating dans le FPGA comme c'est le cas dans certains FPGAs commerciaux de Microsemi (famille de FPGAs IGLOO).

L'adaptation se fait en connectant l'entrée ON/OFF qui coupe l'alimentation au réseau d'interconnexions programmables via un bit de configuration lui-même programmable. Il faut remarquer que le bit ON/OFF ne peut pas être contrôlé par le mécanisme de rafraichissement décrit précédemment. En effet, ce bit contrôle la coupure du bloc de mémoire MRAM, sa configuration doit donc être indépendante du bloc de mémoire MRAM. C'est donc le seul bit de configuration contrôlé par une mémoire SRAM non-volatile (à base de JTMs pour la rendre non-volatile). Sa programmation se réalise comme la programmation d'une cellule MRAM, seule le type de mémoire change. Si la coupure de l'alimentation se fait de façon dynamique par le circuit utilisateur, alors le bloc de mémoire MRAM est programmé comme étant sous tension. Un bit de configuration dont le contenu est stocké dans le bloc de mémoires MRAM permet de véhiculer le signal qui coupe l'alimentation de façon dynamique par le circuit utilisateur. Notons que le transistor qui coupe l'alimentation doit avoir un courant de fuite très faible pour que la coupure de l'alimentation soit efficace c'est-à-dire que les courants de fuite du bloc soient négligeables. Pour que la configuration du mode de coupure de l'alimentation soit complète, il faut un autre bit de configuration déterminant si le rafraichissement doit être maintenu ou non durant la coupure de l'alimentation. Si la coupure de l'alimentation est prévue pour durer longtemps, c'est-à-dire une durée très supérieure à la période de rafraichissement, et que l'éveille du circuit

peut se faire lentement (il faut attendre le temps que les cellules DRAMs soient rafraichies) alors le rafraichissement peut être coupé lors des phases de coupure d'alimentation. C'est le mode très basse consommation. Si la coupure de l'alimentation est prévue pour durer longtemps et que l'éveille du circuit doit se faire très rapidement, c'est-à-dire que l'éveille doit se faire plus rapidement qu'une phase de rafraichissement (quelques dizaines de nanosecondes par exemple) alors le rafraichissement doit être maintenu lors des phases de coupure d'alimentation. C'est le mode basse consommation. Finalement, si la coupure de l'alimentation est prévue pour durer peu de temps, c'est-à-dire que la coupure de l'alimentation dure moins longtemps qu'une période de rafraichissement (quelques dizaines de microsecondes par exemple) alors le rafraichissement doit être maintenu lors des phases de coupure d'alimentation. C'est le mode veille. Cela permet au FPGA d'être très flexible et de pouvoir être configuré facilement en fonction des contraintes de consommation d'énergie de l'utilisateur. Notons qu'il est également possible de couper l'alimentation des LUTs de la même façon, en ajoutant un transistor de coupure à faible courant de fuite afin de couper également les LUTs pour plus d'efficacité. Finalement, les courants de fuite des interconnexions peuvent être fortement diminués en insérant un transistor à très faible courant de fuite dans les buffers attaquant ces interconnexions.

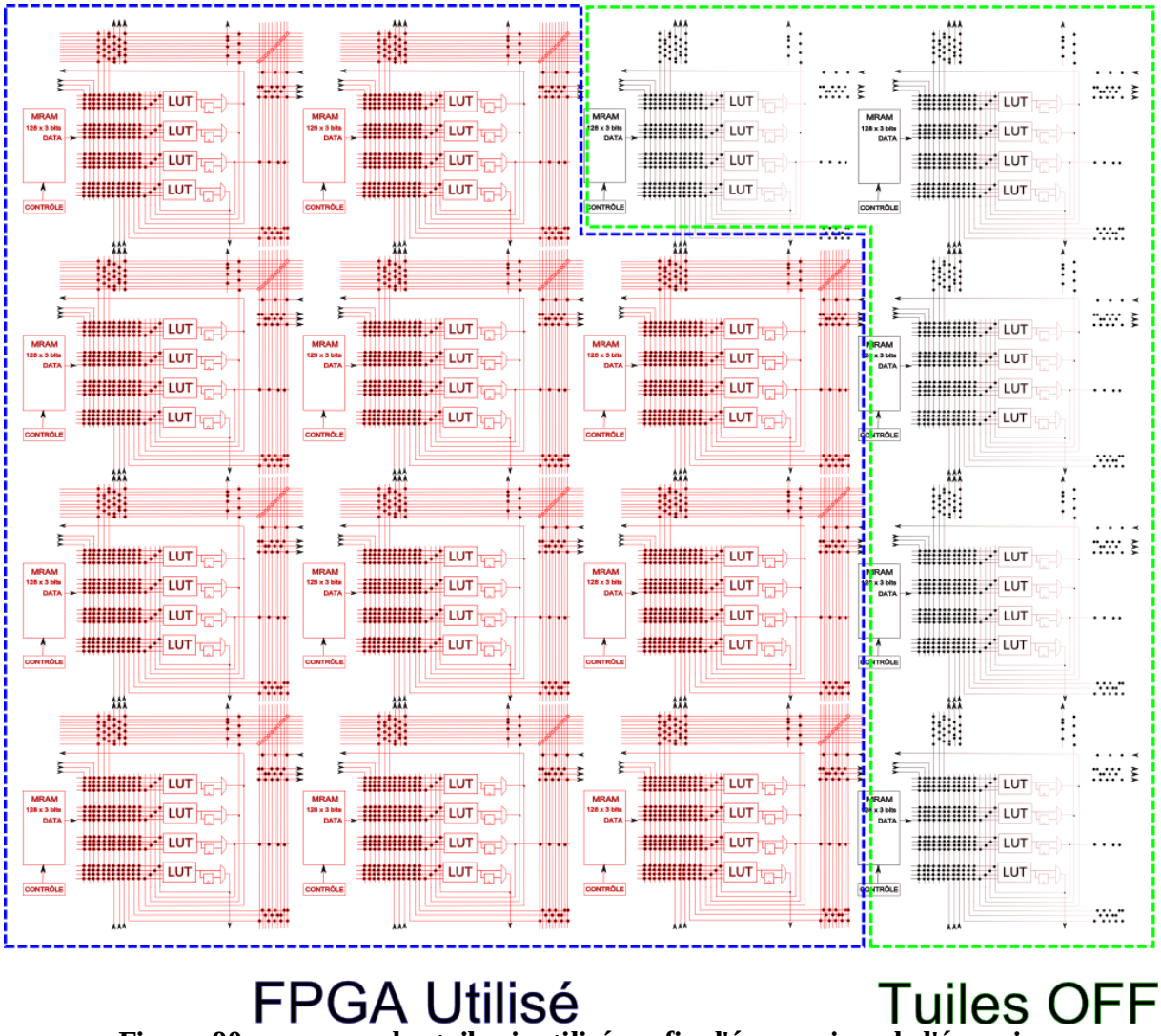


Figure 90 : coupure des tuiles inutilisées afin d'économiser de l'énergie

L'avantage de contrôler la coupure des circuits de façon dynamique est de pouvoir optimiser la consommation du FPGA en fonction de son utilisation. Par exemple, un processeur implémenté dans le FPGA peut contrôler la consommation du circuit. C'est une fonctionnalité qui est possible grâce à l'utilisation de cellules MRAMs qui sont non-volatiles. En effet, on retrouve cette possibilité dans les FPGAs à mémoire Flash de la famille IGLOO de Microsemi (anciennement Actel). Il est en effet possible de couper l'alimentation du FPGA entier. Cependant, on ne peut pas couper des parties du FPGA individuellement. Pour les FPGAs SRAM commerciaux, on ne peut pas le faire aisément. Il faut introduire des circuits complexes à cause de la volatilité des cellules SRAMs et utiliser une mémoire flash externe. Les bénéfices en termes de consommation et de flexibilité sont beaucoup plus faibles qu'avec des cellules MRAMs. Finalement, les ressources inutilisées du FPGA ne consomment pratiquement pas d'énergie lorsqu'elles ne sont pas utilisées grâce à cette technique ce qui n'est pas le cas avec les FPGAs SRAM existant dont les ressources inutilisées participent à la consommation statique.

XV.6.5 Gestion des blocs de MRAM défectueux

Les procédés de fabrication des mémoires MRAM ne sont pas encore matures. Il y a donc de nombreux défauts de fabrication c'est-à-dire que des points mémoires sont défectueux. Dans le cas d'un FPGA à base de MRAMs, cela se traduirait par des blocs de mémoires MRAM défectueux et donc des LUTs et interconnexions défectueuses. Pour y remédier sans introduire de circuits complexes supplémentaires, on peut envisager d'utiliser les blocs de mémoires MRAM inutilisés (Figure 91).

Il faut alors pouvoir véhiculer les signaux du bloc de MRAM vers les tuiles adjacentes. C'est possible grâce à la flexibilité de l'architecture proposée dans cette thèse. En effet, il y a très peu de signaux à véhiculer vers les autres tuiles : le signal lu par le circuit de lecture des cellules MRAMs et un signal pour contrôler le rafraichissement des cellules DRAMs. Il faut ajouter à cela, des bits de configuration afin de déterminer quel bloc de mémoire MRAM sera utilisé pour rafraichir telle ou telle tuile. Cette étape intervient dans la programmation du FPGA ou l'on peut détecter les blocs de mémoire MRAM défectueux. Il suffit de tester les blocs de MRAM comme des blocs de SRAM classiques. Une fois les blocs défectueux identifiés, le logiciel de programmation détermine quel bloc utiliser pour quelle tuile.

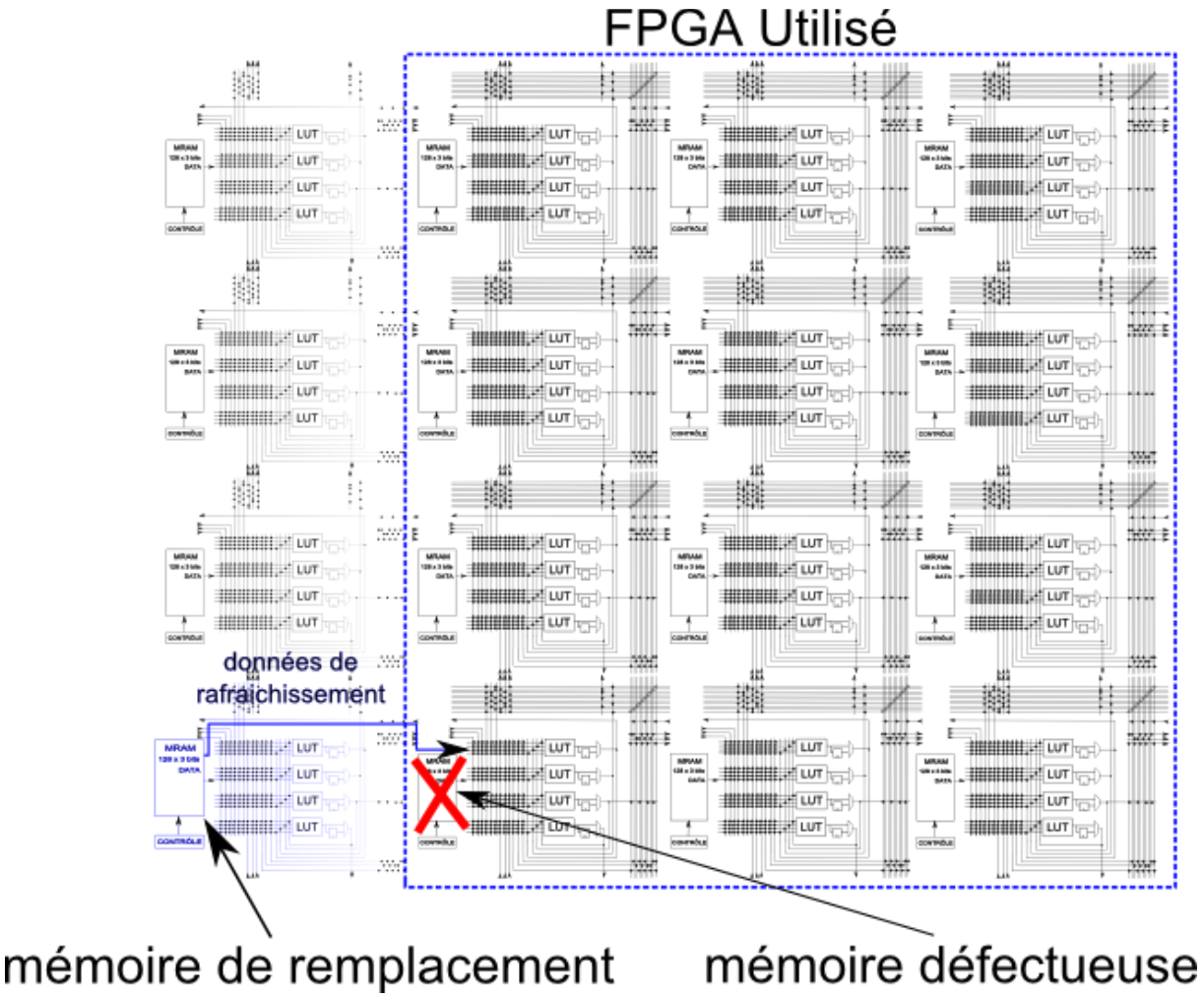


Figure 91 : remplacement des blocs mémoire MRAM défectueux par des blocs inutilisés

XV.7 Evolution possible : FPGA reconfigurable dynamiquement

Une amélioration possible de cette architecture, décrite de façon théorique dans ce chapitre, est de faire en sorte que le FPGA soit reconfigurable dynamiquement. Pour cela, il serait nécessaire de rendre possible l'accès des blocs mémoire MRAM de configuration au circuit utilisateur implémenté dans le FPGA. C'est possible en connectant chaque bloc de mémoire MRAM, adresses et entrées/sorties, au réseau d'interconnexion. Ainsi, interconnexions pourraient être connectées soit aux blocs mémoires MRAM lorsqu'il est en mode reconfiguration dynamique soit aux LUTs en mode normal. C'est le même schéma de principe que précédemment dans la Figure 87 où l'on utilise un bloc de mémoire MRAM comme bloc de mémoire de donnée.

III. IMPLEMENTATION

XVI. IMPLEMENTATION

Le chapitre précédent a décrit le concept du circuit logique programmable à base de JTMs proposé durant cette thèse. A partir de la deuxième année de thèse, plusieurs structures de circuits simples de LUT à bases de cellules MRAM ont été implémentées sous différentes technologies pour étudier les avantages et inconvénients. La première difficulté a été de déterminer quelle structure de FPGA concevoir pour pouvoir comparer les performances avec des FPGAs existants, c'est-à-dire déterminer le nombre et la taille des LUTs, ainsi que la structure du réseau d'interconnexions. Mais les FPGAs commerciaux étaient trop complexes pour pouvoir être implémentés en les adaptant à l'architecture innovante présentée ici. De plus, les FPGAs présentés par des équipes de recherche étaient souvent spécifiques à un domaine, par exemple des FPGAs reconfigurables dynamiquement, ou bien des caractéristiques différentes des nôtres, comme par exemple un FPGA classique alors que le nôtre est durci et destiné au domaine du spatial. Il a donc été décidé de réaliser un FPGA avec des caractéristiques classiques que l'on retrouve couramment dans la littérature comme une LUT à 4 entrées et 4 LUTs par tuile et adapté au domaine du spatial. La comparaison se fait alors avec la même structure mais en remplaçant les cellules DRAM et MRAM par des cellules SRAM. La comparaison s'est faite sur les caractéristiques de consommation, de densité et de fiabilité. Le circuit de configuration d'un FPGA étant le sujet de cette thèse, la rapidité n'a pas été déterminée car elle dépend principalement de la structure du FPGA (nombre d'interconnexions, taille des LUTs, …) et non de sa partie configuration. Différentes structures ont été pensées et testées, principalement durant la deuxième année de thèse. L'implémentation, dans sa forme définitive et présentée dans ce chapitre, s'est déroulée approximativement de la fin de la deuxième année vers la fin de la troisième année coupée par quelques périodes de plusieurs mois comme la conception du démonstrateur en Juin et Juillet 2011, l'absence de licence Cadence en Janvier et Février 2012 et les tests du démonstrateur en Juin et Juillet 2012.

Le choix de la technologie dans laquelle implémenter la tuile a été l'autre difficulté. Les premiers circuits simulés ont été conçus dans la technologie ST CMOS 130nm hcmos9gp avec d'abord des JTMs en technologies TAS puis en STT. Le kit de conception, adapté à la technologie magnétique était disponible tout de suite et était mature car déjà utilisé dans les projets CILOMAG et SPIN, c'est pourquoi il a tout de suite été utilisé et en particulier pour se familiariser avec la technologie TAS. Quelques circuits de bases d'une LUT ont ensuite été conçus dans la technologie CMOS 65 nm de ST aussi bien en technologie TAS que STT. Le passage à cette technologie, plus avancée, visait d'abord à obtenir de meilleurs résultats en termes de consommation à l'écriture des JTMs et de densité des circuits comparée à une technologie plus mature comme la 130 nm. Cependant, il n'a pas été possible d'aller jusqu'au layout car le kit de conception ne prenait pas en compte la partie magnétique et il aurait fallu le modifier entièrement. Notons aussi que les circuits à base de JTMs en technologie STT souffraient souvent de problèmes de convergence lors des simulations. C'est pourquoi, Il a été décidé de repasser à la technologie CMOS 130 nm de ST avec des JTMs en technologie TAS. Cette décision était aussi motivée par le fait qu'il était plus probable de réaliser un démonstrateur dans la technologie 130 nm avec de la TAS que dans une technologie plus avancée. L'implémentation de la tuile présentée dans ce chapitre a donc été réalisée dans la technologie CMOS 130 nm de STMicroelectronics en technologie TAS en ayant l'intention de faire un démonstrateur. C'est pourquoi les circuits

contenant des JTMs ont été conçus de façon très conservative, avec de fortes marges de sécurité qui implique des dimensions élevées des transistors et le fait qu'une mémoire de configuration contienne deux JTMs dont les états sont complémentaires pour plus fiabilité par rapport au procédé de fabrication. Une tuile complète a été conçue et sera décrite dans ce chapitre avec l'architecture de circuit montrée précédemment. Le travail sur la tuile s'appuie donc sur des simulations. Finalement, nous avons eu l'opportunité de réaliser un démonstrateur implémentant une simple LUT-2 et conçu dans la technologie hybride Tower Jazz/Crocus Technology 130nm. Le temps dont je disposais ne m'a pas permis de faire un circuit plus complexe mais il était suffisant pour prouver que le concept innovant était valide. Ce démonstrateur a été testé avec succès et a permis de voir que l'architecture présentée dans la thèse marche. Il sera décrit dans le chapitre suivant. Décrivons donc la tuile implémentée dans la technologie 130 nm de chez ST.

Le travail réalisé sur la technologie hcmos9gp de ST a consisté à réaliser une tuile complète, de la conception au layout puis la simulation post-layout. Les simulations post-layout de la tuile entière n'ont pas pu être faites car la station de travail n'était pas assez puissante. Seuls les circuits de base ont été simulés séparément. Les mesures de consommation ont été effectuées sur chaque bloc séparément puis ont été rassemblées pour estimer une consommation globale. Les estimations ont été faites de façon pessimiste. La consommation présentée est donc approximative mais donne un ordre de grandeur et permet de faire des conclusions sur ses caractéristiques par rapport à l'utilisation d'autres types de mémoire de configuration comme la SRAM. Toute cette étude s'est faite sous CADENCE avec Composer Schematic et le simulateur Spectre pour la partie Front-end. Quant à la partie layout, Virtuoso a été utilisé associé à Assura pour la partie vérification physique (DRC, LVS). Décrivons donc chaque circuit séparément - leur schéma, layout et caractéristiques - puis leur assemblage dans la tuile.

XVI.1 Mémoire de configuration

L'élément le plus important dans un FPGA est la mémoire de configuration comme expliqué précédemment. Dans le cadre de cette thèse, la mémoire de configuration est constituée de cellules DRAM et MRAM. Cependant, la mémoire de configuration « active » directement connectée aux circuits de calcul (Look-Up Table) ou de connexion (transistor d'interconnexion ou de routage) est la DRAM. Elle est constituée d'une capacité qui stocke l'information et d'un transistor de sélection.

Les mémoires DRAM embarquées n'étaient pas disponibles dans le kit de conception de la technologie 130nm de STMicroelectronics disponible à Spintec. Les capacités DRAMs ont donc été réalisées à partir des capacités parasites d'un transistor NMOS : le substrat, le drain et la source sont connectés à la masse tandis que la grille est connectée au transistor de sélection comme le montre la Figure 92.

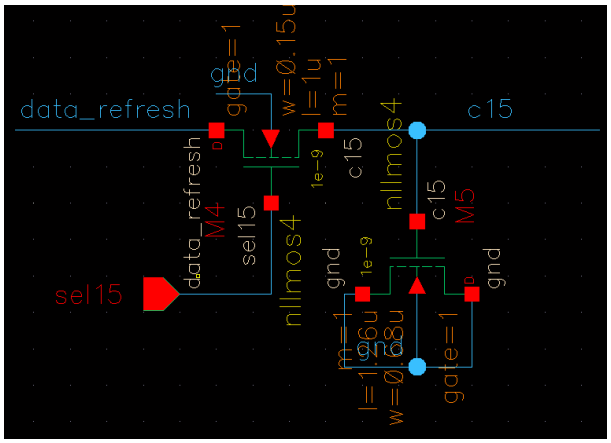


Figure 92 : schéma d'une pseudo cellule DRAM

La capacité peut être faite indifféremment avec un transistor NMOS ou PMOS, cela ne change pas ses caractéristiques. Le choix du type de transistor NMOS ou PMOS de sélection dépend des caractéristiques attendues du FPGA. En effet, la capacité sera reliée à un transistor NMOS d'interconnexion et la tension aux bornes de cette capacité, lorsqu'elle sera chargée, ne sera pas la même suivant que le transistor de sélection sera un NMOS ou PMOS. Si c'est un NMOS, la tension de la capacité sera 0V si le bit est « 0 » et $V_{dd} - V_{th}$ si le bit est à « 1 ». Donc le transistor d'interconnexion ne sera pas saturé lorsque le bit sera à « 1 » et sera donc plus lent qu'un transistor saturé. Mais cette perte de rapidité peut être compensée par le gain en rapidité du à la densité des interconnexions. On peut également augmenter la tension appliquée en entrée afin que la tension stockée soit V_{dd} et non $V_{dd} - V_{th}$. C'est la méthode appliquée dans les mémoires DRAM du commerce. A l'inverse, si c'est un PMOS, la tension de la capacité sera V_{th} si le bit est « 0 » et V_{dd} si le bit est à « 1 ». Donc le transistor d'interconnexion ne sera pas parfaitement bloqué lorsqu'il sera à « 0 » et donc les courants de fuite seront plus grands mais l'augmentation de la consommation statique des transistors d'interconnexion sera compensée par la diminution de la consommation statique des cellules mémoires de configuration. On peut donc utiliser un transistor de sélection de type NMOS si on veut un FPGA basse consommation ou bien de type PMOS pour un FPGA rapide. On peut également faire un compromis entre les deux sachant qu'il faudra adapter les circuits de contrôle.

La valeur de la capacité est déterminée par les dimensions du transistor utilisé comme capacité. On règle W et L des transistors de sélection et de capacité de façon à avoir un temps de rétention suffisamment long. Le temps de rétention est choisi en fonction des caractéristiques attendues du FPGA en termes de fréquence de rafraichissement et de consommation de la phase de rafraichissement. La consommation de cette phase doit être négligeable devant la consommation du circuit utilisateur et la fréquence de rafraichissement doit être assez rapide pour ne pas perdre les données des cellules DRAM et éviter l'accumulation d'erreurs transitoires. Dans notre cas (voir Figure 93), le temps de rétention est de 100µs à 125°C (température du spatial) avec un transistor de capacité de 1µm x 1µm et un transistor de sélection avec un W de 150nm et un L élevé de 1µm pour diminuer les courants de fuite. La durée de rétention a été choisie de façon à ce qu'elle soit prépondérante par rapport au temps de rafraichissement pour que la fréquence de rafraichissement ne soit excessive. Néanmoins, on pourrait choisir une fréquence de rafraichissement plus élevée afin de corriger les erreurs dues aux radiations plus rapidement. Les transistors NMOS utilisés sont standards, ces valeurs peuvent donc être fortement diminuées avec des transistors à faibles courants de fuite. La taille du transistor de capacité détermine également la

consommation lors de la phase de rafraichissement. En effet, plus la capacité est élevée, plus la consommation à l'écriture du bit dans la pseudo DRAM est élevée.

La Figure 93 montre la simulation du schéma de la Figure 94 de la durée de rétention d'une pseudo cellule DRAM. La courbe Vcapa montre la tension sur la capacité. La courbe Dout montre le signal en sortie de l'inverseur qui a en entrée la tension de la capacité. Le signal stocké dans la cellule DRAM est un '1' logique. On voit que la donnée reste durant environ 100 µs. La présence de l'inverseur est nécessaire dans une LUT pour éviter que la capacité se décharge rapidement au fur et à mesure des commutations des transistors. Le fait que la tension Vcapa ne soit pas strictement égale à 0 explique le fait que la tension Dout soit de 1 V au lieu de 1,2V. Il a aussi été possible d'évaluer la consommation à l'écriture de cette cellule. L'énergie nécessaire pour écrire une pseudo cellule DRAM est de 100 fJ.

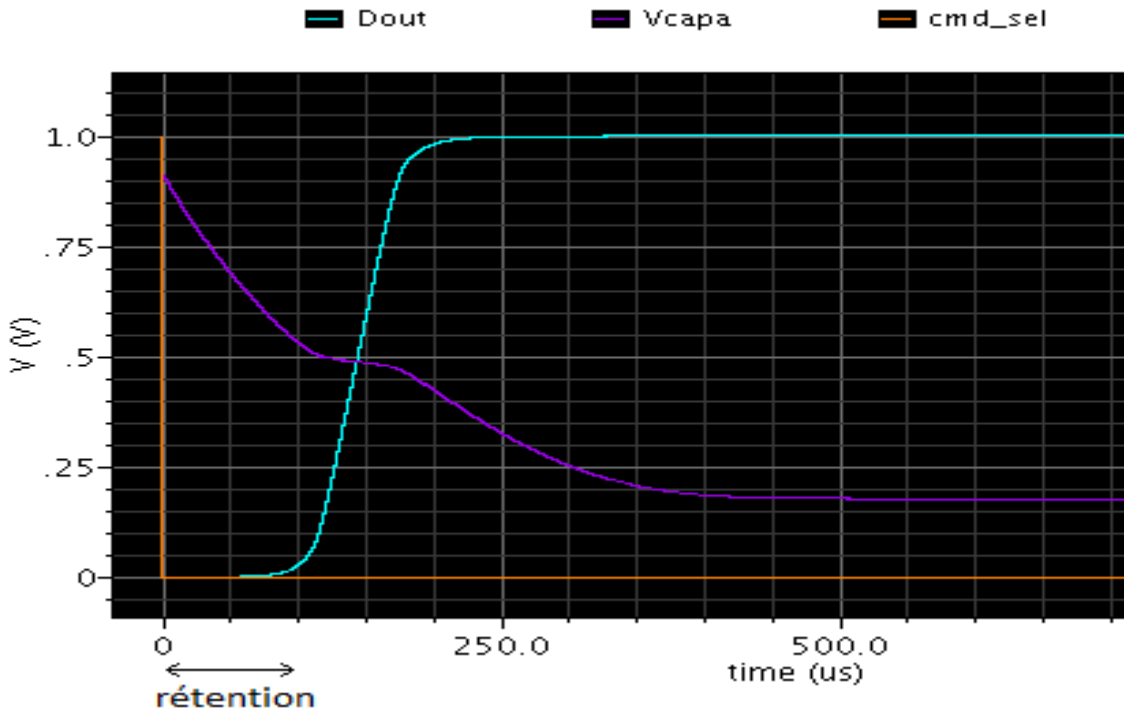


Figure 93 : résultat de simulation de la durée de rétention d'une pseudo cellule DRAM

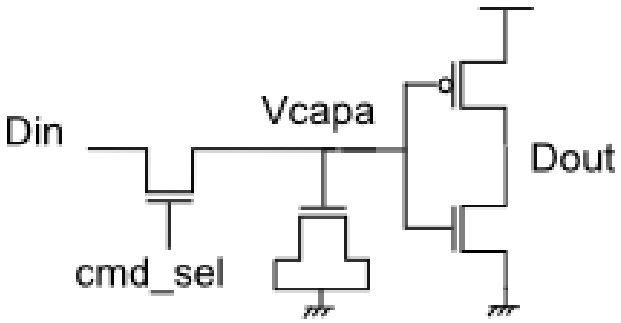


Figure 94 : schéma de simulation d'une pseudo cellule DRAM

La Figure 95 montre le layout d'une pseudo cellule DRAM. Elle a une surface de 6,1 µm². C'est le layout que l'on retrouvera dans les blocs d'interconnexions locales et de routage. On peut voir les deux transistors de sélection et de capacité connectés au transistor d'interconnexion qui a un W de 150nm dans ce cas.

Afin d'avoir une capacité la plus forte possible, les zones de diffusion constituant les source et drain du transistor faisant office de capacité ont été conçues pour occuper le plus de surface possible. De plus, la capacité doit être blindée pour qu'il n'y ait pas de couplage capacitif avec les autres éléments du circuit comme les lignes de métal composant les interconnexions. La capacité a donc été entourée, autant que possible, de couches de métal reliées à la masse. Quant au signal de sélection, la ligne de métal qui le constitue est au niveau du métal 4.

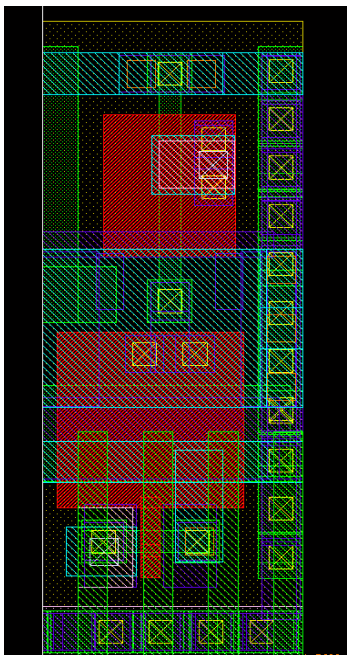


Figure 95 : layout d'une pseudo cellule DRAM

XVI.2 LUT

La LUT est le circuit qui accomplit les calculs lorsque le FPGA est en fonctionnement. Comme décrit précédemment, elle est principalement constituée d'un multiplexeur auquel on connecte des mémoires de configuration. Dans notre cas, on connecte des cellules mémoires DRAM ou pseudo DRAM.

La LUT implémentée est une LUT à 4 entrées. Il y a donc 16 cellules mémoires de configuration. La Figure 96 montre le schéma du multiplexeur. Il a une structure de type décodage par arbre. Elle consiste à utiliser des pass-transistors. Pour sélectionner un signal parmi deux, un pass-transistor est fermé tandis que l'autre est ouvert. C'est donc un multiplexeur 2 vers 1. Pour faire un multiplexeur 4 vers 1, on utilise des multiplexeurs 2 vers 1 en cascade. Ainsi pour un multiplexeur N vers 1, on utilise $2*(2^N - 1)$ transistors. C'est la solution la plus dense. La rapidité est directement liée au nombre de transistors en série. Plus le nombre de transistor en série est élevé (appelé profondeur de l'arbre) plus le délai de propagation est élevé. De plus, l'utilisation de pass-transistors provoque une chute de tension de V_{th} lorsqu'ils sont passants. Donc N pass-transistors en série provoquent, au plus, une chute de tension de $N.V_{th}$ du niveau logique « 1 ». Cette chute de tension est maximum lorsque N pass-transistor de type N sont en série et que le signal propagé est un « 1 » logique. Si on a des transistors P en série, le niveau logique « 0 » ne sera plus à 0V mais 0 - $N.V_{th}$. Dans notre cas, des buffers ont été insérés

pour mettre en forme le signal. Ainsi, quatre pass-transistors sont en série et un buffer est inséré à mi-parcours pour mettre en forme le signal.

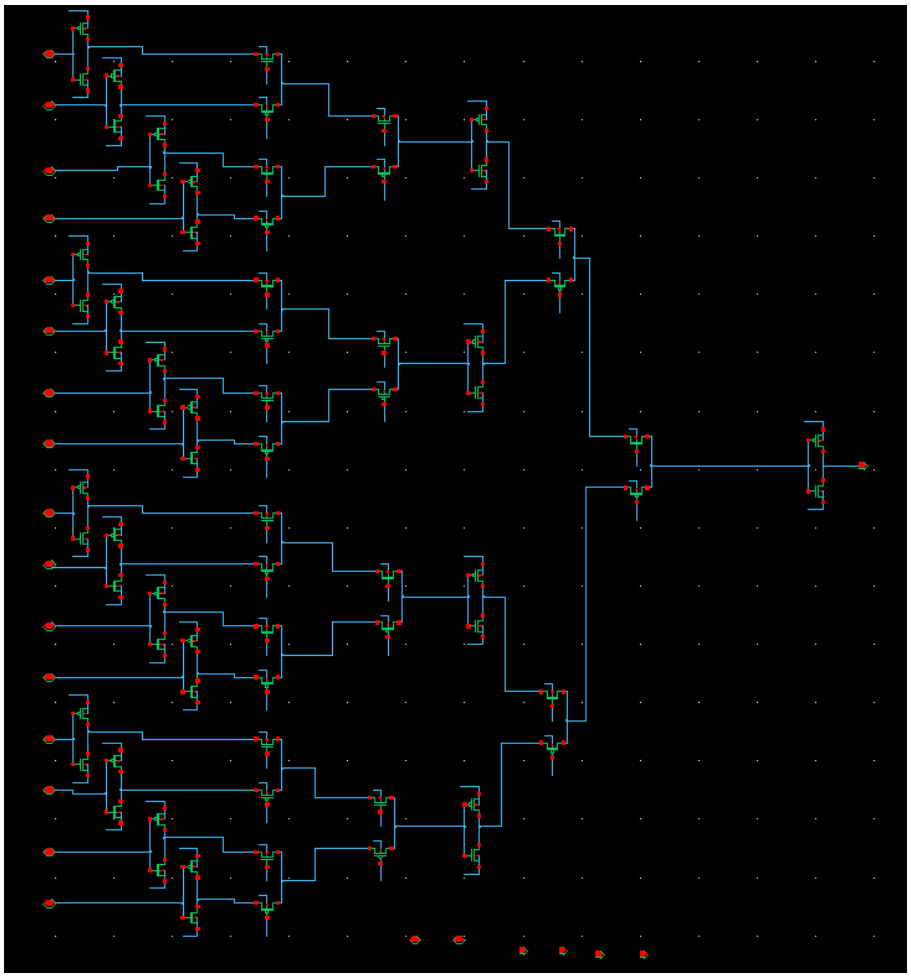


Figure 96 : schéma du multiplexer d'une LUT

Dans le cas de la LUT, il faut noter que les cellules mémoires de configuration DRAM ne doivent pas être directement connectées au multiplexeur. Il faut insérer un inverseur entre les cellules DRAMs et les pass-transistors du multiplexeur car dans le cas contraire, les DRAMs se déchargeraient rapidement au fur et à mesure des calculs. Cela ajoute des transistors mais ce ne sera pas nécessaire pour les mémoires de configuration des interconnexions et la surface d'un FPGA est principalement constituée des interconnexions. Donc le surcout en surface des inverseurs est limité.

La Figure 97 montre le layout de la LUT-4 implémentée. Elle a une surface de 197 µm². Au total, les quatre LUTs occupent une surface de 788 µm² ce qui représente 11% de la surface totale de la tuile. C'est une surface très faible comparée à la surface des interconnexions : 5298 µm² soit 73% de la surface totale de la tuile. C'est en accord avec ce qui est observé dans les FPGAs existants dont ce pourcentage varie entre 50 et 80 %, comme mentionné dans le chapitre sur l'état de l'art. Les cellules mémoires utilisées sont celles décrites précédemment. Le multiplexeur a été placé au centre avec les pseudos cellules DRAM sur les côtés. Pour limiter au minimum les couplages capacitifs des capacités DRAM avec les lignes de métal du circuit, les capacités ont été entourées de couches de métal 2 et 3 pour les blinder. De plus, ce blindage augmente la capacité parasite de cette cellule DRAM. Les lignes de sélection n'ont pas été affichées pour ne pas encombrer l'image. Elles sont placées au niveau du métal 4 et sont verticales. Le

signal véhiculant les données de rafraichissement passe par tous les transistors de sélection et fait le tour de la LUT. C'est la ligne de métal 2 qui entoure le layout. Les entrées de la LUT sont regroupées dans un coin du layout pour être véhiculées vers les interconnexions locales.

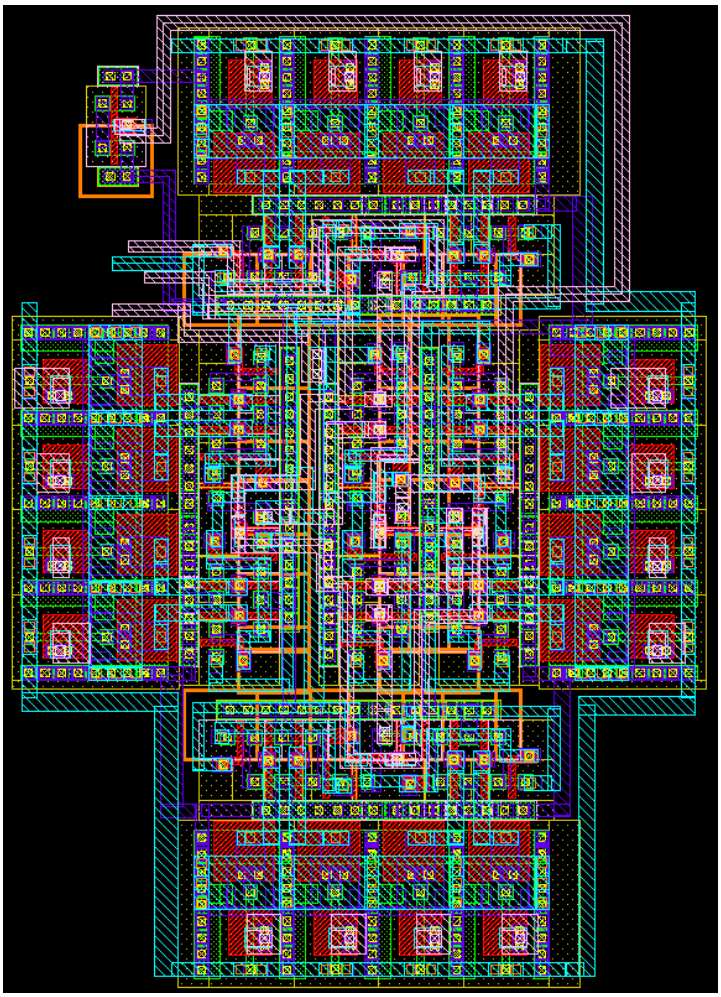


Figure 97 : layout d'une LUT

Bien que le FPGA soit destiné à des applications dans le spatial, les techniques de durcissement aux radiations permettant de supporter les fortes doses de rayonnement, comme l'utilisation de transistor en anneau, n'ont pas été utilisées car elles auraient été trop onéreuse en terme de surface. De plus, elles ne sont pas forcément nécessaires avec cette technologie en 130 nm. En effet, plus la technologie est avancée plus les circuits sont résistants aux doses de radiation, comme expliqué précédemment dans le chapitre sur le durcissement aux radiations.

Tous les transistors utilisés dans le circuit sont des transistors standards. Les avantages en termes de surface auraient été plus évidents avec l'utilisation de transistors à très faibles courants de fuite, pour les transistors de sélection des cellules DRAM, mais ils n'étaient pas disponibles dans le kit de conception disponible à Spintec et la mise à disposition de ces transistors aurait engendré des contraintes trop fortes.

Une simulation d'une LUT-4 a été faite afin de mesurer les caractéristiques de circuit élémentaire (Figure 98). Les 16 signaux de sélection sélectionnent chaque cellule DRAM séquentiellement puis le signal DATA indique la donnée de rafraichissement à écrire qui provient du bloc de mémoire MRAM adjacent. Toutes les entrées possibles sont appliquées séquentiellement à la LUT pour vérifier le contenu de chaque cellule mémoire DRAM afin de voir qu'elles ont bien été écrites. En faisant durer la simulation

durant 1 ms, on peut voir la durée de rétention des données dans les cellules DRAM et donc déterminer la fréquence de rafraichissement nécessaire. Dans notre cas, la durée de rétention est de 100 µs environ à 130°C ce qui donne une fréquence de rafraichissement de 10 kHz. La simulation a permis également de déterminer sa consommation en fonctionnement. La Figure 99 montre le courant afin de déterminer la consommation statique et dynamique. Sa consommation dynamique est de 23 nW/MHz et sa consommation statique est de 62 µW à 130°C. C'est une valeur élevée mais elle est due à la température de fonctionnement très élevée de 130°C. Il faut noter également que le fonctionnement n'est pas affecté par la phase de rafraichissement. Il n'y a donc pas d'arrêt des calculs lors du rafraichissement et c'est transparent pour le circuit utilisateur ce qui est indispensable pour la simplicité de fonctionnement et de programmation.

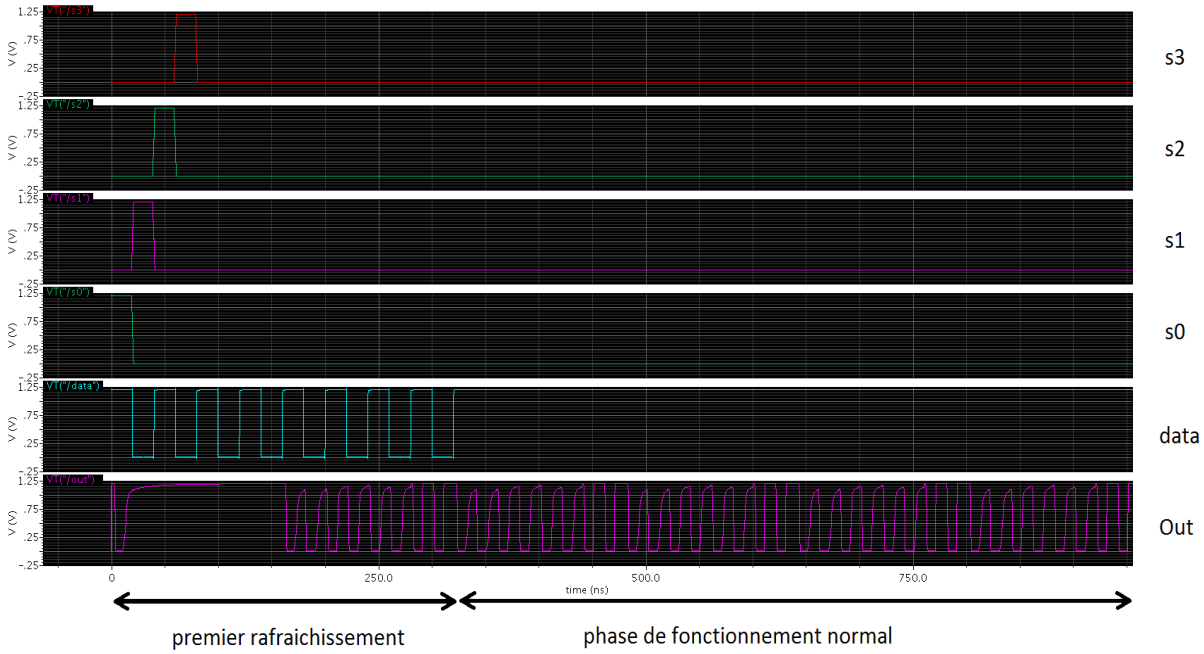


Figure 98 : simulation de la LUT4

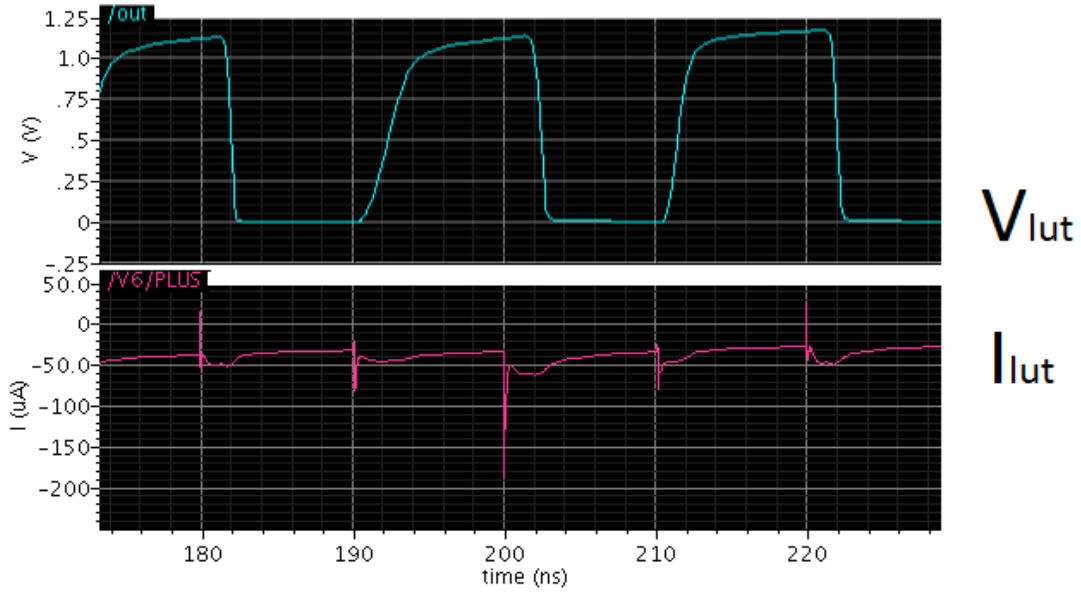


Figure 99 : simulation d'une LUT 4 avec le courant

XVI.3 Bloc de mémoire MRAM

Le bloc de mémoire MRAM stocke la configuration du FPGA de façon fiable et non-volatile. La MRAM est la mémoire de configuration de référence. Ce bloc est principalement constitué de deux circuits : la matrice de JTMs et le circuit de lecture/écriture.

XVI.3.1 Matrice de JTMs

Un bit de configuration est constitué de deux JTMs dont les états sont complémentaires. Cela permet de faire une lecture en différentielle donc plus fiable des données écrites dans les JTMs. La Figure 100 montre quelques JTMs composant le bloc de JTMs. Chaque JTM possède un transistor de sélection et les deux transistors de sélection des JTMs complémentaires sont reliés au même signal de sélection.

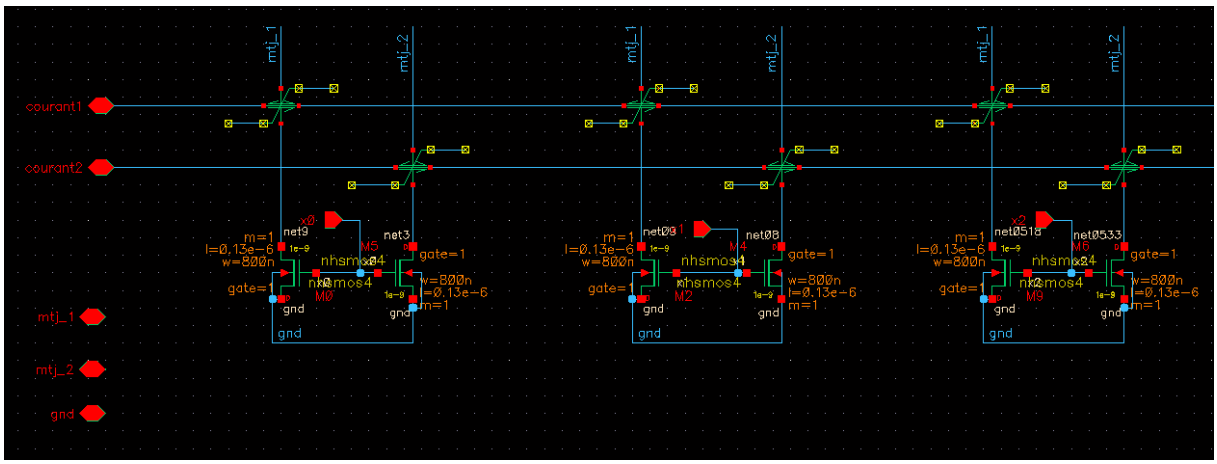


Figure 100 : cellules MRAMs du bloc mémoire MRAM

Les JTMs sont reliées au circuit de lecture et d'écriture par les signaux mtj_1 et mtj_2 qui sont les équivalents des signaux BL et BL_bar dans un bloc de mémoire SRAM classique. Les transistors de sélection sont réalisés avec des transistors à faible Vth pour diminuer leur résistance à l'état passant. Afin de limiter les courants de fuite, des transistors à faible courant de fuite sont introduits pour permettre de couper l'alimentation lorsque les JTMs ne sont pas en phase d'écriture. L'écriture est détaillée au paragraphe suivant. La technologie de JTM utilisée est la TAS. Pour l'écrire, il faut donc faire passer un courant à travers les jonctions pour les chauffer et générer un champ magnétique pour changer l'aimantation de la SL. Cette ligne de champ est partagée par toutes les JTMs d'un même bloc de cellules MRAM. Sa longueur est limitée par sa résistance parasite qui, si elle est trop grande, limite le courant généré. Dans ce cas, le champ magnétique généré serait trop faible pour pouvoir écrire les JTMs. Dans notre cas, la ligne de champ est partagée entre 128 JTMs.

La Figure 101 montre le layout d'une matrice de 64 bit de configuration MRAM et la Figure 102 un détail de la matrice. La surface de cette matrice est de 432 µm². Afin de gagner en surface, les sources des transistors sont mises en communs. Cependant, le plus

contraignant en terme de surface est l'espacement entre les JTMs qui doit être de l'ordre de 1 µm. On peut donc imaginer une surface plus réduite lorsque la densité des cellules MRAMs aura été améliorée car au moment où sont écrites ces lignes, la technologie n'est pas encore mature. Pour avoir un bloc mémoire plus grand afin de programmer tous les bits de configuration d'une tuile, on met plusieurs matrices MRAM en parallèle avec leur propre circuit de lecture et d'écriture.

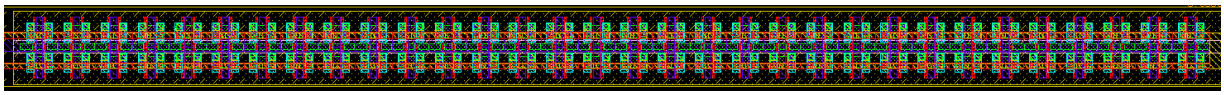


Figure 101 : layout d'un bloc mémoire MRAM 64 bits

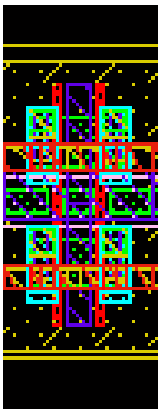


Figure 102 : layout de quatre JTMs de la matrice

XVI.3.2 Circuit de Lecture/écriture

Le circuit de lecture et d'écriture des cellules MRAM est composé d'un latch qui lit les données différentielles des JTMs, des transistors de chauffage et de génération du courant de champ permettant l'écriture et des circuits de control des signaux. La Figure 103 montre ces circuits. Il y a donc un latch composé de deux transistors PMOS. Afin de mettre hors tension le latch lorsque le bloc MRAM est inactif, un transistor PMOS a été inséré pour faire du « power gating » et donc diminuer fortement la consommation statique. Dans l'idéal, ce transistor aurait dû être un transistor à faible courant de fuite mais il n'était pas disponible dans le kit de conception. Ensuite, deux transistors de chauffage sont connectés aux signaux mtj_1 et mtj_2 pour permettre le chauffage des JTMs en phase d'écriture et couper l'alimentation en dehors de cette phase. Ils devraient également avoir un courant de fuite faible. Ensuite des portes NAND permettent de contrôler l'écriture à partir des signaux EN, data et Clk. Ainsi, pour permettre l'écriture, En doit être mis à « 1 ». Ensuite, Clk doit être à « 1 » pour chauffer. Lorsque Clk repasse à « 0 », les JTMs refroidissent. On applique pendant ce temps, le courant qui génère le champ magnétique grâce au signal En qui autorise la génération du champ et Data qui détermine son sens. Le champ est généré par un circuit de type pont en H. Les entrées de control, write1 et write2, permettent de déterminer le sens du courant. La Figure 104 montre les signaux d'entrées pour écrire les données dans les JTMs. Plusieurs données sont écrites. La suite de données écrite est une alternance de 1 et de 0 : 0101010101. La Figure 105 montre une partie des signaux de sélection des JTMs à écrire. Les JTMs sont sélectionnées les unes à la suite des autres.

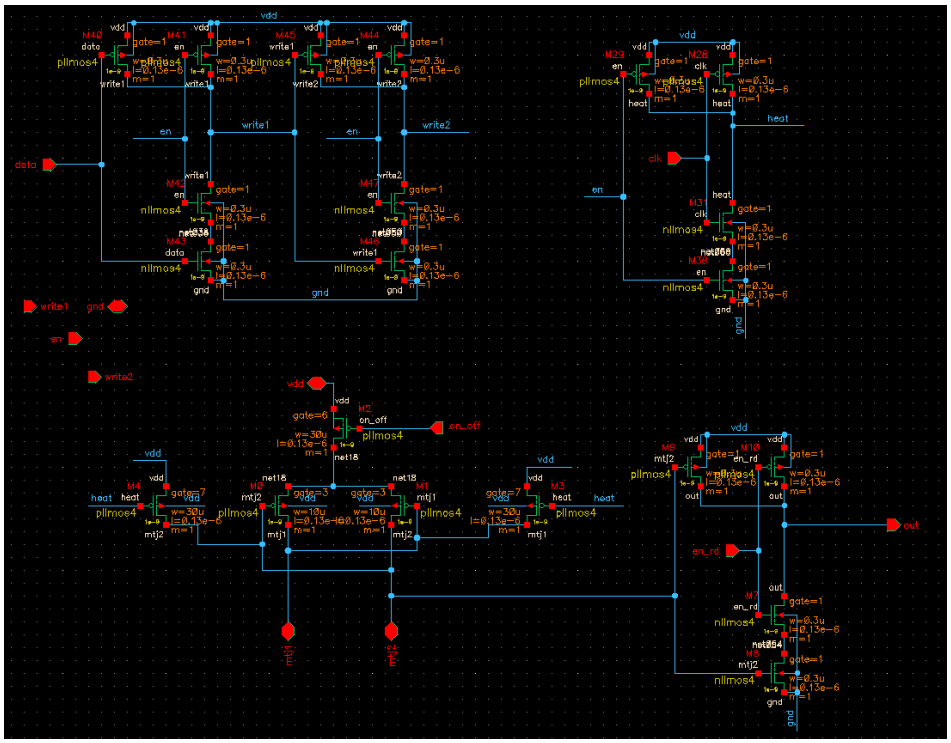


Figure 103 : schéma d'un circuit de lecture et d'écriture d'un bloc mémoire MRAM

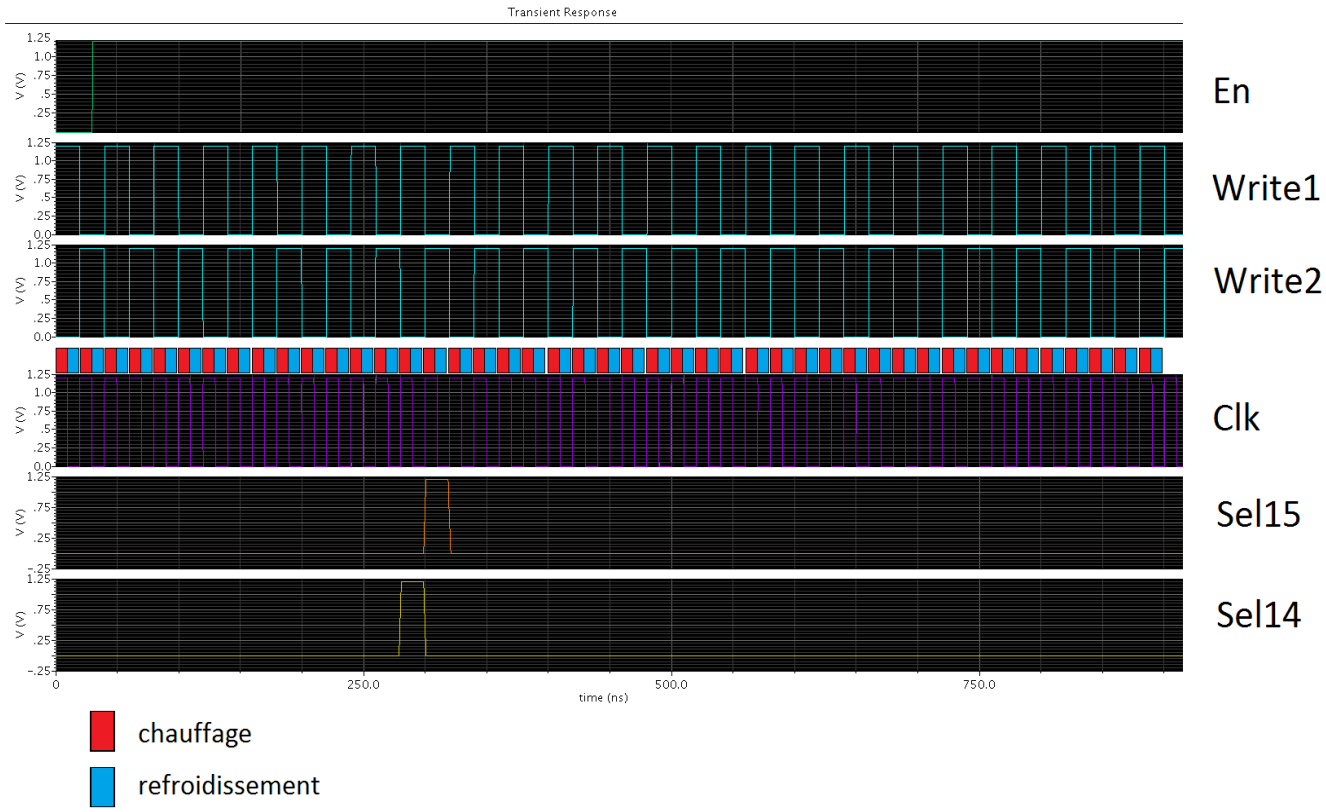


Figure 104 : signaux d'entrée pour écrire des données dans les JTMs

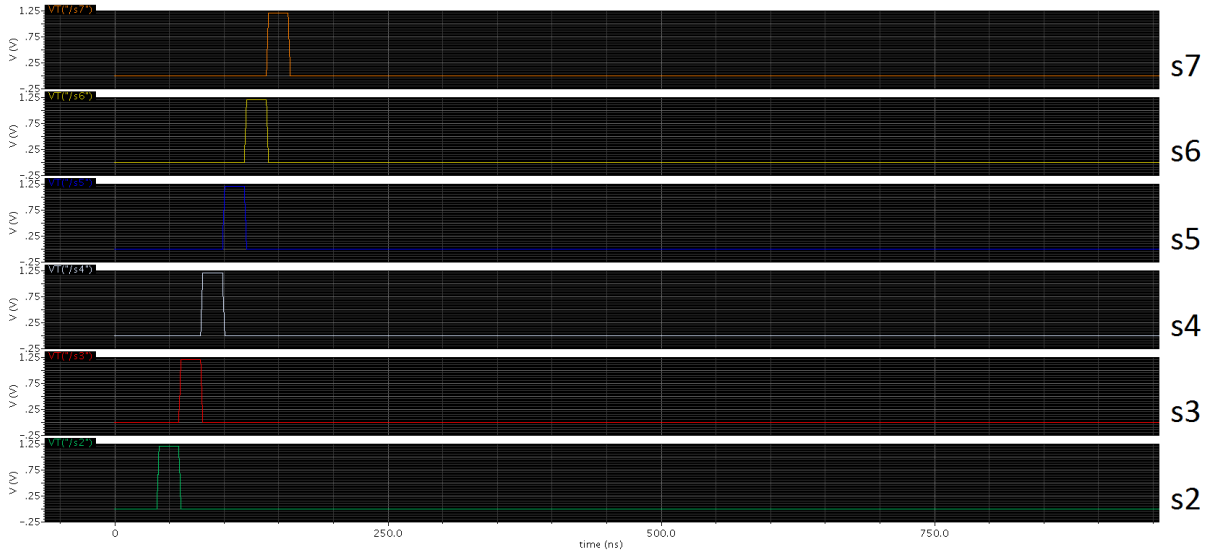


Figure 105 : signaux de sélection des JTMs à écrire

Pour la lecture, une porte NAND supplémentaire permet de mettre en forme le signal lu par le latch pendant la phase de rafraichissement et de maintenir le signal de sortie à « 0 » lorsque le bloc mémoire est inactif. Le schéma présenté n'implémente pas de transistor de limitation de courant. Si c'était le cas, la porte NAND ne serait pas indispensable.

La Figure 106 montre le résultat de simulation de Monte Carlo de la sortie de lecture. Elle montre la lecture de deux bits de configuration. Le premier est à « 0 » après une période de transition. Ensuite, un autre bit est sélectionné. Il est à « 1 » après une période de transition plus courte. Les temps d'écriture peuvent donc être évalués. L'écriture qui comprend les phases de chauffage et de refroidissement peut durer 30 ns avec 10 ns pour le chauffage et 20 ns pour le refroidissement. La vitesse de lecture est comparable à celle d'une mémoire SRAM. Dans notre cas, elle est de l'ordre de la nanoseconde. Quant à la consommation lors de la phase d'écriture, elle comprend la consommation pour chauffer les JTMs qui est de 3,6 pJ, et l'énergie pour générer le champ magnétique d'écriture de 360 pJ. L'énergie totale d'écriture d'un bit de configuration est donc de 363,6 pJ. Cette valeur n'a pas été optimisée dans la mesure où la configuration est écrite rarement et donc que la consommation à l'écriture n'est pas primordiale. Notons que les temps de lecture et d'écriture ne sont pas primordiaux. En effet, la lecture des JTMs qui intervient pendant la phase de rafraichissement, ne demande pas une grande rapidité car le rafraichissement est périodique et la durée d'un rafraichissement est longue. Donc un temps de lecture de l'ordre de la nanoseconde est largement suffisant. C'est pourquoi la rapidité de la lecture n'a pas été optimisée. Seule la consommation durant cette phase doit être optimisée de façon à ce qu'elle ne pénalise pas la consommation totale du circuit. Ensuite, l'écriture de la configuration d'un FPGA intervient très rarement dans la plupart des applications. La configuration est écrite une fois dans une mémoire non-volatile et ensuite elle peut éventuellement être actualisée. Le temps d'écriture des cellules mémoires MRAM n'est donc pas critique. Le seul domaine où cela peut être critique est pour les applications avec reconfiguration dynamique. Une telle reconfiguration est possible avec l'architecture décrite dans cette thèse, cependant le circuit présenté n'a pas été optimisé pour cette application car trop complexe à mettre en œuvre. Cela pourra faire l'objet d'une prochaine thèse. La consommation lors de la phase de lecture est de 0,84 pJ par cellule mémoire de configuration. Rappelons qu'une cellule mémoire de configuration est composée de deux JTMs dont les états sont

complémentaires. L'énergie consommée par les blocs de cellules MRAM (336 cellules mémoires de configuration) lors d'une phase de rafraichissement est donc de 282 pJ.

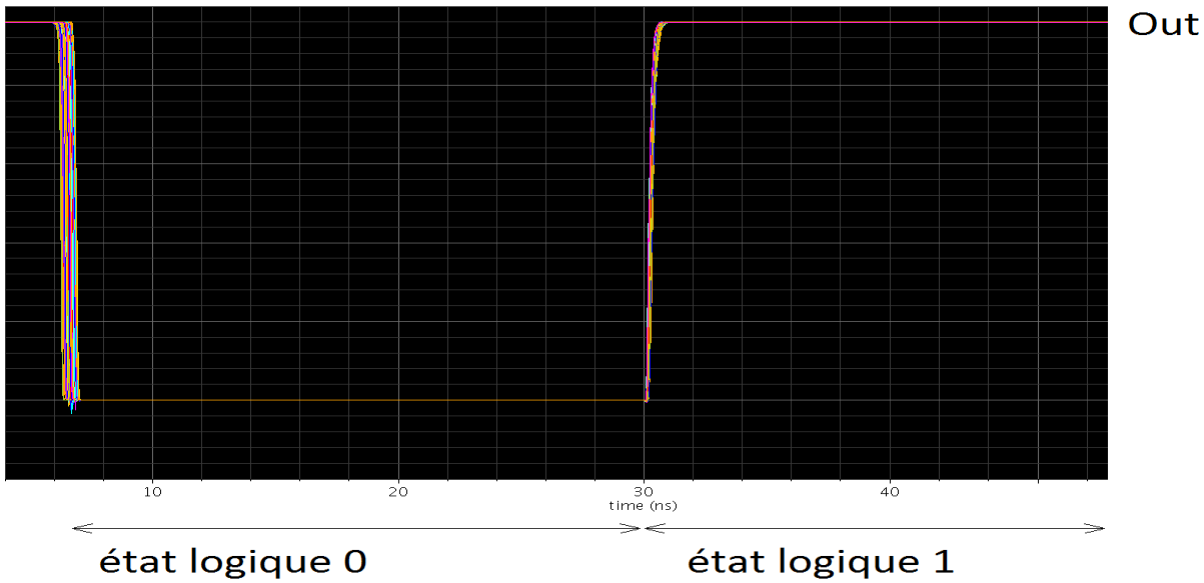


Figure 106 : résultat de simulation de Monte Carlo la lecture de mémoire MRAM

La conception de ces blocs a consisté à dimensionner les transistors d'écriture de façon à générer suffisamment de courant pour chauffer les JTMs et avoir un champ magnétique suffisamment intense pour écrire. Le dimensionnement des transistors de lecture a été réalisé de façon à lire de façon fiable et répétable le contenu des JTMs. Notons que la taille des transistors importait peu car le bloc était partagé entre de nombreuses JTMs donc au final leur taille est relativement faible. De plus, une taille élevée permet d'augmenter leur LET et donc d'accroitre leur fiabilité par rapport aux erreurs transitoires générées par les radiations. Les Figure 107 et Figure 108 montrent les layouts du circuit de lecture et d'écriture ainsi qu'un demi-générateur de courant. Dans le circuit de lecture, le plus critique d'un point de vue layout est le latch notamment à cause des problèmes de mismatch entre les deux transistors. Les mismatch sont minimisés en dessinant les deux transistors dans la même direction, avec bien sûr la même taille, et le plus rapproché possible. De plus, le circuit doit être symétrique, il est donc nécessaire d'ajouter un porte NAND sur les deux nœuds Q et Q_bar.

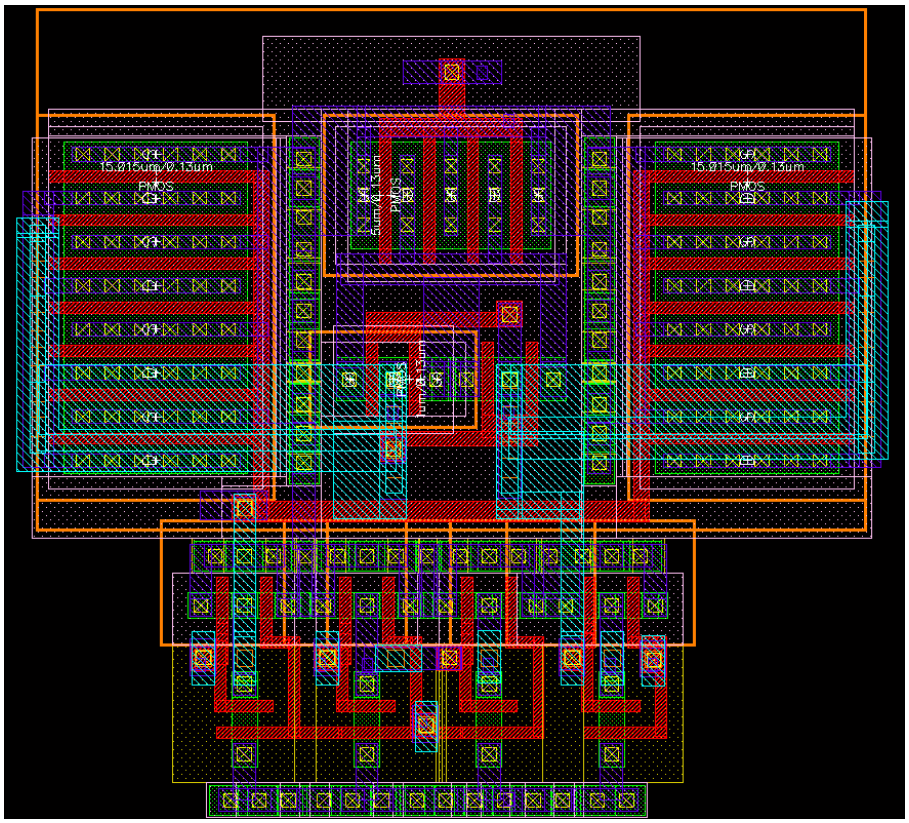


Figure 107 : layout d'un circuit de lecture et d'écriture d'un bloc mémoire MRAM

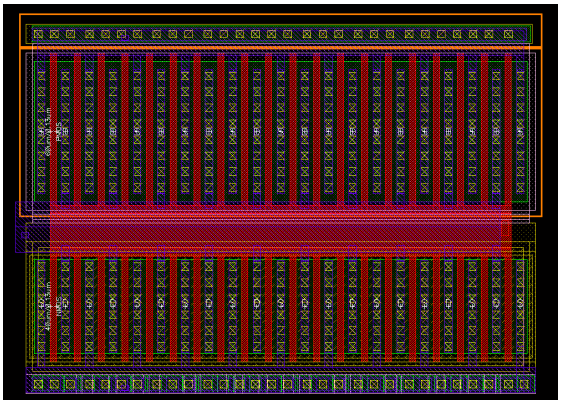


Figure 108 : layout du demi-générateur de courant

XVI.3.3 Interconnexions locales

Les interconnexions locales connectent la LUT au réseau de routage. Elles sont composées d'un pass-transistor NMOS dont la grille est connectée à une cellule mémoire de configuration. Dans notre cas, la mémoire de configuration est composée d'une cellule DRAM ou pseudo DRAM et de son transistor de sélection. Elle a déjà été décrite précédemment. Les interconnexions locales sont donc un réseau de connexion programmable comme le montre la Figure 109.

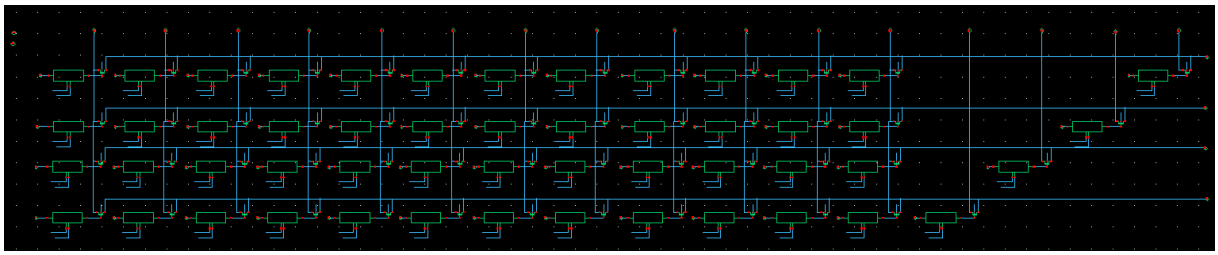


Figure 109 : schéma d'un bloc d'interconnexions locales

Avec cette structure, il y a 12 connexions vers le bloc de routage et 4 connexions pour se connecter à la sortie d'une LUT pour mettre plusieurs LUT en série par exemple. Chacune de ces connexions peut être reliée à chacune des entrées de la LUT. Cela permet un meilleur routage du circuit car l'algorithme de placement routage a plus de possibilités de connexion. Le choix du transistor NMOS comme pass-transistor est dû au fait qu'il est plus rapide que le PMOS. Le réglage de son W est lié a la rapidité souhaitée pour ces interconnexions. Dans notre cas, la rapidité souhaitée est la rapidité standard pour ce type de technologie sachant que la plus value de cette thèse est sur l'architecture de la partie configuration du circuit et non la partie « opérative » du FPGA. La Figure 110 montre le résultat de simulation d'un transistor d'interconnexion de la Figure 111. Le signal In est appliqué sur le drain du transistor et le signal Out est observé sur la source où est connectée une capacité simulant la capacité parasite d'une entrée de LUT. On voit qu'il y a la perte de V_{th} quand le signal propagé par le transistor d'interconnexion est « 1 ». On peut déterminer la consommation liée à ce transistor. Elle est de 1,6 nW/MHz pour la consommation dynamique et 0,96 µW à 130°C pour la consommation statique. La consommation maximale du bloc de connexion locale, c'est-à-dire lorsque toutes les entrées commutent est de 339 nW/MHz pour la consommation dynamique et 203 µW pour la consommation statique.

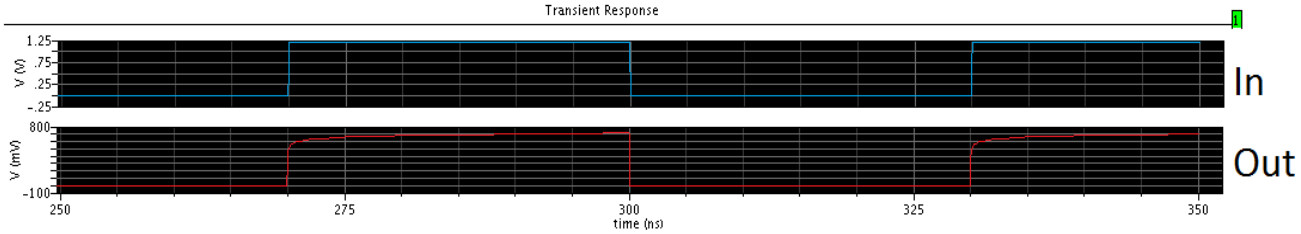


Figure 110 : simulation d'un transistor d'interconnexion

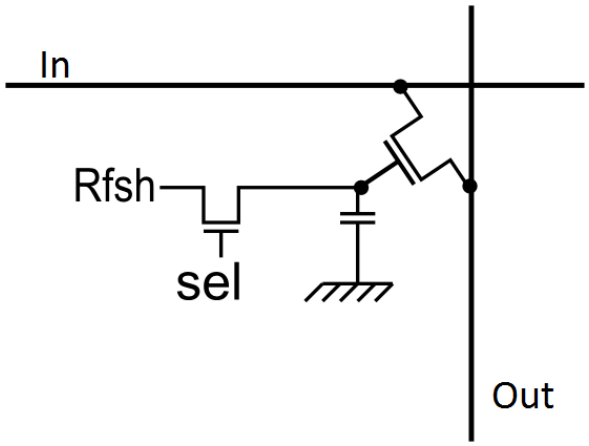


Figure 111 : schéma d'une interconnexion programmable

Comme mentionné précédemment, la tension présente sur la capacité pseudo DRAM est de $V_{dd} - V_{th}$ lorsque la cellule est configurée à « 1 » en raison du transistor de sélection. On peut remédier à ce problème en appliquant une tension Vsel supérieure à V_{dd} sur la grille du transistor d'interconnexion. Notons que cela n'affecte pas la fiabilité du transistor car cette tension est appliquée uniquement lorsqu'il est sélectionné lors du rafraichissement ce qui est peu fréquent mais pose des problèmes d'implémentation puisque deux tensions sont nécessaires. On peut donc implémenter un multiplexeur numérique pour régénérer le signal.

La Figure 112 montre le layout d'une interconnexion programmable. On peut voir la pseudo cellule mémoire DRAM décrite précédemment. Elle est reliée au pass-transistor qui connecte deux lignes. L'une étant une entrée de la LUT et l'autre provenant du bloc de routage. Ce circuit a une surface de 7,9 µm².

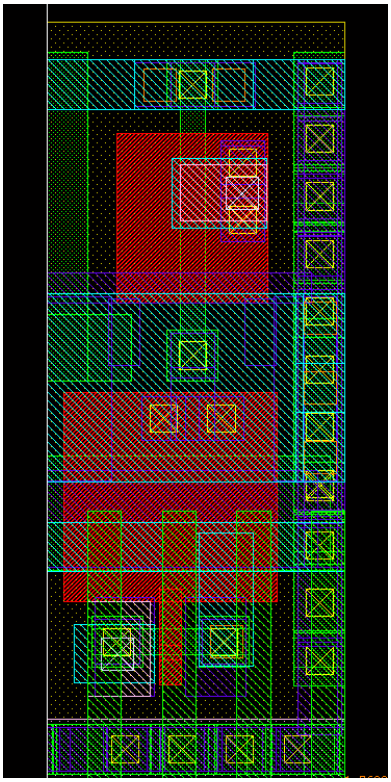


Figure 112 : interconnexion programmable

La Figure 113 montre le bloc d'interconnexions locales complet. C'est une matrice d'interconnexions programmables. Les entrées de sélection sont en métal 4 et disposées verticalement. Le signal de rafraichissement des capacités parcourt toutes les cellules et est en métal 2. Les connexions vers le bloc de routage sont verticales et les connexions vers la LUT sont horizontales. Cette structure permet d'avoir une bonne densité. La surface du bloc d'interconnexions locales est de 489 µm².

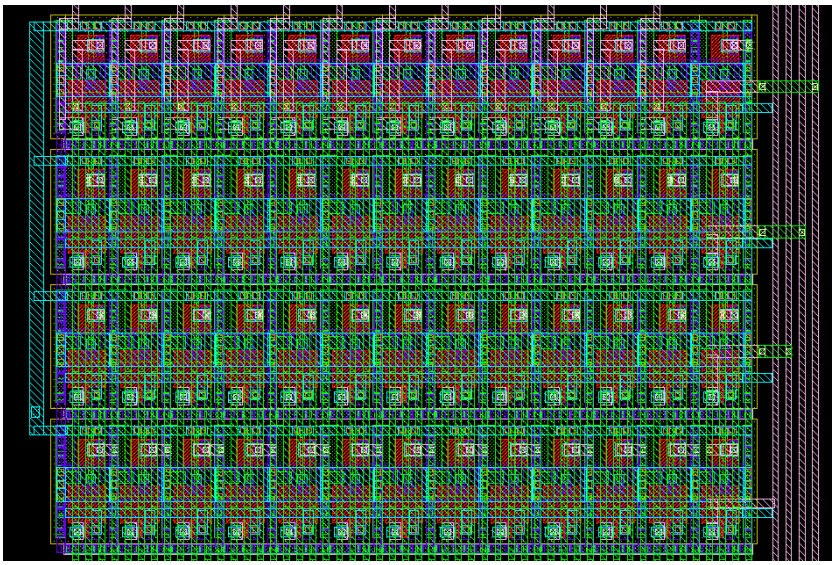


Figure 113 : layout d'un bloc d'interconnexions locales

XVI.3.4 Interconnexions de routage

Un des blocs qui occupe le plus de surface est le bloc d'interconnexion de routage. Il y a deux types d'interconnexions. Il y a d'abord les connexions programmables qui relient les interconnexions locales aux signaux horizontaux et verticaux de routage. Notons que des buffers sont placés pour mettre en forme le signal qui vient des lignes de routage. En effet, les signaux passant par les lignes de routage traversent des transistors NMOS. Lorsque le signal véhiculé est un « 1 », la tension baissera de V_{th}. Donc la tension baisse au fur et à mesure du passage par les transistors NMOS, c'est pourquoi des buffers sont placés pour rehausser le niveau de tension du signal. De plus, des buffers supplémentaires peuvent être placés pour véhiculer des signaux qui passent par un nombre important de bloc de routage. Ces buffers sont appelés répéteurs et sont programmables, c'est-à-dire qu'ils peuvent être utilisés ou pas selon les besoins.

Il y a également les interconnexions programmables de routage qui relient les fils verticaux et horizontaux entre eux. La Figure 114 montre un point de routage programmable. La ligne horizontale Ouest peut être reliée aux lignes verticales Nord et Sud ou bien horizontale Est. Ce point est composé de six transistors NMOS et toutes les configurations peuvent être programmées grâce aux mémoires de configuration reliées aux grilles des transistors : les quatre lignes reliées ensembles, une ligne horizontale connectée à une ligne verticale, etc.

La Figure 115 montre une simulation d'un point de routage. Il a été programmé pour que tous les transistors soient passants. Les mémoires de configuration des six transistors programmables sont donc à « 1 ». Notons que les cellules de configuration sont les mêmes que pour les interconnexions locales. On retrouve donc que la tension présente sur la capacité pseudo DRAM est de $V_{dd} - V_{th}$ lorsque la cellule est configurée à « 1 » en raison du transistor de sélection comme mentionné précédemment.

Ici, le signal vient du nœud Sud et est véhiculé vers les trois autres nœuds. On voit que le signal du nœud Sud est bien carré et les trois autres ont un régime transitoire dû aux transistors N passants. Cela implique que lorsqu'un signal est véhiculé à travers un nombre important de nœuds de routage, le signal doit être mis en forme par un buffer. Au niveau du FPGA, des répéteurs programmables doivent être insérés régulièrement pour pouvoir véhiculer les signaux sur plusieurs tuiles. Il prend la forme d'une chaine

de deux inverseurs et d'un transistor de connexion programmable. Ce type de circuit n'a pas été implémenté car le FPGA entier n'a pas pu être conçu. La consommation dynamique maximale d'une interconnexion programmable a été évaluée à 1,6 nW/MHz, c'est-à-dire que la simulation a été faite avec le nœud Ouest qui envoie les données vers les trois autres nœuds. La consommation dans le pire cas, c'est-à-dire lorsque tous les nœuds commutent, du bloc de routage entier est donc de 16 nW/MHz sachant qu'il y a 10 nœuds de routage.

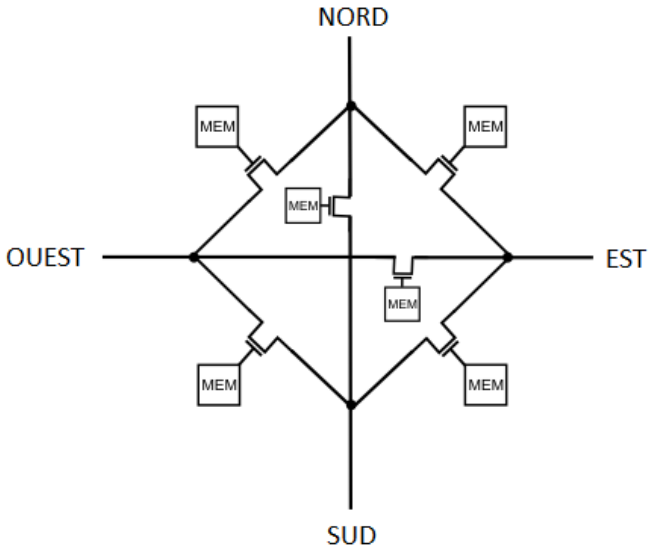


Figure 114 : schéma d'un point de routage programmable

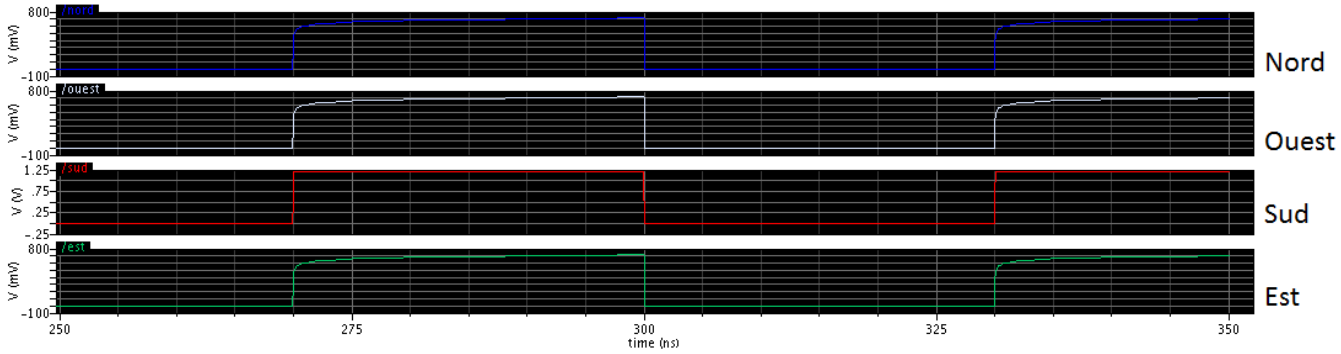


Figure 115 : simulation d'un point de routage

La Figure 116 montre le layout d'un point de routage programmable. Il est composé de mémoires de configuration « pseudo DRAMs » décrites au début de ce chapitre. Il forme une surface rectangulaire de 51,8 µm². Comme dans les autres circuits, les cellules ont été blindées pour ne pas créer de couplages capacitifs parasites avec les lignes de routage afin de ne pas perturber le fonctionnement des mémoires de configuration. La densité pourrait être augmentée en utilisant des transistors à très faible courants de fuite ou bien des cellules DRAM embarquées. On peut voir que les lignes de sélection sont verticales pour être compatibles avec les autres circuits car toutes les lignes de sélection de la tuile sont verticales et en métal 4 (en vert).

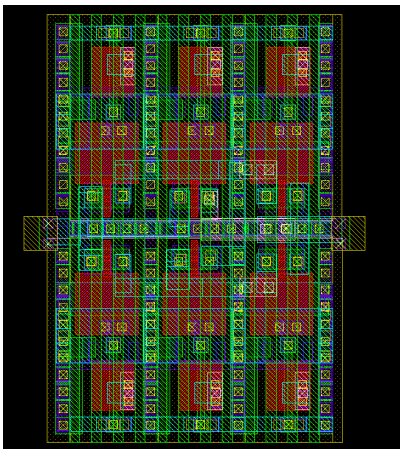


Figure 116 : layout d'un point de routage programmable

La Figure 117 montre le layout d'un bloc de routage. Il a été réalisé pour être le plus compact possible. Les surfaces vides vont être comblées par le layout des LUTs. Le layout a été fait autour des points de routage qui ont été compactés au maximum. Il a été organisé de façon à compacter les lignes horizontales et mettre les LUTs sous les lignes verticales. Les lignes de routage ont été réalisées en métal 5 (en jaune), le niveau de métal le plus élevé (à part les couches magnétiques), afin de limiter au maximum les couplages capacitifs parasites avec les interconnexions programmables. De plus, la couche de métal est plus épaisse que les autres ce qui diminue la résistance parasite de la ligne.

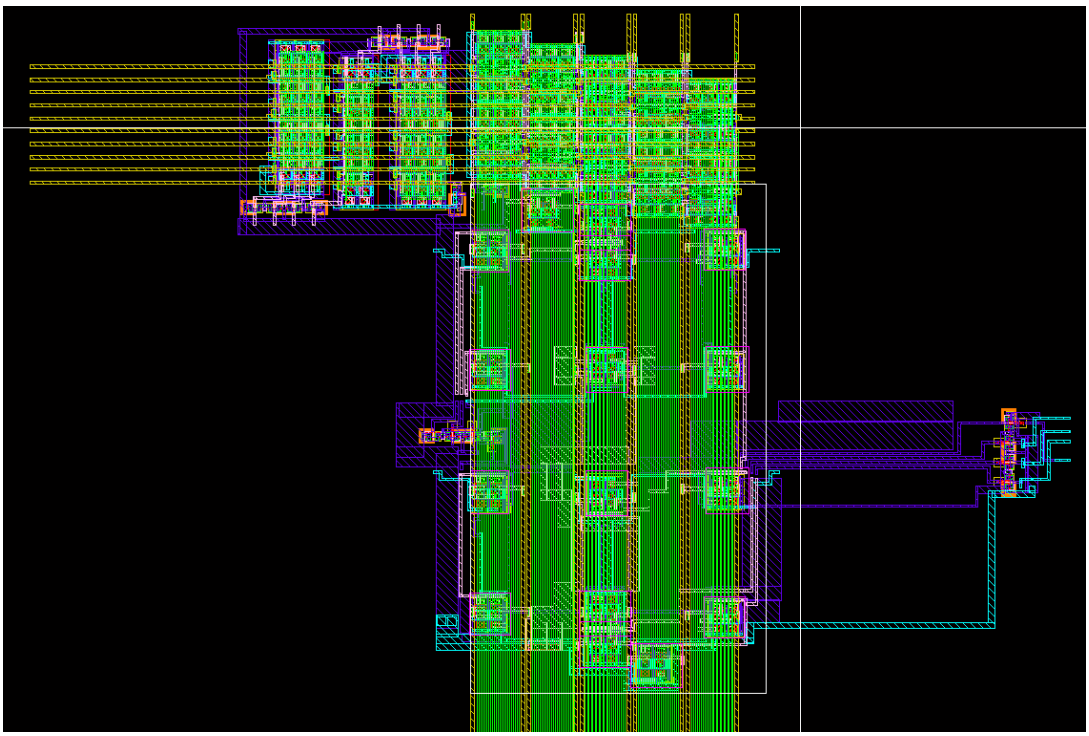


Figure 117 : layout d'un bloc de routage

XVI.3.5 Bascule durcie

La Figure 118 montre le schéma d'une Flipflop durcies aux radiations. Les étages Maître et Esclave sont réalisés à partir de cellules DICE, décrites dans le chapitre sur les

radiations, auxquelles ont été ajoutées des transistors pour les signaux set et reset. Ainsi, la bascule implémentée dans la tuile est résistante aux erreurs transitoires.

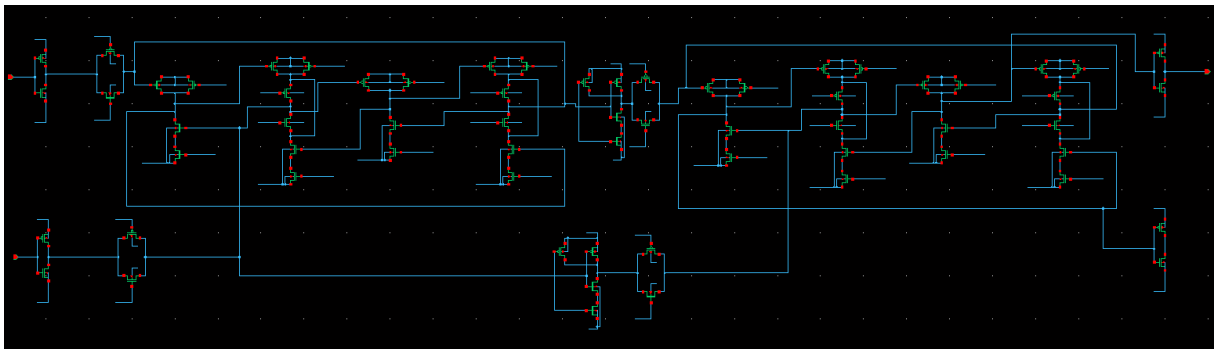


Figure 118 : schéma d'une Flipflop durcie aux radiations

Cette bascule est utilisée dans la LUT dans les cas où le circuit de l'utilisateur est un circuit séquentiel. Elle est également utilisée dans le circuit de control des phases de rafraichissement et de programmation. C'est une machine à état simple. Elle est composée de bascules définissant l'état de la machine et d'un compteur envoyant les adresses des cellules mémoires afin de sélectionner les cellules MRAM et/ou DRAM à écrire ou lire. Pour que les phases de programmation et surtout de rafraichissement se déroulent correctement et ne génèrent pas d'erreur, les bascules doivent être durcies car ce sont les éléments les plus sensibles. En effet, dans le cas contraire, des erreurs de programmation ou de rafraichissement pourraient apparaitre. Une cellule mémoire DRAM pourrait, par exemple, être sélectionnée à la place d'une autre si une particule génère une erreur dans le compteur d'adresse. Ou bien une DRAM pourrait être écrite avec la mauvaise donnée si une particule change les données de rafraichissement.

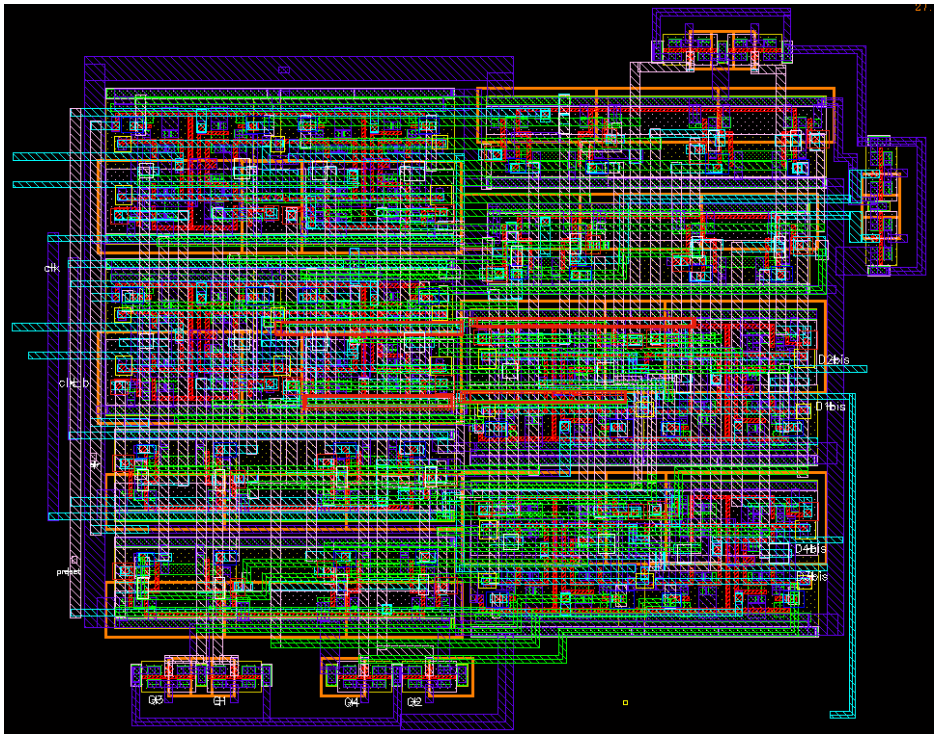


Figure 119 : layout de quatre bascules durcies aux radiations

La Figure 119 montre le layout de quatre bascules durcies. La contrainte la plus importante dans le layout d'une bascule durcie est l'espacement entre les nœuds sensibles du circuit. En effet, comme expliqué dans le chapitre sur les radiations, si une particule touche un des quatre nœuds sensibles d'une cellule mémoire DICE, la donnée peut être automatiquement corrigée grâce à la redondance présente dans les autres nœuds du circuit. Cependant si une particule impacte deux nœuds, la donnée est changée de façon irréversible ce qui provoque une erreur. Il faut donc séparer les nœuds sensibles de façon à ce qu'une particule n'impact pas deux ou plusieurs nœuds du circuit. Pour cela, les nœuds doivent être séparés d'environ 10 µm. Les nœuds des circuits ont donc été séparés de 10 µm mais cette espacement a provoqué des vides dans le layout. Afin de combler ces vides, plusieurs bascules ont donc été imbriquées pour respecter la contrainte d'espacement tout en comblant le vide entre les nœuds d'une même bascule. Cette méthode a augmenté la complexité du layout et a donc nécessité d'utiliser tous les niveaux de métaux disponibles (à part les niveaux de métaux de la couche magnétique). Les couplages capacitifs entre bascules étaient donc forts. De plus, cela augmente la consommation du circuit et diminue sa rapidité mais c'est le prix à payer pour augmenter la fiabilité du circuit. Une bascule a une surface de 187 µm². La surface des quatre bascules est relativement importante comparée à la surface des autres circuits. Elles représentent 11 % de la surface de la tuile.

La Figure 120 montre le schéma de simulation de la bascule. Pour ce test, plusieurs bascules ont été assemblées en un registre à décalage.

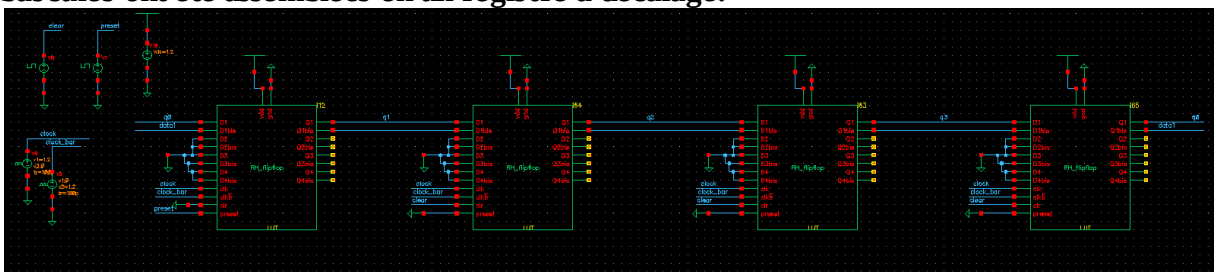


Figure 120 : schéma de simulation de plusieurs bascules en registre à décalage

Cette simulation de registre à décalage a permis de vérifier le bon fonctionnement des bascules. De plus, c'est également un registre à décalage qui permet de véhiculer la configuration du FPGA lors de sa programmation. On peut donc vérifier la programmation du FPGA. La consommation dynamique de la bascule est de 112 nW/MHz ce qui est une valeur élevée dû au fait qu'elle contient un nombre important de transistor en raison de la redondance et les capacités parasites sont élevées à cause des couplages entre les lignes de données avec les autres bascules. La consommation statique, de 52 µW, est également élevée pour les mêmes raisons.

XVI.3.6 Circuits de control des circuits de configuration

Décrivons maintenant les circuits de control des blocs mémoire MRAM et DRAM. Ils consistent principalement à générer les signaux de sélection des cellules MRAM et DRAM ainsi que les signaux de control de la lecture et l'écriture des MRAMs pendant les phases de rafraichissement et de programmation. La génération des signaux de sélection se fait grâce à un compteur qui génère un mot binaire de 8 bits et d'un décodeur qui convertit les mots de 8 bits et un mot de 256 bits où un seul bit est actif afin de sélectionner un seul bit. Il y a 8 bits car il y a 165 signaux de sélection des mémoires de configuration. Pour les sélectionner il faut donc un mot binaire de 8 bits. Avec 7 bits,

on ne peut générer que 128 signaux de sélection. Notons que les signaux de sélection des cellules MRAMs et DRAMs sont légèrement différents. Les cellules DRAMs doivent être sélectionnées une fois que les cellules MRAMs auront été lues. On a donc des signaux de sélection comme le montre la Figure 121. Dans cette figure, le vecteur S sélectionne les cellules DRAM et le vecteur SEL sélectionne les cellules MRAMs.

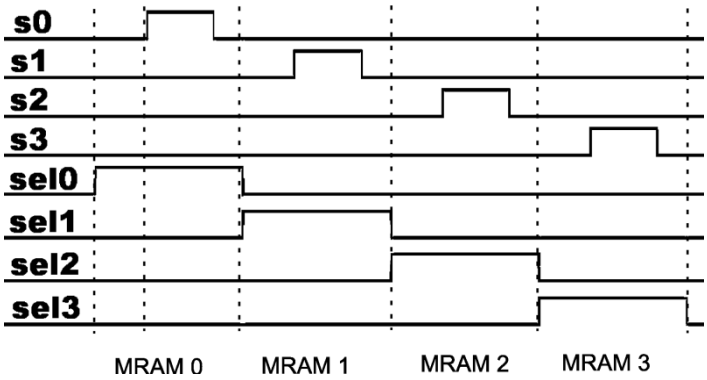


Figure 121 : signaux de sélection des MRAM 0 à 3 lors d'une phase de rafraichissement

Le décodeur génère donc les signaux de sélection des cellules MRAMs. Un circuit supplémentaire, constitué simplement d'une porte NAND pour chaque signal de sélection de cellule DRAM, permet d'activer la sélection d'une cellule DRAM lorsque la lecture de la cellule MRAM a été accomplie. Le compteur est composé des bascules durcies aux radiations décrites précédemment et d'additionneurs afin d'incrémenter le mot binaire à chaque front montant d'horloge. Pour finir, une machine à état simple permet de générer les signaux de control de la lecture (En_rd, On/off) et l'écriture (data, clock, Heat, En) des cellules MRAMs. Cette machine à états n'a pas été conçue au niveau transistor, mais seulement à partir de composants utilisés lors des simulations comme des générateurs de signaux. La programmation est initiée par des signaux globaux au FPGA qui permettent de mettre le FPGA en mode programmation ou fonctionnement. Cependant, ces signaux n'ont pas été implémentés car il n'a pas été possible de faire un FPGA entier.

La consommation a cependant été évaluée à partir des composants constituant ce circuit de contrôle (principalement les bascules et le décodeur) et dont la consommation est connue. La consommation dynamique théorique est donc de 2,5 pJ par signal de sélection, ce qui donne une consommation totale de 412 pJ pour une phase de rafraichissement. Etant donné qu'il y a 10 000 rafraichissements par seconde (fréquence de rafraichissement), la consommation supplémentaire moyenne due à cette phase, qui est alors considérée comme de la consommation statique car ne dépendant pas de la fréquence de fonctionnement, est de 4,12 µW. Concernant la surface occupée par ce bloc, on peut appliquer la même méthode d'évaluation. Le résultat est une surface de 7690 µm² ce qui est très grand comparée à la taille d'une tuile qui a approximativement la même surface, cependant ce circuit est partagé entre plusieurs tuiles. Afin que la surface de ce bloc ait peu d'impact sur la surface du FPGA, on peut partager ce bloc avec au moins 20 tuiles. Il représenterait alors 5% de la surface de la tuile.

XVI.3.7 Tuile complète

Les circuits présentés précédemment ont été connectés ensembles pour former une tuile. La Figure 122 montre le schéma d'une tuile où l'on peut voir les différents blocs la constituant. Il y a donc quatre LUTs avec leur bloc d'interconnexions locales respectif. Chaque LUT a une bascule que l'on peut utiliser pour faire des circuits séquentiels. Un multiplexeur permet de l'utiliser ou pas. Le bloc de routage est connecté aux quatre blocs d'interconnexions locales pour véhiculer les signaux à travers le FPGA. De plus, le bloc de routage et les interconnexions locales permettent de connecter les différentes tuiles du FPGA entre elles. Pour faire un FPGA entier, il faut connecter les tuiles entre elles en ajoutant le nombre de blocs de mémoires MRAM nécessaires et en reliant les entrées de sélection des mémoires de configuration des différentes tuiles. Il faut également ajouter le circuit de control des phases de programmation et de rafraichissement. Ce circuit de control est partagé par plusieurs tuiles de façon à minimiser son coût en surface mais n'a pas été conçu au niveau transistor, seulement à partir de générateur de signaux pour la simulation.

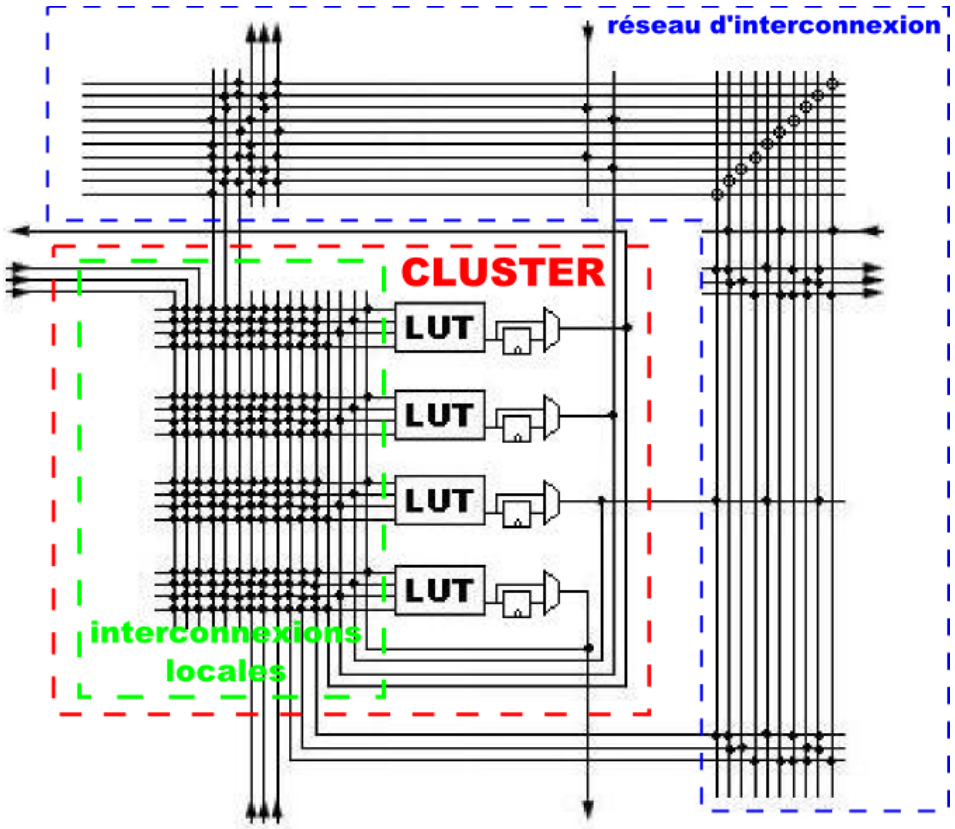


Figure 122 : schéma d'une tuile

La tuile n'a pas pu être simulée car le circuit était complexe et nécessitait donc des moyens de simulations plus performants que ceux qui étaient présents dans le laboratoire. Les résultats sur cette tuile, notamment les consommations et rapidités, présentés ultérieurement sont des estimations basées sur les résultats de simulation (post layout) des différents blocs pris séparément.

La Figure 123 montre le layout d'une tuile sans son bloc de mémoire MRAM et sans les lignes de sélection pour plus de clarté. Les lignes de routage sont en métal 5 (jaune), le métal au niveau le plus élevé et le plus épais pour limiter les couplages capacitifs avec les autres circuits, limiter l'électromigration et pour pouvoir véhiculer de forts courants. On reconnait les lignes verticales et horizontales. On voit également les bascules regroupées dans le bloc en haut à droite du layout. La difficulté dans le dessin

du layout a été de trouver la bonne structure qui permettrait de regrouper les différents blocs de façon à avoir un layout compact. L'autre contrainte était de faire en sorte que l'on puisse connecter directement plusieurs tuiles entre elles, c'est-à-dire sans faire de modification ou d'ajout, pour que les tuiles se collent parfaitement afin de constituer un FPGA entier.

Afin d'économiser en surface, il a fallu intégrer les LUTs dans le bloc de routage. L'autre difficulté, la plus contraignante, a été de placer les circuits de manière à faire coïncider les signaux de sélection des cellules mémoire de configuration (MRAM et DRAM). Les signaux de sélection ont été fait en métal 4 (en vert mais non représenté ici) et verticalement avec la largeur et l'espacement minimale pour qu'ils n'induisent pas de surface supplémentaire dans les circuits. Les lignes de sélection parcourent toute la hauteur de la tuile pour pouvoir se connecter directement aux tuiles Sud et nord.

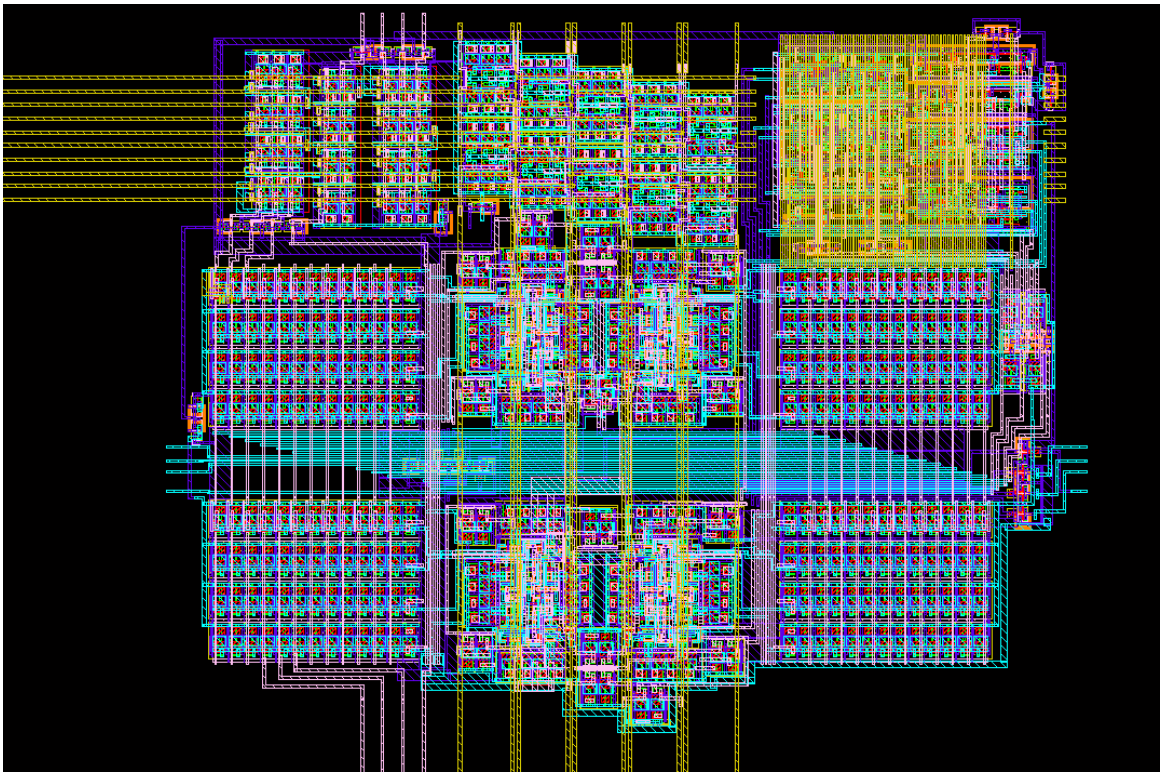


Figure 123 : layout d'une Tuile

La surface de la tuile sans le bloc mémoire MRAM est de 7200 µm². Le bloc mémoire MRAM se place à gauche de la tuile pour que les lignes de sélection des cellules MRAM et DRAM soient parallèles. Ainsi, les signaux de sélection des cellules MRAMs seront directement connectés à ceux des tuiles supérieures et inférieures. La présence du bloc MRAM induit une augmentation de la longueur des lignes de routage horizontale mais cela impacte peu les performances en terme de vitesse car, par ailleurs, l'utilisation de DRAM diminue la taille des interconnexions.

Ce bloc n'a pas pu être simulé complètement en raison de capacité de calcul insuffisante. Néanmoins le circuit de simulation a été implémenté. On retrouve la tuile à laquelle est connecté le décodeur utilisé pour générer les signaux de sélection. Des signaux de test sont appliqués sur les entrées des lignes de routage Est, Ouest, Nord et Sud ainsi que les interconnexions locales. La simulation consiste d'abord à générer les signaux pour la phase de rafraichissement qui configure les cellules mémoires de configuration DRAMs. Pour la simulation, la configuration consiste à laisser passant les tous les transistors de routage et certaines interconnexions locales pour tester

uniquement les interconnexions. Cependant, comme les simulations n'ont pas pu être effectuées avec ces tests simples, le test de la tuile n'a pas été plus poussé. On peut néanmoins estimer que la tuile complète fonctionne car les circuits individuels fonctionnent et ne fait intervenir que des circuits CMOS standards fait dans une technologie mature. On peut donc en déduire que la tuile entière fonctionne. Par ailleurs, le concept a été validé grâce au démonstrateur qui sera décrit dans un chapitre ultérieur.

XVI.4 Compilation des résultats

Ce chapitre décrit les différents circuits constituant la tuile conçue durant la thèse afin de valider le concept de circuit de configuration et de déterminer ses caractéristiques en termes de consommation et de densité par la simulation. Les résultats sont résumés dans les tableaux suivants (Tableau 1 et Tableau 2). Le Tableau 1 permet de comparer les résultats et tirer des conclusions quant à des améliorations possibles. Le Tableau 2 permet de comparer la consommation de la phase de rafraichissement à celle des autres blocs. Discutons donc des différentes caractéristiques en densité, consommation mais aussi rapidité et fiabilité.

XVI.4.1 Densité

Le Tableau 1 présente la surface détaillée d'une tuile. Quatre types de résultats sont présentés. La première colonne détaille la surface mesurée du layout de la tuile ainsi que ses différents blocs. Les cellules mémoires de configuration utilisées sont donc les celllules pseudos DRAM et MRAMs. Le total représente la surface totale du layout d'une tuile sans le bloc de mémoire MRAM. Il ne correspond pas à l'addition des surfaces des blocs constituant la tuile car il faut également prendre en compte les espacements supplémentaires dus à l'assemblage et au routage des blocs. La deuxième colonne montre les surfaces des différents blocs sans leurs cellules de configuration. Elle permet d'abords d'évaluer le poids des mémoires de configuration dans la surface totale de la tuile. Elle sert également à calculer les surfaces des différents blocs avec d'autres types de cellules mémoires de configuration. Elle permet donc de calculer les valeurs deux autres colonnes où l'on remplace les cellules pseudos DRAMs par de vraies DRAM embarquées (colonne 3) et par des cellules SRAM (colonne 4). On peut donc comparer la surface d'une tuile avec différentes technologies de mémoire mais avec la même structure de LUTs et d'interconnexions pour que la comparaison soit pertinente. On considère que la surface d'une cellule mémoire DRAM est de 0,53 µm² et de 2,5 µm² pour une SRAM, comme mentionné dans le manuel de la technologie CMOS 130nm de ST. Il y a 165 signaux de sélection et 336 cellules mémoires de configuration.

Tableau 1 : surface détaillée de la tuile

bloc	Surface mesurée avec pseudo DRAMs + MRAMs (µm²)*	surface sans cellules mémoire de configuration (µm²)	Surface estimée avec DRAMs + MRAMs (µm²)*	Surface estimée avec SRAMs (µm²)*
Cellule mémoire de configuration	6,10	0	0,53	2,5
LUT4	197	77,9	86,38	117,9
Cellule MRAM (deux JTMs)	6,75	-	-	-

Matrice JTMs	**2268**	**-**	**-**	**-**
Circuit lecture/écriture	**89,1**	**-**	**-**	**-**
Générateur de champ	**177**	**-**	**-**	**-**
Circuit de contrôle rafraichissement (partagé entre 20 tuiles)	**7690**	**-**	**-**	**-**
Point d'interconnexion	**7,87**	**1,77**	**2,3**	**4,27**
Interconnexions locales (68 points d'interconnexion)	**489**	**128**	**155,6**	**258**
point de routage	**51,8**	**9,6**	**12,78**	**24,6**
Bloc de routage (10 points de routage)	**518**	**96**	**127,8**	**246**
Point connexion locale - routage	**7,87**	**1,77**	**2,3**	**4,27**
Connexions locales - routage	**535**	**120**	**276**	**410**
Bascule durcie	**209**	**-**	**-**	**-**
Tuile (total) **	**7200**	**1875**	**2205**	**2995**

Pour mettre en évidence le gain en densité de cette architecture, on va donc comparer les densités des cellules DRAM avec les cellules SRAM car les circuits, tels les LUTs et le réseau d'interconnexions, doivent leur densité à l'utilisation de cellules DRAM comme mémoire de configuration « directe » (par comparaison avec les mémoires MRAMs qui sont séparées de ces circuits). Le bloc de mémoire MRAM induit une augmentation de la longueur des lignes de routage horizontales. Cependant, l'impact est limité car l'augmentation est de 24%. Les effets en termes de rapidité et de consommation dynamique sont donc faibles, en particulier pour les technologies matures où les capacités parasites des lignes ont un faible impact. Notons que dans le cas d'un FPGA SRAM, une mémoire externe durcie aux radiations est nécessaire. Lorsque la configuration est stockée dans une mémoire externe commune à plusieurs circuits comme un disque dur, il faut alors un circuit spécial qui fait la transition entre le disque dur et le FPGA. Il est donc difficile de chiffrer la surface supplémentaire due à la mémoire externe durcie. C'est pourquoi seules les surfaces des tuiles sans mémoire non-volatile (MRAM ou Flash) sont comparées.

Le gain en densité de l'architecture présentée dans cette thèse est lié à l'utilisation conjointe des cellules MRAM et des DRAM. Comme le montre la Figure 124, les densités des MRAMs et DRAMs sont parmi les plus élevées. Elles sont, en particulier, beaucoup plus élevées que la densité de la SRAM qui est actuellement la mémoire la plus utilisée dans les FPGAs. Ainsi, les interconnections programmables et les LUTs sont très denses car elles utilisent des DRAMs comme mémoires de configuration. On voit dans le Tableau 1 que ce sont les cellules DRAM qui permettent d'avoir la plus faible surface. Elle est 26 % plus faible que pour la SRAM. Notons qu'il n'y a pas d'avantage en surface pour une pseudo DRAM ce qui s'explique par le fait que les marges prises pour concevoir le circuit étaient élevées car l'un des buts était de faire un circuit pouvant être fabriqué. Notons également que les circuits hors cellules mémoires de configuration représentent 26% de la surface. Ils comprennent, par exemple, les transistors qui connectent les lignes entre elles, le multiplexeur des LUTs et les quatre bascules durcies. Les quatre bascules durcies et les espacements dus à l'assemblage des blocs et aux fils pour les relier occupent une place relativement importante. Les surfaces estimées des FPGAs DRAM et SRAM ne prennent pas en compte les espacements et connexions supplémentaires entre blocs mais elles permettent d'avoir un ordre de grandeur de

l'avantage de l'utilisation de DRAMs associées à des MRAMs que de SRAM associées à des mémoires Flash durcies.

On remarque également que le circuit de contrôle du rafraichissement et de la programmation a une surface de 7690 µm². C'est un circuit imposant mais il peut être partagé entre plusieurs tuiles, 20 dans notre cas. Il est partagé entre 20 tuiles afin de réduire son encombrement à 5% de la surface du FPGA.

Le bloc de mémoire MRAM qui comprend la matrice de JTM, le générateur de champ et le circuit de lecture/écriture a une taille de 2534 µm². La majorité de cette surface, 90%, est occupée par la matrice de JTMs. Les circuits de lecture/écriture et le générateur ont une part très faible car ils sont partagés entre toutes les cellules. On peut largement diminuer la taille de la matrice de JTMs simplement en ayant une seule JTM par bit de configuration au lieu des deux JTMs complémentaires actuelles. Le circuit de lecture changerait mais il occuperait également une faible surface comparée à ceux des autres blocs. On peut également prendre en compte que les améliorations du procédé de fabrication permettront d'augmenter la densité des JTMs afin de réduire leur surface.

La pseudo DRAM conçue n'a pas permis de montrer le gain en densité car les transistors utilisés étaient standards. En effet sa surface est de 6,1 µm² comparée à 2,5 µm² pour une cellule SRAM en technologie 130 nm [16]. En utilisant des transistors à faibles courant de fuite, la cellule aurait eu une surface beaucoup plus faible car le transistor de sélection aurait eu une taille plus faible pour avoir de faibles courants de fuite et la taille du transistor utilisé comme capacité aurait été plus faible pour la même durée de rétention de 100 µs à 130°C.

mémoire		DRAM (embarquée)	SRAM	Flash NOR	Flash NAND	FeRAM	STT-MRAM	TAS MRAM [15]	PCM
Taille (nm)		65	45	90	22	180	65	130	45
densité		(12 - 30)F^2	140F^2	10F^2	4F^2	22F^2	20F^2	(16-40)F^2	4F^2
Rapidité	lecture	2ns	0.2ns	15ns	0.1ms	40ns	35ns	3-20ns	12ns
	écriture	2ns	0.2ns	1µs/10 ms	1/0.1 ms	65ns	35ns	3-20ns	100ns
Rétention		4ms		10 ans	10 ans	10 ans	>10ans	10ans	>10ans
Endurance (cycle)		>10^{16}	>10^{16}	>10^5	>10^4	10^{14}	>10^{12}	>10^{15}	>109
Consommation à l'écriture (J/bit)		4 fJ/bit	0.5 fJ/bit	100 pJ/bit	0.2 fJ/bit	30 fJ/bit	2.5 pJ/bit	300pJ/bit *	6 pJ/bit
Non-volatile		non	non	oui	oui	oui	oui	oui	oui

Figure 124 : tableau des caractéristiques des principales mémoires [20]

*** La consommation de la TAS est principalement due à la génération du champ magnétique. On peut cependant partager la ligne entre plusieurs cellules pour faire diminuer cette valeur. On se rapprocherait de la consommation (environ 20pJ/bit) due au chauffage de la cellule**

Les quatre LUTs occupent une surface de 788 µm² et représente donc 11% de la surface totale de la tuile. C'est relativement faible comparée à la surface des interconnexions qui est de 5298 µm² soit 73% de la surface totale de la tuile. C'est en accord avec ce qui est observé dans les FPGAs existants dont les interconnexions représentent entre 50 et 80 % de la surface totale de la tuile [18]. Ceci montre que les inverseurs ajoutés aux mémoires de configuration des LUTs ont un surcout en surface assez faible. On peut donc considérer que les mémoires de configuration sont constituées

d'une capacité DRAM, d'une cellule MRAM et de leur transistor de sélection respectif, négligeant donc la part de ces inverseurs.

XVI.4.2 Consommation

La consommation du circuit peut être décomposée en deux parties : la consommation statique et dynamique. La consommation statique est la consommation du circuit due aux courants de fuite des transistors, c'est-à-dire lorsque le circuit est au repos. La consommation dynamique est liée à l'énergie nécessaire pour faire commuter les transistors. Le Tableau 2 montre la consommation détaillée, statique et dynamique, des différents blocs.

Tableau 2 : consommation détaillée de la tuile

bloc	Consommation statique (en µW à 130°C)	Consommation dynamique (nW/MHz)
Pseudo DRAM (en fJ/cell)	0,96	100 fJ/cell
Entrée de sélection (en fJ/cell)	-	6,64 fJ/cell
Phase de rafraichissement	3,03	-
Matrice JTMs	-	0,84 pJ/cell
Circuit de contrôle rafraichissement (décodeur + bascule+logique)	5,2	2,5 pJ/cell
LUT4	62	23,0
transistor d'interconnexion de routage	0,96	1.6
transistor d'interconnexion locale	0,96	1,6
Interconnexions de routage (60 transistors d'interconnexion de routage)	57,6	96
Interconnexions locale (212 transistors d'interconnexion)	203	339
Bascule durcie	52	112
Tuile (total) **	529	977 (n'a pas pu être mesurée – valeur estimée)

La tuile entière n'a pas pu être simulée donc il n'a pas été possible d'évaluer précisément le gain en consommation statique. Il a été évalué grâce à la consommation statique de chaque bloc individuellement. La valeur trouvée est de 529 µW à 130°C. C'est une valeur relativement élevée pour un circuit de cette taille. Ceci est dû au fait que le paramètre de température du circuit en simulation est très élevée (130°C) et les courants de fuite augmentent de façon exponentielle avec la température. Il n'a pas été possible de comparer cette consommation avec celle d'un FPGA commercial en raison de la complexité de l'architecture. On peut cependant penser qu'elle serait plus faible que pour un FPGA SRAM car il est possible d'appliquer des techniques basses consommations comme le « power gating » pour faire diminuer la consommation. En effet, la configuration ne sera pas effacée lorsque le FPGA sera mis hors tension contrairement à un FPGA SRAM où la configuration doit être rechargée. Le « power gating » serait donc plus efficace avec ce FPGA MRAM qu'avec un FPGA SRAM. De

plus, les courants de fuite de la cellule SRAM sont plus élevés que ceux de la cellule DRAM car son transistor de sélection a de très faibles courants de fuite. Le gain en consommation statique est amélioré grâce à l'utilisation de mémoires MRAMs qui sont non-volatiles afin de mettre hors tension les tuiles inutilisées.

A cette consommation statique, il faut ajouter la consommation due à la phase de rafraichissement. Elle a été estimée grâce à la consommation individuelle des différents circuits entrant en jeu : les cellules DRAM (pseudo DRAM dans notre cas), lecture des cellules MRAM et le circuit de control (compteur, décodeur, transistors de sélection, machine à états). La consommation totale de ces circuits pour un rafraichissement entier de la tuile est de 303 pJ. Etant donné qu'il y a une période de rafraichissement de 10 kHz, la consommation en une seconde est de 3,03 µJ. La consommation supplémentaire due à la phase de rafraichissement est donc de 3,03 µW dans les conditions maximales de température du spatial c'est-à-dire à 130°C. Dans notre cas, la consommation des DRAMs est faible car contrairement à une mémoire DRAM du commerce, où la donnée est lue et instantanément réécrite par l'amplificateur, on écrit directement la donnée lue du bloc mémoire MRAM. On compense donc les courants de fuite de la DRAM. Ensuite, la consommation du circuit de contrôle peut être négligée car il est partagé par plusieurs tuiles. Les principales sources de consommation sont donc la lecture des cellules MRAMs et les circuits de sélection des cellules DRAM et MRAM. La consommation de la phase de rafraichissement est donc une partie de la consommation statique. Elle est incluse dans la consommation statique car elle ne dépend pas de la fréquence de fonctionnement du circuit. Cette consommation n'est pas présente dans les FPGAs SRAM cependant elle peut être négligée ici dans les cas où la consommation dynamique du circuit utilisateur est grande, c'est-à-dire dans les applications hautes performances. Ceci est détaillé dans le paragraphe suivant. A 130°C, on voit que la part de la consommation statique due à la phase de rafraichissement est faible comparée à celle des autres blocs. Cependant, à température ambiante, cette part pourrait augmenter en raison du fait que la consommation statique des autres circuits baissera grâce à la baisse de température.

La présence de MRAM ne permet pas de faire diminuer la consommation dynamique du circuit directement. En effet, elle est due notamment à la présence des transistors et des capacités parasites des pistes de métal dans les circuits utilisateur (c'est-à-dire les circuits participant aux calculs donc ne comprenant pas le circuit de configuration). Ces circuits ont une architecture classique donc il n'y a pas de gain en consommation dynamique grâce à ceux-ci. Cependant, l'augmentation de la densité des mémoires de configuration permet de diminuer la taille des interconnexions. Leur capacité parasite est donc diminuée et donc la consommation dynamique due aux interconnexions également. La MRAM et la DRAM permettent donc de diminuer la consommation dynamique du FPGA mais de façon indirecte. La consommation dynamique totale a été évaluée grâce à la consommation des différents blocs individuellement. La consommation dynamique de la tuile est de 977 nW/MHz au maximum. Elle a été évaluée en programmant tous les transistors d'interconnexion à l'état passant et en appliquant une alternance de 1 et de 0 sur les signaux de routage. Dans la tuile, la consommation due à la phase de rafraichissement peut être négligée dans les cas où le circuit utilisateur fonctionne à une fréquence supérieure à 100 MHz. Dans ce cas, la consommation supplémentaire due au rafraichissement représente seulement 3 % de la consommation sans compter la consommation statique qui dépend de la température. Il est possible d'améliorer ce rapport, c'est-à-dire diminuer la part du rafraichissement, en optimisant la consommation du circuit de lecture de la cellule MRAM qui représente 90% de la consommation de la phase de rafraichissement. On peut également diminuer la fréquence de rafraichissement en augmentant le temps de

rétention des données comme par exemple en utilisant de vraies cellules mémoires DRAM embarquées qui ont des temps de rétention de plusieurs millisecondes. Grâce à ces considérations, on peut donc imaginer utiliser ce type de FPGA pour des applications plus communes qui requièrent des fréquences élevées, plus de 100 MHz.

XVI.4.3 Tolérance aux radiations

Dans l'architecture présentée, la tolérance aux erreurs transitoires générées par les radiations ionisantes est due à l'utilisation de mémoire MRAM. Les JTMs sont intrinsèquement immunes aux radiations. Elles ont de plus, une grande tolérance aux doses de radiation [17]. On peut donc les utiliser comme mémoires de référence pour la configuration du FPGA. Cela permet de compenser la sensibilité des DRAMs aux radiations. En effet, les DRAMs sont plus sensibles aux radiations que les SRAMs. Une SRAM est composée de deux inverseurs qui forment une boucle de rétroaction. Elle est donc très stable, ce qui n'est pas le cas des DRAMs. Cependant, les sensibilités des SRAMs et des DRAMs ont tendances à se rapprocher avec la miniaturisation des transistors. D'une part, parce que la cellule DRAM a une très faible surface, en particulier grâce au procédé « deep trench », et d'autre part, parce que les cellules SRAM auront une sensibilité suffisamment élevée pour que toute particule, quelque soit son énergie, provoque une erreur, c'est la saturation. Les cellules DRAM et SRAM tendent donc vers la même sensibilité [18]. La sensibilité du FPGA n'a pas pu être mesurée ou simulée mais on peut supposer que les circuits implémentés sur le FPGA auront une fiabilité élevée grâce au mécanisme de scrubbing décrit précédemment. Le fait que la mémoire MRAM soit locale et placée près des circuits de configuration permet d'avoir une fréquence de rafraichissement élevée. Cela permet donc de corriger rapidement les erreurs qui auraient pu se produire.

XVI.4.4 Rapidité

La rapidité d'un FPGA est liée à l'architecture des circuits qui commutent durant son fonctionnement normal. Ces circuits sont les transistors d'interconnexions, les buffers et les multiplexeurs présents dans les LUTs. Par exemple, la présence de nombreuses interconnexions programmables sur un fil va augmenter sa capacité parasite et donc diminuer sa rapidité comme expliqué dans le chapitre sur l'état de l'art. De même, la structure du multiplexeur influence ses performances. Leur architecture n'a pas été l'objet de cette thèse, seule la partie configuration a fait l'objet de recherche. Ces circuits ont donc des structures classiques. L'utilisation des MRAMs et DRAMs ne concernant que la partie configuration, elles n'ont donc pas un impact direct sur la rapidité du FPGA. Cependant, le gain en densité expliqué précédemment permet principalement de diminuer la taille des interconnexions et donc diminuer leur capacité parasite. On peut donc augmenter la rapidité du FPGA de façon indirect, de la même façon que la consommation dynamique est diminuée. Cette augmentation de rapidité est faible pour les technologies matures car leur rapidité est d'abord liée aux capacités parasites et résistances à l'état passant des transistors. C'est dans les technologies avancées que le gain est significatif dans lesquelles les capacités parasites des interconnexions sont élevées.

XVI.5 Conclusion

L'architecture de circuit de configuration pour FPGA présentée dans cette thèse allie les avantages des MRAMs et des DRAMs. Les densités élevées de ces deux mémoires permettent d'avoir des LUTs et des blocs d'interconnexions présentant une faible surface. Elle est 26 % plus faible que pour un FPGA SRAM ayant la même structure de LUTs et d'interconnexions. Cette réduction de la surface induit une diminution de la taille des interconnexions ce qui réduit donc leur capacité parasite. La rapidité des circuits et leur consommation dynamique sont donc réduites. L'utilisation de cellules mémoires MRAM qui sont non-volatiles, permet d'appliquer la technique du power gating pour diminuer la consommation du circuit implémenté. De plus, les tuiles inutilisées du FPGA peuvent être mises hors tension pour diminuer leur consommation statique. La consommation statique de cette architecture de FPGA est donc faible. Ensuite, grâce au rafraichissement des DRAMs, les éventuelles erreurs induites par les radiations ionisantes sont corrigées rapidement, toutes les 100 µs contre toutes les centaines de millisecondes pour les FPGAs SRAM, grâce à l'utilisation de MRAM qui sont résistantes par nature aux radiations (la fréquence de rafraichissement peut être plus courte). Ensuite, les MRAMs étant compatibles avec le process CMOS, il est possible de faire des FPGAs avec des mémoires MRAM embarquées. Il n'est donc pas nécessaire d'utiliser une mémoire Flash externe pour stocker la configuration du FPGA comme c'est le cas pour les FPGAs SRAM. Il y a donc un gain de place sur le circuit imprimé du système grâce à un boitier en moins.

Le principal inconvénient de cette structure est l'utilisation de deux process supplémentaires, DRAM et MRAM, ce qui ajoute un nombre de masques important. Ceci peut être partiellement résolu par l'utilisation des capacités parasites des transistors pour faire une capacité DRAM mais avec une perte en densité. Cela n'a pas pu être mis en évidence dans les simulations et le démonstrateur mais l'utilisation de transistors très basse consommation peut résoudre le problème. Ensuite, la consommation supplémentaire induite par la phase de rafraichissement est nécessaire pour que les circuits implémentés sur le FPGA soient fiables. On peut cependant négliger cette consommation supplémentaire dans les cas où le circuit implémenté fonctionne à haute fréquence, au-delà de 100 MHz, car la consommation de cette phase peut être négligée devant celle du circuit utilisateur comprenant les LUTs et les interconnexions.

XVI.6 REFERENCES

[15] DI PENDINA Gregory. Conception innovante et Développement d'outils de conception d'ASIC pour Technologie Hybride CMOS / Magnétique. Thèse de doctorat en nanoélectronique et nanotechnologies. Université de Grenoble, 2012, 191p.

[16] Design rules manual technologie ST hcmos9gp. STMicroelectronics

[17] CONRAUX Yann. Préparation et caractérisation d'un alliage amorphe ferrimagnétique de GdCo entrant dans la conception de jonctions tunnel magnétiques, résistance des jonctions tunnel magnétiques aux rayonnements ionisants. Thèse de doctorat en physique. Université de Grenoble, 2005, 159p.

[18] Shubu Mukherjee. Architecture Design For Soft Errors. *Elsevier*, February 2008

[19] I. Kuon and J. Rose. Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203–215, 2007.

[20] INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS2011 EDITION EMERGING RESEARCH DEVICES, table ERD3, 2011

IV. DEMONSTRATEUR

XVII. TEST DU DEMONSTRATEUR

Le démonstrateur a été conçu avec la technologie 130 nm de TowerJazz semiconductor pour la couche CMOS et Crocus-Technology pour la couche magnétique. Notre équipe a eu l'opportunité de placer des petits circuits dans les « scribelines » d'un run de Crocus. Afin d'avoir accès au KIT de conception de TowerJazz, nous avons dû travailler sur les stations de travail de Crocus-Technology, en partageant les licences logicielles. Fort heureusement, l'équipe de conception de Crocus étant basée aux Etats-Unis, le décalage horaire nous a permis de travailler en parallèle. Cependant, le nombre de licences étant limité, le temps disponible pour la conception a été très réduit. Dans le cadre de cette thèse, nous avons choisi d'implémenter une simple LUT à deux entrées sur le principe décrit précédemment. La LUT était entourée des mémoires MRAM de Crocus avec un autre circuit démonstrateur (bascule non-volatile innovante) du laboratoire Spintec conçu par Gregory Di Pendina. Nous n'avions ni le temps ni la place de faire un circuit plus complexe. La phase de conception s'est étendue sur un mois et demi en alternance avec Gregory. Elle a compris la phase conception du schéma durant laquelle les transistors ont été dimensionnés comme décrit précédemment. L'optimisation par rapport aux variations de procédé s'est faite par des analyses paramétriques car les fichiers de corners n'étaient pas disponibles dans le kit de conception tower pour une simulation de MonteCarlo. Ensuite, a eu lieu la phase de conception du layout (Figure 126). La taille du circuit n'a pas été optimisée afin de réduire les risques de défaut de procédé. La couche CMOS est composée de seulement 3 lignes de métal en raison du fait que Crocus fabrique des mémoires ce qui nécessite donc peu de couches de métal. Les couches magnétiques sont situées au dessus du métal 3. Cependant le faible nombre de couches de métal n'a pas été handicapant car le circuit était simple. Il n'a été possible de n'utiliser que 2 classes de transistors : 1,2 et 3,3 V standards. Les transistors à faible et à forte tension de seuil n'étaient pas disponibles, seuls les transistors standards l'étaient. Il n'a donc pas été possible de combiner transistor rapide et très faible courant de fuite. Il aurait été préférable, en effet, d'utiliser pour les transistors de sélection des capacités et ON/OFF des transistors à faible courant de fuite. Dans le cas des transistors de sélection des JTMs et du générateur de courant, des transistors rapides pour leur faible résistance à l'état ON auraient été nécessaires. Le circuit est donc largement perfectible. Durant la phase de layout, les DRC et LVS ont permis de vérifier que le circuit était bien dessiné. Notons que pour le LVS, il fallait remplacer les JTMs du schéma par des transistors bipolaires (où la base correspondait à la ligne de champ) pour que la partie magnétique soit également vérifiée car le LVS des composants magnétiques n'était pas pris en compte dans le kit de conception..

XVII.1 Description du démonstrateur

Le circuit testé est une LUT à deux entrées (Figure 125) qui fonctionne sur le principe de LUT décrit précédemment. Il y a donc quatre cellules mémoires de configuration. Chacune est constituée de 2 JTMs dont les données sont complémentaires. Le vecteur S (s0, s1, s2, s3) permet de sélectionner la capacité à rafraichir et le vecteur SEL (sel0, sel1, sel2, sel3) permet de sélectionner les JTM à lire

ou écrire. Le signal HEAT active le chauffage des JTMs à écrire. ON/OFF permet de mettre sous tension le bloc MRAM durant les phases de rafraichissement ou d'écriture. Az est le signal qui permet d'initialiser la lecture des JTMs comme expliqué précédemment. EN est le signal qui permet de fixer Rfsh à '1' lorsque le bloc mémoire est OFF pour ne pas avoir un signal indéfini. Le vecteur E (e0, e1) est le signal d'entrée de la LUT et OUT est sa sortie.

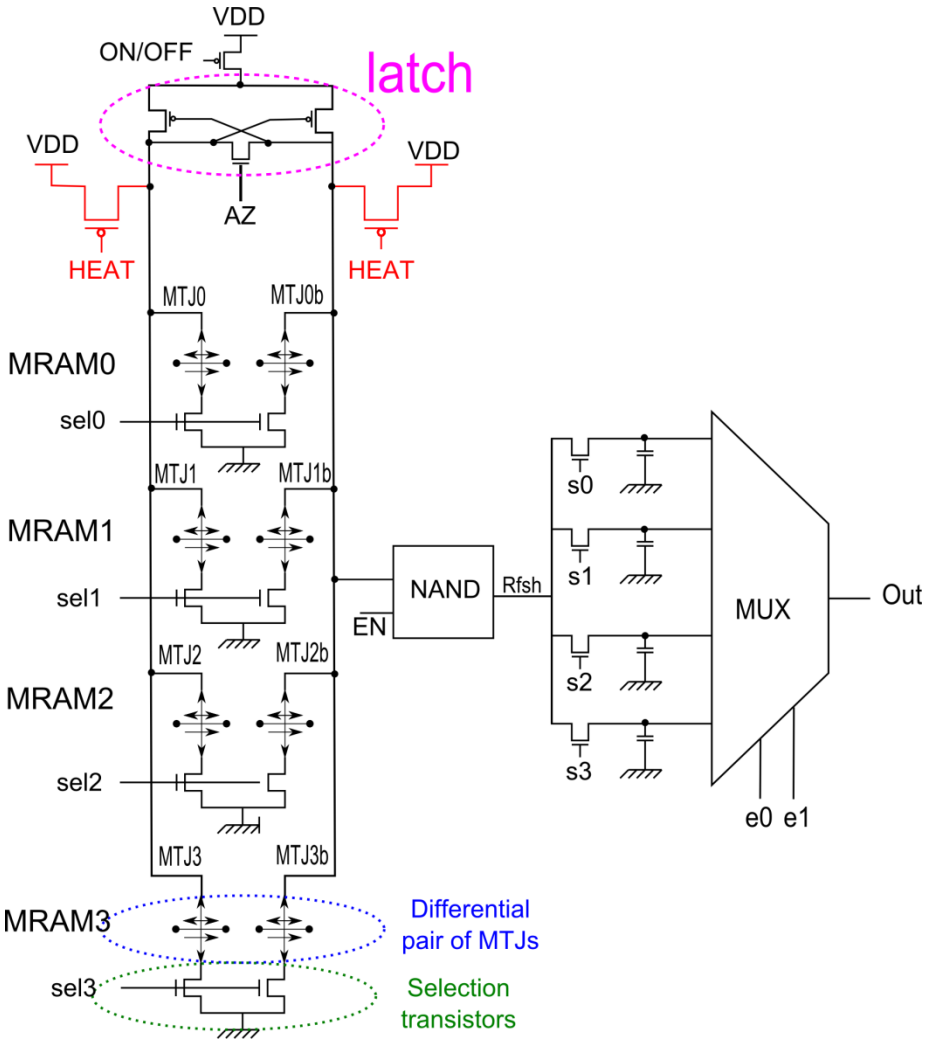


Figure 125 : schéma du circuit intégré dans le démonstrateur

La conception du circuit a consisté à déterminer les dimensions des transistors de façon à pouvoir écrire et lire les JTMs, rafraichir les capacités et faire commuter le multiplexeur. Tout d'abord, les transistors de sélection et de chauffage ont été dimensionnés de façon à chauffer suffisamment les JTMs. Lors de la phase de chauffage, le latch est inutilisé. Une JTM est en série avec le transistor de sélection et de chauffage (HEAT). Pour leur dimensionnement, on peut faire deux remarques : premièrement le transistor de chauffage est partagé entre plusieurs JTMs, il a donc un W élevé de façon à ce que sa résistance à l'état ON soit faible. Deuxièmement, les transistors de sélection sont les plus nombreux donc il faut minimiser leur taille. Dans la pratique, des marges assez élevées ont été prises pour être sûr de chauffer suffisamment les JTMs. De plus, le circuit a été conçu pour fonctionner dans la plage de température des composants destinés au spatial c'est-à-dire de -55°C à 125°C. Donc pour chauffer suffisamment les JTMs, en partant de -55°C, des transistors relativement grands ont été implémentés. Ainsi les transistors de chauffage ont un W de 25 µm et les transistors de sélection un W

de 4 µm. Les autres transistors à dimensionner sont ceux du générateur de courant. Rappelons qu'il génère le champ magnétique lors de l'écriture des JTMs. Il a été dimensionné de manière à générer un courant minimum de 20 mA. Les transistors ont donc un W de 100 µm pour les transistors PMOS et 45 µm pour les transistors NMOS. Ensuite le dimensionnement du latch a été fait de façon à lire de manière fiable le contenu des JTMs en mode différentiel. Grâce à une analyse paramétrique (les modèles pour Monte Carlo n'existaient pas pour les JTMs) le W optimal a été de 8 µm. Concernant le transistor ON/OFF, il doit couper l'alimentation en mode OFF et doit avoir une résistance très faible lorsqu'il est en mode ON pour ne pas ralentir le circuit de lecture. Le W est donc de 7µm. Les capacités et leur transistor de sélection ont été dimensionnés pour avoir un temps de maintien de 10 µs à 125°C. Le transistor de sélection est standard donc les courants de fuite sont élevés mais il n'y avait pas d'autre choix dans la technologie utilisée. Il y a un W de 150 nm et un L de 1 µm pour limiter les courants de fuite. Quant à la capacité, elle a été faite à partir des capacités parasites d'un transistor NMOS, les modèles des cellules DRAM n'étant pas disponibles. Le drain et la source ont été connectés à la masse pour que la capacité soit la plus grande possible. Le transistor a un W de 1µm et un L de 1 µm. Ensuite les transistors des portes NAND et du multiplexeur ont un W de 300 nm et un L de 130 nm pour limiter les défauts liés au procédé de fabrication. Le circuit a donc été conçu de façon non optimale pour assurer sa viabilité. Mais les tests ont permis de valider la fonctionnalité du circuit et des enseignements ont été apportés et seront utiles pour les futurs circuits.

Le circuit a été conçu avec le moins de circuit de contrôle possible pour limiter la complexité et ainsi diminuer le risque d'erreur lors des tests. Il y a donc de nombreux signaux à générer lors de ceux-ci. Au total, il faut compter 22 plots dont les 2 tensions d'alimentation 1,2 et 3,3 V, leur masse respective, les quatre entrées de sélection des JTMs, les sélections des capacités, Heat, On, En, AutoZ, AutoZ_bar, les entrées de la LUT e0 et e1, sa sortie Out et les commandes du générateur e_gene_1 et e_gene_2.

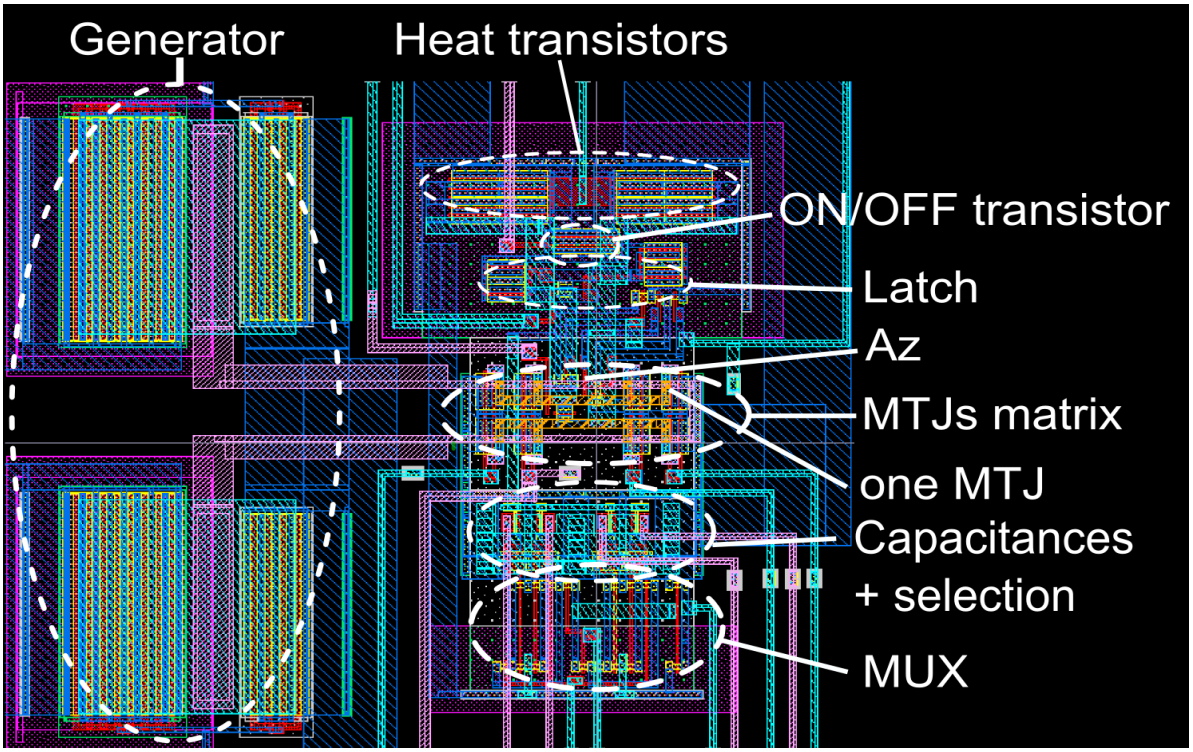


Figure 126 : layout du circuit intégré dans le démonstrateur

La surface totale du circuit est de 1540 µm² dont un peu moins de la moitié est occupée par le générateur de courant. Il faut noter que pour faire une LUT plus grande, par exemple 4 entrées, il suffirait d'ajouter des JTMs et agrandir le Multiplexeur. Le module de lecture et d'écriture sont inchangés. Il est donc intéressant d'agrandir la taille du bloc mémoire MRAM pour partager les blocs de lecture et d'écriture entre plusieurs JTMs pour diminuer leur taille relative. Mais avec une LUT 2, les blocs de lecture et écriture constituent la plus grande surface.

Les tests se sont déroulés sur le testeur Diamond D10 de chez LTX Credence (Figure 127). Il est destiné au test de circuits numériques, analogiques et signaux mixtes. Il peut tester des circuits prototypes pour le debug ou la caractérisation ou du test de circuit en grand volume durant la production. Il peut contenir jusqu'à 10 cartes de test. Les signaux de test numériques peuvent être cadencés jusqu'à 200 Mbps. Le testeur peut gérer jusqu'à 768 signaux numériques et 16 alimentations. A Spintec, nous utilisons 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. Dans notre cas, elles sont utilisées comme alimentations

- DPIN96 : permet de tester des circuits numériques. Jusqu'à 96 signaux numériques indépendants peuvent être testés. Pour chaque canal, les niveaux de tension, de courant, les timing, les formats et 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. Les tests fonctionnels sont programmés avec le langage Standard Test Interface Language (STIL). Chaque canal peut stocker jusqu'à 32M vecteurs en mémoire.

Figure 127 : photographie du testeur Diamond de LTX Credence

Le circuit, une fois découpé, a été intégré dans un boitier (Figure 128). La puce a été découpée en un rectangle qui comprend la LUT2, la bascule non-volatile et une mémoire Crocus car la découpe des circuits de Spintec seuls aurait pu les endommager parce que le rapport des longueurs des côté était élevé. Les puces ont été câblées de manière à pouvoir tester les deux circuits Spintec : la LUT et la bascule non-volatile.

Quatre puces ont été intégrées dans un seul boitier. Comme les deux circuits étaient proches, il n'a pas été possible de câbler les deux en même temps donc 2 puces ont été câblées sur la bascule et 2 autres puces sur la LUT.

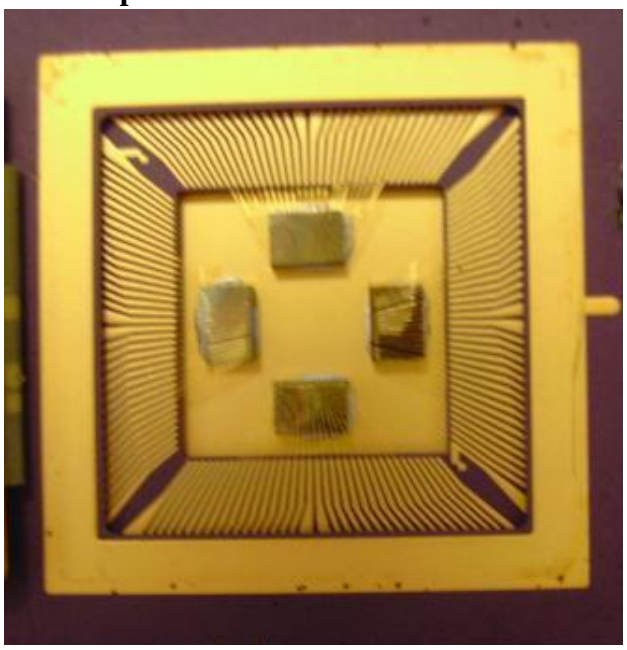


Figure 128 : photographie du démonstrateur comprenant quatre puces

Une fois toutes les puces testées, elles ont été décâblées puis recâblées sur le circuit non testé. Ainsi, 18 puces ont été testées. 7 puces ont fonctionné. Les résultats ont été compilés dans le Tableau 3.

Le dernier dispositif nécessaire au test était la carte de test qui permettait de relier le boitier au testeur. La Figure 129 montre la carte connectée au testeur via des fils. La carte comprend des connecteurs tulipe afin de brancher les fils ainsi qu'un connecteur pour insérer le démonstrateur dont le boitier est de type PGA. Les signaux logiques sont connectés à la carte DPIN96. Des capacités de découplage sont connectées le plus près possible des broches d'alimentation du boitier pour stabiliser l'alimentation du composant. Les broches d'alimentation sont branchées à la carte VIS16 du testeur qui peut être utilisée comme source de tension. Dans le circuit, deux tensions sont utilisées : 1,2 V pour les circuits CMOS logique et 3,3 V pour le circuit de génération du champ magnétique extérieur. Pour que l'alimentation soit plus stable, des fils d'alimentation ont été torsadés. Notons également qu'un deuxième circuit, conçu par Gregroy Di Pendina et implantant une architecture innovante de bascule non-volatile, a également été testé selon la même méthode. On peut voir sur la photo que les fils de la LUT sur la partie gauche de la photo et ceux de la bascule sur la partie droite.

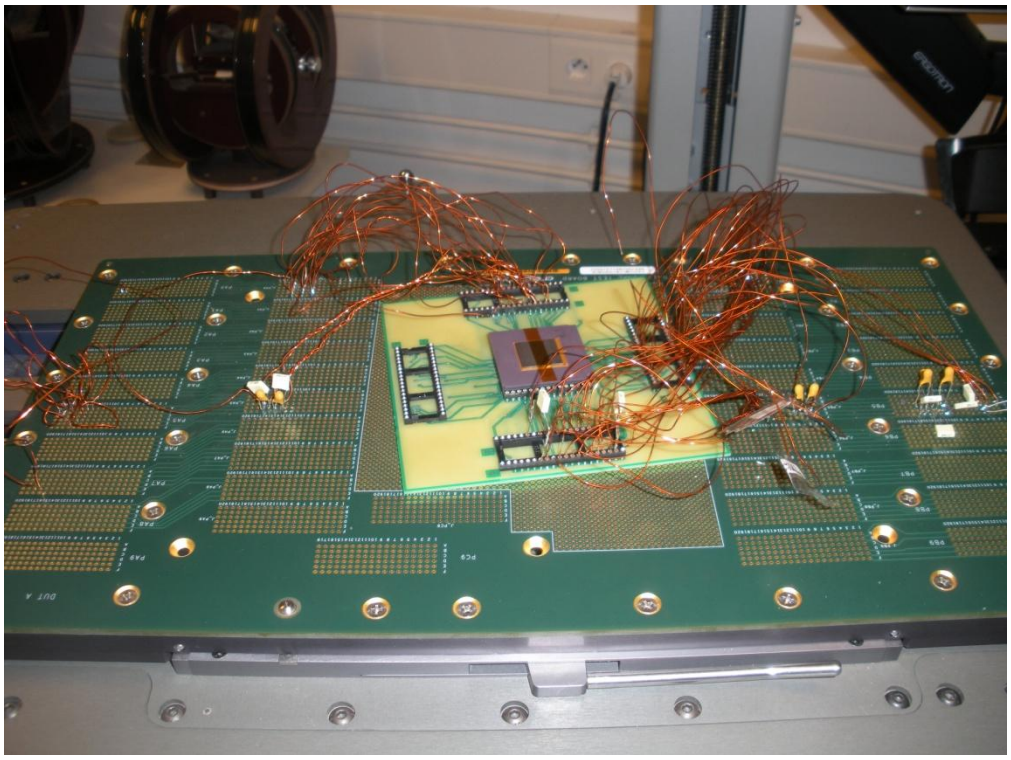


Figure 129 : photo de la carte de test avec le circuit en cours de test

La mise en place du test a tout d'abord commencé par l'écriture du programme de test qui indique au testeur les ressources matérielles à utiliser comme le nom des entrées/sorties et leur emplacement et les alimentations. Le programme doit également mentionner les caractéristiques des signaux et des alimentations comme les tensions d'entrée correspondant aux niveaux logiques 0 et 1, les seuils de tension qui détermine si une sortie est à 0, 1 ou dans un état indéterminé et les temps de cycle des vecteurs de test et en particulier la durée d'une période. Il faut également décrire les vecteurs de test c'est-à-dire définir la suite de 0 et 1 à chaque signaux. Toutes ces informations sont stockées dans un ou plusieurs fichiers au format STIL et un fichier supplémentaire écrit en langage C contrôle toutes les étapes du test c'est-à-dire la mise en route et l'extinction des alimentations et également le lancement des vecteurs de test. Une fois le programme de test mis en place puis débuggé, il restait à connecter la carte de test aux cartes DPIN96 et VIS16 du testeur avec des fils. Cette étape a été fastidieuse et source d'erreur car les tests ont été retardés à cause de faux contacts. Pour réduire ce risque, les fils ont été soudés. Malgré les précautions prises, il y avait du bruit sur les signaux, qui gênait les mesures. Cependant il était suffisamment faible pour permettre d'observer le signal de sortie de la LUT. La Figure 130 montre l'interface graphique qui permet de lancer le test et d'observer le résultat. Pour chaque signal, deux courbes sont représentées. D'une part, on peut voir le signal attendu, c'est-à-dire les vecteurs de test qui ont été programmés pour les entrées et le signal correct pour la sortie et d'autre part, le signal qui est effectivement mesuré par le testeur. En regardant une sortie on peut donc déterminer si le test est correct en comparant ce qui est mesurée à ce qui est attendu ce qui est fait automatiquement par le testeur qui indique les bits passant ou faux (PASS ou FAIL). En observant les entrées, on peut vérifier qu'il n'y a pas de court-circuit entre deux entrées, par exemple, en comparant les signaux qui ont été programmés à ceux qui ont été mesurés. On peut voir sur la Figure 130 qu'en sortie, la présence de bruit se traduit par des pics jaunes ou verts.

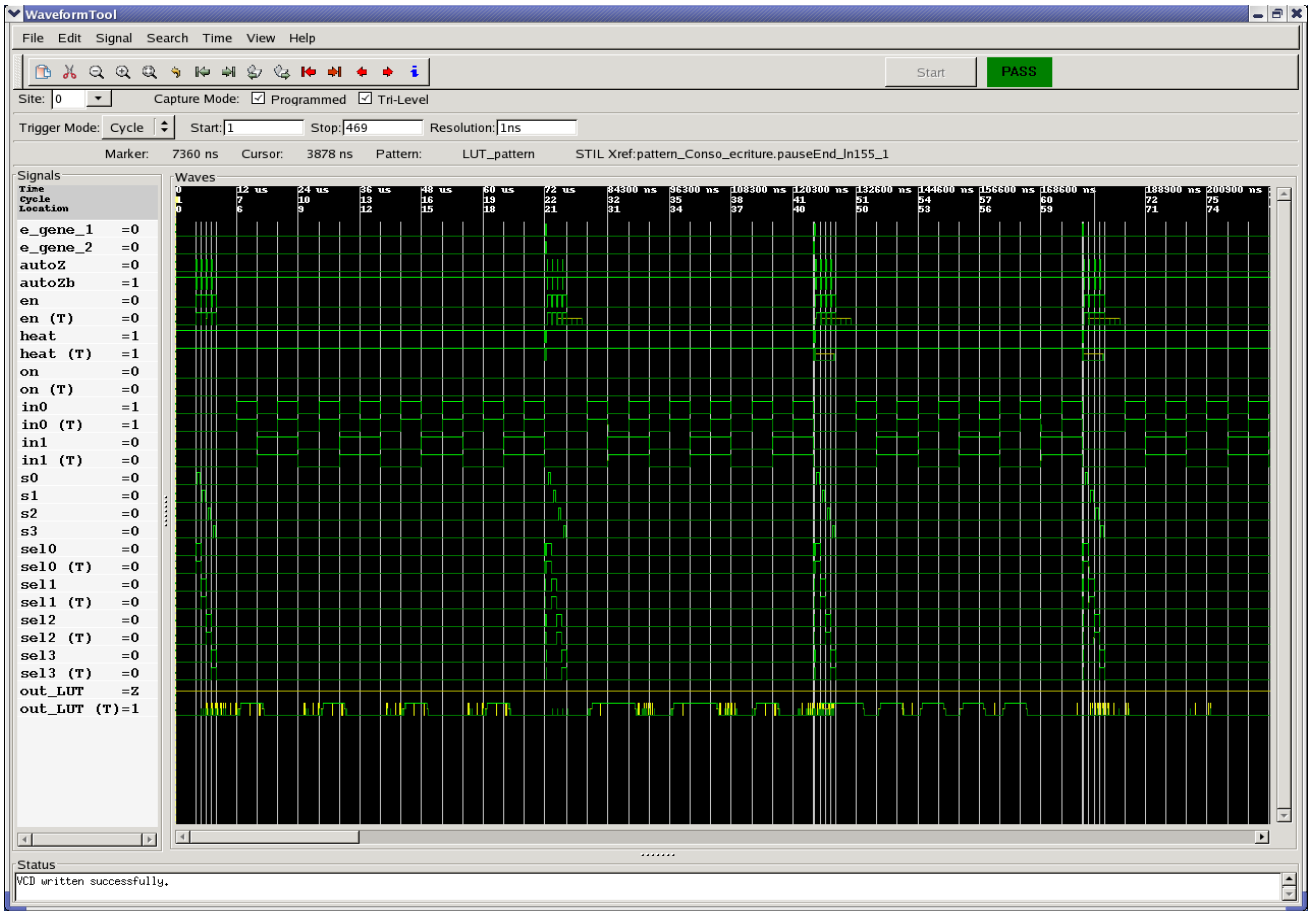


Figure 130 : interface graphique indiquant les résultats du test

XVIII. Résultats des tests

XVIII.1 Résultats des tests fonctionnels

Les tests fonctionnels comprenaient trois phases. La première phase était destinée à écrire une fonction logique dans la LUT. Elle consistait donc à écrire la table de vérité de la fonction dans les JTMs. Ensuite, la deuxième phase faisait un rafraichissement des capacités avec la nouvelle fonction. Pour finir, la troisième phase testait la fonction implémentée dans la LUT c'est-à-dire qu'elle envoyait tous les codes possibles (00, 01, 10, 11) en entrée de la LUT et vérifiait que l'on obtenait bien, en sortie, la table de vérité écrite lors de la phase de programmation. Pour que le test soit complet, toutes les fonctions possibles (16 fonctions) ont été testées successivement. Cela permet de vérifier que les JTMs étaient bien écrites et qu'elles fonctionnaient toutes. Finalement, pour tester la non-volatilité, le circuit était éteint puis remis sous tension et un nouveau test était effectué. Lors du nouveau test, la première opération était de faire un rafraichissement (sans programmation préalable) puis un test de la fonction pour vérifier que l'on retrouvait bien la fonction programmée lors du précédent test.

Le Tableau 3 montre les résultats des tests fonctionnels effectués sur les 18 puces. Le résultat indique simplement si le test s'est déroulé avec succès ou non (PASS ou FAIL), c'est-à-dire si tous les signaux de sortie correspondent aux signaux attendus. Dans les cas où le test est FAIL, la colonne « remarques » indique quelle en est la raison.

Tableau 3 : résultats des tests fonctionnels

boitier	puce	Résultat TEST	Remarques
1	Nord	PASS	Présence de bruit*
	Est	FAIL	Bloquée à '0000'**
	Sud	FAIL	Bloquée à '1000'**
	ouest	PASS	Présence de bruit*
2	Nord	PASS	Présence de bruit*
	Est	Non câblée	-
	Sud	FAIL	Bloquée à '0001'**
	ouest	Non câblée	-
3	Nord	PASS	Présence de bruit*
	Est	FAIL	Bloquée à '0100'**
	Sud	FAIL	Rien ne marche
	ouest	PASS	Présence de bruit*
4	Nord	FAIL	Bloquée à '0000'**
	Est	FAIL	Bloquée à '1100'**
	Sud	FAIL	Bloquée à '0000'**
	ouest	FAIL	Bloquée à '1100'**
5	Nord	FAIL	Bloquée à '0000'**
	Est	PASS	Présence de bruit*
	Sud	FAIL	Bloquée à '0001'**
	ouest	PASS	Présence de bruit*

*** Présence de bruit signifie que le signal de sortie présente de nombreux pics de tension que l'on peut attribuer à du bruit présent dans le circuit de test. En effet, les fils qui connectent les signaux du testeur au boitier sont très longs ce qui favorise l'apparition de bruit. De plus, les capacités de découplage sont situées loin du boitier, ce qui peut générer des erreurs. Ces problèmes pourront être résolus à l'avenir grâce à la réalisation d'une carte de test avec une connectique de meilleure qualité**

**** Bloquée à '0000' par exemple signifie que la table de vérité lue en sortie est '0000', correspondant à la sortie pour chaque code d'entrée : 00, 01, 10 et 11**

Remarque :

- **Les boitiers 1 et 2 avec les puces Nord et Sud, ont été les premiers à être testés. Lors de ces premiers tests, les paramètres optimaux n'étaient pas bien déterminés, en particulier le courant maximal dans le générateur de courant. Celui-ci a été très élevé lors des premiers tests ce qui a pu détruire la ligne de champ et donc rendre impossible l'écriture des JTMs. Ceci a pu se produire pour les puces 1_Sud et 2_Sud et expliquerait leur dysfonctionnement**
- **Les puces des boitiers 1 à 3 ont été découpées au centre du wafer tandis que celles des boitiers 4 et 5 ont été découpées sur les bords. Ceci explique pourquoi il y a des puces correctes pour les boitiers 1 à 3 et très peu pour les boitiers 4 et 5**
- **Les puces 1 Nord et 3 Nord ont fonctionné durant plusieurs jours. A la fin des tests, celle-ci n'ont plus fonctionné peut-être en raison du claquage des JTMs ou de certain transistors lors des tests à des tensions plus élevée (1.3 V) ou bien la ligne a été détruite. Je privilégie la troisième explication car les JTMs ne peuvent pas claquer toutes en même temps et les transistors CMOS peuvent tenir longtemps à tension élevée**

XVIII.2 Description des vecteurs de test

Décrivons d'abord les vecteurs de test de la partie phase de programmation. Elle consiste à sélectionner le couple de JTM à écrire avec le vecteur SEL.

Le vecteur SEL = '0001' sélectionne la cellule 0

Le vecteur SEL = '0010' sélectionne la cellule 1

Le vecteur SEL = '0100' sélectionne la cellule 2

Le vecteur SEL = '1000' sélectionne la cellule 3

Ensuite, on chauffe les deux JTMs en mettant le signal HEAT à 0. Lorsque les JTMs ont dépassé la température de blocage, on applique le champ magnétique extérieur en faisant passer un courant de +/-10 mA dans la ligne de champ. Après avoir dépassé la température de blocage, le signal HEAT retourne à 1 pour que les JTMs refroidissent. Durant cette phase de refroidissement, le champ magnétique est maintenu jusqu'à ce que la température des JTMs redevienne basse. On obtient alors les vecteurs de test de la Figure 131.

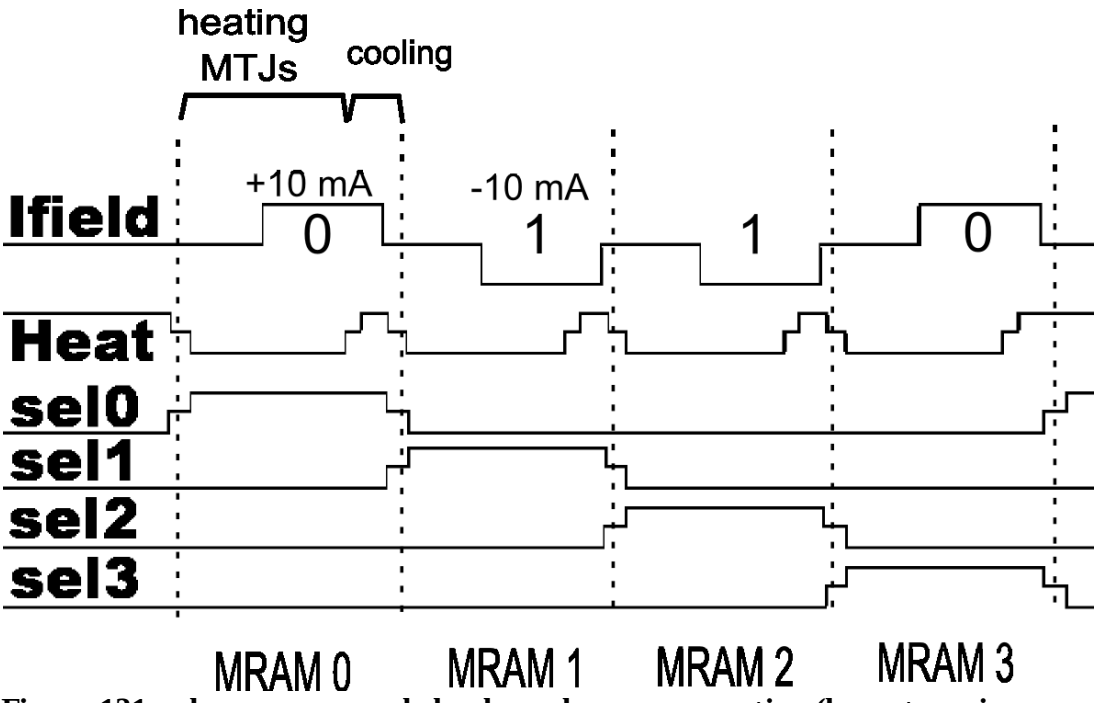


Figure 131 : chronogramme de la phase de programmation (les autres signaux sont inactifs)

La phase de rafraichissement est représentée sur la Figure 132. D'abord, un couple de JTMs est sélectionné par le vecteur SEL comme précédemment. Ensuite la donnée est lue grâce au Latch. Pour initialiser la lecture, un Auto Zéro est effectué grâce au signal Az. Ensuite, on met En à 1 pour autoriser l'écriture des capacités. Puis la capacité correspondante à la cellule MRAM lue est sélectionnée avec le vecteur S. La capacité est alors rafraichie. Pour finir, on désélectionne la capacité puis la cellule MRAM. On peut alors passer au rafraichissement des capacités restantes jusqu'à les avoir toutes rafraichies.

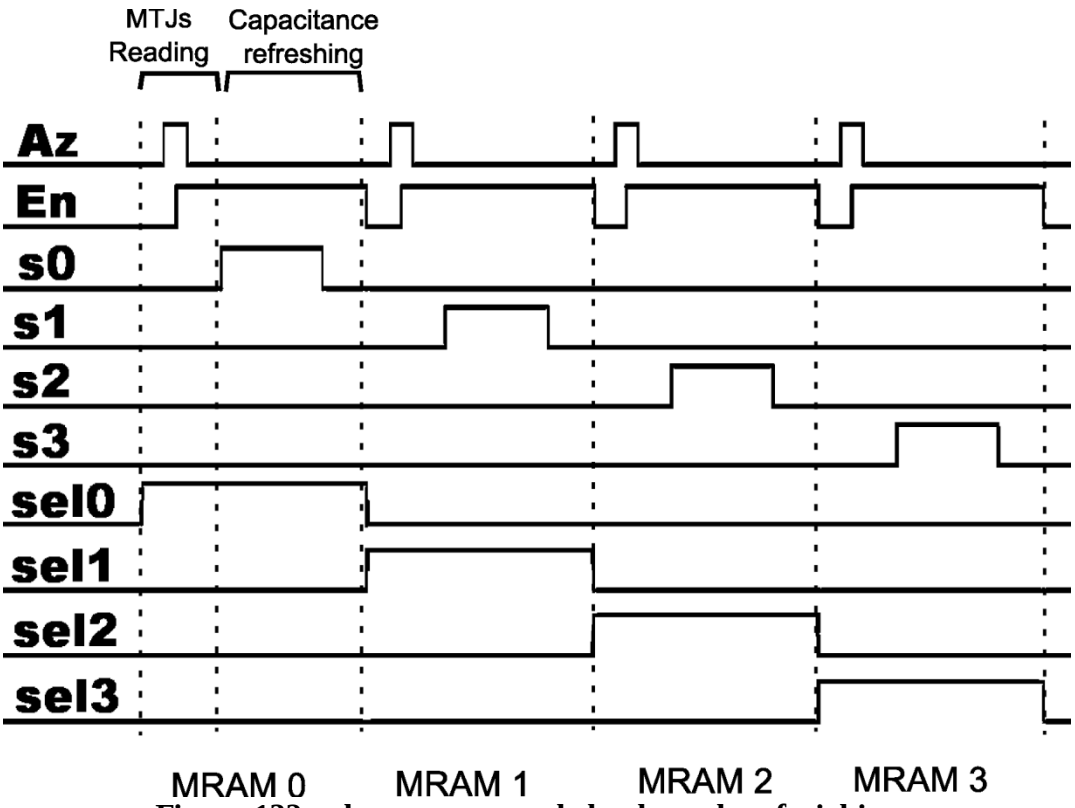


Figure 132 : chronogramme de la phase de rafraichissement

Après les phases de programmation et de rafraichissement, il ne reste plus qu'à tester le bon fonctionnement de la LUT en appliquant les vecteurs d'entrée sur les entrées de la LUT en vérifiant que l'on obtient bien en sortie la table de vérité programmée. On applique toutes les entrées possibles en entrée c'est-à-dire 00, 01, 10 et 11. On obtient la Figure 133.

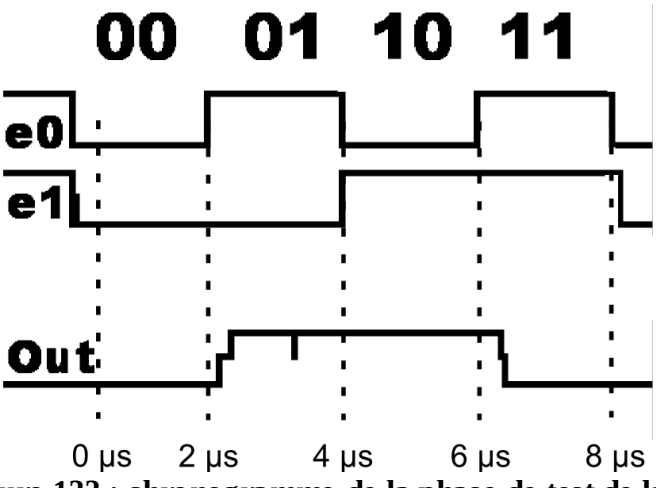


Figure 133 : chronogramme de la phase de test de la LUT

Pour vérifier que les JTMs ont été effectivement bien écrites, on effectue une reconfiguration c'est-à-dire que l'on écrit une autre fonction. La Figure 134 montre cette reconfiguration. On voit que la fonction implémentée précédemment est la fonction « 0010 ». Ensuite, intervient la reconfiguration. Il suffit de répéter la séquence de programmation décrite précédemment : programmation, rafraichissement et test des

entrées. On voit sur la Figure 134 qu'après la fonction « 0010 », on programme une nouvelle fonction : « 0110 ». Ensuite, intervient la phase de rafraichissement puis le test de la fonction. On voit que l'on obtient bien la fonction « 0110 ». Les JTMs ont donc bien été écrites.

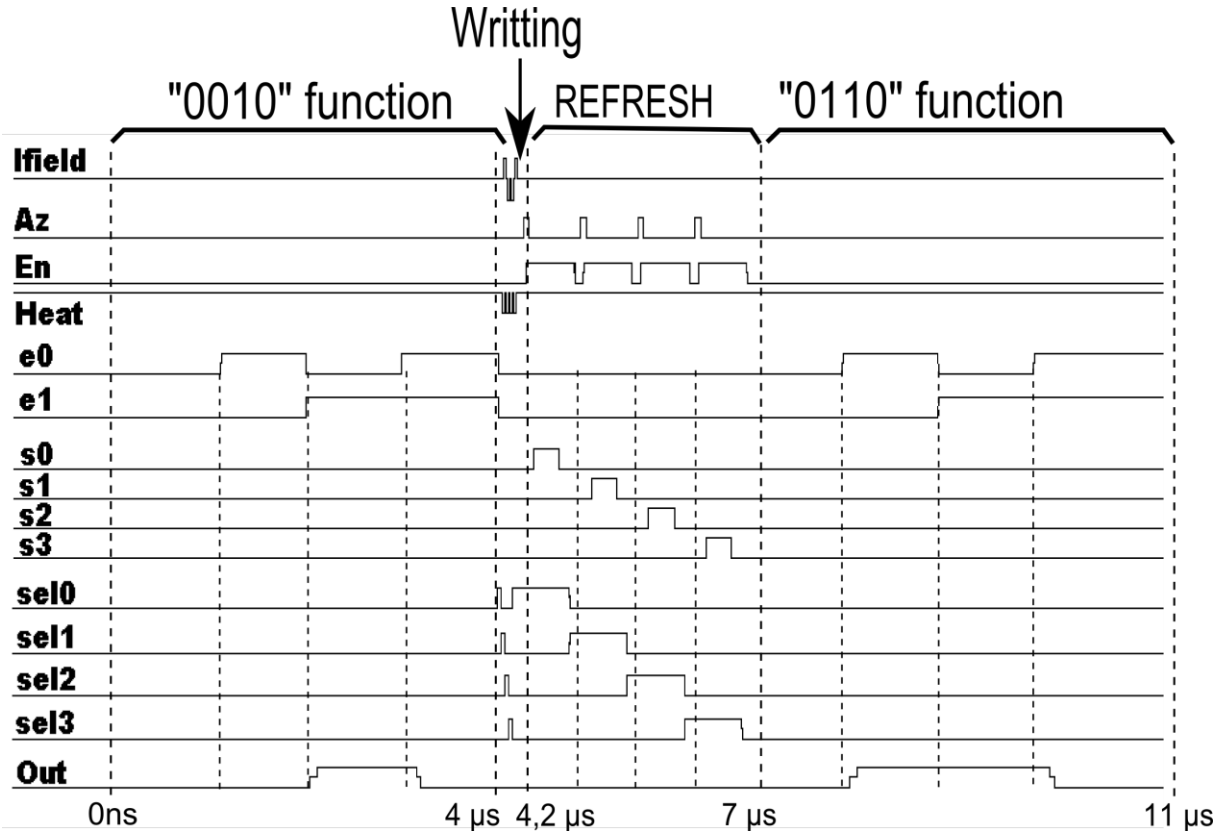


Figure 134 : chronogramme de la reconfiguration de la LUT

Toutes les fonctions possibles ont été écrites pour vérifier qu'aucun bit n'était collé à 1 ou 0. Les tests fonctionnels ont permis de vérifier le bon fonctionnement de la LUT et ainsi de valider le concept.

Note : lors de la conception du circuit, nous n'avions pas accès a des plots numériques, mais seulement des plots de contact métalliques. Il aurait fallu concevoir des buffers de sortie afin d'amplifier le signal pour attaquer l'extérieur et en l'occurrence le testeur, ce que nous avons oublié.. Les temps de montée et de descente du signal de sortie sont donc très long ce qui explique pourquoi l'échelle des temps est comptée en microsecondes. Ce n'est cependant pas important car la rapidité de la LUT en utilisation est liée au multiplexeur qui, dans notre cas, est classique. Donc l'évaluation de la rapidité de la LUT en fonctionnement ne présente pas ici d'intérêt. L'architecture présentée dans cette thèse peut améliorer la rapidité du FPGA mais de façon indirecte car la taille peut être réduite. Le démonstrateur ne permet pas de déterminer cette amélioration car il faudrait un circuit beaucoup plus complet pour prendre en compte les interconnexions programmables.

XVIII.3 Test de consommation et de rapidité

Pour tester la consommation minimale et la rapidité d'un circuit lors de la programmation de la LUT, on peut effectuer un test en faisant varier deux paramètres :

la tension du générateur de courant et la durée de l'impulsion de courant qui permet de moduler la durée et donc l'énergie de chauffage. La Figure 135 montre un test où l'on fait varier ces deux paramètres et l'on observe si le test est **PASS** (JTMs écrites avec succès) ou **FAIL** (JTMs non écrites).

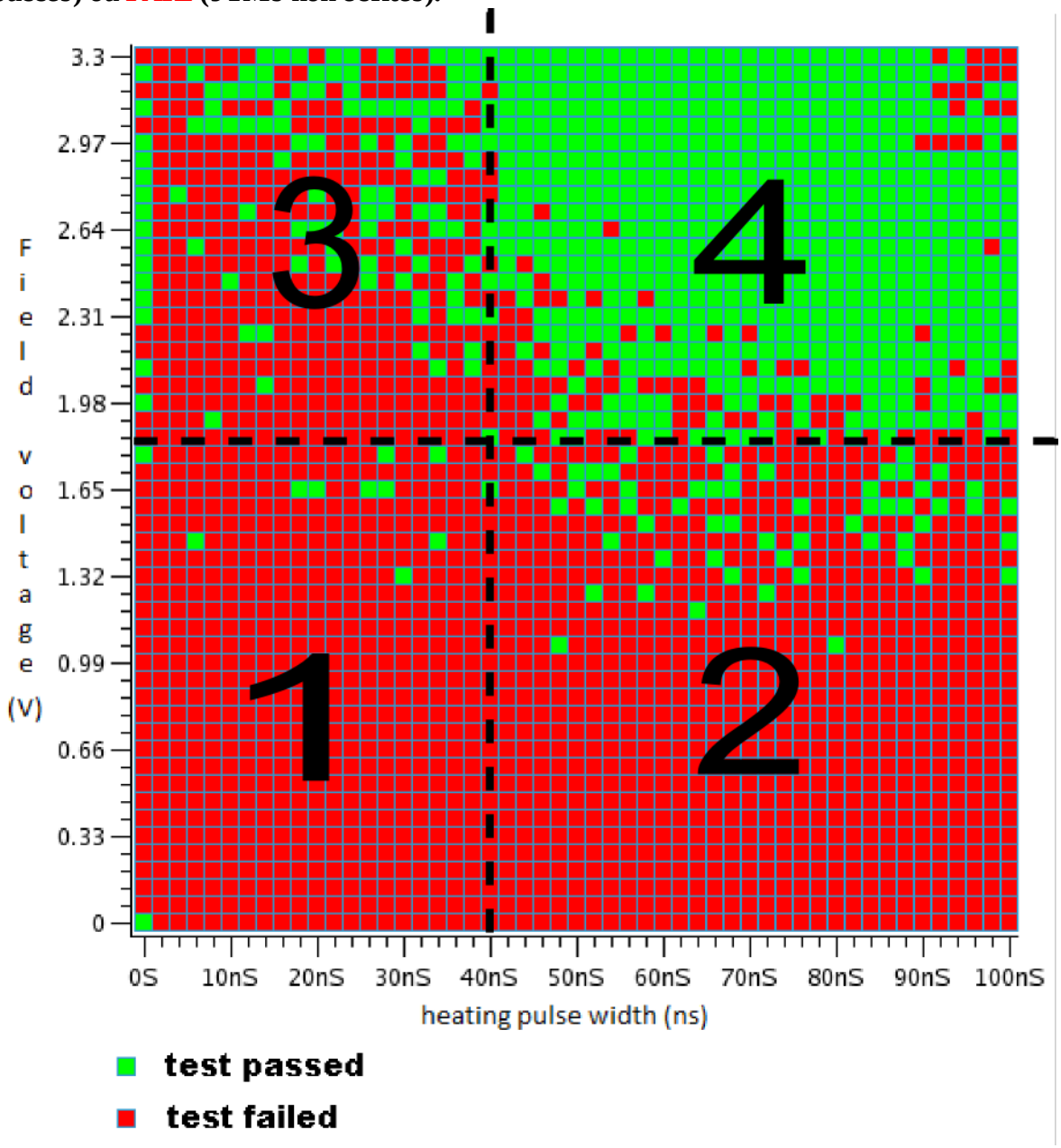


Figure 135 : test fonctionnel en faisant varier deux paramètres, le temps de chauffage et l'amplitude du champ d'écriture, afin de déterminer la consommation et la rapidité de la phase d'écriture des JTMs

On voit qu'il y a quatre zones caractéristiques :

- zone 1 : dans cette zone, la durée de chauffage et le champ sont trop faibles pour pouvoir écrire une JTM. Donc le test est FAIL
- zone 2 : dans cette zone, la durée de chauffage est assez longue mais le champ est trop faible pour pouvoir changer les données donc le test est FAIL
- zone 3 : dans cette zone, le champ est assez grand pour changer les données mais le temps de chauffage est trop court

- **zone 4 : dans cette zone, le champ et le temps de chauffage sont suffisants pour écrire les JTMs. Donc le test est PASS**

On peut en déduire la durée minimale du pulse et la tension minimale pour le générateur de champ : 40 ns minimum pour la durée du pulse et 2,3V pour la tension minimale du générateur.

Il faut noter que les limites entre les zones ne sont pas claires. Certains points ne sont pas cohérents et sont dus à du bruit lors du test ou à des imperfections car la MRAM n'est pas une technologie mature.

On peut déduire de ces mesures la consommation minimale lors de l'écriture d'un couple de JTM. La durée de chauffage est donc de 40ns. La tension appliquée est de 1,2V et le courant mesuré est de 770µA. Pour le générateur, la tension minimale est de 2,3V, le courant mesuré est de 10 mA et le champ est appliqué durant 30 ns. La consommation minimale lors de l'écriture de 2 JTMs est donc de 727 pJ c'est-à-dire 363 pJ par JTM. Cette valeur peut être diminuée en écrivant plusieurs bits en même temps mais cela demanderait un circuit de contrôle plus complexe ce qui augmenterais la surface du bloc mémoire et diminuerais donc l'intérêt des MRAMs. Notons également que les courants nécessaires à l'écriture sont élevés car le diamètre des jonctions est de 220 nm contrairement aux jonctions utilisées pour la tuile décrite précédemment dont la taille était de 130 nm. La consommation des JTMs diminue avec la miniaturisation ce qui est un des avantages de la MRAM. Rappelons également que le manque de maturité de la technologie ainsi que la connaissance approximative que nous en avions nous a obligé à prendre des marges énormes lors de la conception, incompatibles avec une recherche de performances.

Remarque :

Le domaine visé est le spatial. Pour la plupart des applications, la configuration du FPGA est écrite une fois et n'est pas changée par la suite ou bien seulement mise à jour. C'est pourquoi la rapidité et la consommation lors de l'écriture des JTMs n'est pas importante. Cependant cela peut être important pour les applications de haute performance comme dans les télécommunications.

XVIII.4 Comparaison avec les simulations et améliorations possibles

Les simulations fonctionnelles montraient bien sûr que le circuit marchait. Ce ne serait donc pas pertinent de les comparer avec les courbes obtenues lors des tests, d'autant plus qu'elles ne permettraient d'évaluer que la rapidité du multiplexeur ce qui ne nous intéresse pas (le multiplexeur est très classique). Comparons plutôt la consommation du circuit lors du chauffage et le courant du générateur. Le courant simulé dans le générateur de champ est de 50 mA. Ceci est largement assez pour écrire les JTMs. En effet, d'après les mesures, entre 10 et 20 mA sont suffisants. On en déduit donc que la taille des transistors peut être fortement réduite : d'un facteur 2 pour avoir de la marge.

Concernant le courant lors du chauffage, la valeur mesurée lors des simulations est de 740 µA. Dans la pratique, les mesures révèles des dispersions. Rappelons que ces dispersions n'ont pas pu être évaluées en simulation car les modèles des simulations de Monte-Carlo des JTMs n'existaient pas. Les dispersions mesurées étaient grandes en

raison du fait que la résistance des JTMs intervient dans la consommation contrairement au générateur de champ qui n'utilise que la technologie CMOS. Le Tableau 4 résume les différentes valeurs mesurées.

Tableau 4 : mesure du courant consommé par le circuit lors de la phase de chauffage (sans le générateur de courant)

boitier	puce	Résultat TEST	Courant mesuré
1	Nord	PASS	NM*
	Est	FAIL	186 µA
	Sud	FAIL	100 µA
	ouest	PASS	422 µA
2	Nord	PASS	1,4 mA
	Est	Non câblée	NM
	Sud	FAIL	840 µA
	ouest	Non câblée	NM
3	Nord	PASS	NM
	Est	FAIL	268 µA
	Sud	FAIL	930 µA
	ouest	PASS	447 µA
4	Nord	FAIL	NM
	Est	FAIL	4,3 mA
	Sud	FAIL	NM
	ouest	FAIL	3,2 mA
5	Nord	FAIL	NM
	Est	PASS	non pertinent
	Sud	FAIL	NM
	ouest	PASS	non pertinent

*** NM : non mesuré car les puces en question étaient dé-câblées**

On peut voir dans le tableau qu'il y a une grande dispersion des résultats. Comme les couches CMOS sont dans une technologie mature et que les transistors ont une grande taille, on peut supposer que ces dispersions sont dues aux JTMs. Il semble que dans les cas, où les tests sont FAIL, le courant mesuré est soit très élevé (4,3 mA) soit très faible (100 µA). Dans les cas où le courant est élevé, on peut supposer que les JTMs ont claquées et dans le cas où le courant est très faible, on peut en déduire que les JTMs ont une résistance trop élevée pour dépasser la température de blocage. Dans les cas où les tests sont PASS, le courant est aux alentours de 450 µA sauf pour la puce 2_Nord (1,4 mA). Cette valeur est du même ordre de grandeur que la valeur trouvée en simulation. Notons que la puce 2_Nord ne marche plus, sans doute parce que ce composant a été souvent testé notamment avec des tensions d'alimentation élevées.

XVIII.5 CONCLUSION

Le démonstrateur présenté dans cette thèse est le premier circuit logique hybride CMOS/magnétique fonctionnel en Europe. Il a permis de valider le concept innovant de FPGA décrit précédemment et de tirer des enseignements utiles pour les futurs circuits tant du point de vue conception que du test. Les marges en termes de taille de transistor

pourront être réduites à l'avenir et permettront ainsi d'augmenter la densité des circuits ainsi que leurs performances. La taille des transistors s'explique également par le fait que le circuit a été conçu pour fonctionner dans la plage de température des composants militaires et spatiaux : -55 à 125°C. La taille des transistors de sélection des JTMs est donc grande car il faut pouvoir augmenter la température de la JTMs jusqu'à sa température de blocage de 180°C environ quand la température de la puce est de -55°C. De plus, les tailles des transistors pour la capacité et son transistor de sélection sont grandes car la charge des capacités doit tenir 10 µs à 125°C où les courants de fuite sont plus grands.

De meilleures caractéristiques en termes de surface, de tolérance aux radiations et de consommation pourraient être obtenues avec une technologie CMOS plus complète c'est-à-dire avec des cellules mémoires DRAM embarquées ainsi que des transistors à faible et forte tension de seuil. En effet, l'utilisation de DRAM permettrait de diminuer fortement la surface des mémoires de configuration ce qui réduirait également leur section efficace et réduirait donc leur sensibilité aux radiations. De plus, l'utilisation de transistors à faible courant de fuite, permettrait d'augmenter la durée de rétention des capacités et donc diminuer la consommation due à la phase de rafraichissement. L'utilisation de transistors à faible tension de seuil, qui possèdent une faible résistance à l'état passant, permettrait de diminuer la taille des transistors de sélection des MRAMs tout en faisant passer le même courant à travers les JTMs. Ainsi, l'utilisation conjointe de transistor faible et forte tension de seuil permettrait d'optimiser la surface, la fiabilité et la consommation du circuit. L'inconvénient est que cela nécessiterait de nombreux masques supplémentaires pour la fabrication du circuit ce qui augmenterait les coûts et la complexité du processus de fabrication. Cependant, les avantages en termes de consommation, de surface et de fiabilité pourraient les compenser.

XIX. CONCLUSION GENERALE

L'un des objectifs de la thèse était de déterminer un domaine d'application où l'utilisation d'un FPGA à base de MRAM serait très avantageuse par rapport aux FPGAs existants que ce soient les FPGAs SRAM, Flash ou antifuse. Ce domaine d'application est le spatial et plus généralement les domaines où les composants évoluent dans un environnement soumis aux radiations. En effet, dans ce domaine, les MRAMs ont de nombreux avantages par rapport à toutes ces mémoires. Concernant les avantages par rapport aux cellules SRAM, les cellules MRAMs sont non-volatiles ce qui permet de réaliser un FPGA dont la configuration ne s'efface pas lorsqu'il est coupé de l'alimentation. Cela permet également de couper l'alimentation du FPGA sans perdre d'information afin de faire des économies d'énergie ce qui est important dans des systèmes spatiaux où l'énergie est limitée. Ensuite, les FPGAs SRAM nécessitent une mémoire non-volatile externe qui stocke leur configuration. C'est un composant supplémentaire dans le système qui prend de la place et qui augmente son coût. La MRAM étant embarquée, aucun composant supplémentaire n'est requis ce qui limite les coûts. De plus, les MRAMs ont une surface plus réduite que les SRAMs, ce qui réduit encore la surface du composant. Ensuite, les JTMs qui composent les MRAMs sont immunes aux radiations ce qui permet de réaliser un FPGA résistant aux radiations à moindre coût. En effet, les SRAMs sont très sensibles aux radiations en particulier dans les technologies avancées. Il faut donc des techniques de durcissement qui augmentent la complexité du circuit et donc les coûts. Tandis que l'utilisation des JTMs permet de les réduire grâce à des techniques où l'on s'appuie sur leur fiabilité pour corriger les erreurs transitoires apparues dans les circuits CMOS.

Concernant les mémoires Flash, l'avantage des JTMs réside dans une consommation et une rapidité d'écriture beaucoup plus avantageuse ce qui permet de reconfigurer rapidement et avec une faible consommation le FPGA. De plus, l'endurance est quasi infinie ce qui permet de faire de la reconfiguration dynamique. La mémoire Flash est également sensible aux fortes doses de radiation tandis que les JTMs peuvent supporter plus de 100 krad ce qui est suffisant pour la plupart des applications du spatial. Ensuite, le principal avantage des MRAMs par rapport aux mémoires antifuse est le fait de pouvoir reconfigurer le FPGA ce qui permet une grande flexibilité du composant et en particulier de corriger d'éventuelles erreurs de conception et des mis à jour du circuit implémenté dans le FPGA afin de rendre un satellite, par exemple, opérationnel.

La MRAM est polyvalente parce qu'elle regroupe tous les avantages de ces mémoires, ce qui est un avantage pour un FPGA. En effet, ce type de composant est par nature très flexible et doit donc s'adapter à des domaines différents. Les mémoires MRAM sont donc les mémoires de configuration idéales pour un FPGA très flexible.

La solution d'architecture proposée dans cette thèse permet de tirer avantage de tous les atouts décrits précédemment. Le fait de combiner des mémoires MRAM à des mémoires DRAM permet de bénéficier de la haute densité de chacune de ces mémoires pour réaliser un FPGA dense. Cette densité permet de faire des circuits plus complexes car on dispose de plus de LUTs par unité de surface. Ensuite, les interconnexions étant raccourcies, elles sont moins capacitives ce qui réduit la consommation dynamique.

Le fait que les MRAMs soient non-volatiles permet de couper l'alimentation du bloc de mémoire MRAM afin de limiter la consommation statique. On pourrait également imaginer de couper l'alimentation de la LUT et des buffers des

interconnexions pour diminuer encore la consommation statique du circuit. Si le signal de coupure est relié au réseau d'interconnexion, cela permettrait de couper l'alimentation de façon dynamique pour optimiser la consommation en fonction de l'utilisation. De plus, les tuiles inutilisées peuvent être coupées ce qui optimise la consommation du circuit.

L'architecture proposée est très flexible, en raison du fait que les JTMs sont placées à proximité des circuits de calculs comme les LUTs et reliées grâce à seulement quelques fils véhiculant les données de configuration pour le rafraichissement. Cela permet d'optimiser l'utilisation des blocs de mémoire MRAM en les utilisant comme bloc de mémoire de donnée comme c'est déjà le cas dans les FPGAs SRAM. On peut également convertir les LUTs en mémoire de données DRAM ou hybride MRAM/DRAM. Dans les FPGAs SRAM, les mémoires de données constituées à partir des mémoires de configuration sont appelées mémoires distribuées. De plus les blocs mémoires MRAM défectueux peuvent être remplacés par des blocs de MRAM inutilisés.

L'architecture peut également être adaptée pour être reconfigurable dynamiquement. En effet, en connectant le bloc mémoire MRAM au réseau d'interconnexions, la mémoire de configuration peut être écrite par le circuit implémenté sur une partie du FPGA. Cela permet de réaliser des circuits plus complexes.

La fiabilité du FPGA s'appuie sur les mémoires MRAMs utilisées comme mémoires de référence. Le fait de rafraichir régulièrement les cellules DRAM permet de corriger les erreurs transitoires dues au passage d'une particule énergétique. Cette méthode appelée scrubbing est déjà utilisée dans les FPGAs SRAM. Elle est améliorée grâce au fait que la fréquence de rafraichissement peut être très rapide et ajustée en fonction des contraintes de fiabilité. Les techniques utilisées en durcissement de circuit peuvent ainsi être adaptées à la nouvelle architecture. En effet, de nouveaux compromis peuvent être trouvés. La redondance temporelle peut être améliorée grâce à une fréquence de rafraichissement plus élevée. Les calculs peuvent être faits deux fois de suite sur deux périodes de rafraichissement afin de détecter les erreurs dues à une erreur de configuration et refaire le calcul. Les ressources nécessaires peuvent être réduites notamment la quantité de mémoire pour stocker les résultats des calculs successifs. Ensuite, le surcoût en surface de la technique de la TMR peut être réduit en partageant les mémoires MRAM des circuits redondants. Ainsi, l'impact des MRAMs en termes de surface est limité. Le fait que les caractéristiques de cette nouvelle architecture viennent principalement des MRAMs font de cet ensemble un FPGA MRAM. Les mémoires DRAM peuvent ainsi être vues comme des mémoires intermédiaires.

La tuile implémentée durant la thèse et qui matérialise la nouvelle architecture de circuit de configuration a permis de déterminer ses principales caractéristiques et d'en déduire des estimations d'architecture différentes grâce à la découpe du circuit en circuits élémentaires. La tuile n'a pas pu être simulée entièrement car le matériel IT disponible n'était pas assez puissant. Toutes les caractéristiques n'ont malheureusement pas pu être testées et notamment la fiabilité par rapport aux erreurs transitoires. Elle pourra faire l'objet d'une prochaine thèse. Néanmoins, on peut imaginer que la fiabilité est meilleure que pour un FPGA SRAM grâce aux considérations théoriques du chapitre sur la description du concept. Cependant, les circuits élémentaires ont pu être simulés afin d'en déduire les caractéristiques générales de la tuile. La technologie utilisée étant ancienne, les contraintes de fiabilité étant forte (afin de fabriquer un démonstrateur) et les options technologiques étant limitées, des marges de fonctionnement élevées ont été prises qui ont induit un surdimensionnement du circuit. Les améliorations en termes de

densité étaient donc faibles mais les avantages en termes de consommation ont pu être déterminés. De plus, à partir de ces données, il a ensuite été possible de déterminer les caractéristiques d'autres architectures, plus avantageuses, grâce à des estimations, en prenant en compte d'autres technologies telles l'utilisation de vraies mémoires DRAM embarquées. Il a aussi été possible de comparer la même structure de LUT et d'interconnexions avec des SRAMs afin de confirmer les avantages de l'architecture innovante par rapport à un FPGA SRAM destiné également au spatial. Enfin cette architecture peut être adaptée à d'autres types de JTMs comme la technologie STT ou même d'autres nouvelles technologies de mémoires.

Finalement, l'un des principaux résultats est le test du démonstrateur fabriqué dans la technologie CMOS 130 nm de Tower et la technologie TAS de Crocus-Technology. Le circuit fabriqué, une LUT-2, était simple mais il a permis de prouver que le concept fonctionnait. C'est le premier circuit de ce type à fonctionner en Europe ce qui prouve également que ce type de circuit hybride peut être fabriqué et utilisé pour des applications logiques. Il a permis également d'acquérir des connaissances en termes de conception et permettra aux futurs circuits hybrides CMOS/magnétiques de fonctionner avec de meilleurs rendements. Il a permis notamment de déterminer que la taille des transistors d'écriture peut être réduite. Les prochains circuits, fabriqués dans des technologies plus avancées, permettrons d'améliorer les caractéristiques du FPGA et permettront d'acquérir plus de connaissances notamment en termes de fiabilité.

XX. PERSPECTIVES

Grâce à l'évolution des procédés de fabrication, les avantages de cette nouvelle architecture seront de plus en plus flagrants. Ceci est particulièrement vrai pour les technologies avancées car les caractéristiques des JTMs s'améliorent grâce à la miniaturisation. Il faudra en effet moins de courant pour écrire une JTMs, ce qui conduira à des transistors d'écriture avec des tailles réduites et donc un FPGA avec une meilleure densité. On peut également prévoir une meilleure fiabilité du procédé de fabrication et une meilleure densité des JTMs. Ensuite, les SRAMs seront beaucoup plus sensibles aux radiations ce qui rendra le FPGA MRAM plus avantageux par rapport au FPGA SRAM. On pourra également utiliser d'autres types de JTM comme les JTMs en technologie STT planaire ou même perpendiculaire car cette nouvelle architecture est flexible et peut donc s'adapter à tous types de mémoire. Des recherches restent à faire notamment en termes de fiabilité par rapport aux radiations des JTMs en technologie STT perpendiculaire. En effet, bien qu'une particule ne puisse pas changer l'état d'une jonction par elle-même, l'impulsion de courant généré a des caractéristiques proches du courant d'écriture d'une telle jonction, qui devient très faible pour les petites dimensions. Pour les applications du spatial, on pourra cependant sacrifier la consommation en acceptant un courant d'écriture plus élevé afin de rendre les cellules résistantes aux radiations. Pour cela, il suffira d'augmenter la taille des jonctions. Finalement, de plus en plus d'entreprises s'intéressent à la technologie magnétique pour remplacer à terme les mémoires classiques. Cette tendance touche d'abord les fabricants de mémoires et mènera peut être finalement les fabricants de FPGA à changer de types de mémoires de configuration et donc au FPGA MRAM.

XXI. BREVETS ET PUBLICATIONS

XXI.1 Brevet

O.Gonçalves, G.Prenat, Radiation hardened reprogrammable logic device, n°FR 12 53926 déposé le 27 avril 2012

XXI.2 Publications

O. GONCALVES, G.PRENAT and B.DIENY: Radiation hardened LUT for MRAM-based FPGAs in the 2012 Internaltional Semiconductor Conference Dresden – Grenoble, 24 au 26 September 2012.

O. GONCALVES, G.PRENAT and B.DIENY: Radiation hardened MRAM-based FPGA in the 12th Joint MMM/Intermag Conference, Chicago, 14–18 January, 2013

O. Gonçalves, G.Prenat, G. Di Pendina, B. Dieny, Non-Volatile FPGAs Based on Spintronic Devices, DAC conference, Austin, Texas, June 2-6 2013, *accepted*

Olivier GONCALVES, Guillaume P. RENAT, Gregory DI PENDINA, Christophe LAYER and Bernard DIENY, Nonvolatile Runtime-Reconfigurable FPGA Secured through MRAM-based Periodic Refresh, IMW conference, Monterey, California, May 26-29 2013, *accepted*