Taux d'erreurs dues aux radiations pour des applications implémentées dans des FPGAs à base de mémoire SRAM : prédictions versus mesures

G. Foucard

UNIVERSITE DE GRENOBLE
INSTITUT POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque
|__|__|__|__|__|__|__|__|__|__|

T H E S E

pour obtenir le grade de

DOCTEUR DE L'Université de Grenoble
délivré par l'Institut polytechnique de Grenoble

Spécialité : *« Micro et Nano Electronique »*

préparée au laboratoire TIMA

dans le cadre de **l'Ecole Doctorale** *« Electronique, Electrotechnique, Automatisme & Traitement du Signal »*

présentée et soutenue publiquement

par

Gilles Foucard

Le 11 juin 2010

Taux d'erreurs dues aux radiations pour des applications implémentées dans des FPGAs à base de mémoire SRAM : prédiction versus mesures

Raoul Velazco

JURY

M. Michael Nicolaidis	, Président
Mme. Lirida Naviner	, Rapporteur
M. Luis Entrena	, Rapporteur
M. Raoul Velazco	, Directeur de thèse
Mme. Sophie Duzellier	, Examinateur
M. Dan Alexandrescu	, Examinateur

Remerciements

Cette thèse a été réalisée au sein du groupe ARIS du laboratoire Techniques de l'Informatique et de la Microélectronique pour l'Architecture des Ordinateurs (TIMA). A l'issu de ces travaux, je souhaiterais remercier les personnes suivantes:

Monsieur Bernard Courtois, Directeur de Recherche au CNRS et Directeur du Laboratoire jusqu'au 31 décembre 2006 puis Madame Dominique Borrione, Directeur de Recherche au CNRS, qui a pris sa succession et qui ont rendu possible cette thèse en m'accueillant au sein de TIMA.

Monsieur Raoul Velazco, Directeur de Recherche au CNRS et Directeur du groupe ARIS, pour m'avoir proposé un stage il y a 5 années de cela et qui a conduit à cette thèse. Je lui exprime ma profonde gratitude pour son ouverture d'esprit, pour m'avoir fait confiance dans mes décisions techniques ainsi que sa disponibilité au cours de ces années. Je le remercie également pour ses conseils, ses critiques et ses encouragements.

Monsieur Michael Nicolaidis, pour m'avoir fait l'honneur d'accepter de présider le jury de cette thèse.

Madame Lirida Naviner et Monsieur Luis Entrena, pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de ce travail.

J'adresse mes remerciements à Madame Sophie Duzellier et Monsieur Dan Alexandrescu pour leur participation à mon jury en temps qu'examinateurs.

Monsieur Vincent Pouget, chercheur au laboratoire IMS (laboratoire de l'intégration du Matériau au Système), pour sa disponibilité et son aide lors de la mise en œuvre des essais laser. Ainsi que Alexandre Douin pour ses connaissances sur l'effet des lasers sur les circuits intégrés et pour sa dextérité dans le maniement du banc laser.

Monsieur Guy Berger, Ingénieur au Cyclotron de Louvain-la-Neuve, pour ses compétences techniques et sa disponibilité lors des campagnes de test que j'ai pu effectuées au cyclotron. Je remercie également Monsieur Jacques Viatour qui, grâce à ses compétences, à permis d'exploiter au mieux les temps de faisceau mis à disposition de ces travaux.

Je remercie Paul Peronnard pour son amitié infaillible depuis les bancs du lycée. Il a été d'une aide inestimable grâce à ses connaissances et ses conseils techniques, notamment lors des longues nuits de *debuggage*. Merci aussi à ses parents qui m'ont si gentiment accueillis et qui m'ont régalé avec leurs canards élevés en plein air.

Je remercie aussi Fabien Faure qui m'a fait partager ses connaissances et compétences techniques au début de ma thèse, alors qu'à l'époque il rédigeait la sienne.

Une thèse c'est en grande partie un travail de collaboration. Pour cette raison je remercie Gaëtan Canivet (avec qui j'ai partagé les joies et les galères du test de FPGA), Jean-Baptiste Ferron (pour son travail sur le logiciel SEFEA-Prod), Julie Correard (pour sa bonne humeur et son enthousiasme inaltérable), Yann Oddos (de m'avoir fait découvrir le café de la Poste et le Coco Loco). De manière générale j'ai une pensée particulière pour toutes les personnes du laboratoire grâce auxquelles j'ai passé de bon moment au laboratoire TIMA.

Finalement, je remercie tout particulièrement mes parents et ma famille qui m'ont permis de faire cette thèse, mais aussi de m'avoir supportés et encouragés durant toute cette période. Ainsi que mes amis proches : Poulpito (pour sa patience et dont le canapé a été pour moi une terre d'asile de nombreux vendredi soirs), Chili (pour sa bonne humeur et son goût de partager les bons produits du pays savoyard), Biour (pour les fêtes de village à Lentiol), Angel (pour les soirées pasta), et tous les membres de la XTBA, dont je fais parti, pour leur soutien au cours de ces années.

Table des matières

Liste des figures

Annexes:

Liste des tableaux

Annexes:

Liste des abréviations

AMBA	Advanced Microcontroller Bus Architecture
AOP	Amplificateur OPérationnel
AsGa	Arséniure de Gallium
ASIC	Application-Specific Integrated Circuit
BGA	Ball Grid Array
BRAM	Block RAM
CCA	Central Carrier Assembly
CCL	Configuration Control Logic
CE	Chip Enable
CEM	Compatibilité ElectroMagnétique
CEU	Code Emulated Upsets
CLB	Configurable Logic Block
CLK	Clock
CME	Coronal Mass Ejection
COTS	Commercial Off-The-Shelf
CRC	Cyclic Redundancy Check
CREDANCE	Cosmic Radiation Environment Dosimetry ANd Charging Experiment
CREME	Cosmic Ray Effects on Micro Electronics
DCM	Digital Clock Manager
DES	Data Encryption Standard
DICE	Dual Interlocked storage CEll
DIME	Dosimetry Intercomparison and Miniaturization
DRA	Defense Research Agency
DSX	Demonstration & Science Equipments
DUT	Device Under Test
EELV	Evolved Expendable Launch Vehicle
ELDRS	Enhanced Low Dose Rate Sensitivity
EPROM	Erasable Programmable Read Only Memory
ESPA	EELV Secondary Payload Adapter
ExGSCE	Experiment Ground Support Equipment
FPGA	Field programmable Gate Array
FTP	File Transfer Protocol
GNR	General Routing Matrix
GPIO	General Purpose Input Output
GTO	Geostationary Transfer orbit
HIF	Heavy Ion Facility
IOB	Input/Output Block
IP	Intellectual Property
JTAG	Joint Test Action Group
LET	Linear Energy Transfer
LSG	Laser Synchronous Generator
LUT	Look-Up Table

LWS	Living With a Star
MBU	Multiple Bit Upset
MCU	Multiple Cell Upset
MPTB	Micro-electronic and Photonic TestBed
NRL	Naval Research Lab.
OMERE	Outils de Modélisation de l'Environnement Radiatif Externe
PLL	Phase Loop Lock
PLS	Photoelectric Laser Stimulation
RADFET	RADiation sensing Field Effect Transistor
RAM	Random Access Memory
RTL	Register Transfer Language
SCAO	Sous-Système de Contrôle d'Altitude et d'Orbite
SEB	Single Event Burnout
SEE	Signe Event Effect
SEFEA-Prod	Soft Error Functionnal Effect Analysis in Programmable Devices
SEFI	Single Event Functional Interrupt
SEGR	Single Event Gate Rupture
SEL	Single Event Latchup
SET	Single Event Transient
SET	Space Environment Testbed
SEU	Single Event Upset
SHE	Single event Hard Error
SOI	Silicium On Insulator
SRAM	Static Random Access Memory
SSA	South Atlantic Anomaly
STRV	Space Technology Research Vehicles
TID	Total Ionising Dose
TMR	Triple Modular Redundancy
UART	Universal Asynchronous Receiver Transmitter
UCL	Université Catholique de Louvain
USAF	United States Air Force
VLSI	Very Large Scale Integration
WE	Write Enable

Introduction

Bien que la croyance commune appréhende l'espace comme un endroit « vide », il est, en autre, le théâtre d'un ballet incessant de particules énergétiques de diverses natures. Les principaux contributeurs à ce phénomène sont le rayonnement cosmique et le Soleil au travers d'activités permanentes comme les vents solaires ou bien d'événements violents et ponctuels comme les éruptions solaires et les éjections de masse coronale. Les particules éjectées sous forme de plasma viennent frapper continuellement la Terre et se manifestent sous forme d'aurores boréales et australes lorsque l'activité solaire est intense. Si ces phénomènes sont un spectacle pour les yeux, ils sont aussi une source de perturbation pour les circuits électroniques et donc une menace pour les applications. L'atmosphère de terrestre agit comme un bouclier naturel en déviant vers l'espace ou en absorbant une grande partie de ces particules, cependant le reste produit de nouvelles particules par collision avec les molécules de l'atmosphère. Ce processus, appelé *douche cosmique*, se répète jusqu'au niveau du sol. Si les premières mesures de flux de particules en orbite ont été réalisées dès 1958, grâce aux satellites Explorer I et III, ce n'est qu'à la fin des années 70 que sont apparues les notions de *soft errors* [May 1979] et *single event upsets* (SEU) [Guenzer 1979], qui décrivent l'inversion du niveau logique d'un point mémoire dû à l'interaction avec une particule. Cette étude concernait des mémoires DRAM fonctionnant en environnement spatial. A cette époque, la finesse de gravure des transistors rendait immunes aux radiations les composants fonctionnant dans l'atmosphère terrestre et seules les particules hautement énergétiques présentes dans l'espace pouvaient provoquer des événements. C'est ainsi qu'au cours des années 80 les techniques de durcissement des composants ont commencé à être mises au point pour les applications spatiales et aéronautiques critiques.

Dans les années 90, l'explosion du développement de l'électronique grand public a multiplié l'offre des composants COTS (Commercial Off-The-Shelf) qui sont toujours plus performants et économiques et qui sont disponibles en quantité. Cette course est rendu possible par la réduction continuelle des dimensions des transistors. Si intrinsèquement un petit transistor est moins sensible aux radiations qu'un gros, du fait de sa surface sensible plus petite, l'effet inverse est observé au niveau d'un circuit intégré. En effet, la sensibilité des composants augmente alors que la finesse de gravure diminue malgré le gain apporté par la miniaturisation. Les deux principaux facteurs responsables de cette conjecture sont l'augmentation de la densité d'intégration et la diminution des tensions de seuil des transistors. Ainsi depuis une dizaine d'années, des études ont pu mettre en évidence la potentielle sensibilité des circuits intégrés en haute altitude avionique [NORMAND 1998] et même au niveau du sol où le flux de neutrons est 300 fois inférieur à celui rencontré par les vols commerciaux. C'est notamment le cas pour les composants complexes à forte intégration tels que ceux à base de mémoire SRAM (Static Random Access Memory) [Baumann 2005], dans lesquels apparaissent de nouveaux effets comme les MBUs (Multiple Bit Upsets) et MCUs (Multiple Cell Upsets), jusqu'alors peu ou pas rencontrés. D'un autre coté, les composants durcis, qui ne

bénéficiaient pas de la même demande, se sont développés plus lentement. Ceci additionné à la réduction des budgets des projets spatiaux font que la tendance depuis les années 2000 est à l'utilisation de composants COTS dans les applications spatiales et aéronautiques [DiUbaldo 2000] [CHAU 2000].

Le marché des circuits complexes COTS se partage en deux grandes familles : les composants ASICs (Application-Specific Integrated Circuit) qui sont des circuits dédiés et les composants reprogrammables de type FPGA (Field-Programmable Gate Array). Les ASICs offrent généralement de meilleures performances et une consommation énergétique moindre que les FPGAs. De leur côté, les FPGAs sont appréciés pour leur faible coût, leur bonne performance, leur courte durée de mise sur le marché et leur grande flexibilité lors du développement. De plus, les modèles de FPGAs à base de mémoire de configuration de type SRAM sont particulièrement adaptés aux applications spatiales et aéronautiques grâce à leur capacité à se reconfigurer sur site. Malgré ces caractéristiques attrayantes, les concepteurs de systèmes embarqués sont souvent réticents à utiliser ce type de composant pour des applications critiques car les profiles des missions dans lesquels ils devront fonctionner incluent souvent un environnement radiatif sévère. Les phénomènes liés aux radiations les plus communément rencontrés sur les FPGAs à base de mémoire SRAM sont les SEUs et les MBUs qui peuvent affecter la mémoire utilisée par l'application (par exemple une bascule, un registre de machine à état, une mémoire de stockage, etc) ou bien la mémoire de configuration. Dans le premier cas une simple réinitialisation de l'application permet de retrouver un fonctionnement normal. Par contre, une faute dans la mémoire de configuration est rémanente, c'est-à-dire qu'elle est permanente jusqu'à ce que le FPGA soit reconfiguré. De plus cette inversion de bit peut résulter en une mutation des fonctions logiques implantées dans le composant conduisant à des fonctionnements imprévisibles et potentiellement critiques.

De part leur éventuelle dangerosité, les *upsets* dans la mémoire de configuration du FPGA doivent être pris en compte par les concepteurs d'applications critiques qui requièrent un haut niveau de fiabilité en milieu radiatif. Une technique de mitigation des fautes dues aux radiations bien connue, nommée TMR (Triple Modular Redundancy) [Kastensmidt 2005], est souvent adoptée pour rendre les applications plus robustes. Cette méthodologie consiste à effectuer en parallèle trois fois la même opération grâce à trois opérateurs identiques et indépendants, un voteur à la majorité détermine ensuite le résultat correct. En théorie, le point faible du TMR est son comparateur puisqu'il ne peut pas être protégé par de la triplication. De plus, la rémanence des fautes dans la mémoire de configuration provoque un phénomène d'accumulation des *upsets* qui peut corrompre la fonctionnalité de plusieurs opérateurs de calcul et ainsi mettre en échec le TMR.

Des méthodologies permettant d'évaluer la sensibilité des circuits intégrés face aux radiations ont été développées afin de prédire le comportement d'un composant cible dans son milieu de fonctionnement final et donc de décider si des techniques de durcissement doivent être mises en œuvre. Cela permet aussi de vérifier l'efficacité des moyens de protection qui ont été déployés et donc de valider l'application. Les différentes méthodologies peuvent être regroupées en deux catégories : les essais en environnement réel et les tests accélérés au sol. Les essais en environnement réel consistent à exposer le composant cible aux radiations naturelles lors de vols spatiaux [Duzellier 1997], à bord d'avions et de ballons stratosphériques [Taber 1993], ou au sol [Lesea 2005]. Dans tous les cas, ces essais sont longs et requièrent la multiplication du nombre de composants exposés afin de compenser la relativement faible densité de particules présente dans

l'environnement naturel. D'un autre côté, les tests accélérés au sol ont pour but de reproduire les effets des particules dans les circuits intégrés. Les moyens de test matériel les plus couramment utilisés sont les accélérateurs de particules qui sont capables de reproduire la plupart des particules trouvées dans la nature et les faisceaux laser dont la précision permet une cartographie détaillée des zones sensibles d'une application. Enfin, des méthodes matérielles/logicielles d'injection de fautes peuvent être soit purement logicielles, par exemple à l'aide de simulations, ou bien mixtes en insérant des fautes de manière logicielle dans une application fonctionnement sur du matériel. Des efforts de standardisation ont permis de décrire le processus de caractérisation de mémoires SRAM [JEDEC] [JESD89]. Cependant, aucune norme ne régit les tests pour les composants complexes tels les microprocesseurs ou les FPGAs. Cela est compréhensible du fait de la multitude des architectures et du fait que le taux d'erreur est aussi fortement lié à l'application elle-même.

C'est dans ce contexte que s'inscrit cette thèse. Les travaux effectués s'articulent autour de deux axes. D'une part, la mise en place d'une plateforme matérielle et logicielle permettant la caractérisation du composant cible, un FPGA à base de mémoire SRAM, grâce à des mesures effectuées en accélérateur de particules et à l'aide de faisceaux laser ainsi que l'adaptation au FPGA d'une technique matérielle/logicielle d'injections de fautes pour microprocesseur. Le but étant de confronter les résultats des mesures à ceux des prédictions et donc de conclure sur la pertinence de la méthode d'injection de fautes utilisée. Le deuxième axe concerne le développement d'une carte expérimentale destinée à obtenir des données sur la sensibilité d'une application tolérante aux fautes implémentée sur le FPGA à base de mémoire SRAM au cœur de cette thèse. Cette expérience sera embarquée dans le satellite scientifique du projet LWS-SET (Living With a Star – Space Environment Testbed) de la NASA.

Le chapitre 1 présente les sources de rayonnement et l'environnement radiatif naturel depuis le milieu spatial et jusqu'au sol. Les interactions des particules énergétiques avec la matière sont ensuite brièvement décrites. Enfin les effets des particules sur les circuits intégrés et les moyens de s'en prémunir sont abordés.

Le chapitre 2 dresse un état de l'art des différentes stratégies de test pour la caractérisation des circuits intégrés face aux radiations puis des différentes méthodologies telles que les tests en environnement réel, les tests accélérés au sol et les méthodes de prédiction par matérielle/logicielle de fautes. Enfin une plateforme de test générique pour composants numériques est décrite.

Le chapitre 3 présente le composant cible choisi pour cette étude puis la conception de la plateforme permettant la caractérisation du composant et des applications implémentées. Sont ensuite décris les protocoles de test et les résultats obtenus lors de campagnes de test en accélérateur de particules et sous faisceaux laser ainsi que lors de sessions d'injection de fautes. Les résultats des mesures sont confrontés aux prédictions afin de conclure sur la pertinence de la méthode d'injection de fautes.

Le chapitre 4 concerne l'expérience développée pour le projet LWS-SET en présentant le contexte et les contraintes imposées. Les différentes étapes de conception et de développement, depuis la maquette jusqu'à la carte de vol, sont ensuite décrites et illustrées.

Chapitre 1. Présentation du contexte

Dans ce chapitre sont décrit les différents types de radiations présents à la fois dans l'espace et dans l'atmosphère terrestre ainsi que leur interaction avec les circuits intégrés. Les effets résultant de ces interactions puis les moyens pour s'en prémunir seront alors présentés. Enfin les méthodes et outils permettant de prévoir ces effets sur les circuits intégrés seront évoqués.

1.1. Environnement et sources radiatives

L'environnement radiatif spatial provient de différentes sources telles que les ceintures de radiations de Van Allen, l'activité solaire et le rayonnement cosmique. Chacune offre une variété de particules qui lui est propre et chaque type de particule ne provoque pas les mêmes effets sur les composants électroniques.

Les électrons et les protons des ceintures de radiations ainsi que les protons provenant des éjections de masse coronale du Soleil se traduisent par un effet de dose sur les circuits intégrés. Le rayonnement cosmique et les particules chargées éjectés par les éruptions solaires sont responsables des événements singuliers. Une description des sources radiatives est donnée dans les paragraphes suivants, des informations détaillées peuvent être trouvée à la référence [Boudenot 2007].

1.1.1. Les éjections de matière par le Soleil

On distingue ici deux types de phénomènes en relation à l'environnement radiatif :

- L'éjection de masse coronale (CME – Coronal Mass Ejection) est une bulle de plasma pouvant atteindre plusieurs dizaines de rayons solaires. Ces phénomènes peuvent durer plusieurs jours et ils émettent principalement des protons à haute énergie (Figure 1-1).
- L'éruption solaire est un jet de matière ionisée s'élevant à des centaines de milliers de kilomètres d'altitude (Figure 1-2). Il s'agit ici d'ions lourds hautement énergétiques.

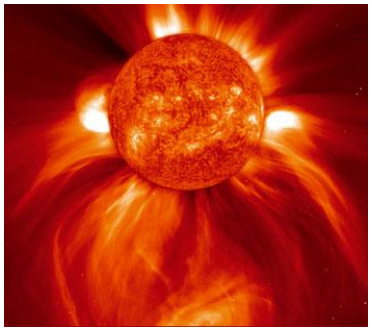

Figure 1-1 : Ejection de masse coronale

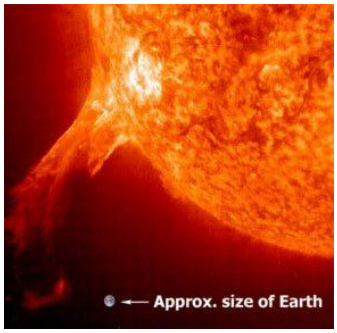

Figure 1-2 : Eruption solaire

1.1.2. Le vent solaire

Le vent solaire prend naissance dans la couronne solaire où les températures dépassent le million de degrés. Ceci confère suffisamment d'énergie aux électrons pour leur permettre de s'échapper du champ gravitationnel du Soleil. En réaction, des protons et des ions lourds sont eux aussi éjectés pour que l'astre conserve une charge électrique nulle. Ainsi le Soleil évacue environ 10^{14} kilogrammes de matière chaque jour. Ce flux de plasma projeté dans toutes les directions vient frapper la Terre à une vitesse comprise entre 300 et 1000 km/h.

1.1.3. Le rayonnement cosmique

Le rayonnement cosmique désigne le flux de particules de haute énergie présent dans tout l'univers. Il est composé de 87% de protons, 12% de noyaux d'hélium, le reste étant principalement des ions lourds. Les particules de faible énergie proviennent du Soleil, celles de moyenne énergie des super novas (ce sont des phénomènes galactiques) et des étoiles à neutrons. Enfin les particules de très hautes énergies correspondent aux sursauts gamma et aux collisions de galaxies, ce sont des phénomènes extra galactiques (l'énergie la plus élevée détectée est de 10^{20} eV). On y trouve les particules les plus énergétiques du système solaire et même le champ magnétique terrestre est souvent insuffisant pour les faire dévier.

1.1.4. Les ceintures de radiations de Van Allen

Les ceintures de radiation de Van Allen sont deux zones toroïdales de la magnétosphère terrestre entourant l'équateur magnétique et contenant une grande densité de particules énergétiques. La magnétosphère est la région entourant un objet céleste dans laquelle les phénomènes physiques sont dominés ou organisés par son champ magnétique. Elle est située au-delà de l'ionosphère, c'est-à-dire au-dessus de 800 à 1 000 km d'altitude. Cette région de l'espace agit comme un véritable bouclier protecteur contre le rayonnement. En effet les vents solaires frappent la magnétosphère pour lui donner une forme de goutte d'eau (Figure 1-3) ; une petite partie des particules est piégée par celle-ci tandis que la plus grande quantité est rejetée sur les cotés. Les périodes d'intense activité solaire donnent naissance à des orages magnétiques et des aurores polaires (boréales ou australes) grâce aux particules qui arrivent à pénétrer l'atmosphère par les pôles.

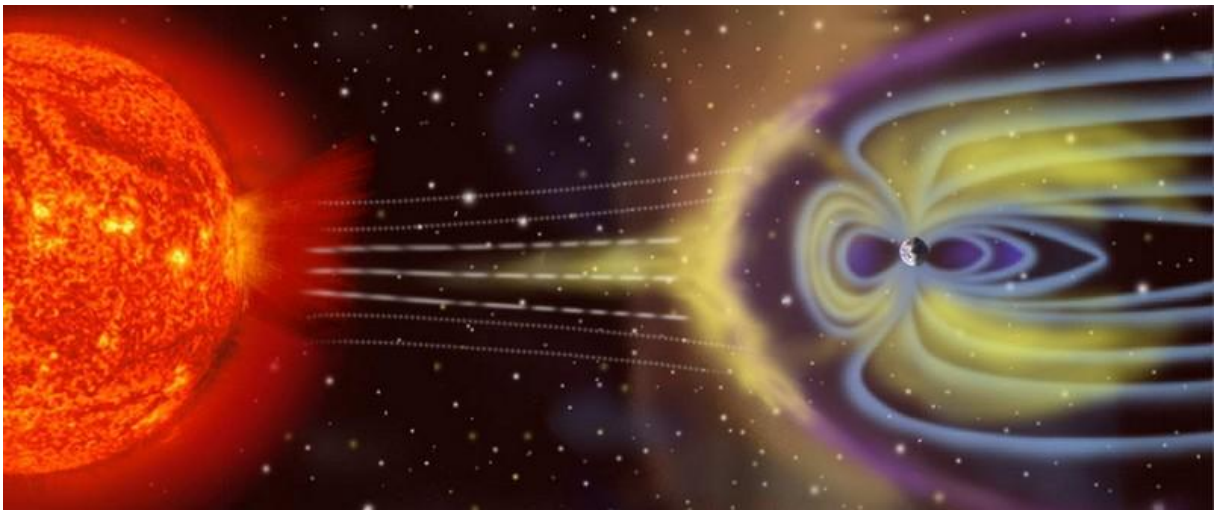

Figure 1-3 : Déformation de la magnétosphère par les vents solaires

La distribution et le type de particules varient suivant la position géographique et l'altitude. La perturbation la plus importante dans leurs dispositions est l'anomalie de l'Atlantique sud (SAA – South Atlantic Anomaly) où ces ceintures s'approchent à moins de 500 kms de la Terre. Les deux ceintures de radiations de Van Allen sont appelées « ceinture intérieure » et « ceinture extérieure » (Figure 1-4) :

La **ceinture intérieure**, située entre 700 km et 10.000 km d'altitude, est constituée principalement de protons à haute énergie (jusqu'à plusieurs centaines de MeV à des fluences de plusieurs dizaines de milliers de protons par centimètre carré et par seconde dans les zones les plus intenses) provenant du vent solaire et du rayonnement cosmique, piégés par le champ magnétique terrestre.

La **ceinture extérieure**, plus large, se déploie entre 13.000 km et 65.000 km d'altitude; elle est constituée d'électrons également à haute énergie (< 5 MeV) à des fluences de l'ordre du millier de particules par centimètre carré par seconde.

Les particules des deux ceintures se déplacent en permanence à grande vitesse entre les parties nord et sud de la magnétosphère.

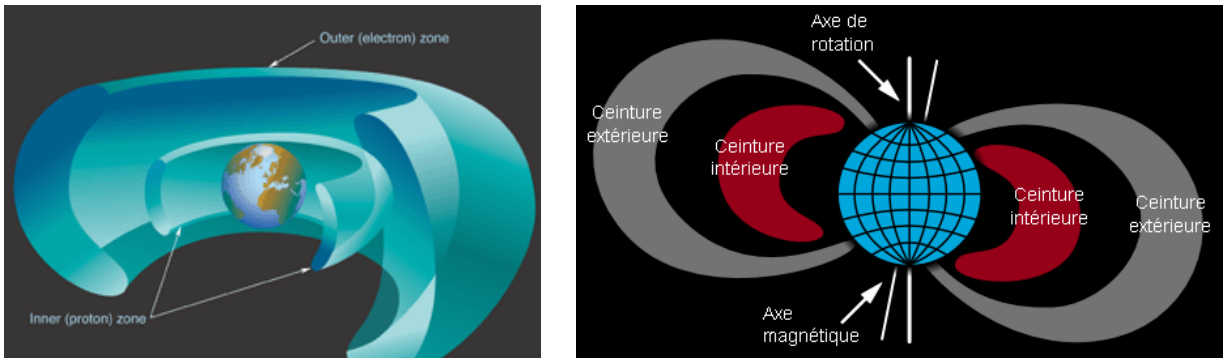

Figure 1-4 : La ceinture de radiations de Van Allen

1.1.5. Les neutrons atmosphériques

Les quatre précédentes sources de particules n'affectent que les personnes et équipements situés en dehors de l'atmosphère terrestre, et jusqu'à ces dernières années elles ne préoccupaient que le monde du spatial. Cependant depuis 1992, année d'observation du premier basculement de bit dans une mémoire en vol atmosphérique, des centaines d'effets similaires ont pu être enregistrés sur des SRAM au cours de plusieurs centaines de vols (civils et militaires) correspondant à plusieurs milliers d'heures de vol. Il a pu être mis en évidence que les particules responsables de ces phénomènes sont les neutrons atmosphériques [Leray 2004].

Les neutrons atmosphériques sont le résultat de la collision du rayonnement cosmique, principalement des protons, avec les atomes présents dans la haute atmosphère de la Terre. Le résultat de ces collisions est, soit la formation de particules ionisées, soit une réaction nucléaire qui produit principalement des neutrons, protons, électrons, etc … Ces phénomènes sont appelés « douche atmosphérique » (Figure 1-5). Le produit résultant de ces collisions est la génération de protons, neutrons, muons, … etc.

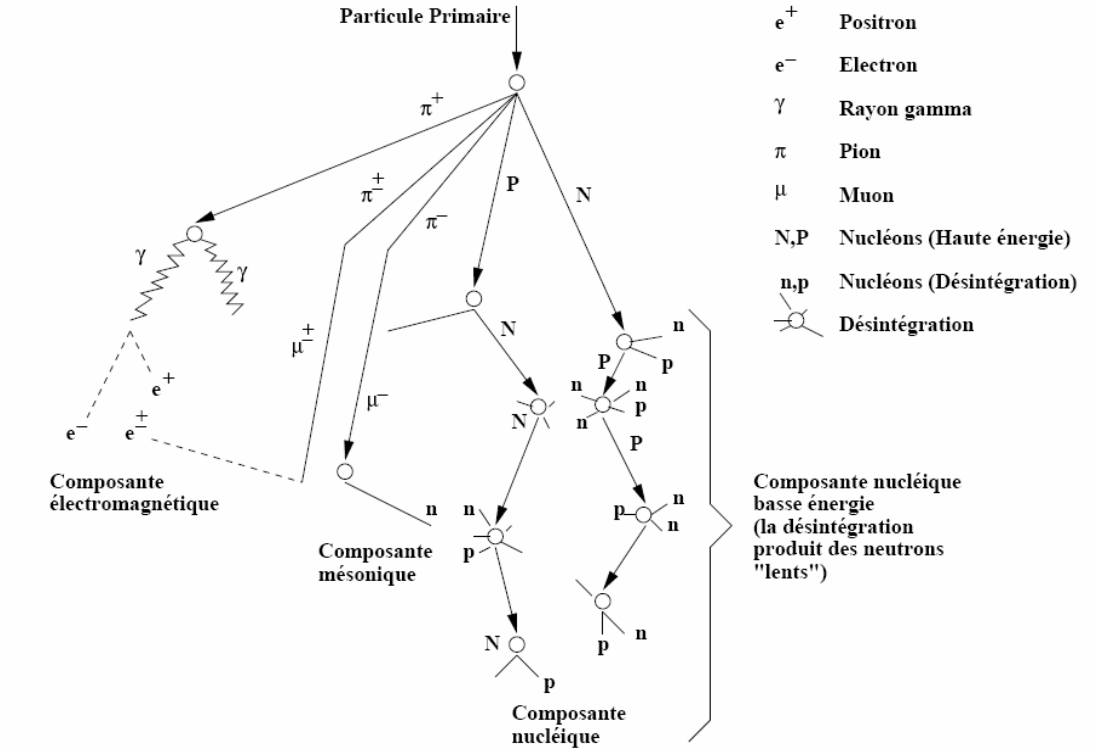

Figure 1-5 : Représentation schématique de la production de neutrons atmosphériques

La concentration maximale de radiations est rencontrée à 18km (60.000 pieds) d'altitude et elle diminue jusqu'à obtenir un flux d'environ 10 particules/cm²/heure au niveau de la mer. A 10.000 mètres, altitude des vols commerciaux, on a un flux de 10^4 particules/cm²/heure. Ces valeurs sont données à titre indicatif car l'activité solaire influence directement la population des particules. En effet, lorsque cette activité est forte le vent solaire augmente, ce qui a pour conséquence de renforcer le champ magnétique terrestre qui repousse aussi plus loin le rayonnement cosmique.

Tableau 1-1 : Récapitulatif des sources radiatives et de leur particule

Sources radiatives	Particules	Energies
Ceintures de Van Allen	Protons Electrons	qq MeV jusqu'à qq 100 MeV[1] < 7 MeV
Ejection de masse coronale	Protons	qq MeV jusqu'à qq 100 MeV
Eruption solaire	Ions lourds	qq 10 MeV jusqu'à qq 100 GeV
Vent solaire	Protons Ions lourds Electrons	< 100 keV < qq keV
Rayonnement cosmique	Protons Noyaux d'hélium	10^2 MeV jusqu'à 10^6 MeV 10 MeV jusqu'à 10^{30} eV
Neutrons atmosphériques	Neutrons	qq 100MeV

[1] L'électron-volt (eV) est l'énergie cinétique d'un électron accéléré depuis le repos par une différence de potentiel d'un volt.

Les neutrons étant des particules non chargées ils ne perturbent donc pas directement le fonctionnement des appareils électroniques. Par contre ils peuvent générer des noyaux de recul dans la matière qu'ils traversent. Lorsque ces ions prennent naissance dans les zones actives des circuits intégrés ils sont capables de modifier le comportement normal des composants.

Le Tableau 1-1 est un récapitulatif des sources de particules rencontrées dans l'espace et dans l'atmosphère terrestre, ainsi que leur nature et leur énergie. Le Tableau 1-2 est un récapitulatif de la nature des particules, pouvant perturber le fonctionnement des circuits intégrés, rencontrées en fonction de l'altitude.

Tableau 1-2 : Nature des particules en fonction de l'altitude

Les particules	L'altitude
Ions lourds	Orbite haute
Protons	Orbite basse
Neutrons, protons (secondaires)	Atmosphère
Neutrons	Au sol

1.2. Les différentes interactions

Les particules naturelles interagissant avec les circuits intégrés sont essentiellement les photons, les électrons, les protons et les ions lourds. Ci-après est décrite succinctement la manière dont ces particules agissent sur la matière, cependant une description plus détaillée peut être trouvée dans [Dos Santos 1998].

1.2.1. Interactions avec les photons

Les photons se déplaçant à la vitesse de la lumière sont capables de pénétrer profondément dans la matière, et cela malgré qu'ils aient une masse au repos et une charge électrique nulles. Ils peuvent interagir avec les atomes du matériau de trois manières différentes :

- **L'effet photoélectrique** : le photon incident communique, à condition qu'il ait une énergie inférieure à quelques dizaines de keV, son énergie à un électron. Si l'électron récupère une énergie suffisante il peut alors se libérer de son atome, on parle d'ionisation, ou bien changer de couche orbitale, on parle alors d'excitation.
- **L'effet Compton** : un photon incident, ayant une énergie de quelques dizaines de keV à plusieurs MeV, transfère une partie de son énergie à l'électron qu'il percute et continu sa trajectoire dans le matériau. L'électron ainsi éjecté de son atome est appelé électron de Compton.
- **La matérialisation** : les photons à haute énergie (> 20 MeV) se transforment en deux particules : un électron et un positron.

1.2.2. Interactions avec les neutrons

Le neutron est une particule sans charge électrique mais dotée d'une masse. En pénétrant dans le matériau une particule rencontre beaucoup de vide, et en cas d'impact elle percute

majoritairement des électrons. Par conséquent la probabilité pour qu'un neutron interagisse est très faible, cependant les effets indirects sont alors importants. En effet il ne peut être arrêté que par une collision avec un noyau lors d'une réaction de spallation[2] qui libère des particules légères très ionisantes. Lors de la collision divers phénomènes peuvent survenir, les trois principaux sont les suivants :

- **La diffusion élastique** : le neutron incident communique une partie de son énergie à l'atome percuté qui peut quitter sa maille cristalline si l'énergie qu'il reçoit est suffisante.
- **La diffusion inélastique** : l'atome capture le neutron incident, récupère une partie de son énergie puis le relâche. L'atome excité peut alors revenir à son état d'origine grâce à l'émission d'un rayon γ, mais en général il quitte sa maille cristalline et interagit.
- **La transmutation** : Le neutron incident est absorbé par le noyau de l'atome percuté. L'atome se mute en un autre élément atomique. Cet effet est peu important sauf dans le cas particulier où le neutron entre en contact avec un atome de $Bore_{10}$ (utilisé pour le dopage des substrats en technologie CMOS). Ce dernier se désintègre en émettant une particule α et un atome de $lithium_{7}$, qui peuvent interagir à leur tour.

Les neutrons sont classées en trois catégories en fonction de leur énergie : les neutrons lents (<1 eV) comprenant les neutrons thermiques (26 meV), les neutrons intermédiaires (entre 1eV et 100keV) et les neutrons rapides (> 100keV).

1.2.3. Interactions avec les particules chargées

Il s'agit des protons, des particules α et des ions. Plus ces particules ont une masse et une énergie importantes, plus elles provoquent des dommages et sont difficiles à stopper ou détourner. Pour la gamme des énergies qui nous concerne, ces particules interagissent par ionisation coulombienne, c'est à dire par ionisation du milieu.

Pour des technologies supérieures à 90 nm les protons ont un pouvoir ionisant insuffisant pour provoquer des effets directs, cependant ils peuvent produire des réactions nucléaires avec les noyaux du matériau. Par contre pour les finesses de gravure de 90 nm et inférieur les protons produisent une ionisation directe.

Les particules α et les ions sont responsables des événements singuliers (voir chapitre 1.3.2.2) en ionisant le matériau traversé par création de paires électrons-trous.

1.3. Les effets des radiations sur les circuits électroniques

1.3.1. La terminologie

Le nombre de particules incidents par unité de surface et de temps est le *flux*, exprimé en particules/cm².s. L'intégration de ce flux sur le temps donne la densité de particules, ou bien *fluence*, donnée en particules/cm². Une particule interagissant avec la matière lui transfère son énergie. La quantité d'énergie déposée ionisation et par unité de longueur de trajectoire est appelée LET (Linear Energy Transfert). Elle est habituellement exprimée en MeV.cm²/mg [Adams 1993]. Par exemple

[2] Réaction nucléaire dans laquelle un noyau atomique est frappé par une particule incidente. Lors de l'impact le noyau se décompose en produisant des jets de particules plus légères (neutrons, protons, etc). Le noyau après la réaction est généralement de masse atomique plus faible que le noyau d'origine.

dans le silicium, une particule ayant un LET de 97 MeV.cm²/mg dépose une charge d'environ 1 pC par micron.

Deux paramètres sont utilisés pour quantifier de façon empirique la sensibilité des circuits intégrés face aux particules. Il s'agit de la *section efficace* et du *seuil de sensibilité* ou charge critique, noté. Ce seuil est relié au LET seuil, noté LET_{th}, qui est le LET minimum qu'une particule doit avoir afin de générer des événements singuliers dans un circuit intégré. Cette valeur est propre au circuit, elle dépend notamment de la technologie de fabrication du circuit. La section efficace, notée σ, est le rapport entre le nombre de perturbations et la fluence reçue par le circuit. Elle peut être exprimée en cm²/composant ou bien en cm²/bit.

La caractérisation d'un circuit intégré est spécifiée par une courbe de section efficace en fonction du LET. Cela permet de déterminer un autre paramètre caractéristique : la section efficace de saturation, notée σ_{sat}, qui représente la surface sensible totale du composant testé.

La Figure 1-6 illustre l'allure typique d'une courbe de section efficace avec ses grandeurs caractéristiques.

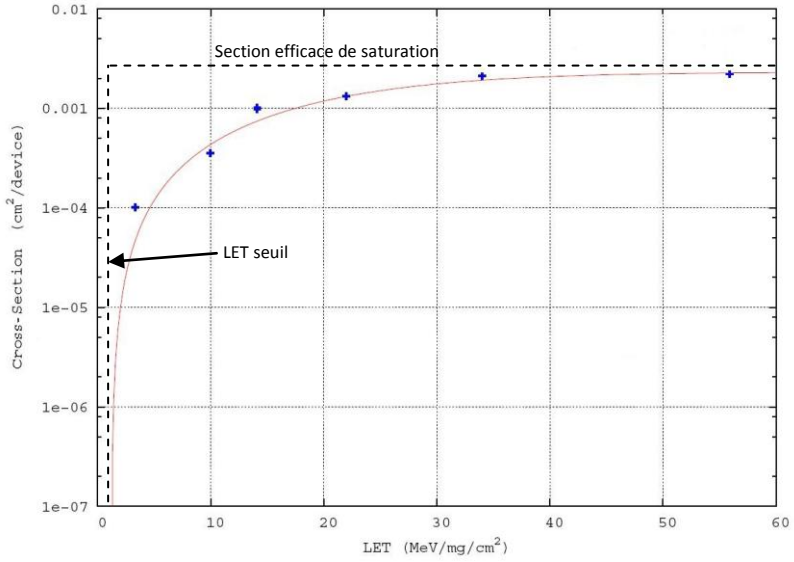

Figure 1-6 : Courbe de section efficace typique

Connaissant le seuil de LET et la section efficace de saturation, il est alors possible de tracer une courbe idéale à l'aide de la distribution de Weibull. La courbe est donnée par l'équation suivante :

$$\sigma = \sigma_{sat}\left[1 - \exp\left(\left(-\frac{LET - LET_{th}}{W}\right)^{s}\right)\right]$$

où W est le paramètre de largeur et S le paramètre de forme. Ils peuvent être déterminés par régression linéaire à l'aide des données obtenues lors des essais sous radiation.

1.3.2. Les différents types d'événements

Il est possible de distinguer deux grandes familles d'effets des radiations sur les composants électroniques : les effets par accumulation comme la dose cumulée et les effets dus à une unique particule aussi appelés événements singuliers.

1.3.2.1. Dose cumulée

L'effet de dose cumulée (TID – Total Ionising Dose) est presque exclusivement imputable aux particules piégées dans les ceintures de radiations et aux protons issus de l'activité solaire. Il s'agit de l'accumulation, dans le temps, de charges au niveau des oxydes isolants dans les circuits intégrés [Mc Lean 1984]. Pour les technologies MOS, cela se traduit par une dérive des paramètres électriques [Bessot 1993-1], comme par exemple le décalage des niveaux de seuils, l'augmentation des courants de fuite. Ces charges sont cumulatives et conduisent à une perte progressive puis totale de la fonctionnalité du composant. Cependant l'effet de dose peut être atténué en protégeant les composants par un blindage. Cette technique est efficace contre les protons de faible énergie et les électrons.

1.3.2.2. Les événements singuliers

Lorsqu'un ion lourd traverse la matière il le fait suivant une ligne droite. Le LET (Linear Energy Transfer) est la grandeur permettant de quantifier l'énergie ionisante cédée à la matière par unité de longueur parcourue et est exprimée en MeV/mg/cm². Lorsque la charge est déposée dans une zone active d'un transistor, elle est collectée par le champ électrique et peut se propager dans le circuit intégré pouvant ainsi provoquer différents effets parasites regroupés sous l'acronyme SEE (Single Event Effects). Ceux-ci peuvent être classés en deux catégories :

Les effets destructifs :

- **SEL** (Single Event Latchup) – mise en conduction d'une structure parasite PNPN (thyristor parasite) résultant en une forte augmentation du courant consommé par le composant, pouvant mener jusqu'à sa destruction par effet Joule. L'effet disparait en arrêtant l'alimentation, il est donc impératif d'inclure des systèmes de protection permettant la détection rapide de ce phénomène pour prévenir la destruction des circuits [Ma 1989]. Dans certains cas cette surconsommation n'excède pas les valeurs maximales spécifiées, on parle alors de microlatchup. De tels événements ont notamment été observés sur certains processeurs [Moran 1995] et microprocesseurs [Buchner 1997].
- **SEB** (Single Event Burnout) – affecte les transistors de puissance en autorisant le passage d'un fort courant, pouvant ainsi causer leur destruction.
- **SEGR** (Single Event Gate Rupture) – Destruction de l'oxyde de grille par la création d'un chemin conducteur dans les MOSFETs de puissance.
- **SHE** (Single-event Hard Error) – un SEU causant un changement irréversible dans un composant. Par exemple le collage d'un bit dans une mémoire.

Les effets non destructifs :

- **SET** (Single Event Transient) – Pic de courant provoqué par une particule et se propageant dans un circuit, pouvant ainsi provoquer une erreur.

- **SEU** (Single Event Upset) – SET qui est capturé par un élément mémoire, provoquant ainsi son basculement [Petersen 1983] [Pickel 1983].
- **MCU** (Multiple Cell Upset) – Une seule particule provoquant le basculement de plusieurs points mémoire géographiquement voisins.
- **MBU** (Multiple Bit Upset) – Cas particulier de MCU où une particule crée plusieurs erreurs dans un même mot logique.
- **SEFI** (Single Event Functional Interrupt) – Une particule provoquant la perte de fonctionnement normal d'un composant. Une réinitialisation du composant est alors généralement requis.

1.3.2.3. Effets directs et indirects des protons

C'est en 1990 que fut observé pour la première fois un SEU produit par un proton. Comme pour les ions lourds, on peut distinguer les effets destructifs comme les SELs et les SEBs, et les effets non destructifs comme les SEUs. Cependant il faut aussi faire la différence entre l'effet direct qui est l'ionisation de la matière traversée par le proton et l'effet indirect qui est l'ionisation produite par l'interaction entre le proton incident et le noyau du matériau.

Les protons engendrent assez rarement un effet direct sur les composants électroniques. Par contre la réaction nucléaire de ces particules avec le Silicium est possible et peut fragmenter le noyau en deux ions de même masse et générer des produits secondaires. Ces ions produits peuvent alors provoquer un SEE.

Suivant le type de composant frappé par une particule, les effets observés ne seront pas les mêmes. En effet, un composant analogique tel qu'un AOP -Amplificateur Opérationnel), ne pourra de toute évidence pas présenter de SEU ou MBU étant donné qu'il est dépourvu de cellules mémoires. C'est donc sous la forme de pics de courant transitoires (SET) que se manifestent les charges déposées par les particules dans les zones actives [10]. Un composant numérique, tel une mémoire, un FPGA, un processeur, etc …, ne permet que rarement l'observation de SET, car ces pics de courant sont le plus souvent capturés par un élément mémoire et se transforment alors en un SEU ou un MBU.

1.3.3. Les moyens de prévention et de protection face aux SEEs

Les moyens de prévention et de protection des circuits intégrés face aux radiations peuvent être appliqués à plusieurs niveaux d'un circuit ou d'un système.

1.3.3.1. Le blindage

Le blindage d'un système a pour but de réduire le flux et l'énergie des particules à l'intérieur d'un satellite. Il s'agit, en général, d'une feuille d'aluminium de quelques dixièmes de millimètres d'épaisseur. Si les électrons sont considérablement ralentis par la présence d'un blindage, l'effet sur les protons les plus énergétiques est moindre [Bourrieau 1991]. De plus les blindages deviennent complètement inefficaces contre les particules issues du rayonnement cosmique qui peuvent avoir des énergies de plusieurs MeV, voire de l'ordre du GeV.

1.3.3.2. Le durcissement des composants

L'amélioration de la finesse de gravure des circuits intégrés apporte de nombreux avantages : augmentation de la densité d'intégration donc diminution des coûts de production, diminution de la consommation et augmentation des performances. Cependant, cela a des conséquences critiques sur la robustesse de ces circuits face aux radiations dont les tendances sont opposées, notamment vis-à-vis des événements multiples comme les MBUs et MCUs. Pour ces raisons de nombreux travaux sont menés pour traiter ce problème de sensibilité aux radiations et deux approches tentent d'apporter des solutions. L'une, au niveau technologique, porte sur le durcissement des technologies de fabrication des composants afin de les rendre plus robustes, l'autre, au niveau conception, se concentre sur la recherche de nouvelles structures et topologies pour les éléments de base des circuits (registres, cellules mémoires, …) mais aussi, à un niveau supérieur, pour les blocs fonctionnels.

La technologie CMOS étant la plus répandue, elle est aussi présente à bord des satellites. Plusieurs technologies ont été développées pour réduire la sensibilité des circuits intégrés CMOS face aux radiations, parmi lesquelles on peut citer :

- **CMOS sur substrat épitaxié** : une mince couche faiblement dopée (donc très résistive) est déposée au dessus du substrat de base, et supporte les transistors MOS. Cela permet de fortement réduire la sensibilité au latchup et à légèrement améliorer la robustesse face aux upsets [Diehl 1983].
- **CMOS sur substrat isolant** : cette technologie, aussi appelée SOI (Silicium On Insulator), supprime la structure parasite PNPN, la rendant ainsi immunisée au latchup. La sensibilité aux upsets est aussi considérablement atténuée. Différentes méthodes existent pour la fabrication de circuits SOI [Musseau 1996] mais leurs coûts, plus élevés que les circuits CMOS standard, les rendent peu répandus parmi les composants COTS dont la tendance actuelle favorise l'utilisation pour les applications satellite.
- **Matériau AsGa** : la vitesse des électrons dans l'arséniure de gallium (AsGa) est deux fois plus élevée que dans le silicium. De plus, Les composants fabriqués avec ce matériau offrent une grande tolérance à l'effet de dose et sont insensibles au latchup dû à l'absence de structure thyristor parasite. Malgré ces avantages ils restent essentiellement réservés à des secteurs de l'industrie bien spécifiques comme les Mosfets, cela à cause de leur coût et de leur consommation élevés [Weatherford 1997].

L'architecture du circuit et la taille des dispositifs jouent un rôle important dans la sensibilité aux SEE du composant. Il existe trois méthodes de durcissement à la conception :

- **Augmentation des dimensions des transistors** : l'allongement de la longueur du canal de jonction permet de diminuer la sensibilité aux radiations. Cependant, cette solution va à l'encontre de la tendance actuelle. Elle peut néanmoins être utilisée pour certain transistors *critiques* du circuit.
- **Atténuation de l'impulsion transitoire de courant induite par une particule** : en première approximation, la charge critique est égale au produit de la tension de seuil de basculement par la capacité des nœuds sensibles. Augmenter une de ces deux valeurs permet d'augmenter la charge critique. Ceci peut être atteint par diffusion de capacité ou bien par l'ajout de résistances de contre réaction. Cependant ces deux solutions ont leurs

inconvénients. L'ajout d'une capacité a pour effet de diminuer les performances dynamiques du composant [Diehl 1983]. D'un autre côté l'utilisation de résistances, afin d'atténuer l'impulsion de courant résultant du passage d'une particule, augmente la tension de seuil [Sexton 1989].

- **Structures bistables durcies** : depuis la fin des années quatre-vingt de nombreuses propositions de durcissement des bascules ont été proposées [Rockett 1988] [Canaris 1991] [Whitaker 1991] [Liu 1992] [Bessot 1993-2] [Velazco 1994] [Salager 1999]. Ces différentes techniques ont en commun l'accroissement de leur complexité comparée aux architectures standard et l'introduction de redondances dans le stockage de l'information. En effet, celle-ci est effectuée sur plus de deux nœuds, comme c'est le cas sur une bascule D classique. La cellule DICE (Dual Interlocked storage CEll), par exemple, stocke l'information sur quatre nœuds et contient un mécanisme de correction du niveau logique du nœud potentiellement perturbé par une particule qui est basé sur les niveaux logiques des autres nœuds [Calin 1996].

1.3.3.3. Le durcissement au niveau système

En plus des moyens de protection face aux radiations décrits précédemment, il est possible d'adopter une approche au niveau système. En voici quatre exemples parmi les plus utilisées :

- **Détection et correction d'erreurs** : cette méthode consiste à ajouter des bits supplémentaires, appelés bits de contrôle, à tout mot mémoire. Différentes techniques permettent de coder ces bits (Hamming, CRC, M parmi N, etc). Elles ont chacune leurs propres caractéristiques concernant le nombre d'erreurs détectées et corrigées par mot, mais elles s'accompagnent aussi d'un surcoût en terme de silicium occupé sur la puce. Un compromis doit donc être trouvé entre la tolérance souhaitée découlant de la criticité de l'application et le coût engendré par la mise en œuvre de telles techniques.
- **La redondance** : elle peut se faire soit de manière matérielle (ou spatiale), soit de manière logicielle (ou temporelle). Dans le premier cas, la méthode consiste à répliquer les ressources du système (processeurs, mémoires, etc). Cela se fait sans perte de performance par contre le coût en terme de place, de poids et de consommation électrique va à l'encontre de la philosophie adoptée pour les systèmes embarqués, en particulier pour des applications spatiales. La méthode la plus utilisée est la triplication modulaire ou TMR proposée par J. Von Neumann et qui consiste à effectuer un vote à la majorité des résultats issus de trois répliques d'un même module [Katz 1994]. La redondance logicielle, qui est l'exécution séquentielle d'une même tâche suivis par un vote, se traduit par une perte de performance sans que la partie matérielle de l'application ne soit modifiée.
- **Le chien de garde** : le chien garde, ou *watchdog* en anglais, est un mécanisme de surveillance. Il peut être matériel ou logiciel et être implanté à différents niveaux (composants, sous-systèmes, cartes électroniques, etc). Son rôle est de vérifier qu'il reçoit un signal dans un intervalle de temps donné. En cas de non réception il effectue alors une opération de récupération (reset, coupure de l'alimentation, envoi d'un rapport d'erreur, etc) [Fucile 1997].
- **Le contrôle à partir du sol** : cette méthode de dernier recours permet de réinitialiser à distance un système défaillant. Un téléchargement régulier des données critiques peut être effectué régulièrement pour prévenir les erreurs [Miller 1994].

Chapitre 2. Méthodes et outils pour le test de circuits intégrés

Les méthodes de prévention citées au chapitre 1.3.3 ne permettent pas, à priori, de garantir une immunité totale aux SEEs, en particulier aux SEUs. Il est donc impératif de pouvoir prévoir le comportement d'une application face à des événements susceptibles de perturber son fonctionnement au cours de sa durée de vie. Pour cela, deux stratégies de test existent : la première servant à quantifier la sensibilité intrinsèque d'un composant et la deuxième permettant d'évaluer un taux d'erreurs réaliste pour une application (composant + son logiciel) donnée. Chacune de ces méthodologies peut être mise en œuvre lors de tests en conditions réelles ou bien lors de tests accélérés au sol. De plus des techniques complémentaires de prédiction de fautes par injections de fautes matérielles/logicielles peuvent être déployées. Enfin une plateforme générique pour le test de circuits intégrés numériques sera présentée.

2.1. Stratégies de test pour la caractérisation des circuits intégrés

Les deux stratégies permettant l'évaluation de la sensibilité d'un composant ont pour but d'obtenir une estimation du nombre moyen de particules nécessaire pour provoquer une faute dans le circuit cible. Connaissant les caractéristiques du milieu dans lequel l'application devra fonctionner, il est alors possible de prédire le taux de pannes. La première stratégie de test, dite *test statique*, à pour but d'obtenir des mesures de sections efficaces pour plusieurs types de particules afin de tracer la courbe de section efficace du composant. La deuxième méthodologie, dites *test dynamique*, est une prédiction obtenue pour une application spécifique.

2.1.1. Test statique

Le test statique est utilisé afin d'évaluer la sensibilité intrinsèque du composant, exprimée grâce à la section efficace. Cette mesure est obtenue à l'aide d'accélérateurs de particules, en exposant le DUT (Device Under test) à un flux de particules.

Une fois les zones sensibles identifiées, c'est-à-dire les points mémoires accessibles à l'utilisateur en lecture et en écriture, un test statique se déroule en trois étapes :

- Initialisation à une valeur connue de toutes les zones sensibles accessibles.
- Irradiation du DUT alors qu'il est sous tension et sans activité (aussi dit en mode *idle* en anglais).
- Lecture des zones sensibles pour détecter les SEUs potentiels.

L'opération est répétée pour plusieurs particules afin d'obtenir une valeur de section efficace pour des LET différents et donc de pouvoir tracer la courbe de section efficace en fonction du LET. Cela permet aux développeurs d'estimer le nombre de SEUs susceptibles de se produire dans le DUT en fonction de l'environnement de fonctionnement.

Le test statique caractérise donc le composant mais en aucun cas l'application s'exécutant sur le composant. Il n'est donc pas suffisant pour obtenir une estimation correcte du taux d'erreurs pour l'application finale car il ne prend en compte que la dimension spatiale, c'est-à-dire le nombre et la nature des ressources utilisées. En effet, la dimension temporelle ne peut pas intervenir dans un test statique du fait de sa nature même. Or une application utilise rarement la totalité des ressources mémoires offertes par un circuit, et celles utilisées ne sont pas forcément considérées comme « critiques » durant toute l'exécution de l'application. Par conséquent, la section efficace statique surestime de manière conséquente la sensibilité d'une application et il convient donc de réaliser des

tests dynamiques sur l'application finale afin d'obtenir des résultats plus proche de la réalité. Pour des composants complexes tels des processeurs, Les études faites à ce sujet ont prouvé que la surestimation peut atteindre 2 ordres de grandeurs [Peronnard 2008].

2.1.2. Test dynamique

Le test dynamique reprend le principe du test statique en y intégrant la notion de « fenêtre temporelle de sensibilité ». En effet lors de l'exécution d'une application, le contenu des cellules mémoires n'est généralement pas critique à chaque instant. La Figure 2-1 décrit les fenêtres de sensibilité pour deux points mémoire P1 et P2. Le contenu de ces cellules est critique uniquement entre sa phase d'écriture et sa phase relecture. Par contre si une particule modifie le contenu lorsque le point mémoire n'est pas initialisé ou bien entre une lecture et une écriture, cela n'aura pas d'effet sur l'application qui vient de toute manière écraser le contenu de la cellule par la suite.

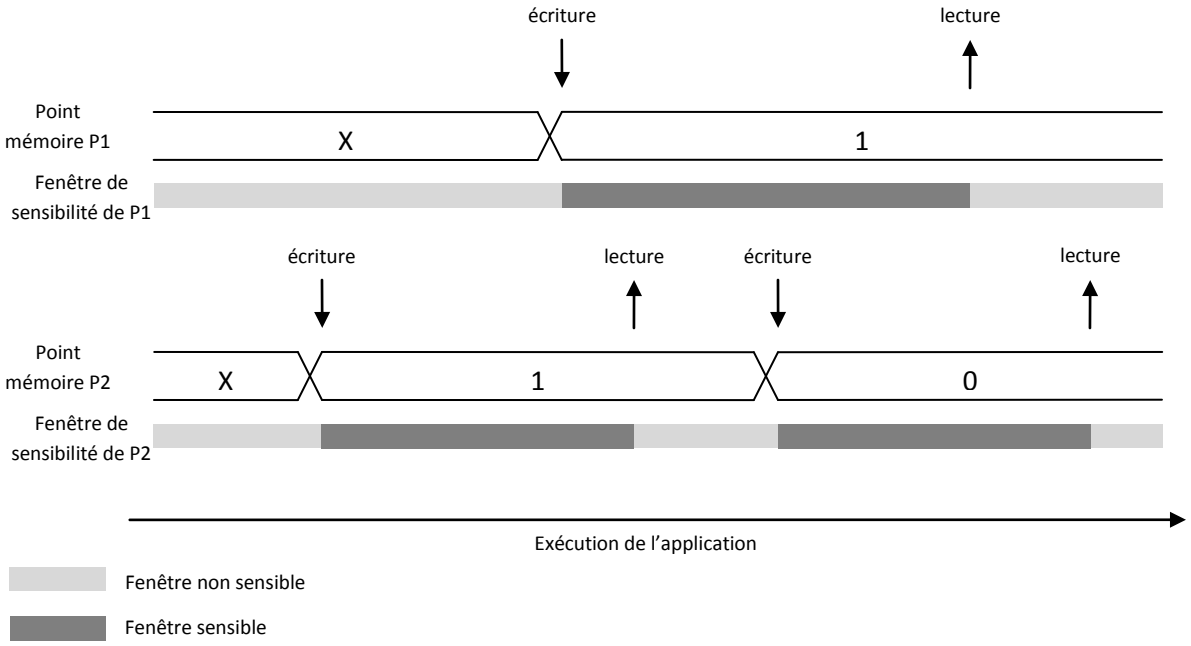

Figure 2-1: Fenêtre de sensibilité de deux points mémoire

A l'issue du test on obtient des données statistiques sur le nombre moyen de particules nécessaires afin d'obtenir une erreur observable sur les sorties de l'application. Il est alors possible d'extrapoler ce nombre pour le système fonctionnant en environnement réel, et ainsi d'en déduire le nombre d'erreurs susceptibles d'apparaître par unité de temps (jour, mois, année, …).

2.2. Tests en environnement réel

Les essais en conditions réelles consistent à exploiter le composant sous test dans un environnement naturel. Pour cela, il peut être mis en œuvre soit dans un environnement spatial, en haute-altitude ou bien au sol. Dans tous les cas, ces expériences requièrent de mettre en œuvre un grand nombre de composants pour compenser la densité de particules relativement faible de l'environnement naturel, comparé aux flux que l'on trouve en accélérateurs de particules (voir le

chapitre 2.3.2). Si ces tests permettent d'obtenir des résultats réalistes, ils comportent aussi un certains nombres de contraintes et d'inconvénients.

2.2.1. Essais en orbite

Il est implicite que ce type d'expérience fournisse des résultats réalistes étant donné que le composant est testé dans les conditions qui seront proches de celles de l'application finale. Les résultats de telles expériences sont disponibles dans la littérature [Falguere 1994], [Duzellier 1997]. Les agences spatiales mettent régulièrement en place des projets afin de tester en ambiance spatiale sévère des nouveaux composants et des nouvelles technologies. Le laboratoire TIMA a déjà participé, en fournissant des cartes d'expériences, au deux projets suivant :

- Le projet **MPTB** (Micro-electronic and Photonic TestBed) [Web MPTB] a été dirigé par le NRL (Naval Research Lab.) [Web NRL] et les opérations en orbite ont commencé en novembre 1997 et il était composé de 24 expériences [Ritter 1997]. Les résultats obtenus dans le cadre de ce projet ont notamment donné lieu aux références suivantes : [Cheynet 1999], [Duzellier 2002], [Barak 2000], [Campbell 2002]. L'expérience développée par le laboratoire TIMA concernait l'étude de réseaux de neurones artificiels [Velazco 1998], [Velazco 1999].
- Le projet **STRV** (Space Technology Research Vehicles) [Web STRV], composé de deux satellites, a été dirigé par le DRA britannique (Defence Research Agency). Les deux engins spatiaux furent placés sur une orbite GTO (Geostationary Tranfer Orbit) en juin 1994. Il donna lieu notamment aux publications suivantes : [Langenbacher 1996], [MacKay 1997], [Daly 1999]. Dans le cadre du projet STRV, le laboratoire TIMA a développé une expérience sur la logique floue. Cependant, suite à un incident technique survenu en orbite, la carte n'a pas pu fonctionner et fournir de résultats.

Si les résultats issus de l'exposition du composant cible à un environnement réel spatial sont les plus représentatifs, ils sont aussi les plus longs, difficiles et coûteux à obtenir. En effet entre le lancement du projet et le lancement de l'engin spatial il s'écoule généralement plusieurs années (trois à cinq ans). Ensuite, compte tenu que les faibles flux naturels de particules et de la surface sensible exposée relativement restreinte, dû aux contraintes de l'expérience embarquée, l'obtention de résultats nécessite aussi plusieurs mois, voire plusieurs années. Ce type d'essais peut donc être envisagé comme démonstrateur des capacités du composant.

2.2.2. Essais en haute altitude

Les essais en haute altitude peuvent avoir lieu sur des durées courtes, expériences embarqués à bord d'avions, ou bien pour des longues durées grâce à des ballons stratosphériques. L'objet de ces expériences peut être la caractérisation du milieu pour établir ou valider des modèles [Normand 2001], ou bien encore pour étudier les effets sur les équipements et les personnels [Chee 2000] [Taber 1993] [Olsen 1993] [Goldhagen 2000] [Sohn 2000].

2.2.3. Essais au sol

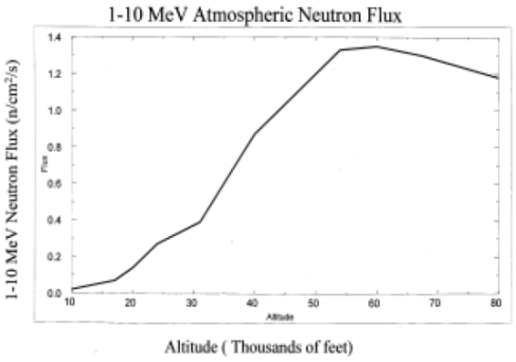

Figure 2-2 : Densité du flux de neutrons atmosphériques en fonction de l'altitude

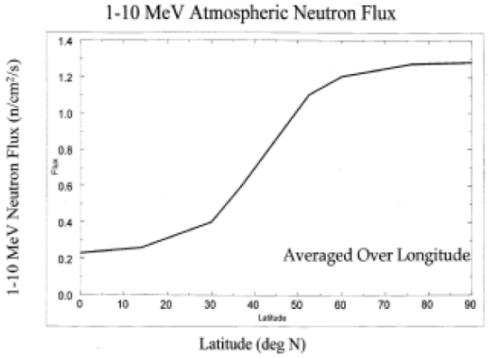

Figure 2-3 : Densité du flux de neutrons atmosphériques en fonction de la latitude

L'expérience Rosetta [Lesea 2005] menée par Xilinx est un exemple de la prise de conscience des fabricants concernant le problème des effets des radiations sur les circuits intégrés. Cette expérience consiste à effectuer des tests statiques en ambiance naturelle sur un grand nombre de composants pour augmenter la surface sensible et ainsi compenser le faible flux de particules au sol. Des plateformes de test ont été installées sur plusieurs sites afin d'obtenir des résultats pour plusieurs altitudes et plusieurs latitudes étant donné que la densité de neutrons n'est pas uniforme [Barth 1997] sur toute la surface terrestre (Figure 2-2 et Figure 2-3).

Tableau 2-1 : Lieux d'implantation des expériences Rosetta avec leur altitude

Structure d'accueil	**Localisation**	**Altitude** (m)
Xilinx SJ	San Jose, CA, USA	0
Xilinx ABQ	Albuquerque, NM, USA	1554
WMRS[3]	White Mountain, CA, USA	750
CSO[4]	Mauna Kea, Hawaï, USA	975
IM2NP[5]	Marseille, France	124
IRAM[6]	Pic de Bure, France	2552
LSBB[7]	Rustrel, France	-550

La liste des sites où est installée l'expérience Rosetta est donnée dans le Tableau 2-1.

Chaque banc de test est composé de cents composants (Figure 2-4) et un site peut abriter plusieurs de ces bancs.

[3] White Mountain Research Station
[4] Caltech Submillimeter Observatory
[5] Institut Matériaux Microélectronique Nanoscience de Provence
[6] Institut de Radioastronomie Millimétrique
[7] Laboratoire Souterrain Bas Bruit de Rustrel-Pays d'Apt

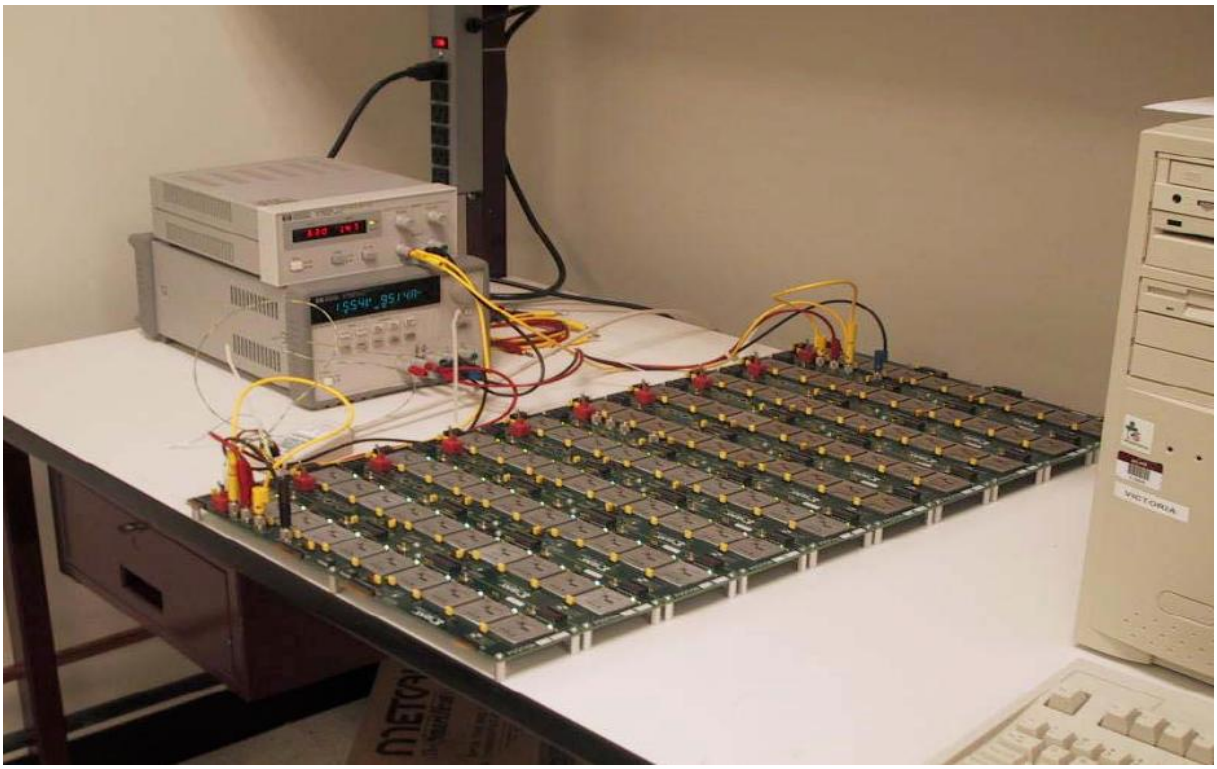

Figure 2-4 : Banc de test de l'expérience Rosetta

2.3. Tests accélérés au sol

Comme leur nom l'indique, les tests accélérés permettent d'obtenir des résultats beaucoup plus rapidement que ce que l'on peut attendre de tests en conditions réelles. Pour cela on fait appel à des machines générant des flux de particules bien supérieurs à ce que l'on peut trouver dans la nature. Ces flux « artificiels » peuvent être issus de diverses plateformes telles que des sources radioactives, des accélérateurs de particules ou bien des faisceaux laser.

2.3.1. Sources radiatives

L'utilisation d'une source radioactive est une méthode peu coûteuse et qui fournie des informations préliminaires sur la sensibilité d'un composant. Le Californium 252, par exemple, émet des particules alphas, deux types d'ions lourds possédant respectivement des LETs de 45 et 46 MeV.cm²/mg ainsi que des neutrons en faible quantité. Une autre source couramment utilisée est l'Américium 242 qui produits des particules alpha ayant une énergie de 5 MeV (LET = 0,6 MeV/mg/cm²).

Cependant cette source génère des particules de faible énergie par rapport à celles que l'on peut trouver en environnement spatial ou bien en accélérateur de particules. Ceci influence directement la distance de pénétration de la particule qui se retrouve ainsi limitée à un parcours dans la matière d'environ 15 µm. Cette distance peut être insuffisante pour des composants possédant des couches superficielles importantes, et cela même en pratiquant un amincissement.

Néanmoins cette méthode de test permet, lorsque le composant l'autorise, de valider la plateforme matérielle et logicielle de test avant de pratiquer à des essais en accélérateurs de particules qui sont plus coûteux et qui ne laissent gère de place l'incertitude.

2.3.2. Accélérateurs de particules

Différents types d'accélérateurs de particules (accélérateur linéaire, cyclotron, etc…) peuvent produire un large éventail de faisceaux telles que des ions lourds, des protons et des neutrons. L'avantage de ces installations est la taille du faisceau (quelques centimètres) qui permet l'irradiation de toute la surface d'une puce en une seule fois. Cependant, pour la plupart des faisceaux, les essais aux ions lourds doivent se dérouler à l'intérieur d'une enceinte sous vide. Ceci impose des contraintes, qui doivent être prises en compte lors du développement de la plateforme, sur les connectiques vers l'extérieur de l'enceinte.

La reproduction exacte des espèces d'ions lourds présents dans l'environnement spatial n'est pas réalisable avec des équipements au sol à cause de la diversité des particules que l'on y trouve, de leur haute énergie et de leur incidence omnidirectionnelle. Des accélérateurs « faible » énergie sont donc utilisés afin de reproduire un dépôt de charges équivalente en jouant sur le paramètre de LET des particules. Ces particules issues d'accélérateurs sont généralement peu pénétrantes, ce qui ne pose pas de problème pour les composants dont les zones actives sont accessibles par la face avant. Mais ceci n'est pas le cas avec le Virtex-II monté en boitier *flip-chip*. Les particules doivent traverser le substrat de la puce, il faut donc procéder à un amincissement de ce substrat.

La réalisation des tests en accélérateur de particules se sont déroulés dans les installations HIF (Heavy ion Irradiation Facility) de l'UCL[8] (Université Catholique de Louvain). La machine est capable de produire deux « cocktails » de particules regroupées en fonction de leurs caractéristiques en terme d'énergie des particules et de pénétration dans la matière. Etant donné le boitier du DUT étudié dans ces recherches, c'est le cocktail haute pénétration qui doit être utilisé. Le Tableau 2-2 donne la liste des ions disponibles avec leurs caractéristiques.

Tableau 2-2: Caractéristiques des ions présents dans le cocktail haute pénétration

Ions	Energie (MeV)	Pénétration (µm Si)	LET (MeV/mg/cm²)
$^{13}C^{4+}$	131	266	1,2
$^{22}Ne^{7+}$	235	199	3,3
$^{40}Ar^{12+}$	372	119	10,1
$^{58}Ni^{17+}$	500	85	21,9
$^{83}Kr^{25+}$	756	92	32,4

2.3.3. Les faisceaux lasers

Alors que les accélérateurs de particules apportent une approche globale à la caractérisation des composants vis-à-vis des SEEs, ils ne peuvent fournir aucune information sur la localisation des fautes

[8] Situé à Louvain-la-Neuve en Belgique.

détectées. Par contre, le laser allie un faible diamètre de faisceau (de l'ordre du micron) et une précision de déplacement submicronique, il est ainsi possible d'effectuer une cartographie précise de certaines zones sensibles.

Malgré ses avantages, le test à l'aide d'un faisceau laser est soumis à deux contraintes :

- Le faisceau est réfléchi par les couches de métallisation, et cela pose un réel problème pour les composants complexes dotés de plusieurs niveaux de métallisation. Il est alors indispensable de procéder à des attaques par la face arrière.
- L'autre limitation du laser est que l'énergie du faisceau ne peut pas être corrélée avec l'énergie des ions lourds [Pouget 2001].

Par conséquent les tests à l'aide de faisceaux laser sont complémentaires aux mesures en accélérateurs de particules grâce à leur précision de positionnement du faisceau et la possibilité d'effectuer des tirs « coup par coup ».

2.4. Méthode de prédiction par injection de fautes matérielle/logicielle

Cette méthodologie fut mise au point pour palier le problème du test dynamique qui impose une nouvelle campagne de test sous radiations pour chaque application, ou même après chaque modification d'une application. Réaliser de tels essais est fort coûteux en terme de moyens et de temps. La prédiction par injections de fautes se base sur les résultats du test statique, c'est-à-dire la courbe de section efficace, qui caractérise uniquement le composant et est donc indépendante de l'application. Le but de l'injection de fautes est de simuler l'effet d'un SEU en faisant basculer de manière logicielle un point mémoire au cours de l'exécution de l'application.

2.4.1. Présentation de la méthode CEU

La méthode d'injection de fautes CEU (Code Emulated Upsets), présentée pour la première fois en 2000 [Velazco 2000], permet d'obtenir une prédiction réaliste de la sensibilité d'une application, c'est-à-dire un composant de type processeur ou microprocesseur accompagné de son logiciel.

Supposons qu'une méthode permettant de provoquer des inversions de bits, aléatoires dans l'instant d'occurrence et la cible affectée, puisse être implémentée sur un composant programmable cible de type processeurs. Si ces inversions de bits sont concurrentes avec l'exécution d'un programme, alors elles peuvent être assimilées à des SEUs et un taux d'erreur, appelé τ_{inj}, peut être dérivé du rapport entre le nombre d'erreurs d'application observé et le nombre de SEU injectés.

$$\tau_{inj} = \frac{\#\ Erreurs}{\#\ de\ SEU\ injectés}$$

Par conséquent, τ_{inj} peut être interprété comme le nombre moyen d'inversions de bits requis pour provoquer une erreur dans l'application. De plus, sachant que la section efficace, σ_{SEU}, issue d'un test statique, effectué au moyen d'accélérateurs de particules, est l'expression du nombre moyen d'un type donné de particules nécessaire pour provoquer le basculement du contenu d'une des cellules mémoires du composant testé, la sensibilité aux SEUs d'une application, τ_{SEU}, peut alors être obtenue grâce au produit de la section efficace statique par le taux d'erreur issue de la prédiction par injection de fautes.

$$\tau_{SEU} = \sigma_{SEU} * \tau_{inj}$$

Concrètement, la méthode CEU consiste à exploiter une interruption externe du processeur afin d'interrompre aléatoirement le flot normal d'exécution d'un programme. La routine d'interruption alors appelée se charge de modifier un bit d'un mot mémoire du composant (registre, mémoire cache, mémoire interne, …) choisi lui aussi aléatoirement. De retour de la routine d'interruption, l'exécution du programme reprend son cycle normal. L'application et ses sorties sont observées afin de détecter la propagation de l'erreur injectée, erreur qui peut se traduire par des résultats faux, une perte de séquencement, …

La principale difficulté de cette approche, appliquée aux processeurs, réside dans l'émulation de SEUs qui doit refléter le plus possible les conséquences d'une particule traversant une zone sensible du composant. La précision de la prédiction du taux d'erreur comparé à une mesure effectuée en accélérateur de particules dépend fortement du nombre de points mémoire présent dans le composant et qui ne sont pas accessible par le jeu d'instruction du processeur. Une étude [Peronnard 2008] mettant en œuvre la méthode CEU à permis de vérifier que les résultats issus de la méthode de prédiction sont très proches de ceux obtenus en accélérateur de particules (ions lourds et protons), cela même pour des circuits complexes comme le Power PC 7448 et des programmes issus d'applications spatiales réelles.

2.4.2. L'injection de fautes sur FPGA à base de mémoire SRAM

Sur les composants de type FPGA à base de mémoire SRAM, composants au cœur de cette thèse, il est possible d'effectuer l'injection de fautes à trois niveaux différents :

- **Dans la mémoire de configuration** : deux alternatives sont envisageables. La première consiste à injecter la faute directement dans le fichier de programmation du FPGA. Cela est possible du fait de la nature rémanente des SEUs dans la mémoire de configuration (voir chapitre 3.1.4.2). Dans la deuxième méthode, le FPGA est configuré avec un fichier sain puis l'exécution de l'application est interrompue aléatoirement pour mettre le FPGA « en veille ». C'est-à-dire qu'il est toujours alimenté mais toutes ses fonctions logiques sont arrêtées. Il est alors possible de pratiquer une relecture de la configuration du composant, grâce à la fonction de *readback* décrite au chapitre 3.1.3.5, sans risquer de corrompre les données contenues dans la mémoire utilisée par l'application alors que celle-ci tenterait d'y accéder simultanément. Le FPGA est ensuite reconfiguré avec les données qui viennent d'être lues mais avec l'inversion d'un bit choisi aléatoirement parmi tous ceux de la configuration. Enfin le FPGA est « remis en ligne » et l'exécution de l'application reprise ou elle s'était interrompue.
- **Dans la description RTL** (Register Transfer Language) : en modifiant directement le code source VHDL ou bien Verilog de l'application.
- **Dans l'application** : à condition que l'application soit de type processeur et ait au moins une interruption externe, l'injection de fautes peut alors se faire directement par l'application dans les registres et mémoires caches du processeur grâce à la méthode CEU décrite dans le chapitre 2.4.1. Il s'agit donc à l'origine d'une technique développée pour les ASICs puisqu'elle agit uniquement sur les ressources utilisateur. Les limitations de cette technique sont la précision de l'instant d'injection et l'impossibilité d'accéder à l'ensemble des cellules mémoires. En effet un SEU peut survenir à n'importe quel instant, alors que le signal

d'interruption est lui pris en compte uniquement lorsque l'exécution de l'instruction en cours est terminée. De plus cette méthode ne permet d'injecter des fautes que dans les registres accessibles depuis le jeu d'instructions processeur.

2.5. THESIC+ : une plateforme de test générique pour composants numériques

L'activité test de composants en environnement radiatif au sein de l'équipe TIMA a fait naître le besoin de s'équiper d'un testeur capable :

- De s'interfacer avec un grand nombre de composants numériques (mémoires, processeurs, FPGAs…)
- De s'adapter aux différentes installations de test au sol (accélérateur de particules, banc laser, exposition à l'environnement naturel…)

Il doit aussi faciliter la mise en place du test en diminuant les temps de développement matériel et logiciel et en réduisant les coûts.

C'est dans cette optique que la plateforme THESIC+ (Figure 2-5), développée au laboratoire TIMA [Faure 2002], a été mise au point spécialement pour le test de composants sous faisceau de particules afin de mettre à disposition une large palette de fonctionnalités.

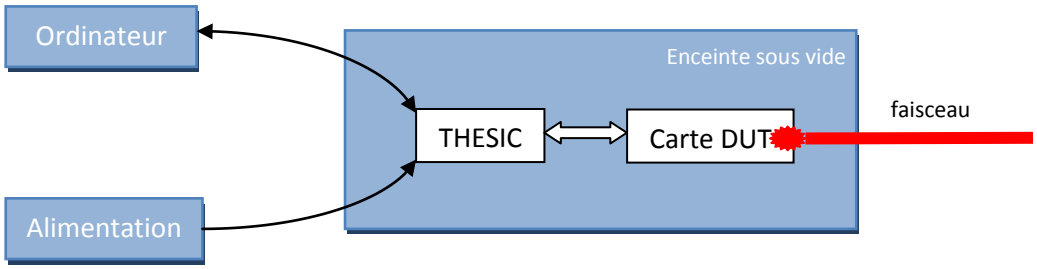

Figure 2-5: Représentation schématique de la plateforme THESIC+ et de ses périphériques

Le testeur THESIC+ est architecturé autour de deux FPGAs (Figure 2-6). Le premier FPGA, appelé FPGA COMM, embarque un processeur LEON qui prend en charge les communications avec l'ordinateur. Les données sont transférées via une interface Ethernet 100Mbits. Le deuxième FPGA, appelé FPGA DUT, permet l'interfaçage entre les ressources mémoire du testeur et le composant à tester.

Le DUT a sa propre alimentation indépendante du testeur et un circuit de protection contrôle en permanence la consommation pour détecter des éventuels latchups et ainsi éviter d'endommager le DUT en coupant rapidement son alimentation.

Enfin, une librairie de fonctions permettant la gestion de la communication avec le testeur est à disposition des développeurs.

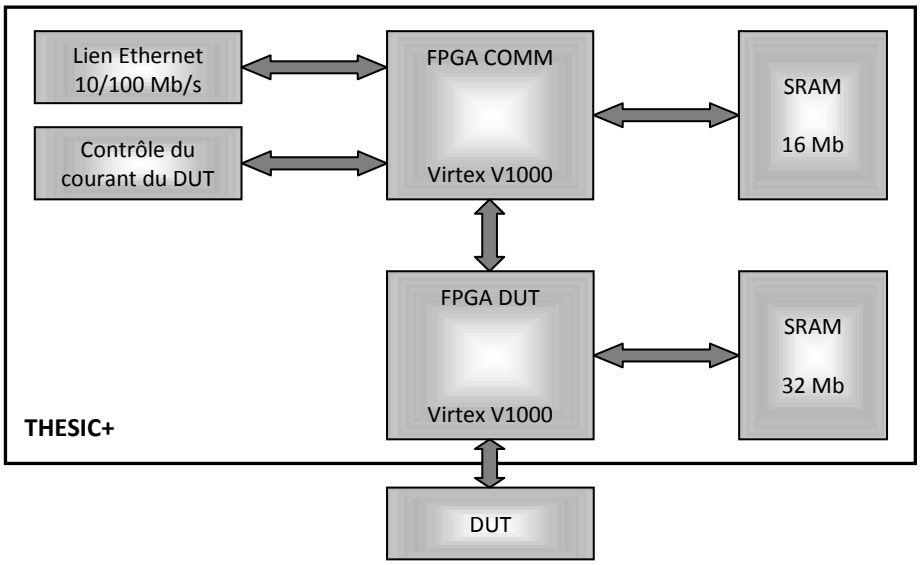

Figure 2-6 : Architecture du testeur THESIC+

Grâce à cette architecture l'effort de développement matériel se réduit à créer une carte fille embarquant les éventuelles alimentations requises par le DUT et à connecter ses entrées/sorties sur le connecteur de THESIC+. L'aspect logiciel consiste à décrire en VHDL ou Verilog l'interface entre le DUT et le testeur.

Chapitre 3. Etude de la sensibilité aux radiations du FPGA Virtex-II

Dans ce chapitre est présenté le composant cible utilisé pour les travaux effectués au cours de cette thèse ainsi que les effets des radiations spécifiques à ce type de composant reprogrammable. La plateforme de test est par la suite décrite. Les différentes campagnes de test, au laser, aux ions lourds et les injections de fautes matérielles/logicielles, sont ensuite décrites et les résultats sont exposés. Enfin une confrontation entre les résultats issus de mesures sous radiations et les prédictions sont confrontées.

3.1. Le composant cible : le FPGA Virtex-II

3.1.1. Les composants programmables de type FPGAs

Les FPGAs, nés au milieu des années 80, sont des composants VLSI (Very Large Scale Integration) entièrement configurables. Comme illustré dans le Tableau 3-1, ces circuits sont à mi-chemin entre les processeurs généralistes et les ASICs (Application-Specific Integrated Circuits).

Tableau 3-1: positionnement des FPGAs par rapport au marché du traitement de données

processeurs			**circuits**	
généralistes		**spécifiques**	**programmables**	**Dédiés**
classiques	**Micro-contrôleurs**	**DSP**	**FPGA**	**ASIC**
Pentium PowerPC Alpha Mips …	80C51 68HC11 …	ADSP-21xx TMS320xx DSP56xx …	Xilinx Altera Actel atmel …	Puces dédiées

Les FPGAs sont typiquement utilisés dans les cas suivants :

- Pour le prototypage rapide : la reconfigurabilité assure une mise au point plus rapide que la simulation et des résultats plus proches de la réalité.
- Lorsque les petites séries ou les coûts de fabrication d'un ASIC ne peuvent pas être amortis. De plus le FPGA assure une mise rapide sur le marché.
- Lorsqu'un produit nécessite la modification de son application afin de le mettre à jour ou bien pour lui ajouter des fonctionnalités au cours de sa durée de vie. Cela est possible grâce à la reconfiguration.
- Pour l'accélération de certains calculs.

3.1.2. Les différentes familles de FPGAs

Il existe actuellement trois grandes familles de FPGAs offrant chacune des avantages et des inconvénients.

- **FPGAs à base d'anti-fusible** : la mémoire de configuration de ces FPGAs est constituée de fusibles. Lors du processus de configuration les fusibles sont laissés intacts ou bien sont détruits par l'application d'une tension adéquate, ceci afin de déconnecter deux éléments. La programmation est définitive et non-réversible, ce qui rend ces composants attrayants pour des applications destinées opérer en milieu radiatif. Par conséquent ils ne peuvent pas être

reprogrammés. Parmi les trois familles, celle-ci est celle qui offre les capacités les plus faibles en terme de ressources logiques. Les deux principaux acteurs sur ce segment de marché sont Actel et Lattice.

- **FPGAs à base de mémoire SRAM** : la configuration du composant est stockée dans une mémoire SRAM. Etant donné sa nature volatile, ces composants doivent être reprogrammés à chaque mise sous tension. Par conséquence les données de configuration sont stockées dans une mémoire externe non-volatile qui est lue par le FPGA au démarrage. L'avantage de cette technologie est sa densité d'intégration, ce qui permet de produire des composants de forte capacité tout en maintenant des prix bas. De plus, la facilité et la rapidité de programmation en font des outils appréciés par les développeurs. Par contre la sensibilité intrinsèque des cellules mémoire SRAM [Baumann 2005] face aux radiations les rend inadaptés pour une utilisation dans les applications spatiales sans prendre au préalable un certain nombre de précautions. Les principaux constructeurs sont Xilinx, Altera et Atmel.
- **FPGAs à base de mémoire Flash** : la mémoire de type flash est la technologie au croisement du fusible et de la SRAM. En effet elle allie la non-volatilité des fusibles et la reprogrammabilité de la SRAM. Un tel composant ne nécessite pas de mémoire externe pour stocker sa configuration. Les progrès effectués sur les mémoires flash durant ces dernières années permettent d'obtenir des densités d'intégration offrant une alternative intéressante aux FPGAs à base de SRAM, sans pour autant égaler ces derniers. Enfin la mémoire flash n'est pas aussi sensible aux *upsets* comme l'est la SRAM. Par contre des problèmes de claquage des oxydes minces ont pu être observés. Actel et Lattice proposent des composants basés sur cette technologie.

3.1.3. Le choix du composant candidat

Au début des travaux de cette thèse il fut décidé que le composant cible serait un FPGA à base de mémoire SRAM car c'était la technologie qui offrait le meilleur rapport performance/capacité/coût, ce qui en faisait un candidat attrayant pour les applications spatiales. Les FPGAs à base d'anti-fusible sont certes robustes mais leur coût est élevé du fait des faibles volumes de production et des multiples étapes de certification de chaque composant. D'autre part, ils offrent une capacité, en termes de ressources disponibles, bien inférieure à celles proposées par la SRAM ou la mémoire flash. Quant à la technologie flash, les produits disponibles à l'époque étaient loin d'être aussi matures que leurs homologues à base de SRAM.

3.1.3.1. Description

Le composant utilisé comme véhicule de test dans le cadre de ces travaux est un FPGA à base de mémoire SRAM Virtex-II XC2V1000 de la société Xilinx. Cette famille de composants est fabriquée en technologie CMOS 150 nm à 8 couches de métallisation avec des transistors haute vitesse gravés en 120 nm. La puce est montée en flip-chip, c'est-à-dire le substrat vers le haut, dans un boitier BGA (Ball Grid Array) métallique au pas de 1.00 mm et doté de 896 billes. Les boitiers de type flip-chip offrent un plus grand nombre d'entrées/sorties et une meilleure dissipation thermique que leurs homologues à base de wire-bonding (câblage par fil). Un autre avantage de ce type de boîtier pour notre étude est sa facilité d'ouverture, permettant une mise à nu aisée de la puce.

3.1.3.2. L'architecture

Une description de l'architecture du composant est donnée en annexe A.

3.1.3.3. Les caractéristiques

Les principales caractéristiques du FPGA Virtex-II XC2V1000 sont les suivantes :

- 1280 CLBs (Configurable Logic Blocks) organisées en une matrice de 40 x 32 éléments.
- 5120 slices. Chaque CLB contient 2 slices et une slice est composée, entre autre, de deux registres, deux LUTs (Look-Up Tables) et de multiplieurs.
- 160 kbits de mémoire RAM dans les CLBs.
- 40 multiplieurs 18 bits x 18 bits.
- 40 blocs de mémoire selectRAM soit une capacité maximale de 720 kbits.
- 8 blocs DCM (Digital Clock Manager).
- 432 ports d'entrées/sorties disponibles pour l'utilisateur.

3.1.3.4. Mémoire de configuration et mémoire utilisateur

Un FPGA à base de mémoire SRAM contient deux mémoires distinctes :

- **La mémoire utilisateur** est celle qui est disponible pour les applications. Elle se présente sous la forme de blocks selectRAM, de registres tels que ceux présents dans les slices et éventuellement des générateurs de fonctions (LUT) configurées comme des mémoires RAM.

- **La mémoire de configuration** sert uniquement à configurer les ressources du FPGA. C'est une matrice de cellules SRAM disposées de manière à ce que chaque cellule soit à proximité de la ressource qu'elle configure. Cette mémoire ne peut pas être utilisée comme ressource utilisateur dans une application.

Cependant ces deux mémoires ne sont pas physiquement dissociées car la mémoire de configuration permet d'initialiser le contenu des cellules mémoires utilisateur. La mémoire de configuration donne donc accès à absolument toutes les ressources du FPGA.

3.1.3.5. Configuration et Readback

Comme décrit précédemment les FPGAs à base de mémoire SRAM doivent être reconfigurés à chaque démarrage, pour cela les données sont lues depuis une mémoire externe. La famille des Virtex possède trois interfaces de configuration dont deux proposent deux modes soit un total de cinq méthodes de configuration :

- Le JTAG (Joint Test Action Group) qui est une interface série bien connue dans le monde du test de composants.
- L'interface série qui peut être configurée, soit en mode maître (le FPGA impose l'horloge à la source de données), soit en mode esclave (une source externe impose l'horloge au FPGA).
- L'interface SelectMap est une interface parallèle de 8 bits. Comme l'interface série, elle peut être configurée en maître ou en esclave.

L'interface retenue tout au long de ces travaux est une interface parallèle sur 8 bits nommée SelectMap. Elle est utilisée en mode *esclave*, c'est-à-dire que le FPGA reçoit l'horloge servant à cadencer la configuration depuis une source externe. D'une part elle offre la plus grande rapidité de configuration. Ceci est déterminant car tous les essais sous faisceau de particules et les sessions d'injection de fautes vont être basés sur des opérations de configuration et de relecture de la mémoire du FPGA. De plus cette interface est facile à mettre en œuvre.

La fonction dite de *Readback* permet la lecture du contenu de la configuration du FPGA sans en altérer le contenu. Elle s'effectue avec la même interface que la configuration. Les conséquences des particules frappant la mémoire de configuration peuvent être observées grâce à cette fonctionnalité.

3.1.4. Impact d'un événement singulier sur un FPGA à base de SRAM

3.1.4.1. Identification et localisation des zones sensibles

Le comportement face aux effets des fautes radiatives d'un nouveau composant débute généralement par l'identification des zones sensibles, c'est-à-dire les points mémoire. Si on considère une même application implantée dans un ASIC et dans un FPGA (occupé à 100%), alors ces deux composants auront en commun les mêmes cibles potentielles : la mémoire utilisateur. Par contre, il ne faut pas oublier que le FPGA possède en plus, et c'est la grosse différence par rapport à l'ASIC, une mémoire de configuration souvent de taille importante par rapport aux autres mémoires. Enfin, le FPGA est doté d'une logique de contrôle de configuration (CCL) dont la composition n'est qu'en partie documentée. Il est ainsi possible de savoir qu'elle est, entre autres, composée de quelques registres qui sont accessibles à l'utilisateur. Les éventuelles autres ressources non documentées n'étant de toute manière pas accessibles leur sensibilité ne peut de toute manière pas être mesurée. La Figure 3-1 donne une vue schématique des zones sensibles.

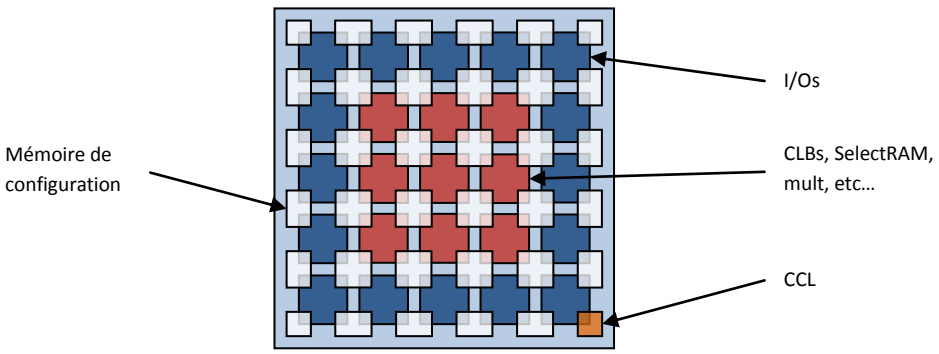

Figure 3-1 : vue schématique du FPGA

Il y a donc dans le Virtex-II deux grandes catégories de cibles potentielles. D'une part la mémoire utilisateur qui se compose comme suit :

- La mémoire SelectRAM : 40 blocks de 512 x 36 bits, soit 737.280 bits.
- Les registres à l'intérieur des slices : 5120 slices * 2 registres = 10.240 bits.
- Les LUTs à l'intérieur des slices : 5120 * 2 tableaux de 16 bits = 153.600 bits.

Soit au total une capacité de 901.120 bits pour la mémoire utilisateur.

Et, d'autre part, la mémoire de configuration organisée en 1104 trames de 106 mots (32 bits) chacune, soit 3.744.768 bits. Cependant, la mémoire utilisateur est configurée depuis la mémoire de configuration, donc en supposant que le contenu d'une cellule mémoire est configuré par un seul bit la mémoire de configuration est donc de 2.843.648 bits (mémoire de configuration – mémoire utilisateur).

Sur un total de 3.744.768 cibles potentielles, la mémoire utilisateur représentant 24% du nombre total de bits.

3.1.4.2. Fautes transitoires, rémanentes et permanentes.

Un upset dans la mémoire de configuration ne donnera pas le même type d'erreur qu'un SEU dans la mémoire utilisateur. En effet, un SEU dans cette dernière se comportera, de la même façon que pour un ASIC, comme une faute qui pourra se propager dans le circuit et éventuellement se manifester sur une sortie en donnant un résultat faux. C'est ce que l'on appelle une **faute transitoire** car elle peut être éliminée lorsque le point mémoire est réécrit par l'application, ou bien en réinitialisant l'application.

D'un autre côté, les fautes dans la mémoire de configuration sont dites **rémanentes,** car elles ne peuvent pas disparaître. Ceci signifie que même en effectuant une réinitialisation de l'application le résultat fourni sera toujours faux. Il est nécessaire de reconfigurer le composant pour éliminer une faute rémanente.

Enfin, le composant peut être physiquement endommagé, la faute se traduisant ainsi par un collage à la valeur logique « 0 » ou « 1 » de la cellule mémoire. L'origine de ce phénomène est encore mal connue. Dans ce cas une reconfiguration du composant ne règle pas le problème. Une « guérison » en température peut être tentée mais sans garantie de résultat et un remplacement du circuit est à prévoir. C'est que l'on appelle une faute **permanente**.

3.1.5. Conséquences sur l'application d'un SEU dans la mémoire de configuration

Un SEU peut intervenir dans une ressource non utilisée par l'application et dans ce cas cela n'aura aucun impact direct sur son fonctionnement. Par contre si le basculement d'état logique modifie une ressource utilisée, cela peut modifier le comportement de l'application en fonction de la nature de la ressource et de la façon dont elle a été modifiée.

Par exemple, pour une connexion les effets d'un SEU peuvent être les suivants. Si la connexion est établie à l'état initial, alors il faut considérer quatre mutations possibles : la connexion peut être modifiée, supprimée, ajoutée ou bien rester inchangée. Par contre, si la connexion est absente à l'état initial, les deux seules mutations sont la création d'une nouvelle connexion ou bien aucune modification. Donc sur six mutations, deux n'ont aucun effet, deux peuvent potentiellement influencer l'application (la création et l'ajout) si la nouvelle connexion se fait avec un signal utilisé et enfin deux affectent forcement l'application (la modification et la suppression).

Un SEU peut ne pas avoir d'influence sur les interconnexions car elles sont différentes en fonction des ressources qu'elles connectent et de la distance physique entre ces ressources [Maingot 2007]. En effet, certaines interconnexions nécessitent un seul bit (9,5%), alors que d'autres peuvent utiliser deux bits (90,3%) voir même trois bits (0,2%).

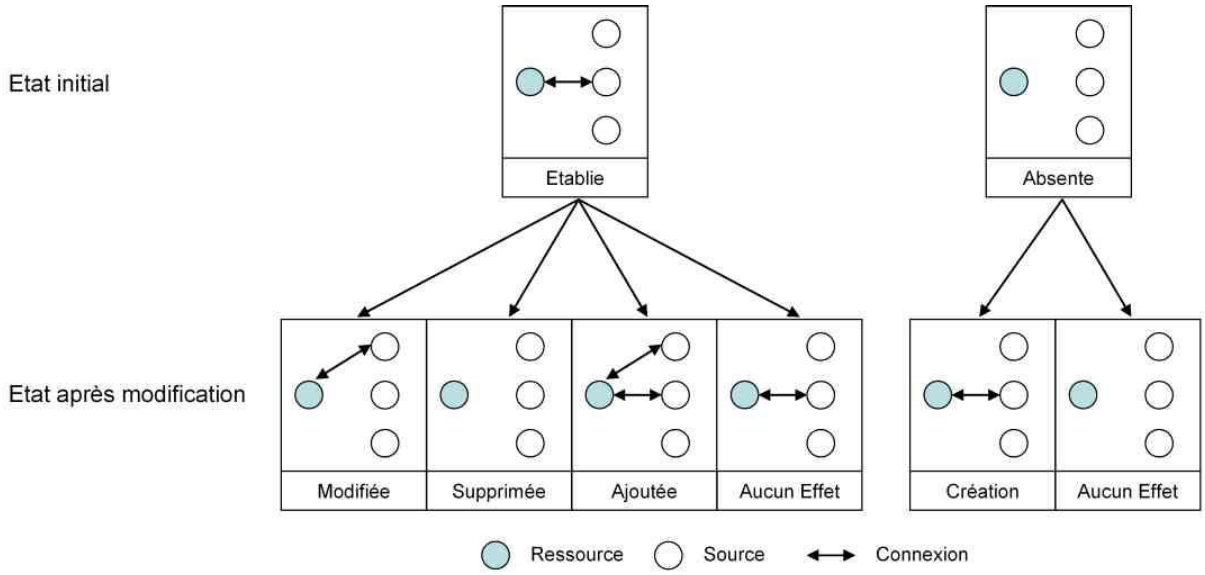

Figure 3-2: Mutations possible d'une connexion suite à un SEU

3.1.6. Préparation du composant

Les outils dont nous disposons ne permettent pas d'effectuer des mesures en gardant le composant dans son boitier d'origine. En effet les accélérateurs ne génèrent pas de particules ayant une énergie suffisante, comme c'est le cas dans l'espace, pour traverser le capot métallique et le substrat. Il en va de même pour le faisceau laser qui est stoppé.

Il est donc nécessaire de décapoter le FPGA. Cela peut être facilement effectué en appliquant une pression à la base du capot métallique afin de le décoller. Ensuite il faut amincir le substrat car la puce étant montée en flip-chip, les zones actives sont accessibles uniquement par la face arrière. Or l'épaisseur du substrat d'environ 700 µm ne permet pas aux particules et au laser d'atteindre les transistors. Le procédé mis en œuvre par la CNES est une érosion mécanico-chimique qui permet de réduire le substrat à une épaisseur d'environ 70 µm. Descendre à des valeurs inférieures est très compliqué et risqué à cause de la taille importante de la puce (~ 1 cm²). En effet, un défaut de planéité lors du collage de la puce sur les plots du boitier a un impact non négligeable sur une distance de 1 cm. De plus, un relâchement des contraintes pourraient endommager la puce.

Etant donné la nature du composant étudié, un FPGA à base de mémoire de configuration de type SRAM, dans ce qui suit seront pris en compte uniquement les événements singuliers non destructifs. Et plus particulièrement les SEUs, MBUs et MCUs.

3.2. Conception et réalisation de la carte fille Virtex-II

Une fois le composant préparé pour les tests accélérés au sol, il a été nécessaire concevoir et réaliser une carte fille afin d'interfacer le DUT avec le testeur THESIC+. La carte ne comporte que peu de composants car son rôle est de relier pin à pin le DUT au FPGA du testeur. Au final, seule les alimentations du cœur (1,5V) et des entrées/sorties (3,3V) du FPGA sont fabriquées sur la carte fille grâce à deux régulateurs linéaires.

La carte Virtex-II (voir Figure 3-3) comporte, en plus du DUT et des alimentations, l'empreinte de la mémoire non-volatile pouvant stocker le fichier de configuration du FPGA, ceci dans l'éventualité d'un test qui aurait requit cette fonctionnalité. Le connecteur JTAG pour le programmateur Xilinx est aussi présent sur la carte. De plus, les signaux de configuration du FPGA sont déportés sur des connecteurs Header afin d'en faciliter l'accès pour les sondes de l'oscilloscope ou bien de l'analyseur logique utilisé pour la mise au point des applications. Enfin, trois interrupteurs permettent de choisir le mode de configuration du FPGA.

La carte a été réalisée en classe 5, c'est-à-dire que l'isolement minimal entre deux conducteurs est de 0,15 mm, et elle comporte six couches :

- Deux couches internes pour les alimentations du cœur et des entrées/sorties du FPGA.
- Deux couches pour les plans de masse.
- Les deux faces extérieures sont réservées aux signaux.

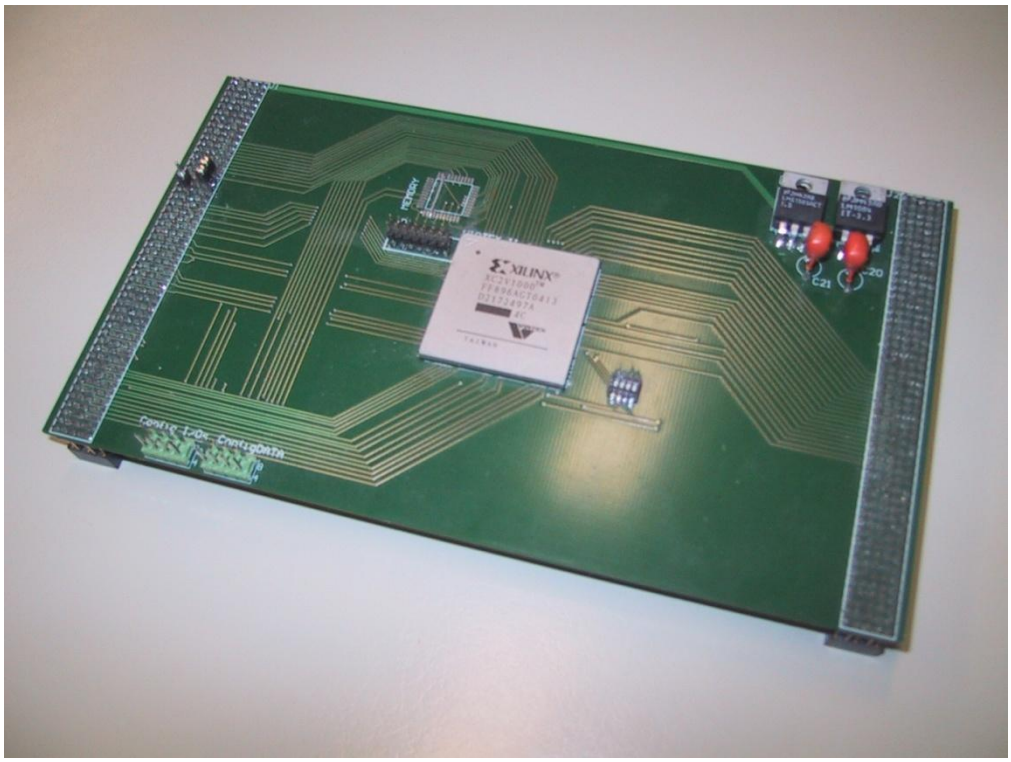

Figure 3-3 : Carte fille du testeur THESIC+ pour l'interfacage du FPGA Virtex-II

3.3. Les campagnes de test en accélérateur de particules

Le but des campagnes d'essais en accélérateur de particules est de mesurer la section efficace statique du composant, c'est-à-dire sa sensibilité intrinsèque. Par la suite il est possible d'en déduire le nombre de SEUs auquel il faut s'attendre lors du fonctionnement dans le milieu final. Néanmoins, cette étape nécessite d'avoir une connaissance des caractéristiques physiques de l'environnement final : le types de particules, leur énergie, leur flux, … La population des particules est relativement bien connue pour les orbites usuelles empruntées par les engins spatiaux, bien que fortement

dépendant de l'activité solaire. Pour aider les ingénieurs, des outils gratuits sont à la disposition de la communauté. Parmi ceux-ci on peut citer :

- CREME96 [Web CREME96] développé par le Naval Research Laboratory [Web NRL] avec le support de la NASA [Web NASA]. Le modèle de CREME96 fut décrit dans une publication en 1997 [Tylka 1997]. CREME96 est une mise à jour du code source de CREME (Cosmic Ray Effects on Micro Electronics). Ce logiciel regroupe un ensemble de programmes permettant la création de modèles numériques de l'environnement radiatif ionisant, pour des orbites proches de la Terre et pour l'évaluation des effets des radiations sur les engins spatiaux.
- OMERE (Outil de Modélisation de l'Environnement Radiatif Externe) est développé par la société TRAD [Web TRAD] avec le soutien du CNES. OMERE permet de calculer l'environnement spatial en terme de flux de particules chargées. Il calcule aussi les effets des radiations sur l'électronique en terme de dose, de déplacement atomique, d'effet singulier et de dégradation des cellules solaires.

L'estimation du nombre de SEUs par unité de temps dans le milieu final requiert deux informations qu'il faut fournir au logiciel : D'une part les données relatives à l'orbite de l'engin spatial et d'autre part la sensibilité intrinsèque du composant. La sensibilité, section efficace statique, peut être spécifiée pour le composant ou bien par unité de volume sensible (par bit), dans ce cas il faut aussi fournir le nombre de volumes sensibles.

3.3.1. Mesure sous ions lourds de la section efficace statique

La mesure de section efficace s'effectue en exposant le composant au flux de particules alors qu'il est alimenté et sans activité particulière. Ce qui veut dire, par exemple, dans une boucle pour un processeur, ou bien configuré avec une application vierge, sans aucune ressource utilisée, pour un FPGA. Préalablement à l'exposition du composant au faisceau, tous les bits sensibles sont initialisés. Pour un processeur et pour un FPGA l'initialisation se fait simplement lors de sa configuration. Le nombre de SEUs générés par les particules est ensuite déduit par la relecture des bits sensibles : il s'agit de ceux qui ont changé d'état. Dans un FPGA à base de mémoire SRAM, tous les bits sensibles sont accessibles par la mémoire de configuration.

L'accélérateur de particules HIF, de Louvain-la-Neuve retenu pour faire les campagnes de mesure, est doté d'un obturateur de faisceau nommé *shutter* et piloté par une commande pneumatique. Une fois en place dans le tube du faisceau il stoppe complètement les particules. Etant donné que la mesure de fluence se fait en aval du *shutter*, cela ne fausse pas la valeur car seules les particules qui frappent réellement le composant sont prises en compte. L'avantage de ce système est qu'il a permis de masquer complètement le faisceau durant les phases de configuration et de relecture du FPGA.

Lors d'un test accéléré aux ions lourds, outre le LET, il n'est possible de régler que deux paramètres :

- Le flux de particules, c'est-à-dire le nombre moyen de particules qui frappent une surface par unité de temps.
- La fluence étant définie comme l'intégrale du flux sur un intervalle de temps donné, il s'agit du nombre de particules reçues au cours de la durée d'exposition.

Cependant le temps de manœuvre du *shutter* requiert quelques centaines de millisecondes, il a donc fallu le prendre en compte afin qu'il soit négligeable par rapport au temps d'exposition. De même la configuration du composant nécessite 200ms et la relecture 300ms. Pour ces raisons le temps d'exposition retenu est de 25 secondes afin de minimiser l'impact de ces périodes de temps « perdues ». Le flux est par la suite réglé, pour chaque type de particule utilisé, afin que le nombre de SEUs obtenus au cours de l'exposition soit aux alentours de un pour mille du nombre total de bits sensibles. Un nombre de SEUs inférieur à cette valeur augmenterait d'autant le nombre d'heures de faisceau nécessaires pour l'obtention des résultats et donc l'impact sur le coût de cette étude. Tandis qu'une valeur supérieure augmenterait les risques d'obtenir deux événements singuliers sur un même bit (effet d'empilement), et donc de ne pas le détecter.

La Figure 3-4 illustre la procédure de test qui consiste à envoyer dans THESIC+ le fichier de configuration du DUT puis de configurer le FPGA alors que le *shutter* bloque le faisceau de particules. Ensuite le *shutter* est retiré pour permettre une irradiation du composant durant 25 secondes. Enfin le testeur procède à l'opération de relecture de la configuration du FPGA et la stocke dans sa mémoire pour être ensuite récupérée par l'ordinateur. Une nouvelle séquence est lancée depuis la configuration du DUT.

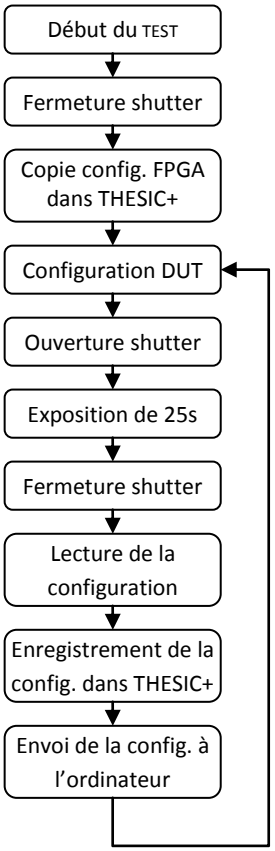

Figure 3-4 : Diagramme de la procédure de test statique

Les résultats obtenus pour les ions Carbone, Argon, Nickel et Krypton sont donnés dans le Tableau 3-2. Les sections efficaces pour les 3 particules possédant les LET le plus élevés sont sensiblement égales, par contre le DUT est nettement moins sensible au faisceau de particules de

carbone dont le LET est faible. Il faut noter que trois mesures avec l'ion Argon ont été effectuées afin de contrôler la cohérence des résultats d'une campagne à l'autre. On peut aussi remarquer que les sections efficaces obtenues sont conformes à celles publiée dans [Yui 2002].

Tableau 3-2 : Sections efficaces statiques sous ions lourds

Ions	**Energie** (MeV)	**Pénétration** (µm Si)	**LET** (MeV/mg/cm²)	**Section efficace** (cm²/bit)
$^{13}C^{4+}$	131	266	1,2	7,47 E-10
$^{40}Ar^{12+}$	372	119	10,1	1.45 E-8
$^{40}Ar^{12+}$	372	119	10,1	1,52 E-8
$^{40}Ar^{12+}$	372	119	10,1	1,66 E-8
$^{58}Ni^{17+}$	500	85	21,9	1,00 E-8
$^{83}Kr^{25+}$	756	92	32,4	1,23 E-8

Le logiciel OMERE est capable, à partir des résultats précédents, de fournir les paramètres de Weibull caractérisant au milieu la courbe de section efficace. Les valeurs fournies par OMERE sont les suivantes :

W = 7.8526 $\sigma_{sat} = 1{,}66^{E}\text{-}8$ cm²
S = 1.64496 LET_{th} 0,59 MeV.cm²/mg

La courbe de section efficace statique du Virtex-II, donnée dans la Figure 3-5, est obtenue en remplaçant les quatre valeurs ci-dessus dans l'équation suivante. Le traçage de la courbe est effectué par une application développée en langage PHP [Web PHP].

$$\sigma = \sigma_{sat}\left[1 - \exp\left(\left(-\frac{LET - LET_{th}}{W}\right)^{s}\right)\right]$$

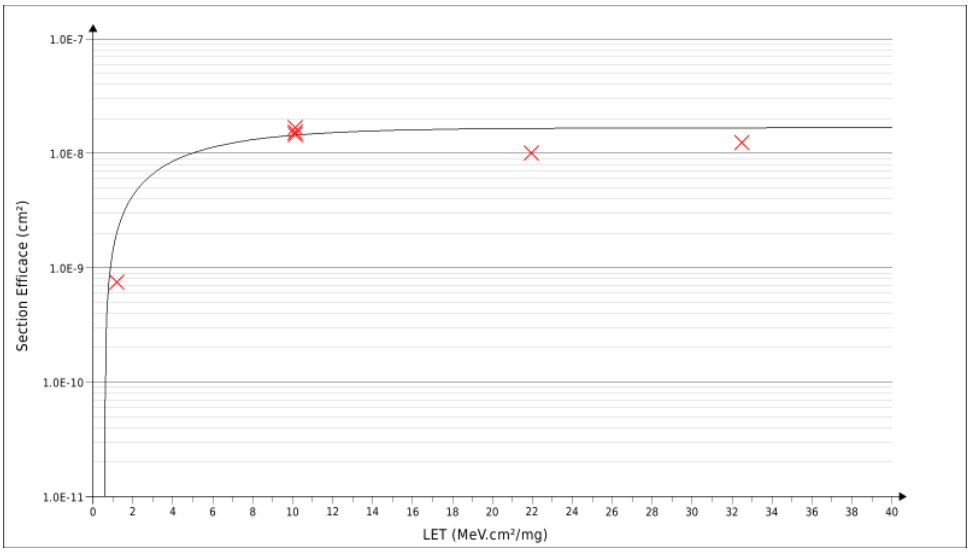

Figure 3-5 : Courbe de section efficace statique du Virtex-II

3.3.2. Mesure du taux d'erreur d'applications en dynamique sous ions lourds

L'étape suivant la mesure de la sensibilité intrinsèque du composant est l'évaluation du taux d'erreur d'une application. Le test dynamique est relativement similaire au test statique à la différence que le composant exécute une application alors qu'il est soumis au flux de particules. Les sorties de l'application sont comparées aux valeurs de référence. Si aucune erreur n'est observée alors l'application est exécutée à nouveau, ceci jusqu'à l'apparition d'une faute sur les sorties. La configuration du FPGA et les valeurs de sortie erronées sont stockées dans le testeur jusqu'à ce que l'ordinateur ne les télécharge. Le diagramme décrivant le protocole de test est donné en Figure 3-6.

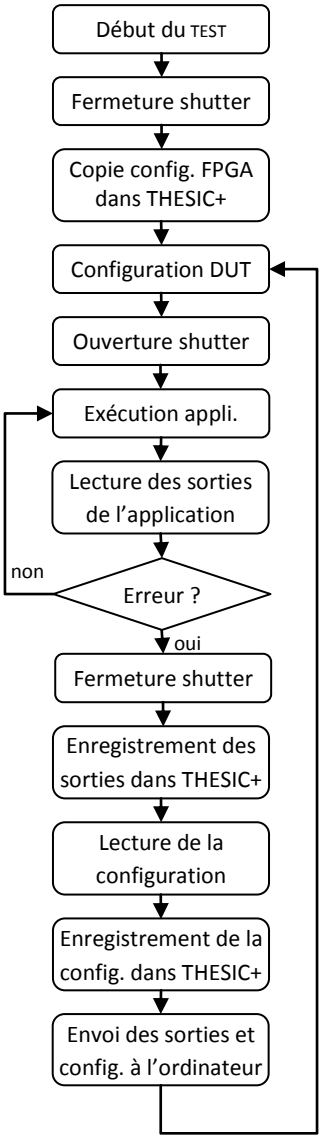

Figure 3-6 : Diagramme de la procédure de test dynamique

Deux applications ont été retenues :

- Le **TMR du cryptocore DES3** : cette application est une variante de celle utilisée pour la carte satellite COTS2, décrite en détail au paragraphe 2.4.4.1. La différence vient de l'ajout d'un registre de 3 bits pour stocker le code d'erreur détecté par le voteur du TMR. Il s'agit du numéro de la branche qui contient une erreur suite à la comparaison. Ce registre peut contenir les valeurs suivantes :
    - « 0 » lorsqu'aucune erreur n'est détectée.
    - « 1 », « 2 » ou « 3 » lorsqu'une erreur survient respectivement sur la branche 1, 2 ou 3.
    - « 4 » lorsque les résultats sur les trois branches divergent.
    - « 5 », « 6 » et « 7 » ne sont pas utilisées.
- Un **duplex de processeur LEON** décrit dans ce qui suit.

3.3.2.1. L'application Duplex de LEON

L'application Duplex de LEON est basée sur l'IP (Intellectual Property) LEON3 qui est d'une évolution de l'IP utilisée par ATMEL pour son AT697E (voir paragraphe 2.3). Deux processeurs LEON3 font simultanément les mêmes calculs. La comparaison des résultats identifie si une erreur s'est produite. Une architecture duplex telle que celle implémentée ici ne permet pas de déterminer quel résultat est juste. Cependant une fois l'erreur détectée, il est possible de reconfigurer le FPGA et de refaire le calcul.

Le processeur LEON3 est entièrement configurable et adaptable au besoin de l'utilisateur. Pour cela il est architecturé autour d'un bus AMBA (Advanced Microcontroller Bus Architecture) sur lequel viennent se connecter tous les modules dont l'utilisateur a besoin. La Figure 3-7 illustre l'architecture LEON3 avec les différents périphériques compatibles AMBA et qui apporte la compatibilité avec diverses interfaces comme le RS232, le JTAG, le Spacewire, le CAN, etc.

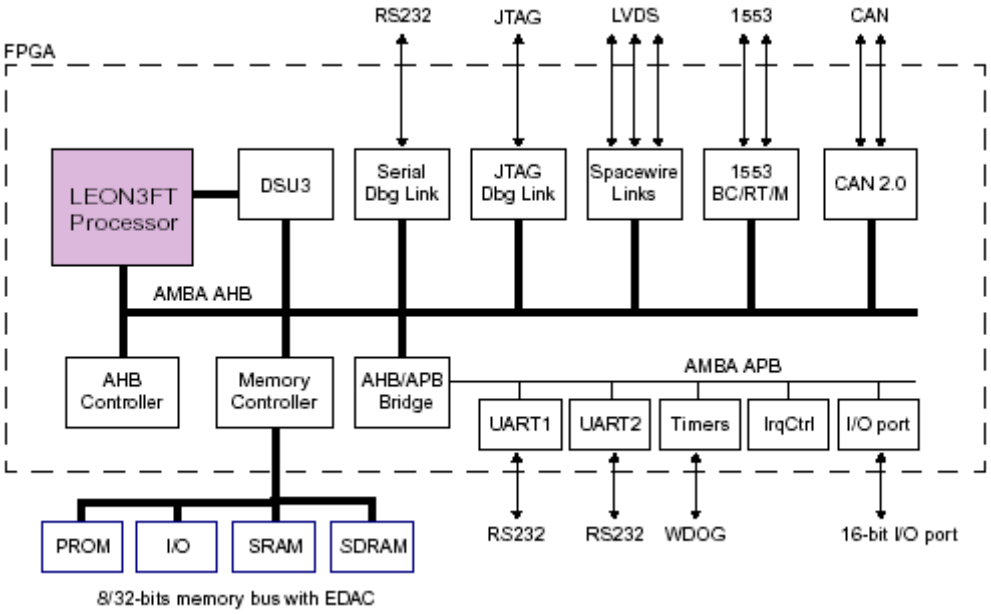

Figure 3-7 : Architecture LEON3

Dans l'architecture retenue, chaque processeur possède sa propre mémoire ROM et sa propre mémoire RAM à l'extérieur du FPGA. Ces mémoires sont physiquement hébergées dans la mémoire SRAM du testeur THESIC+. Etant donné que le testeur dispose de deux bancs de 32 bits accessibles indépendamment, chaque processeur dispose ainsi de sa zone mémoire (Figure 3-8). Chaque banc est découpé en trois zones : la ROM, la RAM et enfin le stockage du fichier de configuration du FPGA dans le banc 0 et le fichier de readback dans le banc 1.

Le testeur se charge de comparer les valeurs produites par les deux processeurs et il interrompt le test dès qu'il détecte une différence.

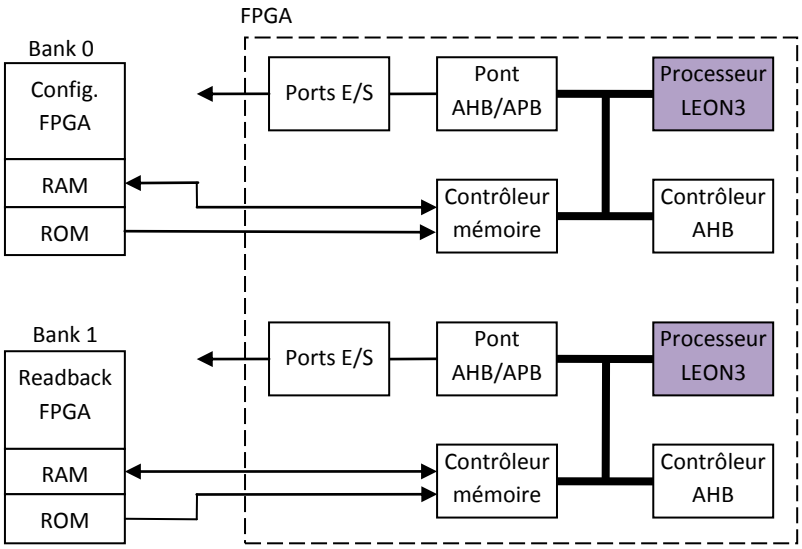

Figure 3-8 : Architecture de l'application Duplex de LEON

3.3.2.2. Résultats du test dynamique de l'application DES3

Comme pour la campagne de test statique, les essais en dynamique se sont déroulés à l'accélérateur de particules de Louvain la neuve. Deux types de particules différentes ont été utilisés : le carbone et l'argon. Cependant les premières mesures au carbone ont montré un taux d'erreur faible (dû au faible LET de la particule). Il a donc été décidé de passer à l'argon afin d'obtenir un nombre significatif d'erreurs dans la durée de la session de test.

Les essais au carbone ont permis d'exécuter l'application plus de 187 millions de fois pour une fluence totale de 158.543 particules. 51 erreurs de cryptage affectant une seule branche sur les trois du TMR ont été détectées par le TMR qui a pu fournir le résultat correct à partir des deux branches non affectées.

Les essais à l'argon ont permis d'exposer le FPGA à 437.095 particules alors que l'application a exécutée plus de 750 millions de cryptages. Cette exposition étant plus longue que celle au carbone, elle a pu fournir des résultats plus intéressants. En effet il a été possible de dénombrer quatre différents types d'erreurs :

- Les erreurs dites « inattendues ». Il s'agit du cas où le code d'erreur du TMR contient une valeur non utilisée (« 5 », « 6 » ou « 7 »). Le danger est que cette erreur peut masquer une erreur réelle sur le résultat du cryptage.
- Les erreurs n'affectant qu'une seule branche du TMR et donc sans conséquence car le TMR le tolère en fournissant un résultat correct à partir des deux branches non fautées.
- Les erreurs sur les trois branches du TMR.
- Les erreurs dites « critiques » non détectées par le TMR. C'est-à-dire lorsque le TMR délivre un résultat faux alors que le registre d'état ne donne aucune erreur.

Le Tableau 3-3 présente le nombre d'exécutions de l'application qui ont donné lieu à une erreur de cryptage détectée soit par le TMR, soit par le contrôleur externe présent dans THESIC+.

Durant cette campagne de test d'environ une heure le DUT a été exposé au faisceau de particules de carbone pour une durée totale de 80 secondes pour une fluence totale de 158.543 particules qui ont permis d'observer 51 erreurs de type « 1 branche » et aucune autre erreur. Cela s'explique par le temps d'exposition relativement court et par le faible LET de la particule qui provoquent peu d'erreurs dans l'application.

Par contre l'exposition à l'argon a duré 319 secondes pour une fluence de 437.095 particules. Cela a permis l'observation les quatre types d'erreurs :

- 1278 erreurs de type « 1 branche » tolérable étant donné qu'elles ne faussent pas le résultat.
- 1 erreur de type « 3 branches » dont le résultat ne peut pas être pris en compte car il n'est pas correct. Cependant le problème étant identifié il est possible d'y remédier.
- 3 erreurs « inattendues », comme précédemment le système peut gérer ce genre d'erreur.
- 34 erreurs « critiques » qui ne sont pas détectées par le TMR et un résultat de l'application faux est fourni sans que le système en soit averti. Il s'agit donc de la mise en évidence du talon d'Achille de la méthodologie TMR alors qu'elle est implémentée dans un FPGA à base de mémoire SRAM, une faiblesse qui a été évoquée dans [Sterpone 2005].

Tableau 3-3 : Nombre d'exécution de l'application contenant des erreurs lors des tests sous ions lourds

Particules	Erreurs 1 branche	Erreurs 3 branches	Erreurs inattendues	Erreurs critiques	Nombre d'exécutions	Temps d'exposition	Fluence totale
Carbone	51	0	0	0	187.469.275	80 sec.	158.543
Argon	1.278	1	3	34	750.688.226	319 sec.	437.095

Le nombre moyen d'exécutions de l'application, le temps moyen d'exécution et le nombre moyen de particules nécessaire à l'apparition de chaque type de faute est donné dans le Tableau 3-4.

Tableau 3-4 : Nombre d'exécutions, temps d'exécution et nombre moyen de particules nécessaire pour provoquer chaque type de fautes

		Carbone	Argon
Erreurs 1 branche	Nb. exécutions	3 675 868	587.393
	Tps d'exécution (s)	1,56	0,25
	Nb. particules	9 647	352
Erreurs 3 branches	Nb. exécutions	N/A	750 688.226
	Tps d'exécution (s)	N/A	319,04
	Nb. particules	N/A	449.783
Erreurs inattendues	Nb. exécutions	N/A	250.229.409
	Tps d'exécution (s)	N/A	106,35
	Nb. particules	N/A	149.928
Erreurs critiques	Nb. exécutions	N/A	22.079.065
	Tps d'exécution (s)	N/A	9,38
	Nb. particules	N/A	13.229

Etant donné que les erreurs de type « 3 branches » ne permettent pas d'obtenir un résultat correct, on peut donc les classer dans les erreurs de type « critiques » et on obtient donc plus que trois catégories d'erreurs qui sont résumés dans le Tableau 3-5.

Tableau 3-5 : Nombre d'erreurs détectées, d'erreurs faussement détectées, erreurs non détectées et de SEUs dans le readback

Particules	Erreurs détectées	Fausses détection d'erreur	Erreurs non détectées	Nb. SEUs dans le readback	Fluence
Carbone	51	0	0	1.373	492.000
Argon	1.278	3	35	21.431	450.000

A partir du Tableau 3-5, le taux d'erreurs de l'application DES pour chacune des deux particules peut être obtenu par le rapport entre le nombre d'erreurs et la fluence relevée lors des essais. Ces résultats sont présentés dans le Tableau 3-6.

Tableau 3-6 : Probabilité d'observer une erreur sur les sorties de l'application lorsqu'un SEU apparaît dans la mémoire de configuration

Particules	Erreurs détectées	Fausses détection d'erreur	Erreurs non détectées
Carbone	$1{,}04\ E^{-4}$	N/A	N/A
Argon	$2{,}84\ E^{-3}$	$6{,}67\ E^{-6}$	$7{,}56\ E^{-5}$

3.4. Les campagnes de test sous faisceau laser

Trois campagnes de test laser se sont déroulées sur le banc de test PLS (Photoelectric Laser Stimulation) de la plateforme laser ATLAS du laboratoire IMS (Intégration de Matériau au Système) de l'université de Bordeaux. La première série de tests à démontré la capacité du banc laser à produire des SEUs dans le DUT et donc d'établir le point de fonctionnement du laser pour ce composant, c'est-à-dire les réglages de longueur d'onde, focalisation du faisceau, etc. à appliquer. A partir de ces observations et mesures il a été possible de développer la méthodologie et les outils afin de mener une seconde campagne dans le but de cartographier de larges zones du composant pour obtenir la distribution de sensibilité des différentes structures composant le FPGA. Enfin la synchronisation des plateformes ATLAS et THESIC+ a permis, lors de la troisième campagne, de tracer une cartographie de la sensibilité de la mémoire du composant ainsi que d'observer des effets du laser sur une application en cours d'exécution.

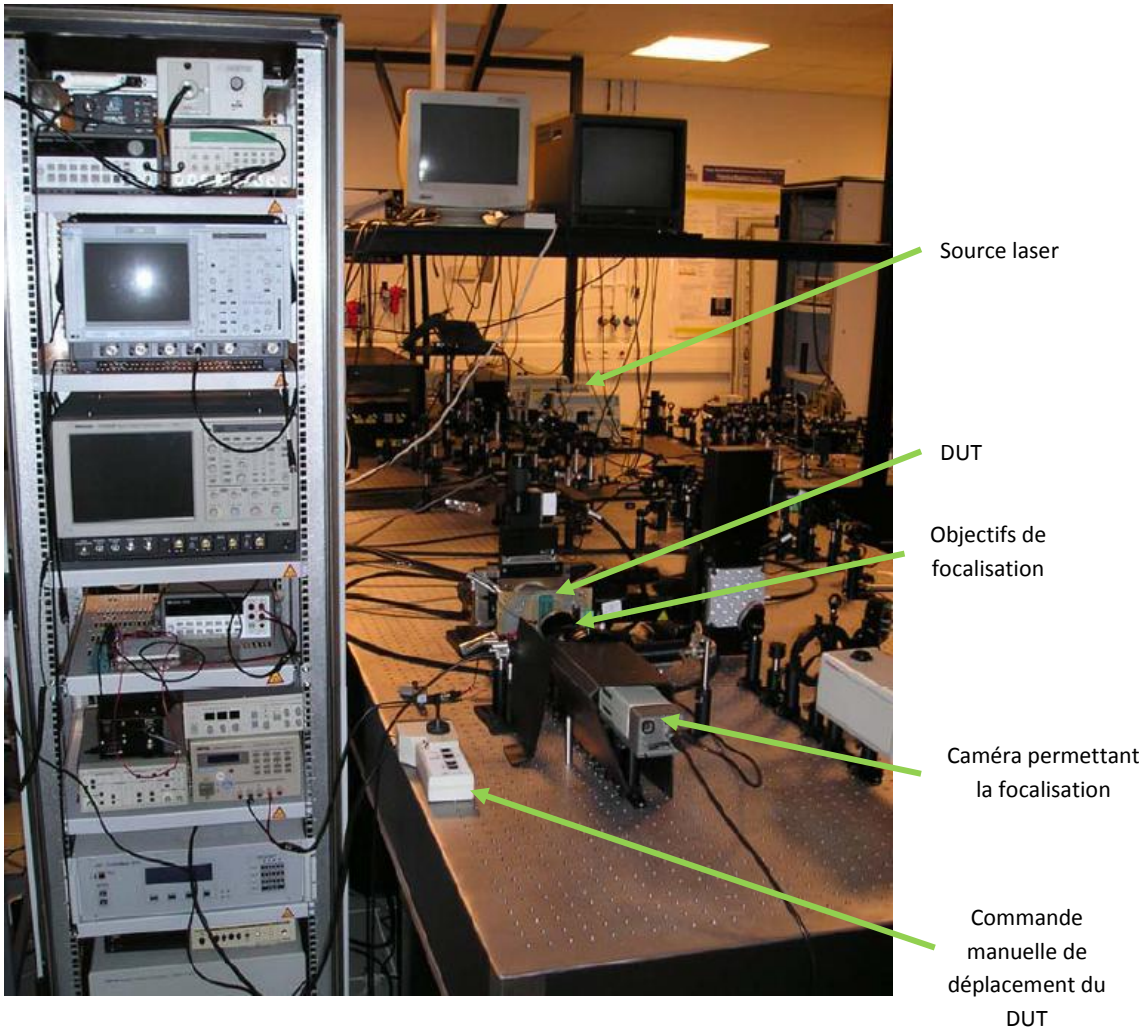

Figure 3-9 : La plateforme laser ATLAS

La source laser utilisée est un oscillateur Ti : Sapphire capable de délivrer des impulsions laser de 1 ps à une fréquence de 80 MHz. La longueur d'onde du laser a été réglée à 980 nm afin d'obtenir une pénétration optimale dans le silicium pour des attaques en face arrière. Un sélecteur d'impulsion, placé à la sortie de la cavité laser, permet de réduire la fréquence des impulsions laser

jusqu'à obtenir un tir mono-coup. Le faisceau est focalisé sur le DUT à l'aide de différentes objectifs de microscope afin de produire un spot de 1, 2 ou 5 µm. La position en XYZ du composant sous le faisceau est contrôlée avec une résolution de 100nm.

Un programme dédié en C++, nommé SEEM [Pouget 2007 SEEM], contrôle l'automatisation complète du banc de test, y compris les communications avec le testeur THESIC+.

La Figure 3-9 est une photographie de la plateforme ATLAS. Elle montre la source laser en arrière plan et le plateau équipé de jeux de miroirs permettant l'acheminement du faisceau laser jusqu'au DUT. Une armoire équipée d'appareils de mesure (alimentations, oscilloscopes, générateurs de signaux, etc) permet l'alimentation en énergie du DUT ainsi que l'observation des signaux électriques. Une caméra fournie une image de la surface du composant et un réticule permet de connaître la position du faisceau laser. Cette image permet entre autre :

- De donner au logiciel SEEM une coordonnée lui servant de point d'origine pour la navigation sur le composant.
- De vérifier l'orthogonalité de la surface du DUT avec le faisceau laser afin de garantir la même focalisation aux quatre coins de la puce.
- D'ajuster la focalisation.

3.4.1. Validation de la plateforme de test

Cette première campagne d'essai au laser a eu pour but de collecter des données qualitatives, et non quantitatives, sur l'efficacité et la flexibilité de la plateforme de test ATLAS/THESIC+. En effet la capacité du laser à produire des SEUs dans la mémoire de configuration du FPGA a pu être vérifiée. Cela a donc confirmé que l'amincissement du composant et la puissance du laser étaient suffisants pour apporter l'énergie provoquant le basculement des points mémoire.

3.4.2. Distribution de sensibilité des différentes structures du FPGA

L'objectif de la deuxième campagne de test a été l'obtention de la distribution de la sensibilité des différentes structures du FPGA grâce à un test statique. Pour cela il a fallu balayer de larges zones de la puce puis identifier la nature des bits modifiés par le faisceau laser dans la configuration.

Le logiciel SEFEA-Prod (Soft Error Functional Effect Analysis in Programmable Devices) a été développé au laboratoire TIMA [Maingot 2007] [Canivet 2008] pour analyser la mémoire de configuration des FPGAs à base de mémoire SRAM ainsi que les effets des injections matérielles/logicielles de fautes. SEFEA est capable de détecter des SEUs dans la mémoire de configuration d'un Virtex-II en comparant un bitstream issue d'une session d'injection de fautes avec un bitstream de référence. L'analyse faite par le logiciel produit une distribution des erreurs en fonction des différentes ressources du FPGA (CLB, IOB, BRAM, etc).

Le protocole de test statique consiste en premier lieu à régler les paramètres du laser afin de produire des SEUs dans la mémoire de configuration du DUT. Ensuite, les coordonnées des deux points définissant la zone rectangulaire à bombarder sont entrées dans SEEM. De son coté, le FPGA est configuré et l'application est laissée oisive. La Figure 3-10 illustre cette procédure de test.

Pour cette campagne les plateformes ATLAS et THESIC+ n'étaient pas synchronisées car le DUT n'était pas reconfiguré après chaque tir. Donc une fois le DUT configuré, la cartographie est lancée manuellement. La lecture de la configuration est aussi exécutée manuellement à la fin des opérations de tir laser.

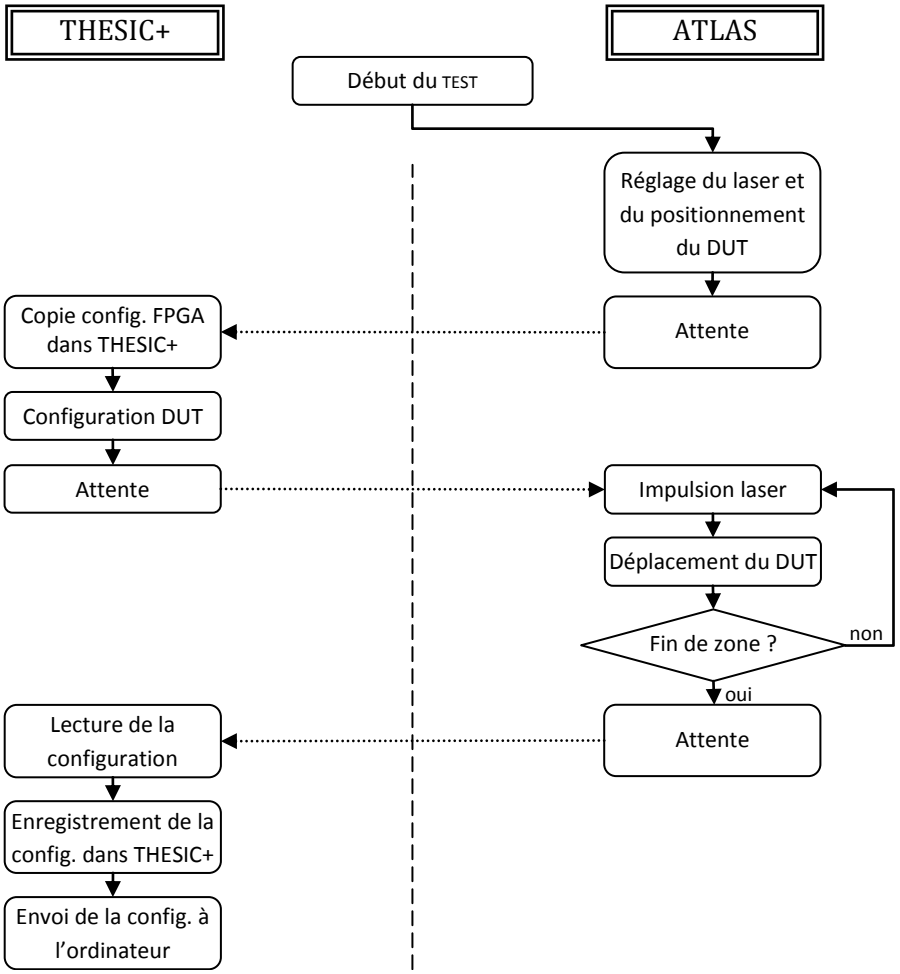

Figure 3-10 : Protocole pour le test statique sous faisceau laser

Le Tableau 3-7 présente la distribution des erreurs, en nombre et en pourcentage, en fonction des différents types de ressources pour la cartographie d'une zone de 1mm x 1mm dans un angle du DUT. Cette session de tirs laser a produit 4 583 SEUs principalement dans les CLB et la BRAM.

Tableau 3-7 : Répartition des bits erronés en fonction des ressources du FPGA

Type de ressource	CLB	CLBE	GCLK	IOB	IOI	BRAM	BRAMI
Nombre de SEUs	3956	0	0	123	6	445	53
	81,97 %	0 %	0 %	2,74 %	0,13 %	9,90 %	1,18 %

En regardant plus précisément dans les blocs CLB, il apparait que les bits contrôlant le routage des signaux sont les principaux contributeurs aux erreurs observées. Ce résultat pouvait être attendu étant donné que cette famille de bits représente une large population des bits de configuration. De plus, un nombre non négligeable de bits erronés a été observé parmi les ressources logiques (entre autre les LUTs). Le Tableau 3-8 résume la distribution des bits erronés en fonction des ressources présentes dans les tuiles CLBs. La configuration des CLBs est organisée en 22 trames. Compte tenu que le rôle exact de la 22ème trame n'est pas complètement défini les SEUs survenant dans cette trame sont classés séparément.

Tableau 3-8 : Répartition des bits erronés à l'intérieur des tuiles CLBs

Type de bit	Logique	Interconnexions	22ème trame	Total
Nombre de SEUs	1680	2122	154	3956
	36,66 %	46,30 %	3,36 %	81,97 %

3.4.3. Influence de l'énergie du faisceau sur le taux de génération des SEUs

Un autre axe d'étude de cette campagne d'essais a été l'observation de l'effet de la puissance du laser sur le taux de génération des SEUs dans la mémoire de configuration. Pour cela la cartographie d'une même ligne a été réalisée plusieurs fois en augmentant à chaque fois l'énergie du faisceau laser. Les premières erreurs sont apparues pour une énergie incidente de 760 pJ sur la face arrière du composant. A partir de cette valeur il est possible d'estimer la valeur du LET seuil équivalent pour un ion en connaissant exactement l'épaisseur et les caractéristiques de dopage du substrat [Pouget 2001]. Pour cette étude nous nous somme intéressés à la variation relative de l'énergie donc cette calibration énergie-LET n'a pas été nécessaire.

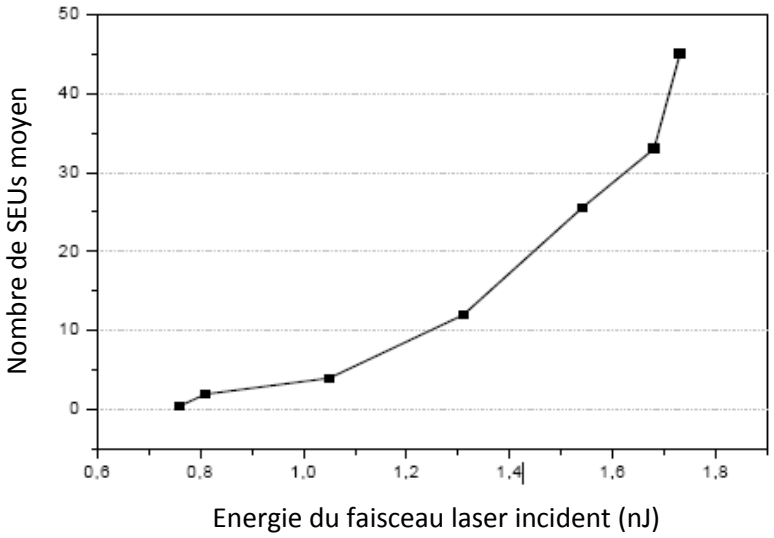

Figure 3-11 : Influence de l'énergie du laser sur le taux de génération des SEUs

Il a pu être observé, voir Figure 3-11, que le nombre total des erreurs augmente avec l'énergie et ne semble pas tendre vers une saturation. Cela pourrait être expliqué par la plage restreinte

d'énergie qui a été explorée (moins d'une décade), par l'importance de la diffusion dans cette technologie et par la contribution de structures de natures différentes.

Le Tableau 3-9 illustre l'évolution du nombre d'erreurs et leur distribution parmi les différents types d'éléments présents dans le FPGA. Les résultats montrent que seuls les CLBs et la BRAM ont été affectés mais cela dépend bien sûr de la zone balayée par le faisceau. Le Tableau 3-10 présente la répartition des bits erronés à l'intérieur des tuiles CLBs pour la même session de tirs. Ces résultats indiquent une sensibilité supérieure des bits de la 22ème trame de configuration par rapport à celle de la logique et des interconnexions. Ceci n'étant pas en accord avec le Tableau 3-8, cela est probablement du à la localisation géographique de la zone scannée dans cette campagne particulière qui est différente de la campagne précédente.

Tableau 3-9 : Distribution moyenne des bits erronés en fonction de l'énergie laser

Type d'élément / Energie (nJ)	CLB	CLBE	GCLK	IOB	IOI	BRAM	BRAMI
0,76	1	0	00	0	0	0	0
0,81	1	0	0	0	0	0	0
1,05	2	0	0	0	0	0	0
1,31	2	0	0	0	0	1	0
1,54	8,4	0	0	0	0	7,6	0
1,68	16,5	0	0	0	0	16,5	0
1,73	18	0	0	0	0	25,3	0

Tableau 3-10 : Distribution moyenne des bits erronés à l'intérieur des tuiles CLBs en fonction de l'énergie laser

Type d'élément / Energie (nJ)	Logique	interconnexions	22ème trame	Total
0,76	0	0	1	1
0,81	0	0	1	1
1,05	0	0	2	2
1,31	0	0	2	2
1,54	0	0	8,4	8,4
1,68	0	0	16,5	16,5
1,73	0	0,67	17,33	18

3.4.4. Influence de l'application présente dans le FPGA

Enfin, le dernier axe d'étude fut l'analyse de l'importance de l'application présente dans le FPGA sur le taux de génération des SEUs. Pour cela une zone rectangulaire de 0,25 mm x 1 mm a été cartographiée quatre fois en changeant l'application à chaque fois. Les quatre applications sont les suivantes :

- L'application « vide » dans laquelle aucune des ressources du FPGA n'est utilisée. Elle permet d'avoir 99,99 % des bits de configuration à l'état logique « 0 ».

- L'application « FF_0 » dans laquelle toutes les flip-flops du FPGA sont utilisées et leur entrées et sorties sont fixées à l'état logique « 0 ».
- L'application « FF_1 » dans laquelle toutes les flip-flops du FPGA sont utilisées et leur entrées et sorties sont fixées à l'état logique « 1 ».
- L'application « additionneur » 200 bits qui occupe 100 slices regroupées dans un coin du FPGA afin d'augmenter la densité de ressources utilisées dans cette zone et donc la probabilité que le laser frappe une de ces ressources.

Les résultats de cette session de tirs laser sont donnés dans le Tableau 3-11. Les deux applications avec les flip-flops instanciés permettent de générer le plus grand nombre de SEUs, avec un léger avantage pour la version avec les entrées et sorties à « 0 ». Ces deux véhicules de test permettent de générer respectivement deux et trois fois plus de bits erronés que l'additionneur et l'application vide. Cela semble clairement montrer une dissymétrie, en terme de sensibilité, entre les deux états logiques des bits de configuration. En conclusion, cela montre que la sensibilité d'une application est fortement liée au type de ressources qu'elle utilise et à la manière dont elles sont utilisées.

Tableau 3-11 : Impact de l'application sur le nombre de SEUs générés

Application	Vide	FF_0	FF_1	Additionneur
Nombre d'erreurs	78	280	240	135

3.4.5. Impact du faisceau laser sur une application en fonctionnement

La dernière campagne a eu pour objectif d'étudier l'impact des tirs laser sur une application en cours de fonctionnement. L'un des buts recherché fut, dans un premier temps, la synchronisation des plateformes ATLAS et THESIC+ afin de positionner précisément l'impulsion laser dans l'exécution de l'application.

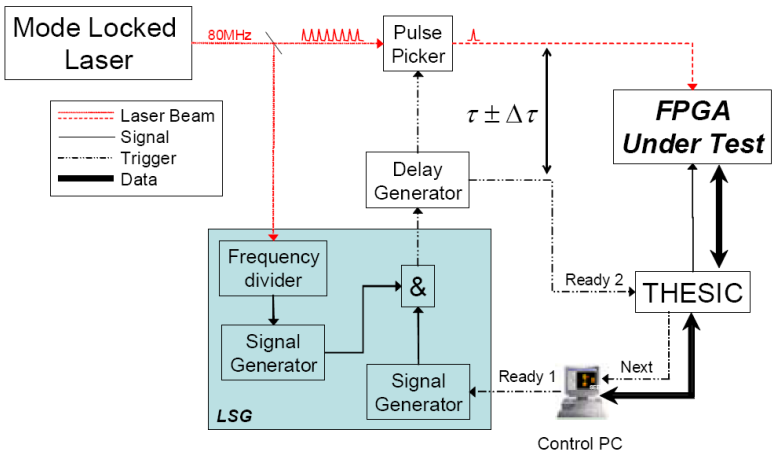

Figure 3-12 : Diagramme du principe de synchronisation des plateformes ATLAS et THESIC+

La Figure 3-12 présente le banc de test utilisé pour cette campagne. Etant donné que le laser est un oscillateur avec sa propre phase, qui ne peut pas être contrôlée suffisamment rapidement pour ce test, il a donc été utilisé comme horloge maître pour la synchronisation des autres éléments du banc de test.

La plateforme ATLAS dispose d'un outil de synchronisation dédié nommé Laser Synchronous Generator (LSG) qui permet de contrôler le retard entre le déclenchement d'une impulsion laser et l'instant choisi. Ceci est effectué avec une résolution supérieure à 50 ps [Douin 2007]. Une petit partie du faisceau laser est déviée et transformée en un signal numérique de fréquence 80 MHz, c'est-à-dire égal à la fréquence de répétition des impulsions laser. Cette fréquence est ensuite divisée par 8 pour obtenir un signal à 10 MHz afin de pouvoir verrouiller l'entrée de la PLL (Phase Loop Lock) d'un générateur de signaux qui délivre un signal à 1 kHz synchrone avec les impulsions laser. Une fonction logique ET est ensuite utilisée pour produire une seule impulsion synchrone avec le laser lorsque le signal « Ready 1 » issu par l'ordinateur de contrôle est au niveau logique « 1 ». Cela permet de déclencher le générateur de retards toujours en synchronisme avec les impulsions laser. Enfin ce générateur déclenche d'une part l'exécution de l'application grâce au signal « Ready 2 » et d'autre part le « pulse picker » qui laisse passer une impulsion du laser et qui frappe le DUT. Le retard τ entre ces deux signaux peut être réglé avec une résolution de 5 ps et le jitter Δτ pour cette configuration précise a été mesuré comme étant inférieur à 130 ps. La période de l'horloge de l'application testée dans le FPGA étant de 12,5 ns, cela assure un contrôle précis de l'instant d'injection des fautes par rapport au cycle d'exécution de l'application.

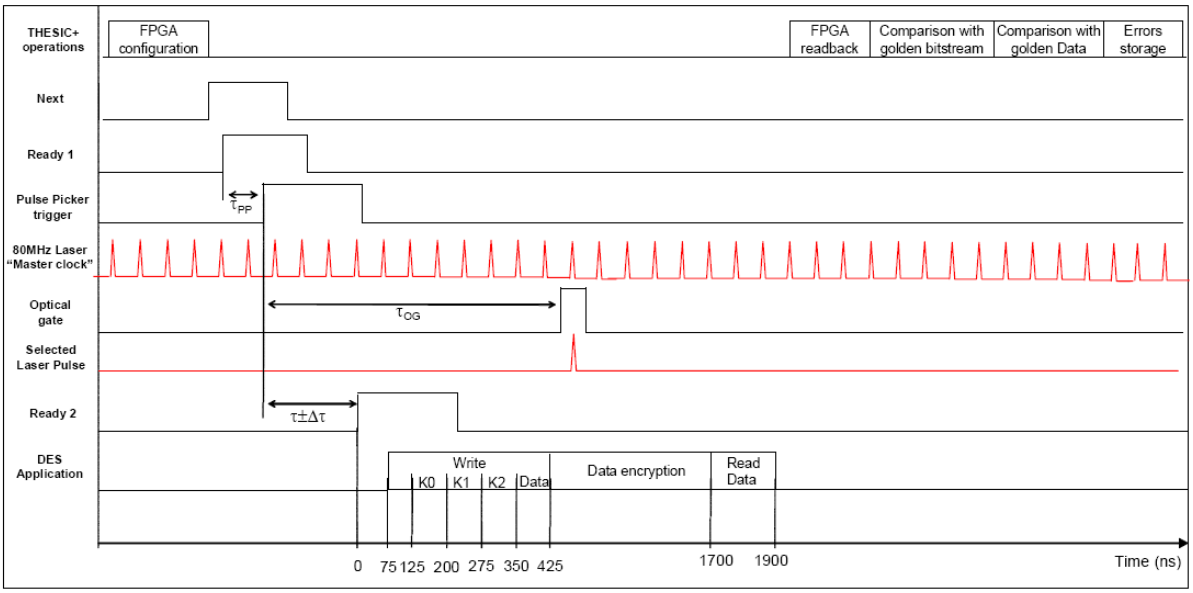

Figure 3-13 : Diagramme temporel de la séquence de test montrant la synchronisation des plateformes ATLAS et THESIC+

La Figure 3-13 décrit la séquence de test qui est effectuée pour chaque tir laser. Les fonctions logicielles permettant le contrôle du testeur THESIC+, du FPGA et de l'application ont été intégrées dans le logiciel SEEM. Il peut ainsi piloter intégralement le test en commençant par l'initialisation et la configuration du FPGA. THESIC+ peut alors générer le signal « Next » indiquant que l'application DES est prête à démarrer et est maintenue en attente grâce à son signal « reset ». Le logiciel SEEM

pilote ensuite le déplacement ensuite le DUT vers le prochain point devant être scanné puis produit le signal « ready 1 ». Grâce au LSG ce signal est traduit en une impulsion synchrone avec les impulsions laser qui déclenche le générateur de retards. Ce générateur déclenche alors le « pulse picker » qui ouvre une fenêtre optique de 12 ns afin de laisser passer une seule impulsion laser qui est acheminée jusqu'au FPGA. Le retard τ_{OG} est fixe et le retard τ_{PP} est réglé précisément de manière à ce que la fenêtre optique soit centrée sur l'impulsion laser. Le générateur de retards génère aussi le signal « ready 2 » contrôle le démarrage de l'application du FPGA.

Le générateur de retards permet de contrôler le retard au démarrage de l'application avec le retard τ et ainsi de positionner l'application par rapport à l'impulsion laser. Sa durée d'exécution étant de 1 900 ns, le retard τ_{OG} est choisi supérieur à cette valeur afin d'injecter une faute sur toute la plage d'exécution. Une fois l'application terminée, THESIC+ relit la mémoire de configuration du FPGA et la compare avec la référence afin de détecter d'éventuels SEUs engendrés par le laser. De plus si le résultat du cryptage est différent de la valeur attendue, THESIC+ indique qu'une erreur est apparue et enregistre la valeur de sortie. Ce résultat, ainsi que les SEUs dans la mémoire de configuration, sont stockés dans la mémoire de THESIC+. Une fois cette opération effectuée, qui nécessite quelques millisecondes, le testeur reconfigure le FPGA puis génère le signal « next » indiquant à SEEM qu'il est prêt pour le tir laser suivant. Cette méthodologie est répétée jusqu'à avoir scanné tous les points de la zone choisie.

Ce n'est qu'une fois la cartographie de la zone achevée que SEEM télécharge les résultats depuis THESIC+. Ceci afin de réduire au maximum les échanges Ethernet car il est bien plus efficace de transférer toutes les données en un paquet à la fin plutôt que d'interroger le testeur après chaque tir pour quelques octets. Cela est notamment dû à la latence de Microsoft Windows pour répondre à une requête ethernet. Le testeur surveille après chaque tir la mémoire restante et peut demander à SEEM de vider les résultats avant de continuer. Cependant, l'inconvénient de ce choix est que les résultats ne sont pas disponibles en temps réel, mais qu'une fois la cartographie achevée. Pour cette raison il vaut mieux s'assurer des réglages du laser avant de lancer une session de tirs qui peut durer plusieurs heures.

Une fois les résultats des tirs téléchargés depuis la mémoire de THESIC+, SEEM traite les données et génère les cartographies de sensibilité de la mémoire de configuration et des erreurs survenues dans l'application.

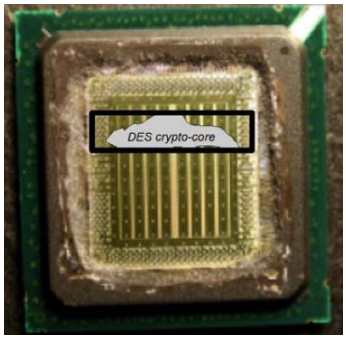

Figure 3-14 : Localisation de l'application cryptocore DES implémentée dans le FPGA

3.4.1. Test de l'application DES en mode dynamique

La Figure 3-14 est une représentation de la localisation des ressources utilisées par l'application dans le FPGA. Les sessions de tir laser vont donc se focaliser sur cette zone.

Test Statique pour la cartographie des ressources du FPGA

La première étape de ce test a consisté à scanner le FPGA sur une large zone afin de localiser précisément les zones d'intérêt. C'est-à-dire les zones comportant les ressources utilisées par l'application. Cette identification s'est déroulée en mode statique, l'application DES étant donc oisive. Par conséquent seules des erreurs induites dans la mémoire de configuration ont été cartographiées afin de déterminer la position exacte des différentes structures du DUT.

Une zone de 10mm, soit la pleine largeur du composant, par 400µm de haut a été scannée. La Figure 3-15 présente le nombre de mots de 8 bits de la mémoire de configuration contenant au moins un SEU provoqué par le laser, cela pour une zone de scannée couvrant la pleine largeur du composant. Grâce à l'adresse des bits erronés dans le fichier de configuration il est possible d'identifier la nature des ressources touchées. Un jeu de couleurs permet de différencier ces ressources. Les zones non-scannées apparaissent en noir et les tirs n'ayant pas induit d'erreur sont en bleu foncé. Les tirs ayant provoqués une erreur et deux erreurs ou plus sont représentées respectivement en bleu clair et jaune. Il est ainsi aisé de distinguer différentes structures comme les quatre colonnes de BRAM et leurs interconnexions (IBRAM), les 32 colonnes de CLBs, deux colonnes d'IOs et l'unique colonne de distribution des horloges (GCLK).

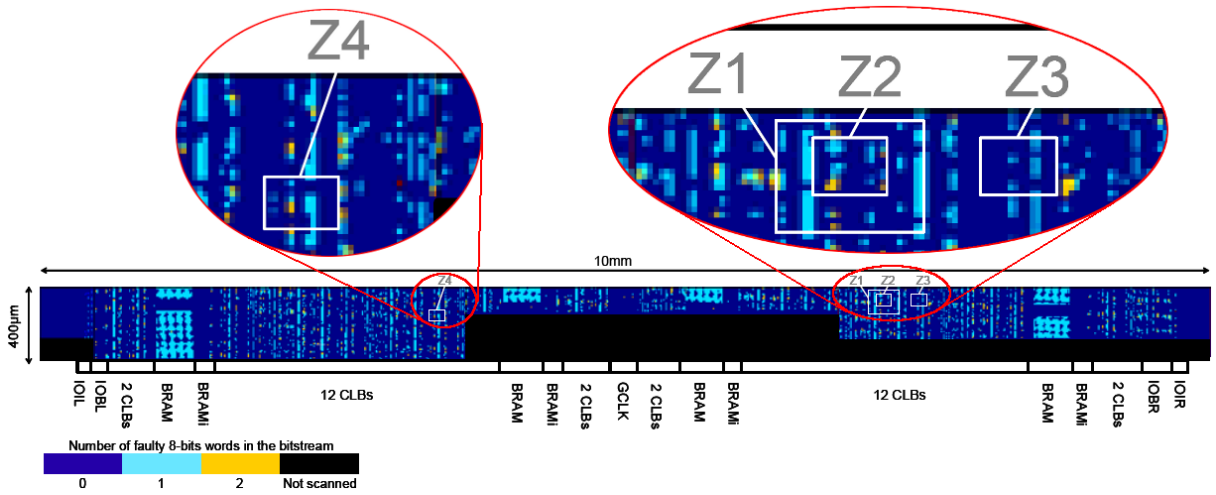

Figure 3-15 : Cartographie du nombre de SEUs induits dans la mémoire de configuration du FPGA

Test dynamique de l'application cryptocore

L'étape suivante a été de répéter la cartographie de la zone identifiée précédemment avec l'application DES activée selon la séquence de test définie par la Figure 3-13. Cette opération s'est montrée délicate du à l'absence de gestion des pertes de séquencement par le contrôleur de la séquence de test. Cela s'est traduit par des pertes de communication entre SEEM et THESIC+. Cela

s'est produit lorsque le laser frappait des zones pouvant engendrer des pertes de séquence de l'application DES comme c'est le cas de la colonne GCLK ainsi que dans quelques CLBs très spécifiques à l'application. Ces contraintes ont obligé à exclure du test certaines zones (voir les zones noires de la Figure 3-15).

Quatre zones d'intérêt, nommées de Z1 à Z4 sur la Figure 3-15, on été définies. La zone Z1 a été scannée avec un spot laser de 5 µm de diamètre alors que pour les zones Z2, Z3 et Z4 le spot avait un diamètre de 1 µm. Ces zones ont été scannées plusieurs fois avec des instant d'injections τ différents.

Des SEUs dans la mémoire de configuration ainsi que des erreurs sur les sorties de l'application DES ont pu être observées dans les quatre régions. Il est important de noter toutes les erreurs sur les sorties de l'application étaient liées à une erreur ou plusieurs SEUs dans la configuration, par conséquent aucune erreur purement transitoire n'a pu être clairement observée.

La Figure 3-16 présente, pour les quatre zones, le nombre d'exécutions de l'application ayant produit un résultat erroné en fonction du retard τ appliqué à l'impulsion laser. Ces valeurs sont normalisées en fonction du nombre maximum d'erreurs observées dans chaque zone.

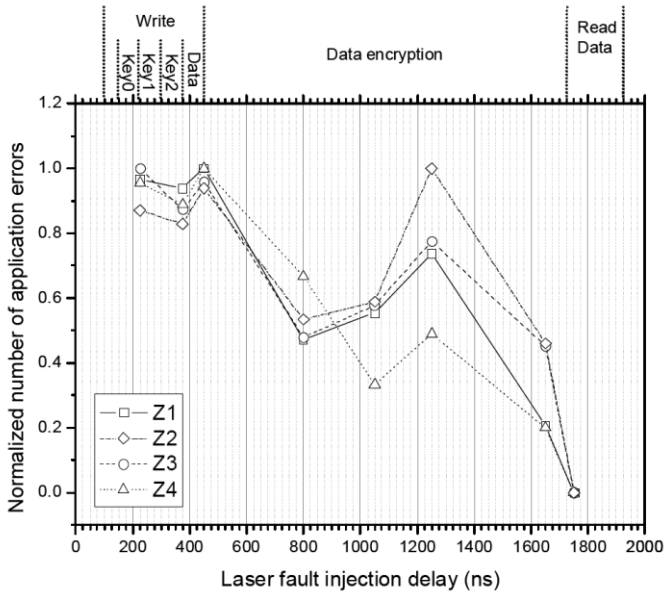

Figure 3-16 : Nombre d'erreurs de l'application en fonction de l'instant du tir laser

La première remarque est que les résultats des quatre zones suivent la même tendance. De plus, la probabilité d'apparition d'une erreur est maximale lorsque l'injection de la faute apparaît dès le début de l'exécution de l'application, c'est-à-dire lors du chargement des données. Ensuite l'impact des fautes injectées diminue de plus de 40% jusqu'au premier quart de la phase de cryptage pour Z1, Z2 et Z3 et jusqu'au premier tiers pour la zone Z4. Par la suite le nombre d'erreurs d'application augmente de nouveau jusque vers les deux tiers du cryptage pour diminuer jusqu'à atteindre zéro au début de la phase de lecture.

Le fait que les quatre courbes ne présentent pas une tendance décroissante monotone est assez surprenante si l'on considère que toutes les erreurs d'application sont dues à des SEUs dans la

mémoire de configuration du FPGA. En effet, les erreurs dans la configuration sont rémanentes durant tout le cycle d'exécution. Il semble donc naturel qu'une faute injectée dès le début de l'exécution de l'application ait une forte probabilité de perturber son fonctionnement. De même, plus la faute apparait tard, plus la probabilité qu'elle provoque une erreur sur la sortie diminue car à contrario la probabilité que la ressource impactée ne soit plus utilisée, et donc n'intervienne plus dans le calcul, augmente. Par conséquent, l'explication concernant l'augmentation du nombre d'erreurs, pour un tir compris entre 1.000 ns et 1.400 ns, n'est pas évidente. Une explication possible pourrait être la présence dans l'algorithme de mécanismes de compensation visant à réduire l'impact des fautes injectées durant la première moitié du cryptage.

La Figure 3-17.a présente la cartographie du nombre de bits de configuration erronés pour la zone Z3. Cela a permis de déterminer les adresses des bits erronés dans le fichier de configuration et de créer une grille virtuelle délimitant l'emplacement physique de chaque mot de huit bits.

Les Figure 3-17.[b, c, d] illustrent les tirs laser, représentés par un point noir, ayant provoqué une erreur sur les sorties de l'application DES pour un retard τ de 375 ns, 800 ns et 1250 ns. Il apparait clairement que le nombre et l'emplacement des erreurs d'application dépend du retard appliqué sur l'injection. Il est ainsi possible de remarquer que des tirs qui ont provoqués une erreur à 375 ns en ont aussi provoqué à 1250 ns mais pas à 800 ns (par exemple pour les mots 60.311 et 61.158).

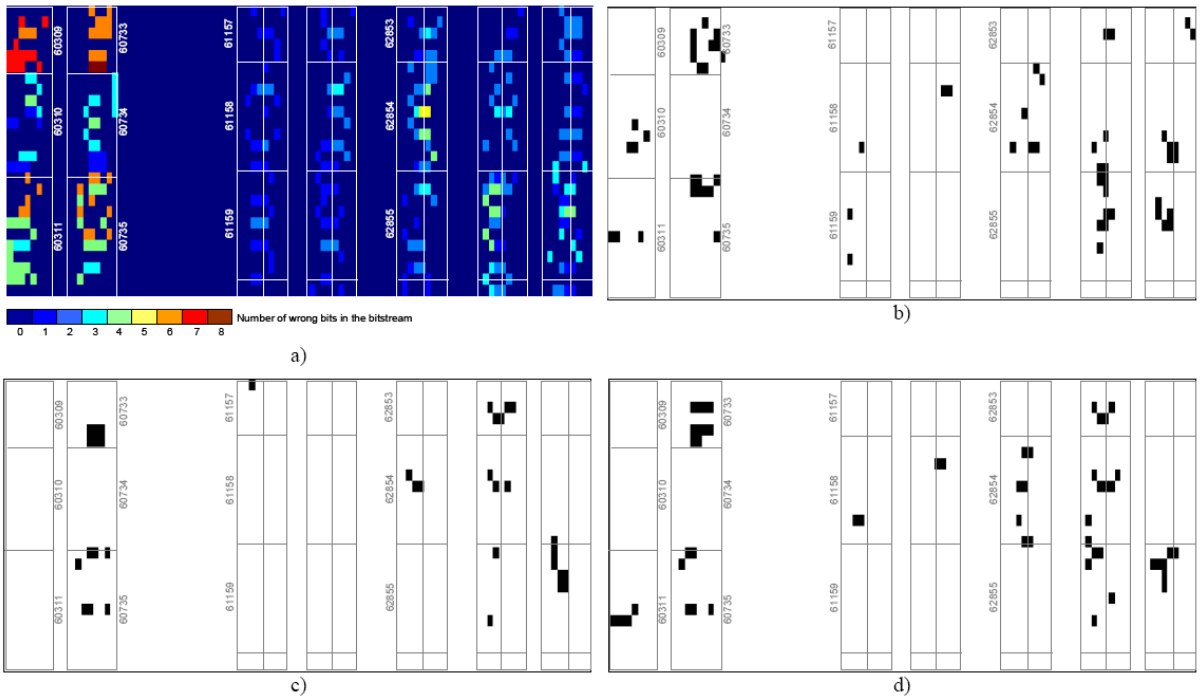

Figure 3-17 : a) cartographie du nombre de bits de configuration fautés pour chaque tir laser. Cartographie des erreurs d'application pour un retard d'injection de fautes de b) 375 ns c) 800 ns et d) 1250 ns. La zone cartographiée mesure 101 x 52 µm et la taille d'un pixel est 1 x 2 µm.

Le Tableau 3-12 présente les résultats pour les trois retards sur l'injection de fautes. La première remarque est la faible dispersion du nombre d'erreurs dans la mémoire de configuration, ce que confirme la bonne reproductibilité d'une cartographie à l'autre. L'autre point marquant est le fait

que pour une zone donnée et quelque soit l'instant d'injection, seulement 15 % des erreurs injectées produise une erreur sur les sorties de l'application.

Tableau 3-12 : Résultats des tirs laser sur la Z3 pour trois retards de l'injections

Retard à l'injection de faute (ns)	Nombre de mots de 8 bits erronés dans la configuration	Nombre d'exécution de l'application ayant donné un résultat faux
375	390	62
800	389	34
1250	394	55

3.4.2. Conclusions des campagnes de test au laser

Les campagnes de test menées au début des travaux réalisés au cours de cette thèse à l'aide d'un faisceau laser ont permis de confirmer que le composant choisi est sensible aux effets des particules et que la plateforme de test est capable de générer des événements dans la mémoire de configuration du FPGA puis de les détecter.

Les cartographies de la mémoire de configuration, qui ont été réalisée en mode statique, fournissent une reproduction fidèle de l'organisation des ressources logiques du composant et sont utiles pour positionnement du faisceau laser sur une zone sélectionnée. De plus ces tests ont fournis des informations sur la répartition des bits erronés en fonction des différentes ressouces.

Enfin les premiers essais en mode dynamique ont mis en évidence la capacité de la plateforme à injecter des fautes à l'endroit et à l'instant désiré, permettant ainsi des cartographies exhaustives dans l'espace et dans le temps. Enfin ces tests ont montré un comportement inattendu même si la tendance générale reste conforme aux prédictions. Etant donné que de tels essais sont peu documentés dans la littérature, il semble qu'il serait intéressant d'effectuer de plus amples expériences en mode dynamique lors de travaux futurs.

3.5. Les injections matérielles/logicielles de fautes

3.5.1. Principe de la méthode

Dans ce qui suit seront présentés les travaux portant sur la mise en œuvre d'une méthode de prédiction du taux d'erreurs d'une application implémentée sur une FPGA à base de mémoire SRAM en combinant la section efficace statique issus de tests en accélérateur de particules avec les résultats issus des campagnes d'injection de fautes effectuées par une méthode matérielle/logicielle. Le but final est la confrontation, pour l'application choisie, des prédictions avec les mesures issues d'essais en accélérateurs de particules. La technique d'injection de fautes mise en œuvre est une adaptation de la méthode CEU présentée au chapitre 2.4.1. En effet la théorie permettant d'obtenir le taux d'erreur est valide pour n'importe quel composant, seul le mécanisme d'injection des fautes doit être adapté à l'architecture du composant cible. Des travaux effectués dans ce domaine sont disponibles dans [Lopez 2005] par exemple.

La précision des prédictions issues de la méthode CEU dépend en grande partie du nombre de cellules sensibles accessibles par le jeu d'instruction par rapport au nombre total de cellules sensibles

du composant. Pour un FPGA à base de mémoire SRAM, ces zones sont la mémoire de configuration qui est entièrement accessible à l'utilisateur. Ainsi, la prédiction appliquée au FPGA ne devrait pas souffrir de cet inconvénient.

Pour un FPGA à base de mémoire SRAM, le mécanisme d'injection de faute doit agir sur la mémoire de configuration étant donné qu'elle représente l'ensemble des cellules sensibles. Pour cela deux méthodes sont possible :

- **L'injection de fautes par interruption de l'application** comme décrite dans la méthode CEU se pratique en interrompant l'exécution de l'application afin d'injecter la faute. Une fois l'application en sommeil la mémoire de configuration du FPGA est relue puis le FPGA est reconfiguré avec la faute. L'exécution de l'application est alors reprise ou elle s'était arrêtée. L'avantage de cette méthode est qu'elle prend en compte le facteur temporel pour l'injection.
- **L'injection de la faute à la configuration** du FPGA se fait directement lors de la première configuration de composant. Etant donné qu'un upset dans la mémoire de configuration est rémanent, son effet est similaire à celui obtenu lors d'une injection de fautes par interruption. L'avantage avec cette méthode est le gain de temps obtenu en supprimant les opérations de relecture de la mémoire de configuration et de reconfiguration du FPGA. Bien que le paramètre temporel de l'injection de fautes n'intervient pas pour une ressource applicative, il a cependant une importance pour un point mémoire de l'application dont le contenu est susceptible évoluer au fur et à mesure de l'exécution de l'application. Etant donné que ces ressources représentent 24% des bits de la mémoire de configuration du FPGA, l'approximation engendrée par cette méthode peut ne pas être négligeable en fonction du pourcentage de ces ressources utilisées par l'application.

Ainsi, le choix de la méthode dépend de l'application finale et du nombre de cellules mémoires qu'elle utilise. Les résultats issues d'une campagne d'injection de fautes sont exprimés en terme de nombre de faute (ou SEU) nécessaire pour provoquer une erreur sur les sorties de l'application. Conformément à la méthode CEU, il est alors possible d'obtenir la prédiction du taux d'erreur du couple composant + application pour chaque type de particules en multipliant ce nombre par la section efficace statique du composant obtenue à partir de mesures en accélérateur de particules.

Cependant, un des effets néfastes de la miniaturisation des circuits intégrés est l'augmentation du nombre d'erreurs de type multiple comme les MBUs et MCUs. Or la méthode d'injection actuelle ne permet pas d'injecter plusieurs fautes à la fois car il serait nécessaire d'avoir les informations concernant la physique du composant. Il serait ainsi possible de déterminer quels bits du bistream sont physiquement voisins sur la puce et donc de générer les vecteurs de test représentatifs de la réalité. Une autre méthode pourrait être d'obtenir les informations sur la physique en effectuant l'ingénierie inverse du composant grâce à un laser moyennement un coût humain et financier conséquent.

3.5.2. Mise en place de la méthode

Le testeur THESIC+ a servi de plateforme pour l'injection de faute étant donné qu'un certain nombre de fonctions avait déjà été développé pour les campagnes de test précédentes aux ions lourds et sous faisceau laser. Les objectifs de cette campagne sont triples :

- Effectuer la prédiction de fautes sur une application et deux variantes mettant en œuvre deux méthodes de mitigation des fautes : le duplex et le TMR.
- Mettre en évidence une éventuelle faiblesse dans le TMR qui permettrait à un résultat faux de sortir de l'application.
- Confronter les résultats de la prédiction par injection de fautes aux résultats des mesures effectuées sous ions lourds.

L'application retenue pour être implémentée dans le FPGA est toujours le cryptocore Triple-DES dans un souci de cohérence de l'ensemble des travaux et pour permettre la confrontation des résultats de la prédiction avec les résultats des mesures sous ions lourds. En plus de l'application de base, deux variantes implémentant les techniques de mitigation duplex et TMR ont été créées.

La Figure 3-18 présente les schémas blocs des trois versions de l'application triple-DES. Les trois variantes nécessitent un contrôleur de lecture et d'écriture des données dans l'application. Sa tâche est de remplir les registres, contenant la donnée à crypter (64 bits), les trois clés de cryptage (56 bits chacune), au démarrage de l'application et de fournir le résultat (64 bits) à la fin. Grâce au contrôleur un seul bus bidirectionnel de 64 bits est requis au lieu de 296 entrées/sorties.

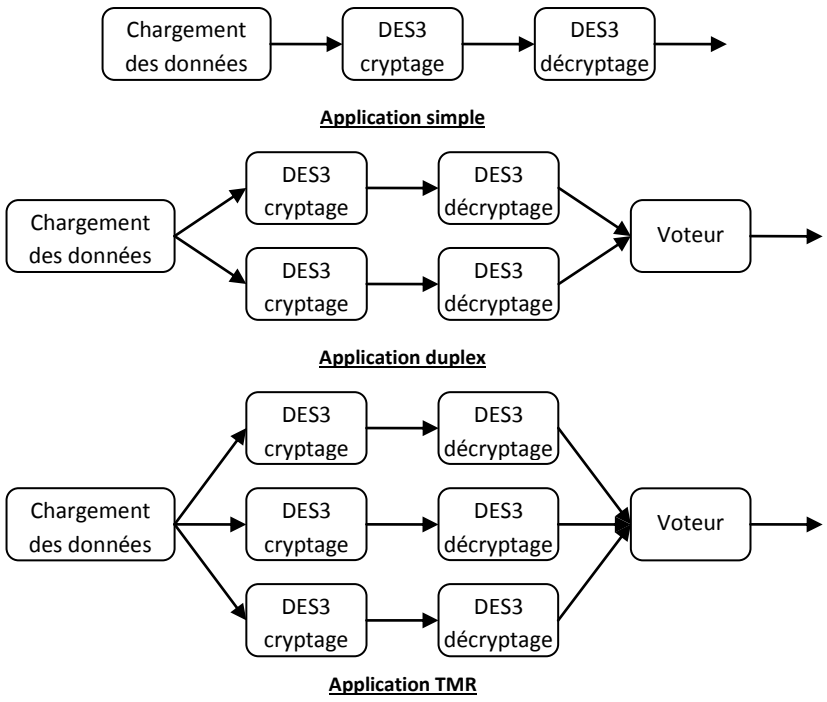

Figure 3-18 : Schéma blocs des trois variantes de l'application Triple-DES

L'application simple est donc composée du contrôleur de chargement des données et d'une branche de calcul composée de deux Triple-DES en calcul. Le premier cœur crypte la donnée puis la transmet au deuxième qui effectue l'opération inverse. La valeur de sortie doit donc être égale à celle d'entrée.

L'application duplex comporte deux branches de calcul identiques à celle de l'application simple. De plus, un voteur en sortie informe si les deux résultats sont identiques en plaçant la valeur « 0 »

dans un registre de quatre bits lisible par le testeur THESIC+. Lorsque les résultats divergent le registre prend la valeur « 1 ». Les autres valeurs (« 2 » à « 7 ») ne sont pas utilisées.

L'application TMR est composée de trois branches de calcul identiques à celles de l'application simple. Ici aussi un voteur à la majorité sélectionne le résultat jugé correct et positionne un registre de quatre bits en fonction de ce qu'il détecte :

- « 0 » lorsque tous les résultats sont identiques.
- « 1 » lorsque le résultat issu de la branche numéro 1 est différent des deux autres.
- « 2 » lorsque le résultat issu de la branche numéro 2 est différent des deux autres.
- « 3 » lorsque le résultat issu de la branche numéro 3 est différent des deux autres.
- « 4 » lorsque les trois résultats sont différents.
- Les valeurs « 5 » à « 7 » ne sont pas utilisées et ne devraient donc jamais être rencontrées.

Le choix de mettre deux Triple-DES en cascade dans chaque branche permet de maximiser la place occupée dans le FPGA. Trois cœurs de cryptage dans chaque branche pas pu être réalisée car le TMR ne disposait plus alors d'assez de ressources.

Le Tableau 3-13 résume la répartition des ressources logiques du FPGA pour les trois variantes de l'application Triple-DES. On peut remarquer que le duplex nécessite deux fois plus de ressources que l'application simple et le TMR trois fois plus. Ceci est conforme au nombre de branches de calcul présentes dans l'application (une pour l'application simple, deux pour le duplex et trois pour le TMR). Cette règle s'applique pour tous les types de ressources sauf pour les flip-flops présentes dans les slices. Cette exception peut être expliquée par l'importante utilisation de ce type de ressources qui est faite par le contrôleur de lecture/écriture des données dans l'application. En effet celui-ci occupe 532 Flips-Flops, ce qui laisse 266 Flip-Flops pour la fonction de cryptage. Or 266 est exactement ce qu'il faut ajouter pour passer de l'application simple au duplex, mais aussi du duplex au TMR.

Tableau 3-13 : Utilisation des ressources du FPGA par chaque variante de l'application Triple-DES

Type de ressource	DES simple		DES duplex		DES TMR	
Utilisation de la logique:						
Flip-Flops dans les Slices	798	(7 %)	1064	(10 %)	1330	(12 %)
LUTs	2176	(21 %)	4397	(42 %)	6831	(66 %)
Distribution des ressources logiques :						
Slices	1434	(28 %)	2607	(50 %)	3868	(75 %)
Nombre total de LUTs	2180	(21 %)	4409	(43 %)	6849	(66 %)
Détail de l'utilisation des LUTs :						
LUTs utilisées pour la logique	1920		3885		6063	
LUTs utilisées pour le routage	6		12		18	
LUTs utilisées comme ROMs	254		512		768	

Les deux méthodes d'injection de fautes présenté au chapitre 3.5.1 ont été testées afin d'évaluer l'erreur introduite par l'injection de fautes à la configuration du FPGA. Les deux protocoles de test sont ainsi illustrés en Figure 3-19.

L'injection de fautes avec interruption de l'application consiste dans un premier temps à générer de manière aléatoire les trois vecteurs de test. C'est-à-dire le vecteur « BYTE » qui correspond à l'adresse de l'octet dans le bitstream de configuration où la faute est injectée, le vecteur « BIT » qui est le numéro du bit dans l'octet et enfin le vecteur « ADDR » qui contient le numéro du cycle d'horloge durant lequel l'application doit être mise en sommeil pour permettre la modification de la mémoire de configuration. Les vecteurs sont ensuite copiés dans THESIC+. Le contrôleur de test dans le testeur lance alors la session d'injection de fautes en configurant le FPGA. L'application est ensuite exécutée puis interrompue lorsque le cycle d'horloge correspondant au vecteur « ADDR » est rencontré. Un readback du composant est effectué et le masque contenu dans BIT est appliqué à l'octet désigné par BYTE. Le bitstream ainsi altéré est remis en forme afin de reconfigurer le FPGA. L'exécution de l'application reprend alors son cours. Enfin lorsque celle-ci s'achève le testeur enregistre les résultats fournis et indique à l'ordinateur que le test est terminé et qu'il peut récupérer les données. Etant donné que les valeurs d'entrée et les valeurs de sortie attendues du cryptocore sont connues, il est donc possible de savoir si le TMR a laissé passer un résultat faux.

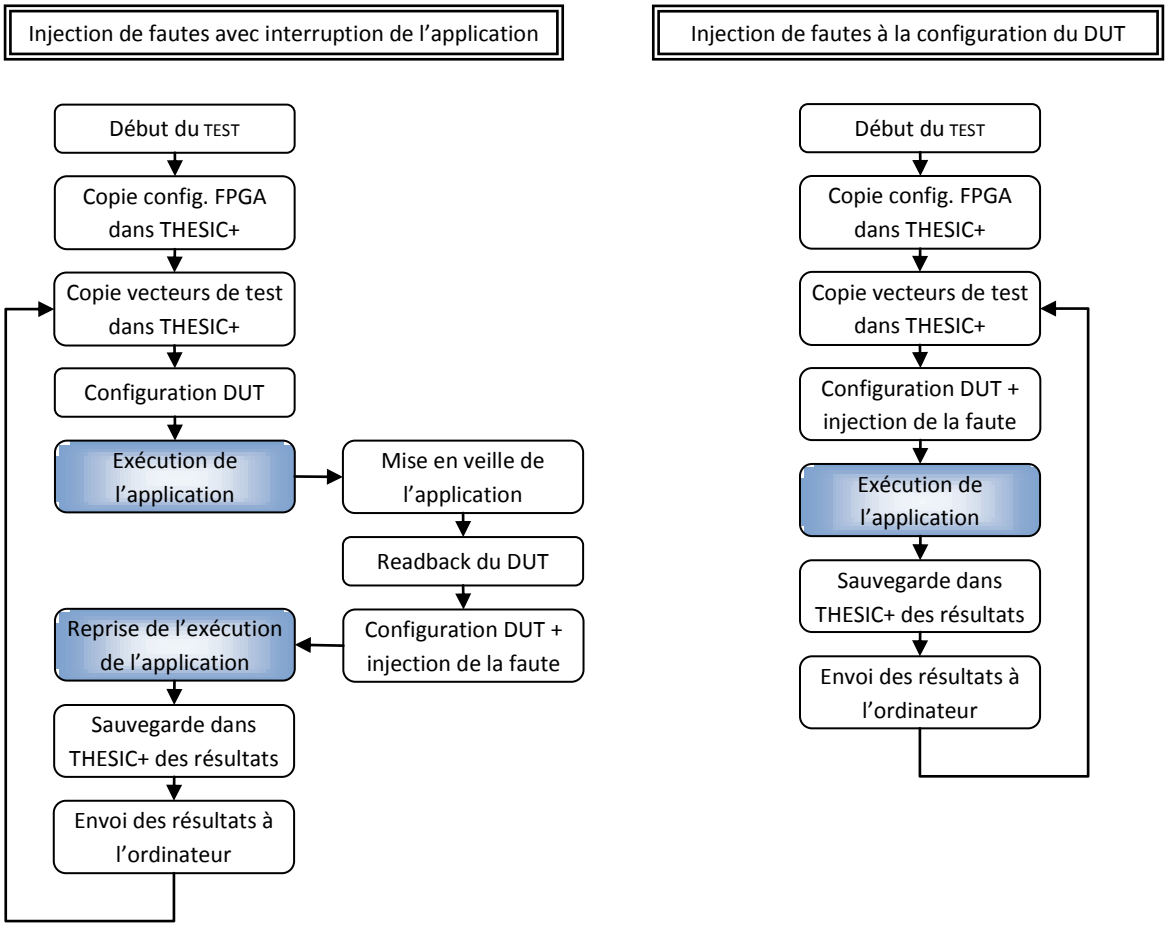

Figure 3-19 : Diagrammes des protocoles de test pour l'injection de fautes avec interruption de l'application et pour l'injection de fautes directement à la configuration

Le deuxième protocole de test avec injection de la faute directement dans la configuration du FPGA consiste à générer plus que deux vecteurs de test : « BYTE » et « BIT ». Ils sont ensuite envoyés dans THESIC+ avec le fichier de configuration. La séquence d'injection de fautes peut alors débuter

par la configuration du DUT avec injection de la faute à la volée à l'endroit spécifié par les vecteurs de test. L'application est alors exécutée puis le résultat du cryptage est sauvegardé dans la mémoire de THESIC+ avec la valeur du registre indiquant la pertinence du résultat. Enfin le contrôleur de THESIC+ compare la valeur cryptée avec la valeur attendue et indique une éventuelle différence. L'ordinateur peut alors récupérer ces données puis passer à l'injection suivante en générant deux nouveaux vecteurs de test…

3.5.3. Etude de la rémanence des fautes dans le FPGA

Afin d'étudier la rémanence des fautes dans le FPGA une campagne spécifique d'injection de fautes a été menée. Au cours de celle-ci une faute est injectée dans l'application suivant le mécanisme d'interruption, en prenant donc en compte le paramètre temporel. Une fois l'exécution de l'application achevée, les résultats sont stockés en mémoire, comme lors des campagnes décrites au chapitre 3.5.2, puis l'application est exécutée une deuxième fois et les nouveaux résultats sont conservés. Cette démarche est décrite dans le diagramme blocs donné en Figure 3-20.

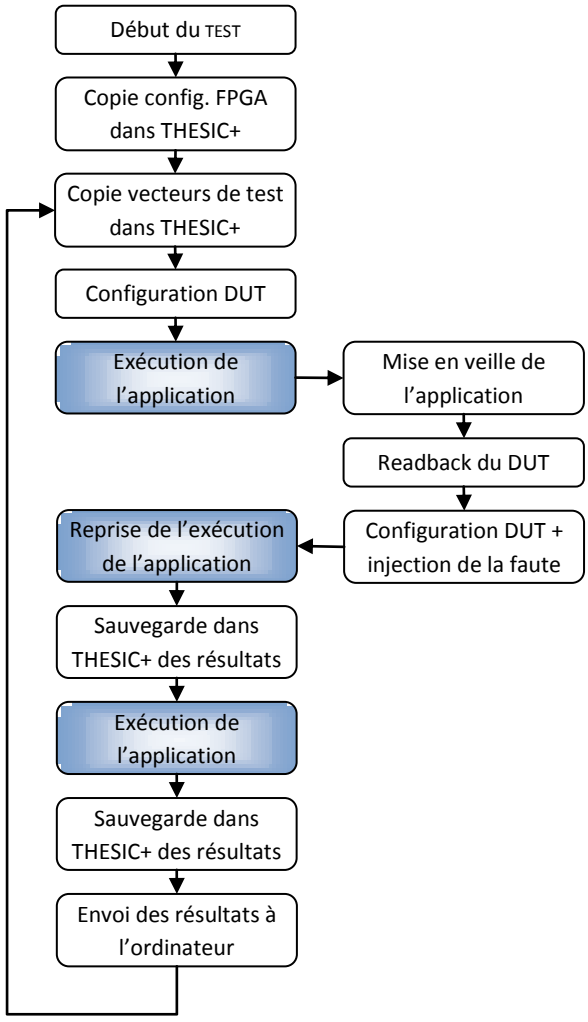

Figure 3-20 : Diagrammes des protocoles de test pour l'injection de fautes avec interruption de l'application et pour l'injection de fautes directement à la configuration

Ce protocole d'injection de fautes a été appliqué à l'application DES3 dans sa version TMR et les résultats sont présentés dans le Tableau 3-14. Ainsi pour 326.328 fautes injectées, 14.218 (soit 4,36 %) ont permis d'observer au moins une erreur dans au moins une des deux exécutions consécutives de l'application. De plus un nombre plus important d'erreurs dans l'application est observé lors de la seconde exécution, à part pour les erreurs faussement détectées. Cela pourrait s'expliquer par le fait que ces erreurs affectent un registre, qui est donc un ensemble de point mémoire de l'application, et le paramètre temporel est non négligeable dans ce cas.

Un nombre d'erreurs plus important lors de la deuxième exécution est le résultat attendu puisque la faute est présente dès le début de l'exécution de l'application, par conséquent elle affecte forcement le résultat.

Tableau 3-14 : Type est répartition des erreurs lors de la campagne daa

	Erreurs détectées	Fausses détection d'erreur	Erreurs non détectées	Nombre total d'erreurs
A la fin de la 1ère exécution de l'application	11.237 *(3,443 %)*	178 *(0,055 %)*	235 *(0,072 %)*	11.472 *(3,515 %)*
A la fin de la 2nde exécution de l'application	13.296 *(4,074 %)*	161 *(0,049 %)*	350 *(0,107 %)*	13.646 *(4,182 %)*

Afin de comparer l'efficacité des deux méthodes d'injection de fautes, il semble pertinent de s'intéresser aux injections de fautes qui ont engendrées un résultat issu de la première exécution de application différent de celui issu de la deuxième exécution. Les valeurs fournies dans le Tableau 3-15 permettent d'isoler trois catégories :

- **Les injections de fautes se traduisant par deux résultats faux et identiques** signifient qu'elles affectent des ressources décrivant l'application (par opposition aux ressources initialisant des points mémoire de l'application) et qu'elles ont eu lieu avant que la ressource soit utilisée pour le calcul. Cette catégorie représente 13,32 % des injections de fautes donnant une erreur.
- **Les injections de fautes donnant un résultat faux lors de la première exécution mais un résultat correct lors de la deuxième exécution** affectent des ressources qui initialisent le contenu d'un point mémoire de l'application. Ainsi lors de la deuxième exécution ces SEUs sont « écrasés » par l'application et n'ont pas d'influence sur le résultat. Cela représente seulement 2,82 % des erreurs.
- **Les injections de fautes ne donnant pas d'erreur lors de la première exécution mais un résultat faux lors de la deuxième exécution** correspondent à une injection ayant eu lieu après que la ressource du FPGA ait été utilisée. Elle ne donne donc pas d'erreur sur le calcul en cours. Cependant pour le deuxième calcul elle est présente dès le début donc elle fausse le résultat. Il est important de noter que l'injection par interruption de l'application permet elle aussi de mettre en évidence la faute, c'est juste le moment d'injection qui ne le permet pas. Ce type d'erreur concerne 17,83 % du nombre d'injection de fautes perturbant l'application.

La première catégorie de résultat est indépendant de la méthode d'injection de la faute car dans les deux cas la faute se traduit par une erreur sur les sorties de l'application. Par contre, La deuxième catégorie avantage la méthode utilisant l'interruption de l'application puisqu'elle met en évidence un résultat faux alors que l'autre méthode ne le permet pas. Enfin, la tendance inverse est observée dans le dernier cas. La méthode par injection de la faute lors de la configuration du FPGA introduit donc bien une erreur dans le calcul de la prédiction puisqu'elle ne permet pas d'observer certaine erreurs. Cependant elle représente moins de 3 % de la population des erreurs observées pour cette application. Cela peut s'expliquer par le faible nombre de cellule mémoire utilisée part l'application DES3. En effet, le Tableau 3-13 indique que la variante TMR n'occupe que 1330 flip-flops (soit 12% des 10.240 disponibles dans le FPGA) et seulement 768 LUTs (soit 7,5 % des 10.240 LUTs) utilisées comme ROM.

Tableau 3-15 : Nombre d'injections de fautes donnant deux résultats différents

	Nombres d'injection de fautes	Pourcentage par rapport au nombre total d'injections
Deux résultats faux et identiques	1.894	13,32 %
Résultat 1 faux, résultat 2 juste	401	2,82 %
Résultat 1 juste, résultat 2 faux	2.535	17,83 %
Résultats différents	4.830	33,97 %

En conclusion la méthode d'injection de fautes à la configuration du FPGA offre un réel gain de performance, au détriment de la précision de la prédiction. Cependant, cette erreur dépend fortement de la nature de l'application et du type de ressources qu'elle utilise. Ainsi, après avoir fait cette évaluation, il est possible de choisir la méthode appropriée en fonction de son application.

3.5.4. Résultats des injections de fautes sur l'application Triple-DES

Trois sessions d'injections de 267.000, 212.000 et 426.000 fautes respectivement dans les applications « simple », « duplex » et « TMR » ont été menées et le Tableau 3-16 présente en six colonnes les résultats obtenus :

- La liste des applications
- Les « erreurs détectées » sont définies de la manière suivante :
    - Par le code d'erreur « 1 » pour l'application duplex. La valeur du résultat n'entre pas en compte étant donné que le voteur ne peut pas connaître le résultat correct. La valeur de sortie de l'application prend donc toujours le résultat issu de la 1ère branche du duplex. D'ailleurs la statistique fait que dans 50% des cas le résultat est juste, cela à cause des vecteurs de test générés de manière aléatoire. Ce qui prouve aussi l'efficacité de la fonction aléatoire.
    - Par les codes « 1 » à « 3 » pour l'application TMR et une valeur de sortie juste. Dans ce cas le TMR rempli complètement sa fonction en détectant l'erreur tout en fournissant un résultat correct.
- Les « fausses détection d'erreur » ne s'appliquent qu'à l'application TMR. Elles se traduisent par un résultat juste mais un code d'erreur différent de « 0 ».
- Les « erreurs non détectées » sont définies de la manière suivante :

  - Dès que le contrôleur de THESIC+ détecte un résultat faux pour l'application Triple-DES simple.
  - Dès que le contrôleur de THESIC+ détecte un résultat faux pour l'application duplex alors que le code d'erreur indique « 0 », autrement dit qu'il n'a pas détecté d'erreur.
  - Dès que le contrôleur de THESIC+ détecte un résultat faux pour l'application TMR alors que le code d'erreur indique « 0 », autrement dit qu'il n'a pas détecté d'erreur.
- Le nombre de fautes injectées, exprimé en pourcentage du nombre total de fautes injectées, qui ont donné lieu à une erreur visible sur la sortie.
- Le nombre total de fautes injectées au cours de la session.

Pour chaque type d'erreur est donné, entre parenthèses, la proportion des erreurs par rapport au nombre de fautes injectées, puis la proportion des erreurs par rapport au nombre total d'erreurs détectées.

Tableau 3-16 : Résultats des injections de fautes dans l'application Triple-DES et ses trois variantes

Applications	Erreurs détectées	Fausses détection d'erreur	Erreurs non détectées	Nb. de fautes injectées provoquant une erreur	Nombre total de fautes injectées
Simple	N/A *(0 %)* *(0 %)*	N/A *(0 %)* *(0 %)*	1.339 *(0,50 %)* *(100 %)*	1.339 *(0,50 %)* *(100 %)*	267.438 *(100 %)*
Duplex	4.853 *(2,28 %)* *(97,18 %)*	N/A *(0 %)* *(0 %)*	141 *(0,07 %)* *(2,82 %)*	4.994 *(2,35 %)* *(100 %)*	212.530 *(100 %)*
TMR	14.564 *(3,42 %)* *(96,32 %)*	237 *(0,06 %)* *(1,57 %)*	319 *(0,08 %)* *(2,11 %)*	15.150 *(3,55 %)* *(100 %)*	426.217 *(100 %)*

Le pourcentage de fautes injectées et provoquant une erreur visible sur la sortie de l'application s'élève à 0,50 % pour l'application simple, 2,35 % pour le duplex et 3,55 % pour le TMR. Une telle tendance était attendue puisque le nombre de ressources utilisées dans le FPGA augmente lui aussi donc la probabilité qu'une faute affecte une ressource suit. Cependant la croissance n'est pas proportionnelle au surcroit de ressources occupées.

Il faut noter la bonne efficacité des deux méthodes de mitigation qui permettent de détecter respectivement 97,18 % et 96,32 % des erreurs pour les applications duplex et TMR.

On note aussi une nette amélioration du nombre d'erreurs critiques (non détectées) dès l'application duplex qui permet de passer de 0,50 % à 0,07% par rapport au nombre total de fautes injectées, et cela malgré l'augmentation précédemment évoquée. L'application TMR ne fait pas mieux avec 0,08%. Le réel avantage du TMR sur le Duplex est que les erreurs détectées n'ont pas besoins d'être recalculées.

Le Tableau 3-17 résume les taux d'erreurs, pour les trois variantes de l'application Triple-DES, calculés à partir des valeurs du Tableau 3-16. Le taux d'erreurs est obtenu à partir du rapport entre le nombre de fautes détectées et le nombre total de fautes injectées.

Tableau 3-17 : Taux d'erreur des trois variantes d'application Triple-DES

Applications	Erreurs détectées	Fausses détection d'erreur	Erreurs non détectées
Simple	N/A	N/A	5,01 E^{-3}
Duplex	2,28 E^{-2}	N/A	6,63 E^{-4}
TMR	3,42 E^{-2}	5,56 E^{-4}	7,48 E^{-4}

Les sessions d'injections matérielles / logicielles de fautes permettent d'obtenir le taux d'erreurs pour une application donnée, c'est-à-dire le nombre moyen de fautes nécessaire pour obtenir une erreur visible sur la sortie. Ayant mesuré, en accélérateur de particules, la sensibilité intrinsèque des cellules mémoires du composant, cela pour plusieurs types d'ions lourds, il est alors possible de mettre en relation ces deux grandeurs enfin d'obtenir une prédiction du taux d'erreur de l'application pour un environnement donné. Le Tableau 3-18 présente la prédiction d'erreur des trois variantes d'applications pour les particules de carbone de d'argon dont les sections efficaces (rappelées dans le tableau) sont respectivement 2,79 E^{-3} cm²/composant et 5,68 E^{-2} cm²/composant. La prédiction est obtenue grâce au produit de la section efficace statique par le taux d'erreurs issu des essais sous ions lourds.

Tableau 3-18 : Prédiction d'erreur des trois variantes d'application Triple-DES pour les particules de carbone et d'argon

Applications	Erreurs détectées	Fausses détection d'erreur	Erreurs non détectées
$^{13}C^{4+}$: 2,79 E^{-3} cm²/composant			
Simple	N/A	N/A	1,40 E^{-5}
Duplex	6.19 E^{-5}	N/A	1,85 E^{-6}
TMR	9,54 E^{-5}	1,55 E^{-6}	2,09 E^{-5}
$^{40}Ar^{12+}$: 5,68 E^{-2} cm²/composant			
Simple	N/A	N/A	2,85 E^{-4}
Duplex	1,26 E^{-3}	N/A	3,77 E^{-5}
TMR	1,94 E^{-3}	3,16 E^{-5}	4,25 E^{-5}

3.6. Confrontation des mesures en tests accélérés aux prédictions par injections de fautes

Le Tableau 3-19 récapitule les taux d'erreurs mesurés sous ions lourds et les prédictions obtenues par injection de fautes pour l'application TMR et pour la particule d'argon. Le volume des mesures obtenues pour le carbone ne permet pas une statistique fiable, ils ne sont par conséquent pas mentionnés ici.

Tableau 3-19 : Confrontation du taux d'erreur mesuré sous ions lourds à la prédiction par injections de fautes pour l'application TMR

Taux d'erreurs	Particules	Erreurs détectées	Fausses détection d'erreur	Erreurs non détectées
Mesure	Carbone	1,04 E^{-4}	N/A	N/A
	Argon	2,84E^{-3}	6,67 E^{-6}	7,56 E^{-5}
Prédiction	Carbone	9,53 E^{-5}	1,55 E^{-6}	2,09 E^{-6}
	Argon	1,94 E^{-3}	3,16 E^{-5}	4,25 E^{-5}

L'observation de ces résultats montre que la prédiction est proche de la mesure. En effet, le taux d'erreur obtenu par prédiction est deux fois inférieur à celui mesuré pour les erreurs détectées et les erreurs non détectées. En revanche, la tendance s'inverse pour les erreurs faussement détectées où un taux mesuré quatre fois inférieur. Ceci pourrait être expliqué par le faible nombre d'erreurs de ce type observées sous ions lourds.

Au final, les prédictions fournies par la méthode d'injections de fautes s'avèrent être proches de la réalité avec un rapport de seulement deux. L'efficacité de la méthode sur ce type de composant est d'autant plus concluante qu'une méthode similaire appliquée à un processeur ASIC permet d'obtenir un écart entre prédiction et mesure de un à deux ordres de grandeurs [Peronnard 2008].

Chapitre 4. Application satellite COTS2

4.1. Description du projet LWS-SET

4.1.1. L'objectif de la mission

Alors que le soleil permet la vie sur la Terre, il produit aussi une quantité de particules hautement énergétiques et de radiations pouvant affecter les organismes vivants. Comprendre les changements du soleil et ses effets sur le système solaire, la vie et la société est le but du projet spatial LWS (Living With a Star). L'objectif de ce projet est de répondre aux trois questions suivantes :

- Pourquoi et comment varie le Soleil ?
- Comment les planètes, et la Terre en particulier, répondent à ces variations ?
- Quels sont les effets sur l'humanité ?

Au même titre qu'il existe des changements de climat sur Terre, il existe aussi des changements de « climat » dans l'espace, et c'est le Soleil qui est responsable de ces modifications. En plus d'émettre en continu un flux de plasma, appelé vent solaire, le Soleil éjecte périodiquement des millions de tonnes de matière : les éjections de masse coronale (CME – Coronal Mass Ejection). Ces immenses nuages de matière causent d'énormes tempêtes magnétiques dans la magnétosphère et la haute atmosphère de la Terre. Le terme « climat spatial » fait généralement référence aux conditions sur le Soleil et dans les vents solaires, la magnétosphère, l'ionosphère et la thermosphère qui peuvent influencer les performances et la fiabilité des équipements spatiaux et au sol, ainsi que mettre en danger les êtres vivants.

Les tempêtes magnétiques produisent les effets observables suivant [Web LWS] :

- Les aurores boréales et les aurores australes.
- Les perturbations dans les communications.
- Les risques d'irradiation des astronautes et des engins spatiaux.
- Les pics de courant dans les lignes électriques.

Deux groupes d'engins spatiaux sont prévus dans le cadre du projet LWS de recherche sur le « *climat spatial* » :

- Le premier porte sur l'étude de la dynamique du soleil, c'est-à-dire l'observation depuis l'héliosphère des turbulences qui se forment à la surface de l'étoile.
- Un second engin placé dans la magnétosphère et l'ionosphère pour caractériser la réponse du proche espace[9] aux variations du soleil et aux vents solaires.

Le projet SET (Space Environment Testbeds) effectuera des vols et des investigations pour le projet LWS dans le but de comprendre comment l'interaction entre le Soleil et la Terre affecte l'humanité. SET se base sur des données existantes ainsi que de nouveaux résultats issus de missions spatiales à bas coût afin de :

- Préciser les mécanismes induits par l'environnement spatial et ses effets.
- Réduire les incertitudes dans les définitions de l'environnement et les effets induits sur les engins spatiaux et de leurs charges utiles.

[9] Cela comprend les hautes couches de l'atmosphère, l'ionosphère et la magnétosphère.

- Améliorer les flots de conception et des protocoles de test, afin que les pannes et anomalies de fonctionnement dues à des effets de l'environnement sur les engins spatiaux pendant les opérations soient réduites.

4.1.2. Description de l'engin spatial

Une description du satellite est donnée en annexe B

L'engin spatial oscillera entre une orbite base à 6000 km et une orbite haute a 12.000 km avec une inclinaison de 28°. Le lancement était initialement prévu pour fin 2007 / début 2008 mais dû à des retards, il n'aura pas lieu avant 2011.

4.2. Déroulement de l'étude

Dans le cadre du projet LWS-SET nous avons étudié, conçu, réalisé et mis au point une carte électronique afin qu'elle soit embarquée dans le satellite DSX. Le but de cette expérience est d'étudier le comportement d'un COTS, en fonctionnement en environnement radiatif, dans lequel il a été implémenté une application tolérante aux fautes. En l'occurrence le composant cible est le FPGA Xilinx Virtex-II. Cela implique :

- L'observation des résultats issus de l'application et la comparaison avec ceux attendus.
- La relecture de la configuration du FPGA et la comparaison avec le bitstream de référence afin détecter les upsets provoqués par les particules présentes dans l'environnement spatial.

Pour cela l'étude a été décomposée en six étapes:

1. Etude du cahier des charges.
2. Définition de l'architecture de la carte.
3. Etude, conception, réalisation et mise au point de la maquette et de son banc de test. La maquette permet de valider les solutions techniques retenues lors de l'étape précédente.
4. Etude, conception, réalisation et mise au point d'un prototype et de son banc de test. Le prototype se veut être le plus proche possible de la carte finale.
5. Réalisation et mise au point du modèle de vol qui sera mis en orbite.
6. Intégration dans le satellite et tests environnementaux de la carte.

L'étude du cahier des charges est présentée dans le paragraphe suivant où sont également décrits l'environnement fourni par le satellite et son architecture ; à savoir les fonctionnalités, comme la gestion des communications avec le sol, mais aussi les contraintes électriques et mécaniques.

4.3. Définition de l'architecture de la carte COTS2

Le but de l'expérience est d'observer le comportement d'une application exécutée sur le Virtex-II alors qu'il est soumis au rayonnement naturel. Cela implique d'avoir un contrôleur externe robuste afin qu'il ne soit pas lui-même perturbé par les radiations. Il doit être capable de gérer deux activités :

- Effectuer les fonctions de configuration et de readback sur le FPGA et de détection d'erreur sur les sorties de l'application.

- Répondre aux commandes envoyées par le satellite. Cette tâche est absolument prioritaire car, en cas de non réponse répétée, la carte serait considérée comme défectueuse et désactivée jusqu'à analyse du problème et prise de décision par les équipes de maintenance et de développement de la carte COTS2 au sol.

Deux pistes ont été examinées et testées afin de trouver une solution convenable pour le contrôleur externe.

- La première s'est orientée vers l'utilisation d'un FPGA anti-fusible d'Actel. Cependant une alerte concernant la fiabilité des composants RTSX-S et SX-A gravés en 25µm avec la technologie MEC/Tonami a été émise par la NASA qui nous a demandé de trouver une solution alternative. De plus, le contrôleur de communication occupait à lui seul 90% du FPGA, donc il aurait fallu s'orienter vers une plateforme multi-FPGAs avec les problèmes de synchronisation des puces, de consommation et de place occupée que cela aurait induit.
- La deuxième solution est l'utilisation d'un processeur générique et c'est celle qui a été retenue.

La partie contrôle de la carte est allouée au processeur durci Atmel AT697E [Atmel AT697E] basé sur le processeur LEON[10]. Il s'agit d'un processeur 32 bits basé sur l'architecture SPARC V8.

Tous les composants, à l'exception du DUT, ont été choisis « durcis » afin de ne pas être perturbés par le milieu radiatif dans lequel ils vont devoir fonctionner. La Figure 4-1 décrit l'architecture haut niveau de cette carte. On peut y voir les éléments suivant :

- Le processeur durci Atmel AT697E en boitier MCGA 349. Ce boitier ressemble à un boitier BGA à la différence que les billes sont remplacées par des colonnes.
- Une mémoire durcie de type ROM contenant le code du processeur et une mémoire durcie de type RAM stockant les variables du programme.
- Le DUT, c'est-à-dire le Xilinx Virtex-II XC2V1000.
- Des mémoires flash Xilinx XQR18V04 qui sont tolérantes aux fautes. Il est nécessaire d'avoir deux puces pour chaque configuration du FPGA : une pour la configuration et l'autre pour l'image de référence de la mémoire afin d'effectuer la comparaison avec le *readback* du FPGA.
- Deux thermistances et un dosimètre.
- L'interface de communication, c'est-à-dire le driver et le récepteur RS-422, sont eux aussi durcis.
- Un bloc d'alimentation permettant de générer les tensions de 1,5V et 1,8V qui sont respectivement les tensions de cœur du FPGA et du processeur.

A l'origine trois jeux de configuration étaient prévus afin de tester différentes applications. Les tests en accélérateur de particules, effectués a posteriori de l'étude de l'architecture, ont permis d'obtenir l'estimation du taux d'erreur attendu lors du vol. Ces résultats étant relativement faibles, ils ne permettent pas d'établir une statistique fiable pour chacune des trois configurations. Il a donc fallu se limiter à une seule configuration sur la carte de vol.

[10] IP développée par Aeroflex Gaisler[Web Gaisler] pour le CNES (Centre National d'Etudes Spatiales).

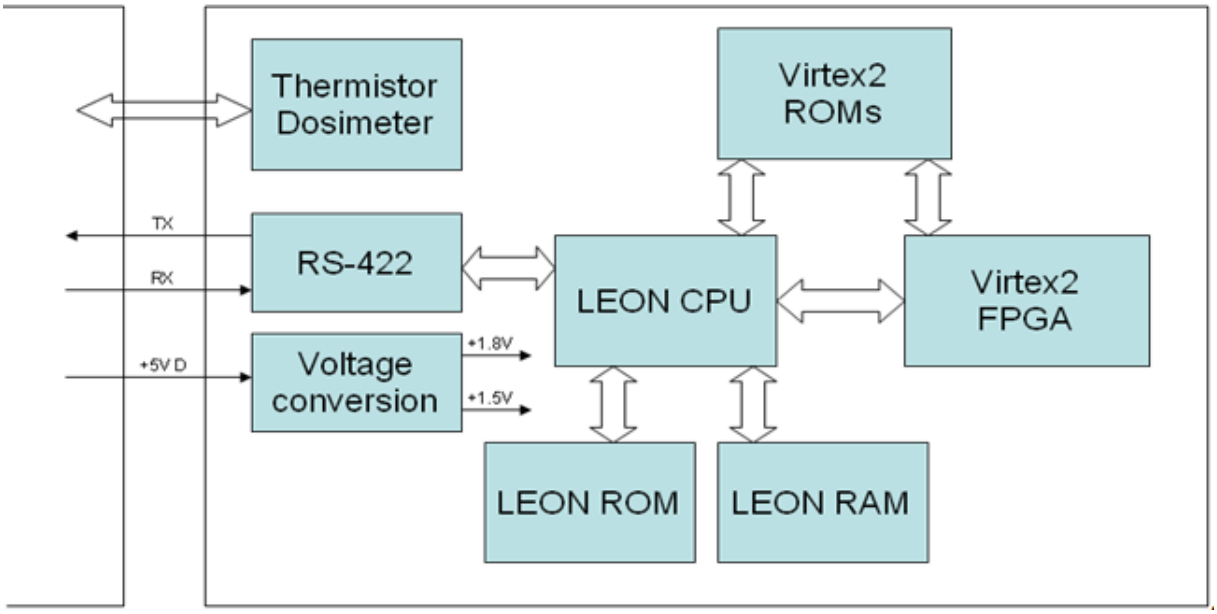

Figure 4-1 : Architecture haut niveau de la carte COTS2

4.4. Etude, conception, réalisation et mise au point de la maquette

Le but de la maquette est de vérifier la validité des solutions techniques retenues, cela en construisant une carte se rapprochant le plus possible de la carte de vol finale. Cependant, l'utilisation de composants qualifiés pour l'environnement spatial n'est pas envisageable pour une maquette car ils sont extrêmement couteux et surtout parce qu'il faut compter six à douze mois pour les obtenir. Il a donc fallu trouver des composants commerciaux équivalents non durcis.

Les objectifs de la maquette sont les suivants :

- Initialiser et faire fonctionner le processeur LEON.
- Etablir une communication avec l'ordinateur en respectant le standard HDLC fixé par le cahier des charges.
- Configurer le FPGA à partir du processeur.
- Effectuer le Readback à partir du processeur.
- Avoir la possibilité d'effectuer des tests accélérés au sol.

Il a été décidé de rajouter comme objectif la possibilité de pratiquer des essais en accélérateur de particules et sous faisceau laser, ceci afin de valider la capacité de détection des SEUs. De plus, cette carte a permis d'effectuer une campagne préliminaire de caractérisation du composant en accélérateur de particules en attendant que la plateforme pour les tests au sol soit opérationnelle. La carte a donc été équipée d'un FPGA décapoté et aminci pour permettre les essais au sol. Le processeur et le débit de la liaison série sont adéquats pour le volume de données à traiter dans l'espace. Cependant, un test sous radiation produit un nombre de résultats beaucoup plus grand et il faut prévoir une unité de traitement en conséquence. Pour cela une interface de connexion au testeur THESIC+ a été réalisée. Dans ce cas le processeur est inhibé et c'est THESIC+ qui se charge de conduire le test. Les données sont alors transmises à l'ordinateur via le lien Ethernet 100Mbits.

Ces contraintes appliquées à l'architecture exposée au chapitre 4.3 ont permis de dériver la variante proposée ci-après.

4.4.1. Architecture de la maquette et du banc de test

La Figure 4-2 illustre l'architecture de la maquette et de son banc de test. Le banc de test reçoit les tensions +5V numérique et +/- 15V analogiques depuis une alimentation stabilisée de laboratoire. Les régulateurs linéaires ajustables produisent les tensions +3,3V numérique et +/- 5V analogiques identiques à celles fournies par le satellite.

Les tensions symétriques analogiques alimentent le module de détection de *latchup*, de la carte COTS2, donc le rôle est d'interrompre la distribution du 3,3V de toute la partie numérique lorsqu'un latchup est détecté. Ceci afin d'éviter la destruction partielle ou complète d'un composant.

Deux régulateurs linéaires produisent les tensions requises par le cœur du processeur et le cœur du FPGA.

Le processeur lit son code dans la mémoire ROM et utilise la mémoire RAM pour les variables temporaires. Aucune ROM en 3,3V certifiée pour le spatial n'existait au moment de la conception. Il a donc fallu opter pour un composant alimenté en 5V et donc prévoir des buffers (non représentés) pour effectuer l'adaptation entre les niveaux 5V de la ROM et les niveaux 3,3V des entrées/sorties du processeur.

Le bus de configuration du FPGA est connecté à la fois au processeur, aux mémoires Xilinx et à THESIC+. Cela permet dans un fonctionnement spatial de commander la configuration du FPGA et d'effectuer le *Readback* depuis le processeur. Pour un fonctionnement en test accéléré au sol, le processeur et les mémoires ROMs sont désactivées et toutes les opérations sont commandées depuis THESIC+. Les mémoires Xilinx sont conçues et optimisées pour configurer les FPGA de la marque et elles n'ont donc pas de bus d'adresses. A la place un compteur interne est incrémenté à chaque coup d'horloge et la ROM fournit la donnée suivante sur son bus de données. L'autre particularité est que ce compteur est réinitialisé lorsque l'on désactive les sorties de la puce. Or, le processeur lit à tour de rôle les données issues de la mémoire de configuration du FPGA et les données de référence stockées dans la Rom Xilinx. Les deux sources utilisent le même bus de données donc elles doivent pouvoir être isolées l'une de l'autre afin de ne par créer de conflit. Pour cela des buffers 3 états (non représentés sur la Figure 4-2) sont employés.

Le processeur intègre une UART (Universal Asynchronous Receiver Transmitter) capable de fournir le signal adéquat pour la liaison série. Il faut néanmoins un driver et un récepteur afin de transformer le signal numérique en signal analogique en respectant la norme retenue, ici la RS-422. La connexion se faisant à un ordinateur qui utilise la norme RS-232, un couple driver/récepteur RS-232 est donc retenu à la place du RS-422.

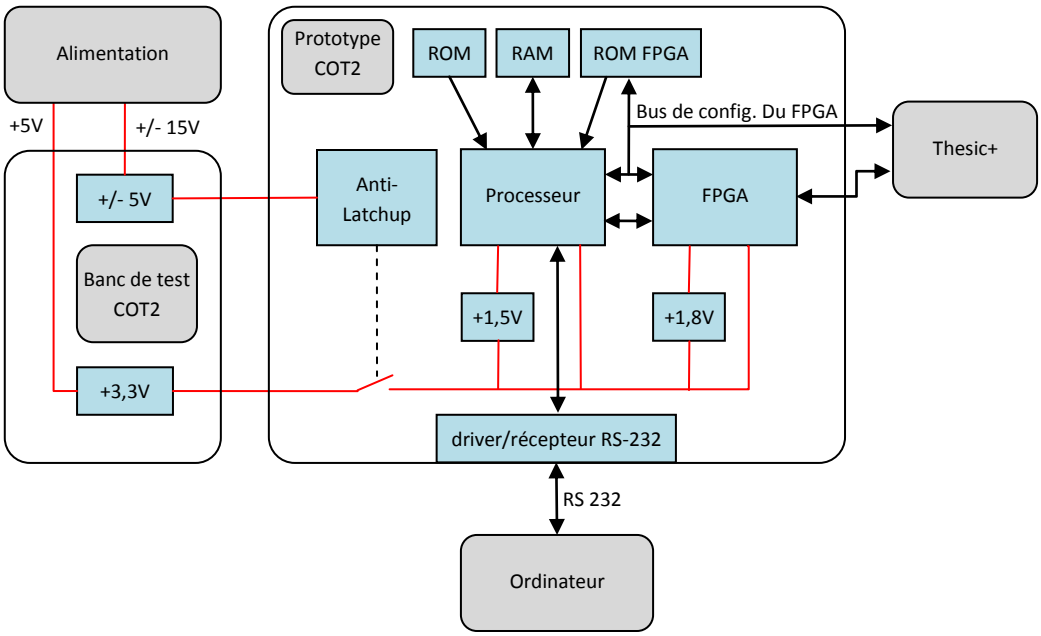

Figure 4-2 : Architecture de la carte prototype COTS2

4.4.2. Conception de la maquette et du banc de test

La complexité de la maquette, due aux deux composants en boitier BGA, a imposé la fabrication d'un circuit imprimé multicouches. Le banc de test est lui par contre réalisé sur une plaque à trous, supprimant ainsi les coûts élevés de l'outillage requis pour la fabrication d'un circuit imprimé à exemplaire unique. Principalement deux logiciels ont été utilisés pour la conception et la réalisation de la carte prototype :

- LTspice [LTspice4] à l'époque dans sa version 3 est un outil gratuit de simulation Spice développé par Linear Technology. Il a permis notamment de simuler la partie analogique du module anti-latchup.
- Altium Designer pour l'édition de schéma, la conception du circuit imprimé et la génération des fichiers pour la réalisation de la carte.

L'ensemble des schémas de la carte comprend 31 pages. Le circuit imprimé est une carte huit couches en classe 6 et dont les dimensions sont celles imposées par la NASA. Il est composé de quatre couches de signaux, deux couches de plans d'alimentation, distribuant les tensions 1,5V, 1,8V, 3,3V et 5V, et deux couches de plans de masse.

4.4.3. La réalisation de la maquette

Une partie du montage-câblage de la maquette est assurée par une société spécialisée équipée pour braser les boitiers de type BGA. Ils ont donc mis en place le FPGA et le processeur avant de livrer la carte.

Pendant la fabrication de la maquette nous avons pu câbler et tester le banc de test.

Nous avons brasé le reste des composants car cela ne pose pas de problème particulier malgré la petite taille de certains, mais surtout parce que cela permet de faire la mise en route de la carte étape par étape. C'est-à-dire câbler un module puis le mettre sous tension et le tester avant de passer au suivant.

En parallèle avec la conception de la maquette nous avons conçu et réalisé le banc de test permettant la validation. Ce banc de test doit reproduire l'environnement numérique auquel la carte COTS2 est soumise lorsqu'elle est connectée au module de charge utile.

4.4.4. Ecriture des logiciels

La réalisation de la maquette comprend toute la partie matérielle, mais aussi une grosse partie logicielle. En effet, qui dit processeur, dit application et il en va de même pour le FPGA. De plus un logiciel est aussi nécessaire pour faire fonctionner l'ordinateur afin de tester la communication avec la carte.

4.4.4.1. L'application du processeur

Le processeur doit exécuter deux tâches principales :

- Gérer les expériences menées sur le FPGA. C'est-à-dire le configurer puis observer son activité ; cela en effectuant régulièrement des relectures de la configuration et en contrôlant les sorties de l'application du FPGA. En cas d'erreur un rapport est généré et stocké en mémoire. Il est envoyé au satellite sur sa demande.
- Répondre aux commandes envoyées par le satellite. Cette tâche est absolument prioritaire car, en cas de non réponse l'expérience serait considérée comme non fonctionnelle et serait désactivée. Pour cette raison l'arrivée d'une nouvelle trame HDLC est traitée au travers d'une interruption afin de suspendre toute autre exécution en cours.

Le diagramme fonctionnel donné en Figure 4-3 décrit le déroulement de la boucle principale du programme, boucle dans laquelle sont effectuées toutes les opérations sur le FPGA. La Figure 4-4 illustre les tâches effectuées afin de valider la pertinence d'une commande émise par le satellite et la réponse appropriée à fournir.

Le processeur est interconnecté avec le FPGA grâce à son GPIO (General Purpose Input.Output). Il s'agit d'un bus 32 bits entièrement configurable. Il peut être positionné en entrée, en sortie ou bien en mode bidirectionnel. La configuration et le Readback du FPGA se contrôlent grâce à 15 signaux :

- Un bus de données bidirectionnelles de 8 bits.
- Le signal d'entrée CE (Chip Enable) de validation du FPGA.
- Le signe d'entrée WE (Write Enable) indiquant le sens sur le bus de données.
- Le signal CLK (Clock) permettant de synchroniser les opérations. Ce signal peut être une sortie si le FPGA est positionné en maître ou bien une entrée s'il est esclave. Dans notre cas il s'agit d'une entrée.
- Le signal de sortie BUSY indique lorsque le FPGA n'est pas prêt et que le transfert doit être momentanément suspendu. En pratique il n'est pas utile pour une fréquence inférieure à 50MHz, ce qui est notre cas.
- Le signal de sortie DONE indique lorsque le FPGA est configuré et prêt à fonctionner.

- Le signal d'entrée PROG démarre la séquence d'initialisation de la mémoire de configuration lorsqu'il est maintenu à la valeur logique '0' durant au moins 300ns.
- Le signal de sortie INIT prend la valeur '1' lorsque la phase d'initialisation de la mémoire est achevée.

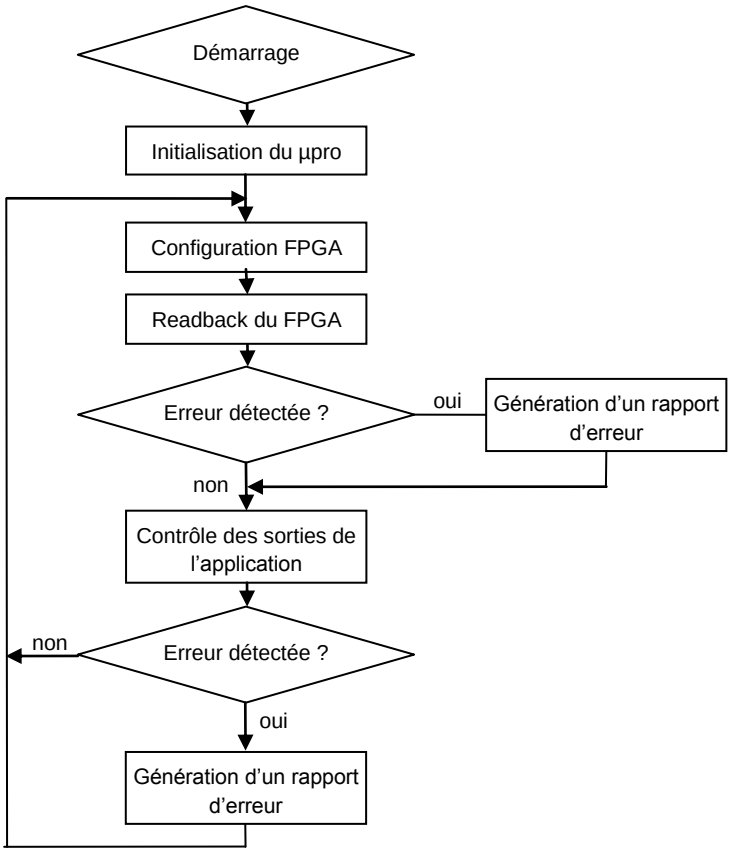

Figure 4-3 : Diagramme d'ordonnancement des tâches effectuées dans la boucle principale

La configuration débute par l'initialisation de la mémoire de configuration. Puis une commande de synchronisation permet d'initier la communication afin de positionner le FPGA en mode configuration. Viennent ensuite les données de configuration et, pour finir, la commande de désynchronisation de la logique de configuration du FPGA. Cela permet d'envoyer par la suite d'autres commandes au FPGA. A l'issue de ces étapes le signal DONE doit passer à '1' pour indiquer la réussite de l'opération.

Le readback débute par l'envoi de la commande de synchronisation puis de l'ordre de readback. Le FPGA passe alors en mode émission pour transmettre le contenu de sa mémoire. Une fois tous les octets lus, le FPGA repasse en mode réception en attente de la commande de désynchronisation.

Le code source du processeur est composé d'environ 3100 lignes réparties comme indiqué dans le Tableau 4-1.

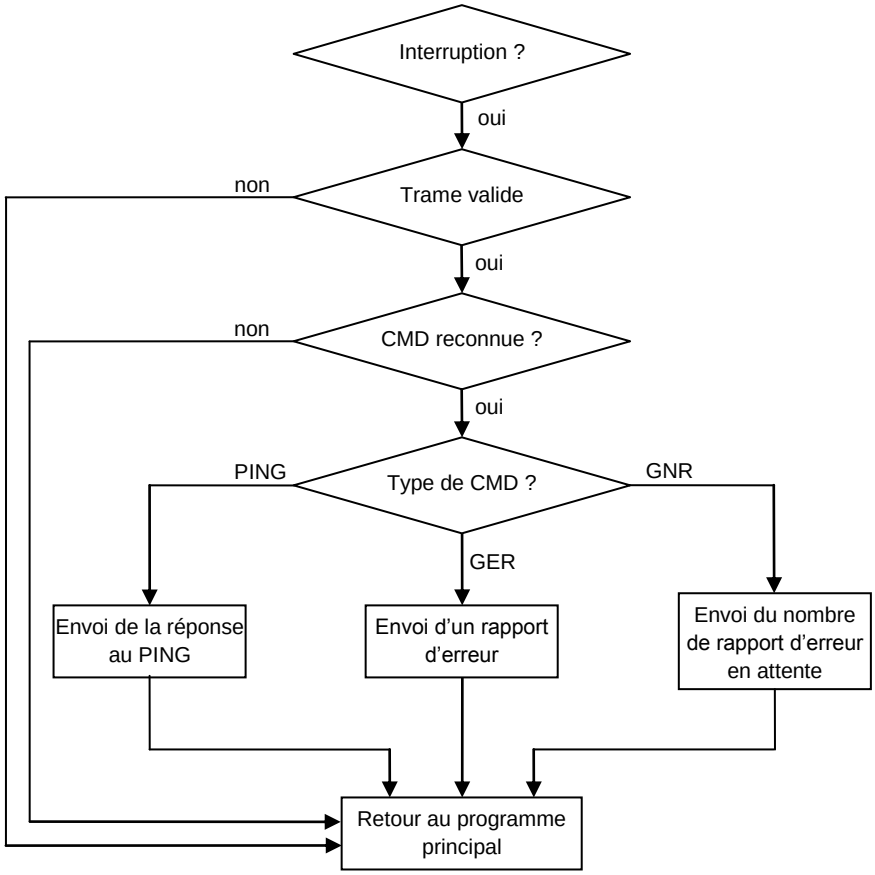

Figure 4-4 : Diagramme fonctionnel de la fonction d'interruption

Tableau 4-1 : Répartition de la taille du code source par fonction

Fonction	Nombre de lignes de code
Initialisation du processeur	150
Protocole de communication	750
Gestion du FPGA	1200
Divers	1000
Total	~3100

4.4.4.1. L'application du FPGA

L'application doit démontrer le fonctionnement du composant dans un environnement réel, sa fiabilité et ses potentiels talons d'Achilles. Pour cela, une application déjà utilisée dans un engin spatial serait souhaitable.

Le CNES a proposé un module du sous-Système de Contrôle d'Altitude et d'Orbite (SCAO) qui est déjà utilisé sur un satellite. Il s'agit d'un programme écrit en langage C qui nécessite par conséquent un processeur pour être exécuté. L'IP du processeur LEON3 peut très bien être implémenté dans le

Virtex-II. D'ailleurs nous avons déjà effectué ce travail pour les essais au sol. Il est décrit dans le chapitre 3. Cependant un tel processeur est extrêmement complexe et faire la relation entre l'inversion d'un bit dans la configuration et son utilisation au sein de l'application demanderait énormément de temps. D'autant que Xilinx ne fournit pas les informations permettant de le faire et il faudrait pour cela recourir à une phase de *reverse engineering*. D'autre part, l'application SCAO représente environ 1600 lignes de code et est composée d'un grand nombre de boucles imbriquées. Là encore l'analyse des résultats s'avèrerait très complexe et très longue.

Pour ces raisons il a été choisi une IP de cryptocore triple DES (Data Encryption Standard), aussi notée DES3. Il s'agit d'un algorithme de cryptage de données de 64 bits qui utilise trois clés de 56 bits. Ce programme est écrit par Rudolf Usselmann [Web TripleDES] et est disponible librement au téléchargement sur le site Opencores [Web Opencores]. L'algorithme simple DES a été adopté comme standard en 1976 puis rendu obsolète en 1999 à cause de la taille de ses clés devenues trop petites par rapport à la puissance de calcul des machines permettant des attaques systématiques en un temps raisonnable. De nos jours, le simple DES ne subsiste plus que dans d'anciennes applications, par contre le triple DES reste encore assez répandu. D'autre part, l'application n'est ni trop complexe, ni trop longue à exécuter et sa taille réduite permet sa duplication et même sa triplication au sein du FPGA.

Dans le DES, la donnée est cryptée suite à 16 itérations de transformation du bloc de 64 bits. Chaque itération utilise une clé partielle de 48 bits. Ces clés sont calculées à partir de huit tables de substitution (dites « S-Boxes » en anglais).

L'application retenue est un triple DES. En pratique elle effectue trois cryptages DES successifs à partir de trois clés. L'opération nécessite donc 48 itérations ou cycles d'horloge. L'inconvénient de cette solution est le nombre élevé d'entrées/sorties (64 bits de données + 3 x 56 bits pour les clés + quelques signaux de contrôle) ce qui ne permet pas l'interfaçage avec le processeur, qui lui ne dispose que de 10 bits. Les données à crypter et les clés sont donc stockées dans le FPGA.

L'application triple DES n'occupe que quelques pourcents du FPGA. Pour maximiser les chances qu'une particule frappe une ressource utilisée, l'application a donc été « tripliquée ». C'est la méthode dite TMR (Triple Modular redundancy) qui consiste à effectuer trois fois le même calcul en parallèle puis à comparer les valeurs de sortie par un vote à la majorité. L'avantage de cette méthode est de supporter la défaillance d'un des modules tout en gardant sa fonctionnalité. Au final le seul élément qui ne tolère aucune erreur est l'organe de vote. Par contre, elle oblige à multiplier par trois les ressources matérielles requises. Malgré l'application du TMR, l'application n'occupe toujours que 25% des slices du FPGA. Il a donc fallu utiliser huit opérations de cryptage/décryptage en cascade afin d'atteindre un taux d'utilisation de 75% des slices. Le diagramme de l'application est donné en Figure 4-5. La sortie du voteur indique si une chaîne de calcul donne des résultats faux ou bien si elle a cessé de fonctionner (timeout).

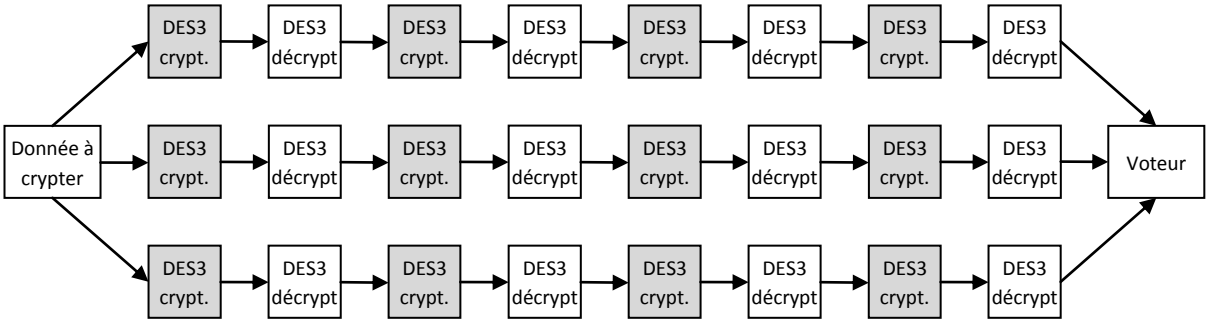

Figure 4-5 : diagramme de l'application triple DES du satellite

4.4.4.2. L'application de contrôle de la carte par l'ordinateur

Le logiciel exécuté depuis l'ordinateur doit générer et envoyer à la carte de vol les commandes comme le fera le satellite. Pour cela on utilise l'interface série RS-232. Le logiciel est écrit en langage C et le compilateur utilisé est Labwindows/CVI de la société National Instruments [Web NI]. Ce compilateur offre la possibilité de mettre facilement en place une interface graphique. Notamment grâce à sa bibliothèque d'objets prête à l'emploi tel que des boutons, des voyants, des graphiques, etc. Cette application nécessite environ 1.300 lignes de code.

La fenêtre du simulateur (voir Figure 4-6) est composée de trois parties :

- La zone située en haut à gauche permet d'initialiser la connexion sur l'ordinateur. Un bouton « Configure » ouvre la fenêtre configuration du port COM et de la vitesse de transmission. Deux afficheurs « Tx » et « Rx » informent sur le nombre d'octets présents respectivement dans les files d'attente d'émission et de réception. Enfin un bouton « Read in Queue » vide la file de réception.
- La zone située en bas à gauche dispose de trois onglets. Le premier permet d'envoyer manuellement la commande souhaitée parmi les trois prédéfinies (PING, GNR et GER). Le deuxième onglet autorise l'envoi d'une commande personnalisée. Enfin le dernier onglet sert à envoyer en permanence les commandes PING et GNR. Cela pour vérifier la robustesse du contrôleur de communication de la carte COTS2 (Figure 4-7). Ainsi des essais sur plusieurs jours ont pu être menés. La carte a toujours parfaitement répondu à plusieurs dizaines de milliers de commandes envoyées et n'a jamais donné de timeout ou d'erreur.
- Toute la moitié droite de l'interface est réservée à l'affichage des événements. C'est-à-dire les commandes envoyées, les réponses reçues et les erreurs. Une copie de ces événements est stockée dans un fichier texte.

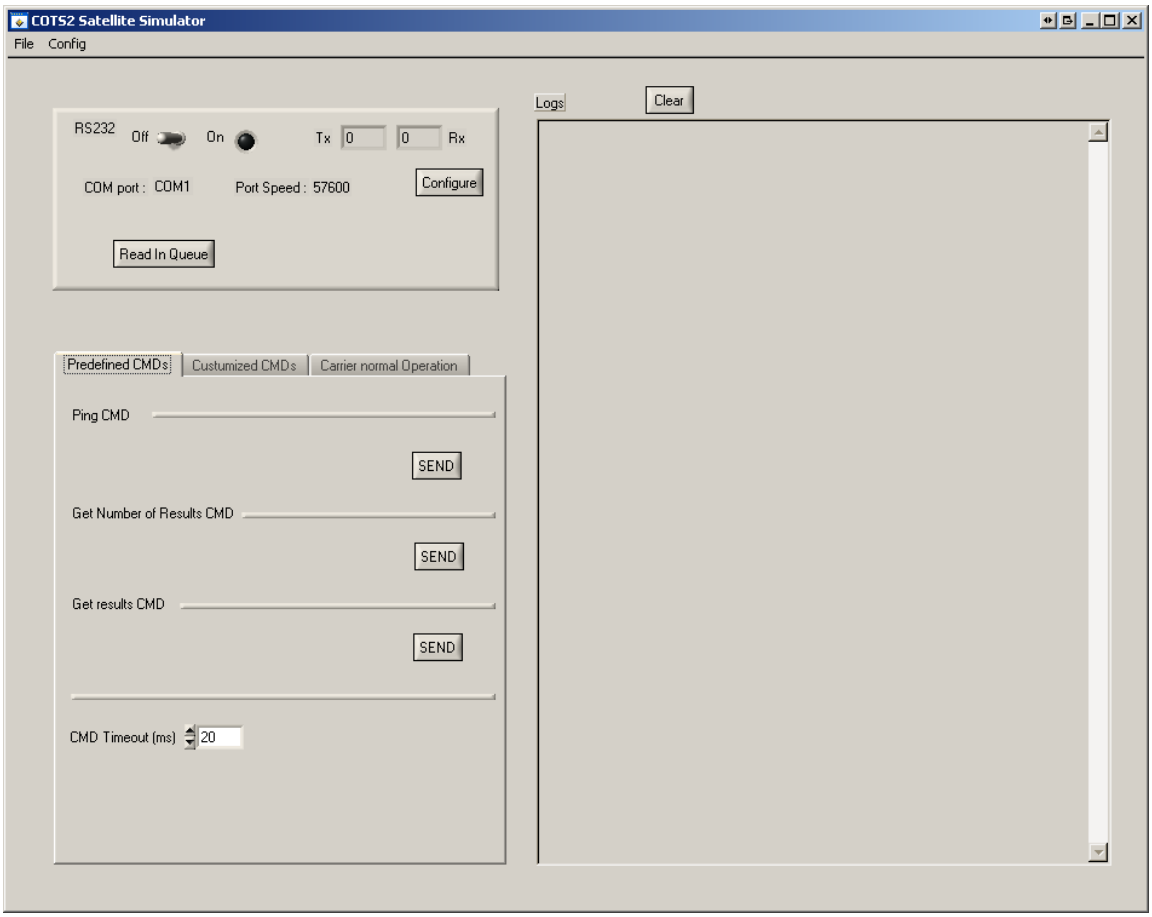

Figure 4-6: Interface graphique de l'application simulateur de satellite

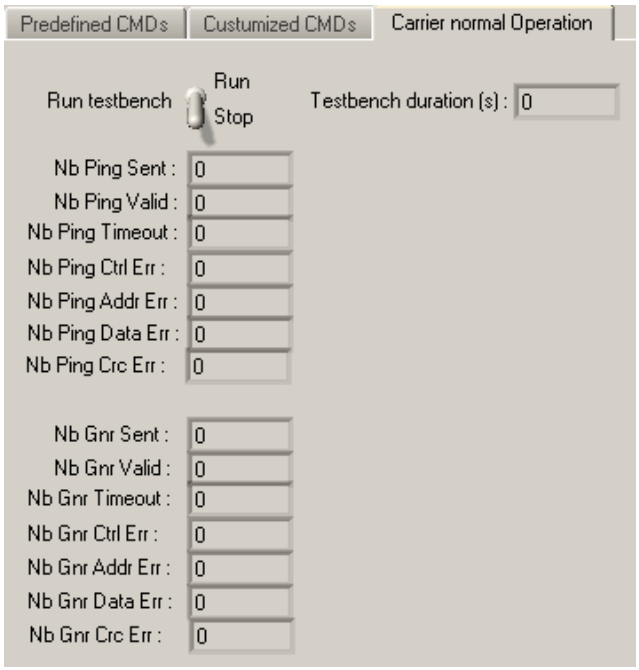

Figure 4-7 : Onglet de test de robustesse de la communication

4.5. Etude, conception, réalisation et mise au point du prototype et de la carte de vol

Cette étape devait à l'origine être décomposée en deux parties. Tout d'abord la conception, la réalisation et la mise au point d'un prototype de vol. Une fois validée, cela aurait pu donner lieu à une mise à jour en prenant en compte d'éventuelles modifications pour le modèle de vol. Les phases d'étude et de réalisation du circuit imprimé ont été effectuées par la NASA afin de respecter les standards du spatial. Les composants utilisés sont donc aussi certifiés « spatial ». TIMA a fourni les schémas en format électronique et une étude thermique a été effectuée en collaboration avec la NASA pour détecter d'éventuels problèmes de dissipation une fois dans le vide. Cette étude a permis aussi de fixer les contraintes pour le placement des composants. La NASA a routé et fait fabriquer le circuit imprimé qui nous a été livré à TIMA avec tous ses composants pour sa mise au point.

Cependant, la phase d'étude et de réalisation du circuit prototype à pris beaucoup de temps (près d'une année au lieu de deux mois prévus), ne laissant ainsi plus suffisamment de temps pour sa mise au point avant la réalisation de la carte de vol. La NASA s'est donc vu contrainte de commander un deuxième circuit imprimé identique à celui du prototype pour le modèle de vol.

4.5.1. Architecture de la carte de vol

L'architecture du modèle de vol est la même que celle du prototype étant donné que les circuits imprimés sont identiques. La Figure 4-8 illustre la carte interfacée avec le module de charge utile.

Il est à noter qu'un système anti-latchup était présent sur le prototype et qu'il a été retiré de la carte de vol car entre temps la NASA a intégré un système similaire à sa carte de contrôle des alimentations, rendant ainsi le module de la carte COTS2 inefficace.

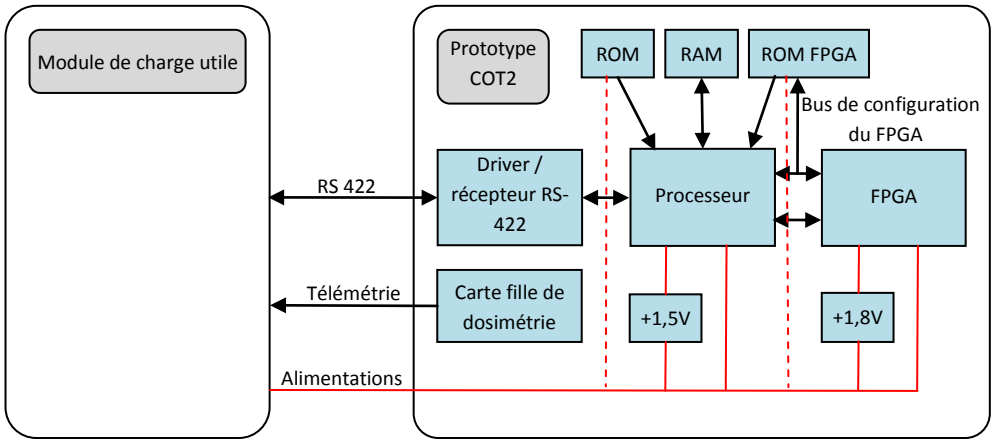

Figure 4-8 : Architecture de la carte de vol COTS2

La Figure 4-9 et la Figure 4-10 montrent respectivement la carte de vol face composant et face soudure.

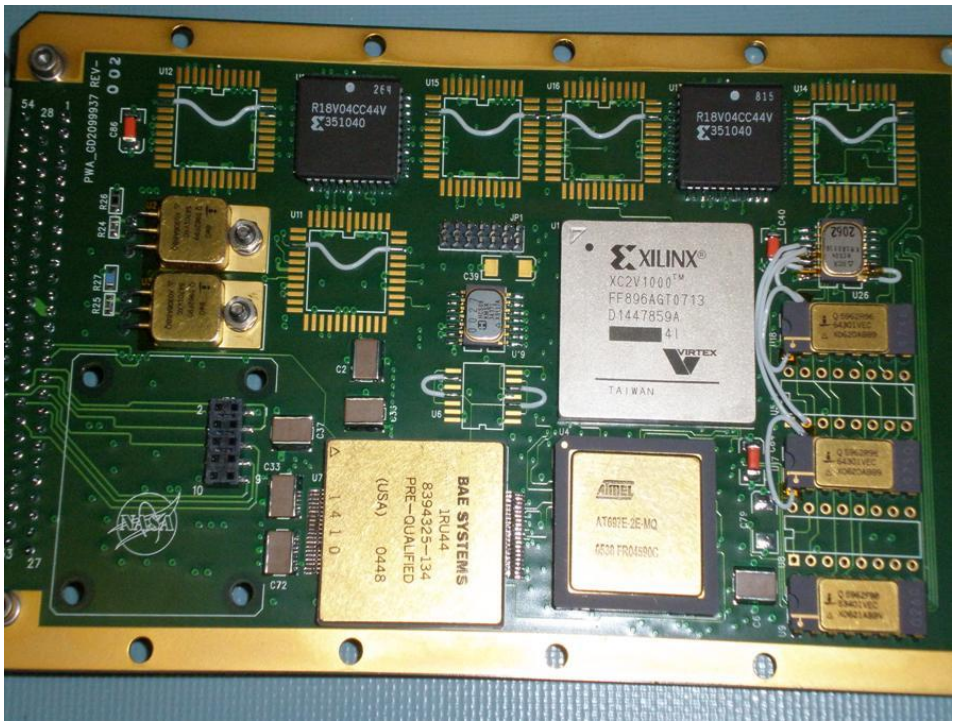

Figure 4-9 : Vue de la face composant de la carte de vol

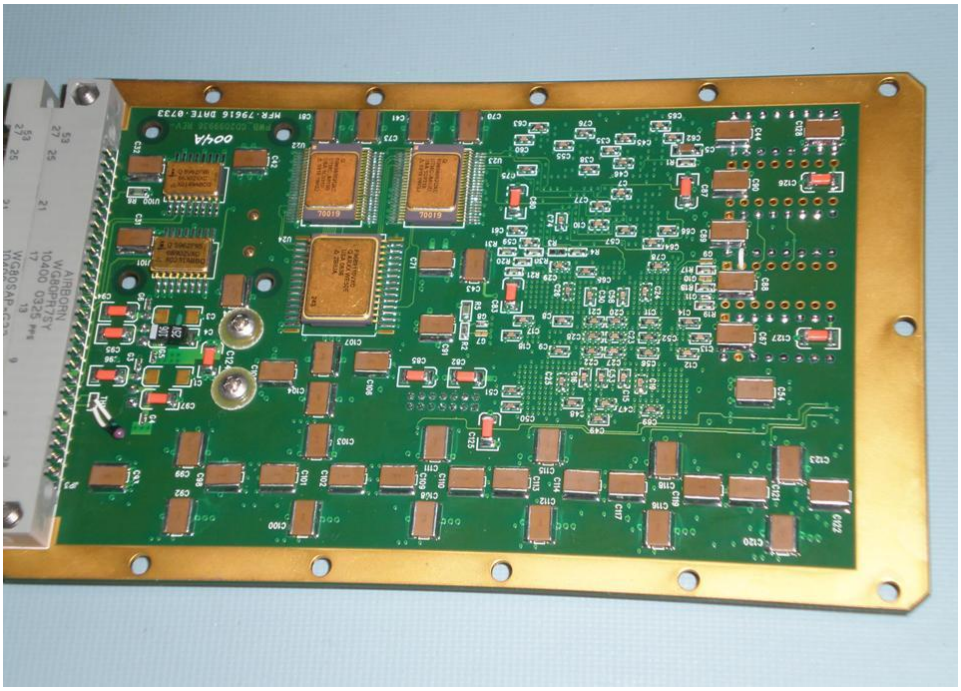

Figure 4-10 : Vue de la face soudure de la carte de vol

4.5.2. Le banc de test de la carte de vol

Le banc de test, nommé ExGSCE (Experiment Ground Support Equipment), doit permettre le test de toutes les fonctions de la carte de vol. Pour cela il doit assurer les services suivants :

- Fournir toutes les alimentations.
- Fournir une connexion RS-422.
- Fournir les signaux analogiques tels que l'horloge et la remise à zéro.
- Etre capable d'envoyer des commandes.
- Etre capable de recevoir et de décoder les réponses de la carte.
- Permettre les tests d'isolation et de continuité de chaque signal allant à la carte.
- Permettre la mesure de courant sur chaque signal.

L'ExGSCE est illustrée en Figure 4-11. Il s'agit d'une carte double face de dimensions 120 mm x 100 mm. Elle est alimentée par une unique tension +7V lui permettant de générer les tensions requises par la carte de vol à l'aide de régulateurs linéaires. Deux horloges de 4MHz et 8MHz sont disponibles au choix. Une conversion RS-422 vers RS-232 permet l'interfaçage avec le port série d'un ordinateur. Enfin chaque signal du connecteur de fond de panier peut être interrompu et une sonde de courant peut être installée.

Figure 4-11 : Le banc de test ExGSCE

4.6. L'intégration du module de charge utile

Une mission de deux semaines dans les locaux de la NASA a eu lieu afin d'effectuer les tests d'intégration. La première étape a été une démonstration de fonctionnement de la carte dans les mêmes conditions que celles lors du développement de la carte au sein du laboratoire TIMA. Puis la conformité des isolements et de la continuité électrique avec les contraintes spécifiées dans le cahier des charges a été démontrée alors que la carte était non connectée.

Ensuite la carte a été branchée sur le module de charge (Figure 4-12) utile pour y effectuer les tests suivants :

- Vérification des isolements et de la continuité électrique.
- Vérification du fonctionnement.
- Contrôle de la communication avec le satellite.
- Vérification du respect du cahier des charges.

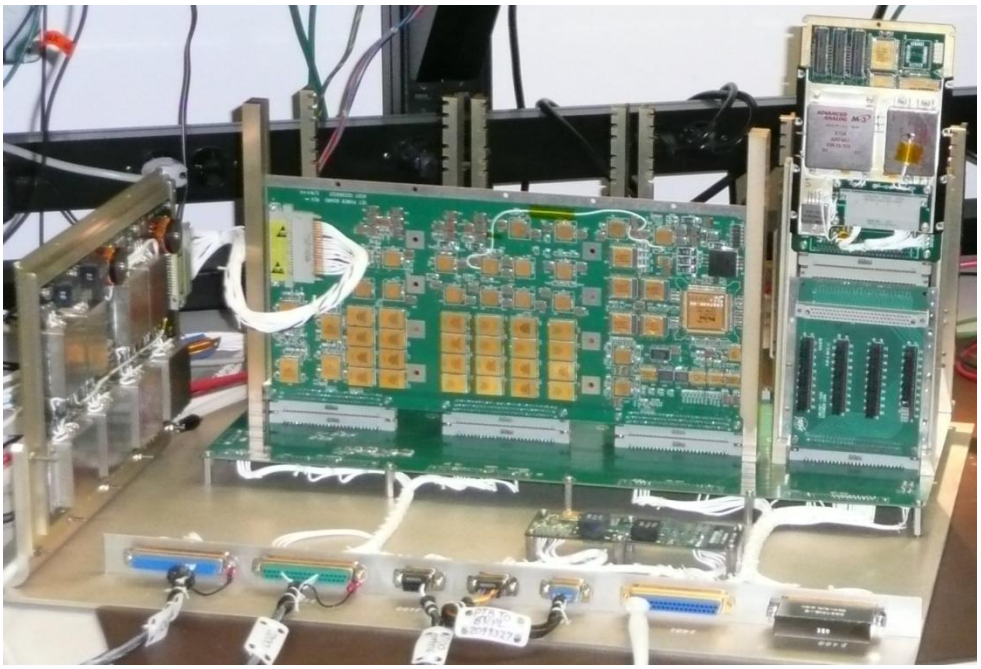

Figure 4-12 : Tests d'intégration de la carte COTS2 au module de charge utile

Les tests d'intégration de la carte ont révélés deux problèmes :

- Un problème mineur dans le calcul du CRC qui donnait un résultat faux. Etant que du logiciel, il a été facile de le corriger.
- Une surconsommation sur le +3.3V au démarrage de la carte (Figure 4-13) due aux capacités trop importantes des condensateurs de filtrage des régulateurs. Ces condensateurs ont été remplacés par des plus petits et le pic de courant est alors rentré dans les spécifications de la NASA.

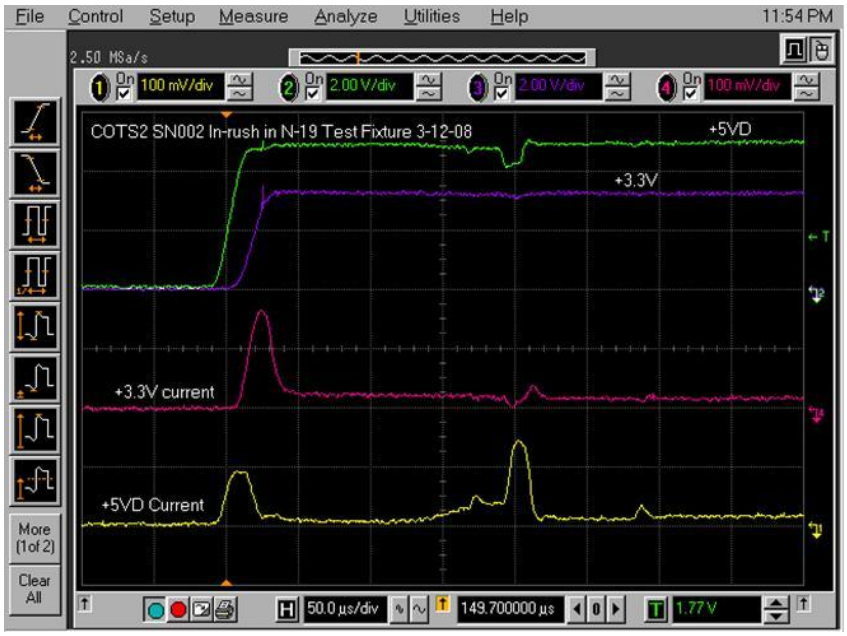

Figure 4-13 : Surconsommation de la carte au démarrage

Au final la carte a passé tous les essais de fonctionnement. Par la suite les tests de compatibilité électromagnétique et les tests environnementaux ont été conduits par la NASA dans leurs installations. Cela regroupe les essais de température et d'humidité sous vide et les essais vibro-acoustiques. La carte a été validée pour tous ces tests.

La photo Figure 4-14 représente le CCA, avec les quatre expériences, prêt pour subir les tests d'intégration au module de charge utile.

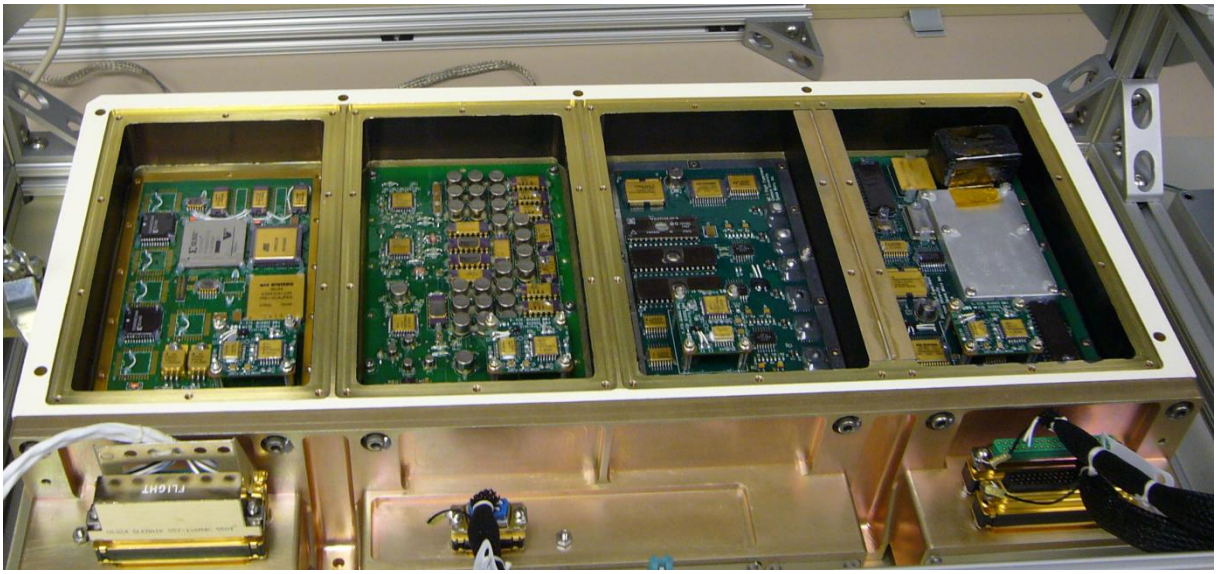

Figure 4-14 : Intégrations des quatre expériences dans le CCA

Conclusions et perspectives

L'augmentation significative de la sensibilité aux effets des radiations présentes dans l'environnement naturel de l'application est l'une des conséquences des progrès constants réalisés dans les procédés de fabrication des circuits intégrés. Alors que jadis cette problématique ne concernait que les applications spatiales, elle doit dorénavant être prise en compte dès lors de la conception des applications dans lesquelles une erreur peut avoir des conséquences critiques. La tendance actuelle conduisant à l'utilisation des composants commerciaux pour les applications spatiales et aéronautiques ne fait qu'accentuer ce phénomène. C'est notamment le cas pour les composants de type FPGA à base de mémoire SRAM qui sont appréciés des concepteurs pour leur nombreux avantages mais qui se montrent particulièrement vulnérables aux effets des radiations à cause de la taille importante de leur mémoire configuration. L'utilisation de tels composants requière donc la mise en œuvre de mesures de prévention et de protection adaptées à l'environnement final de fonctionnement de l'application et à sa criticité. Le développement de méthodologies et d'outils performants, rapides à mettre en place et peu coûteux est donc une nécessité pour la caractérisation des FPGAs et la validation des applications implémentées.

Les objectifs atteints dans le cadre de cette thèse ont été multiples. Une plateforme matérielle et logicielle a été développée autour du FPGA choisi comme véhicule de test dans le but de caractériser sa sensibilité aux radiations grâce à des essais accélérés au sol. Plusieurs campagnes de test, à l'aide de faisceaux laser, ont permis de mettre en évidence la sensibilité du composant mais aussi la capacité de la plateforme à générer des erreurs dans la mémoire de configuration du FPGA étudié. Des cartographiques détaillées de la sensibilité des différentes structures logiques du FPGA ont pu être dressées et des nouvelles pistes ont vu le jour grâce à des sessions de tirs laser sur une application en cours d'exécution. Des campagnes d'irradiation du véhicule de test avec plusieurs faisceaux d'ions lourds, produits par un accélérateur de particules, ont permis de mesurer d'une part la sensibilité intrinsèque de la mémoire de configuration du FPGA et d'autre part le taux d'erreur pour une application implémentant une architecture tolérante aux fautes de type TMR. L'une des contributions importantes de ces recherches a été la validation d'une approche de prédiction du taux d'erreur d'une application implémentée dans le FPGA en combinant les résultats des essais en accélérateur de particules à ceux issus des sessions d'injection de fautes. La confrontation des mesures et des prédictions ont permis la mise en évidence de la pertinence des résultats obtenus par la méthode d'injections de fautes dans la mémoire de configuration du FPGA.

Un objectif majeur a été le développement et la validation d'une expérience architecturée autour du FPGA cible destinée à être embarquée dans un satellite scientifique de la NASA. Cette expérience a pour but l'obtention de données sur le nombre et la criticité des erreurs provoquées par les particules énergétiques incidentes dans la mémoire de configuration du FPGA et leur impact

sur l'application implémentée qui est la même que celle retenue pour les essais au sol. Le but final est la validation des techniques et outils de l'état de l'art pour évaluer la sensibilité aux radiations des circuits et des systèmes à partir des essais au sol et des modèles de l'environnement.

Parmi les perspectives de ces travaux peuvent être mentionnées de nouvelles campagnes d'injection de fautes, à l'aide d'un faisceau laser, sur l'application utilisée dans le cadre de cette thèse afin de valider l'utilisation des équipements laser comme outils de test pour modéliser les conséquences des radiations. Une autre perspective sera l'exploitation des résultats issus de l'expérience embarquée dans le satellite LWS-SET, dont le lancement est prévu en 2011. Enfin, il est important de noter que l'intégration dans les FPGAs commerciaux de fonctionnalités visant à améliorer la tolérance aux fautes des circuits, telles des codes détecteurs d'erreurs dans la mémoire de configuration et l'intégration de voteurs « en dur » afin de sécuriser la méthode de durcissement TMR, est la preuve de la prise de conscience de cette problématique par les fabricants de composants. Une nouvelle étude pourrait ainsi être menée sur un FPGA de dernière génération afin de valider l'efficacité des nouveaux moyens de prévention et de protection implémentés et éventuellement mettre en évidence les faiblesses potentielles de ces systèmes.

Annexe A. Architecture du FPGA Virtex-II

Les composants Virtex-II sont composés de diverses fonctions logiques programmables par l'utilisateur et organisées en matrices.

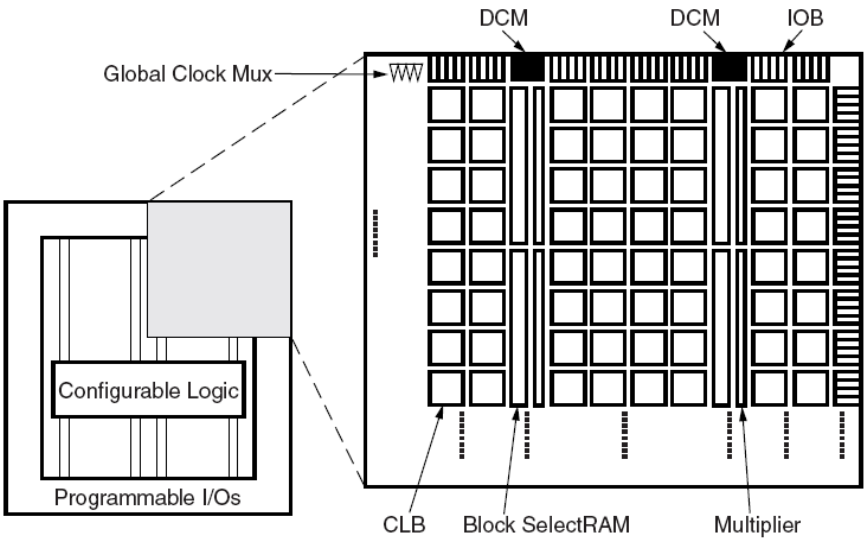

Figure A-1 : Architecture du Virtex-II

La Figure A-1 illustre l'organisation interne des cinq principaux éléments constituant ces FPGA :

- Les CLBs (Configurable Logic Blocks) permettent la réalisation de fonctions de logique combinatoire et synchrone.
- Les bocs de mémoire dit SelectRAM sont des mémoires RAM double port de 18kbits chacun.
- Les multiplieurs dédiés 18 bits x 18 bits
- Les DCMs (Digital Clock Manager) sont dédiées à la mise en forme et à la distribution des horloges dans la puce.
- Les IOBs (Input/Output blocks) sont programmables afin d'adapter les caractéristiques des signaux au monde extérieur au FPGA.

Ces blocs sont interconnectés grâce à une matrice d'interrupteurs, appelée GNR (General Routing Matrix).

Chaque élément programmable, y compris les ressources de gestion des interconnexions, est contrôlé par des valeurs stockées dans des cellules de mémoire statique. Ces valeurs sont chargées dans la mémoire par le processus de configuration du composant.

A.1. Les blocs d'entrée/sortie

Ces blocs peuvent être configurés comme des entrées, des sorties ou bien des ports bidirectionnels. Ils sont compatibles avec la plupart des standards de l'industrie. Parmi eux :

- LVTTL, LVCMOS (3.3V, 2.5V 1.8V et 1.5V)
- PCI-X 133MHz et 66MHz sous 3.3V
- PCI 66MHz et 33MHz sous 3.3V
- Etc…

Deux ports adjacents peuvent être utilisés pour connecter une paire différentielle.

Chaque CLB est composée de quatre slices et deux buffers 3 états.

A.2. La mémoire SelectRAM et les multiplieurs

Ces blocs constitués de 18kbits de mémoire RAM double port peuvent être configurés de 16k x 1 bit jusqu'à 512 x 36 bits. Plusieurs blocs peuvent être cascadés afin d'obtenir des éléments de stockage importants.

Un multiplieur dédié 18 bits x 18 bits est associé à chaque bloc selectRAM et est optimisé pour les opérations sur ces mémoires. Néanmoins le multiplieur peut être utilisé indépendamment de la mémoire qui lui est associée.

A.3. La gestion des horloges

Les DCMs permettent :

- la remise en forme d'horloges venant de l'extérieur.
- l'élimination des retards dans la distribution de l'horloge.
- la génération d'horloges déphasées de 90°, 180° et 270° par rapport l'horloge de sortie.
- La génération d'une horloge avec une fréquence double de celle d'entrée.
- La génération d'une horloge dont la fréquence est égale au rapport M/D ou M et D sont deux entiers paramétrables par l'utilisateur.

A.4. Les blocs CLBs

Ces éléments sont organisés en matrices et sont utilisés pour la construction de fonctions logiques combinatoires et séquentielles. Comme le monte la Figure A-2, chaque CLB est composées de quatre slices arrangées en 2 colonnes de 2 slices.

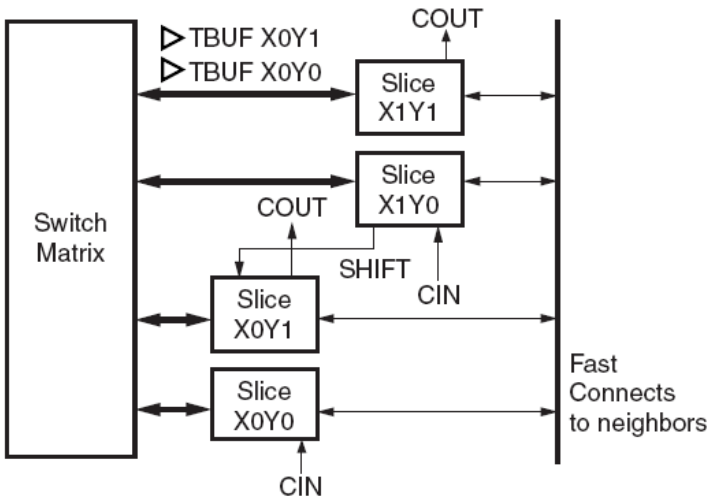

Figure A-2 : Eléments composants une CLB de la famille Virtex-II

Chaque slice est composée de deux générateurs de fonctions à quatre entrées, de portes de logique arithmétique, de multiplexeurs et de deux éléments de stockage. Comme le montre la Figure A-3 chaque générateur de fonctions peut être configuré pour assurer une des trois fonctionnalités suivantes :

- Une LUT (Look-Up Table) à quatre entrées qui permet d'effectuer n'importe quelle opération booléenne de quatre entrées parmi celles prédéfinies.
- Une mémoire RAM de 16 bits.
- Un registre à décalage de profondeur 16 bits.

La sortie de chaque générateur de fonctions peut être appliquée directement à la sortie de la slice ou bien sur l'entrée de l'élément de stockage de type D.

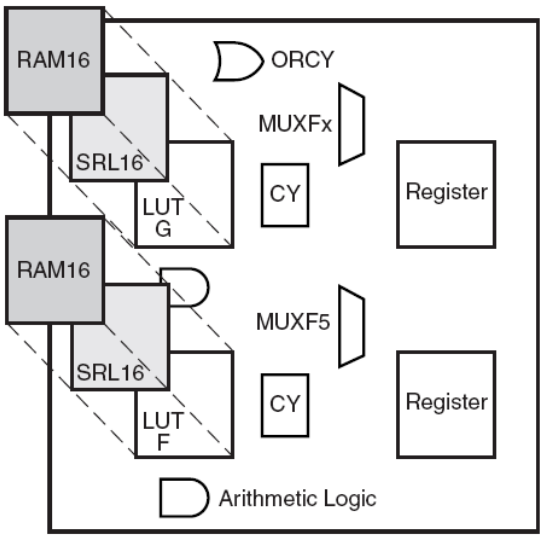

Figure A-3 : Composition d'une slice de la famille Virtex-II

Annexe B. Carte fille Virtex-II pour le testeur THESIC+

Cette annexe présente, pour la carte Virtex-II, les schémas électriques, puis les vues face composants et soudure du circuit imprimé, et enfin une photo de la carte finie.

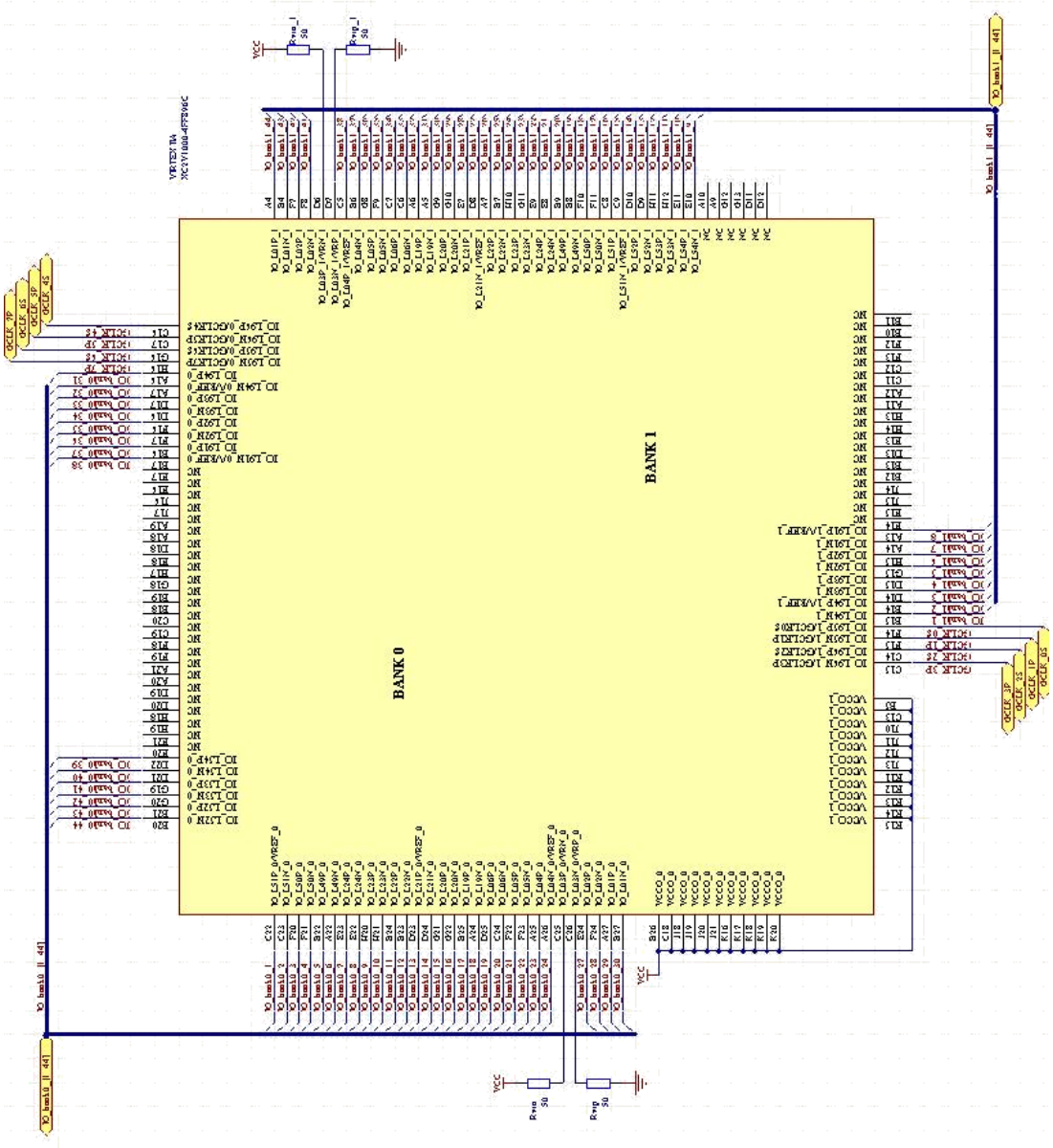

Figure B-1 : Bank 0 et 1 du FPGA

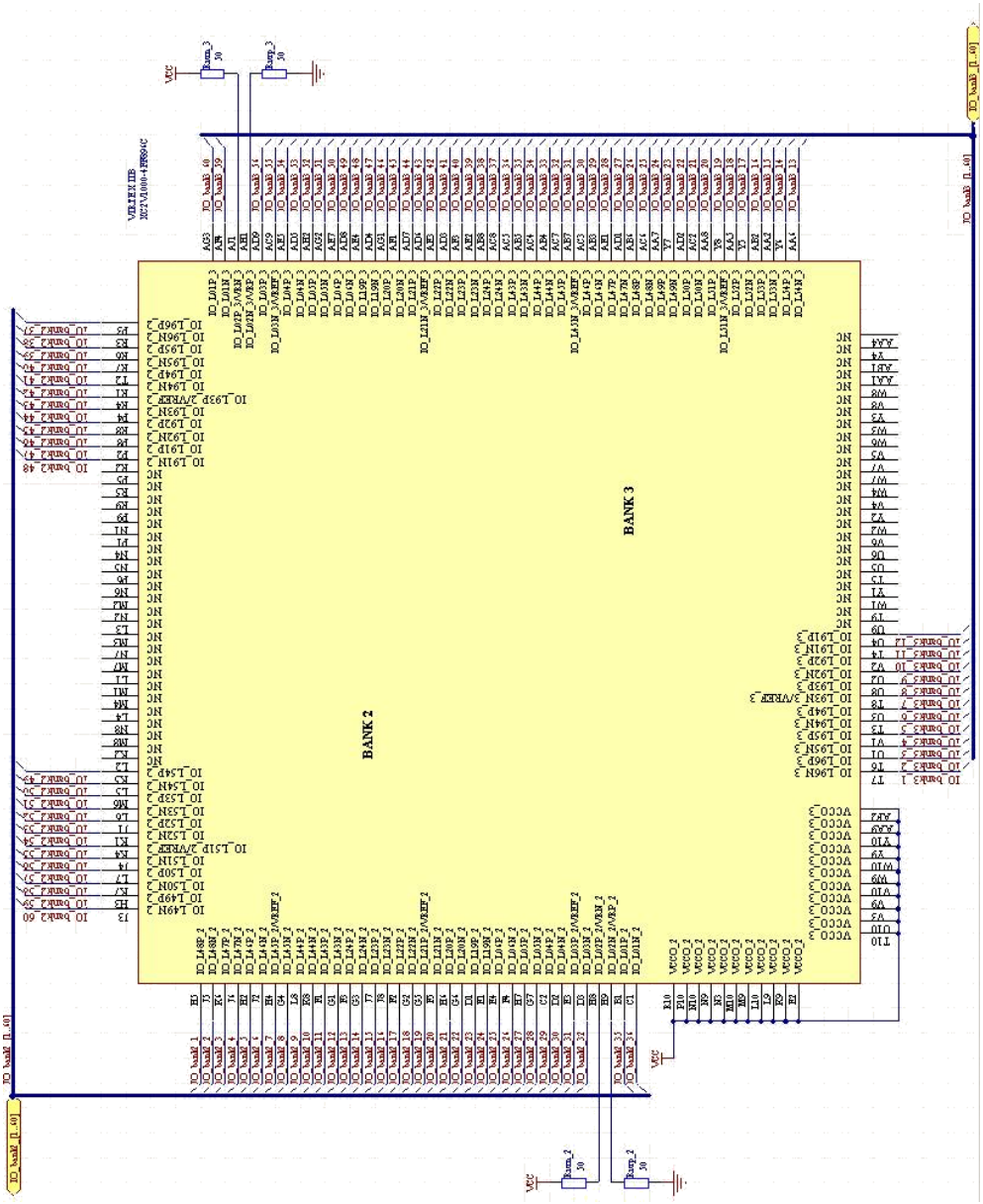

Figure B-2 : Bank 2 et 3 du FPGA

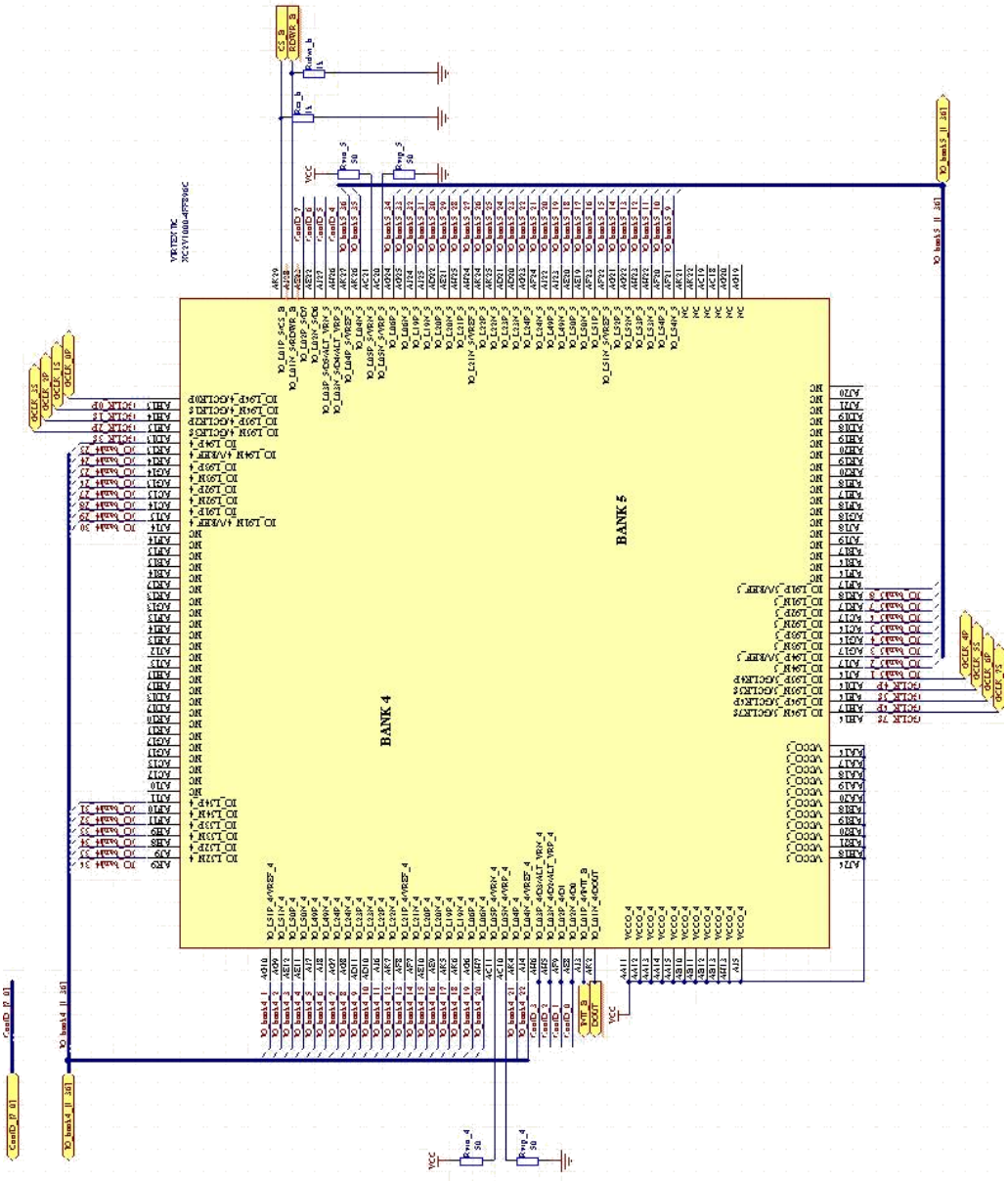

Figure B-3 : Bank 4 et 5 du FPGA

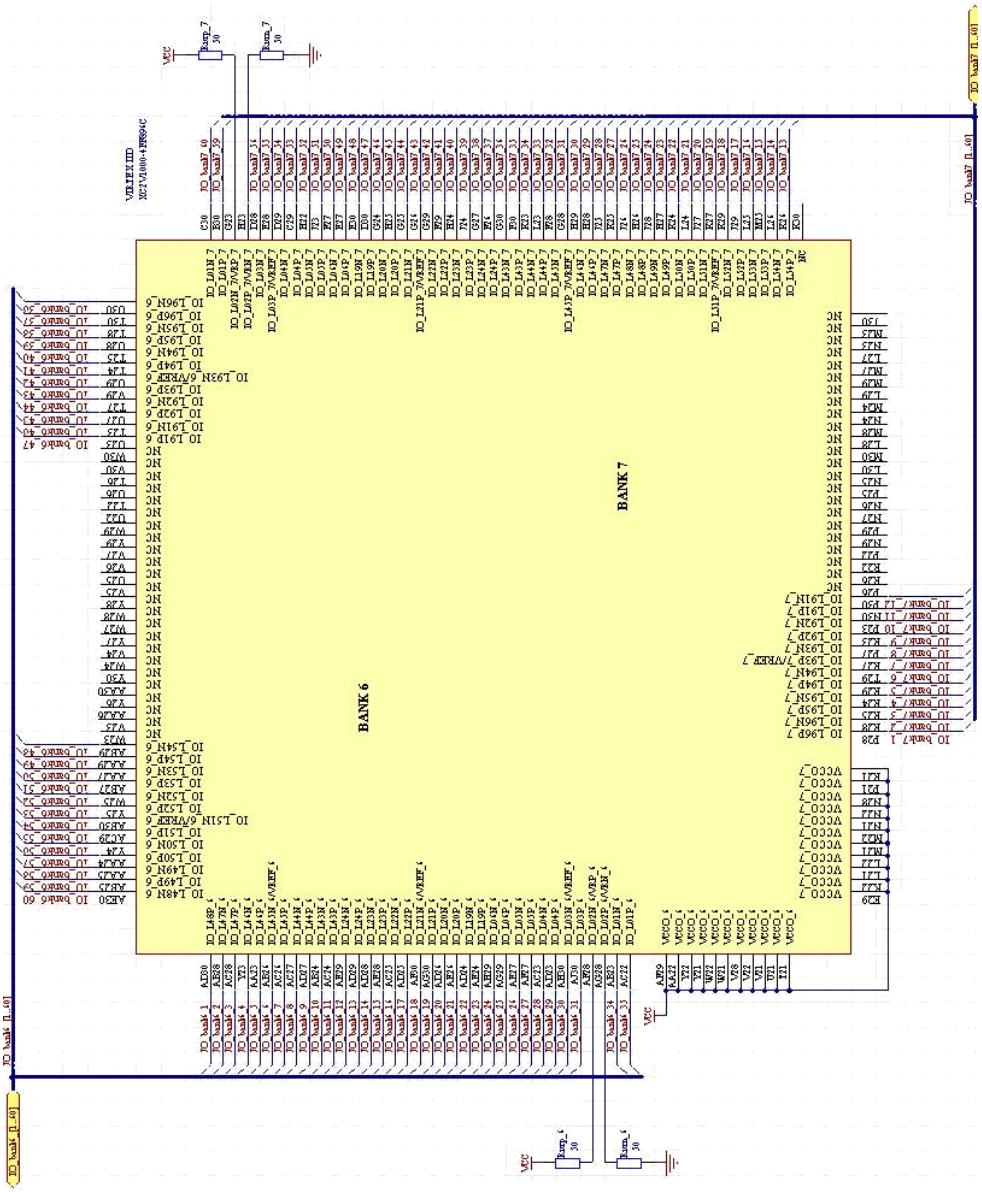

Figure B-4 : Bank 6 et 7 du FPGA

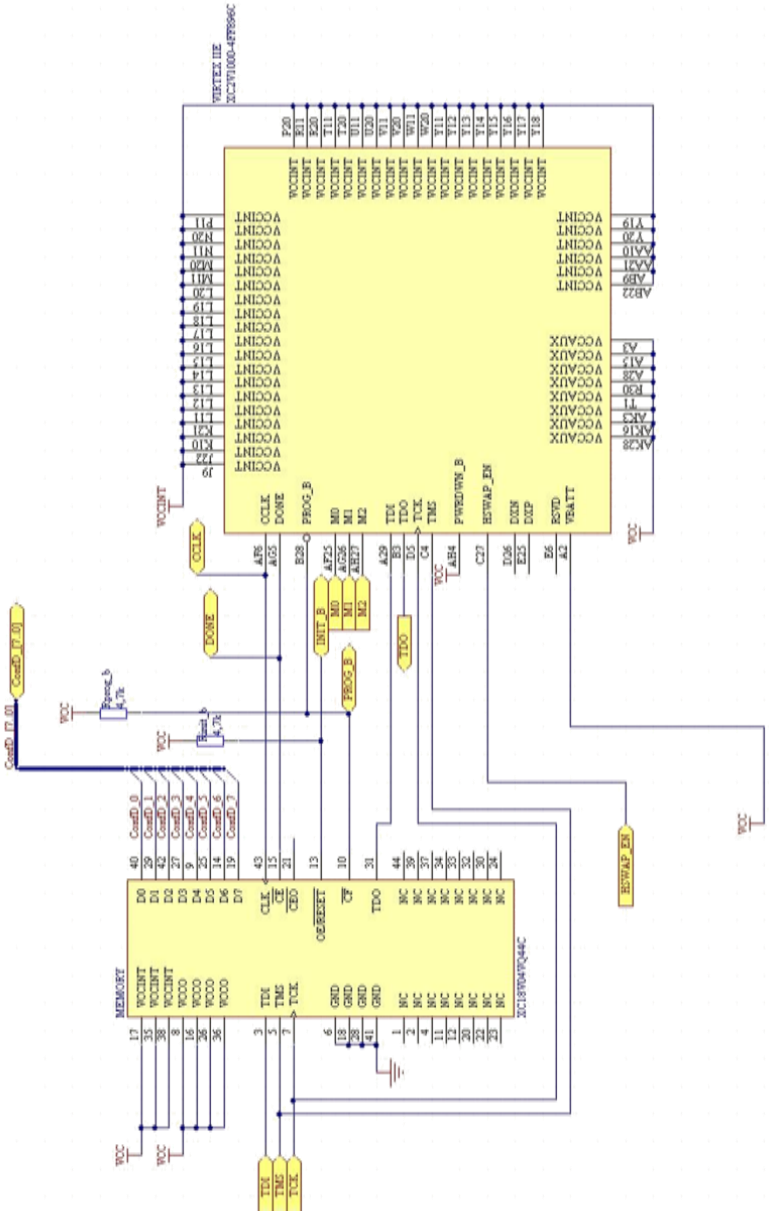

Figure B-5 : Mémoire de configuration externe et signaux de contrôle du FPGA

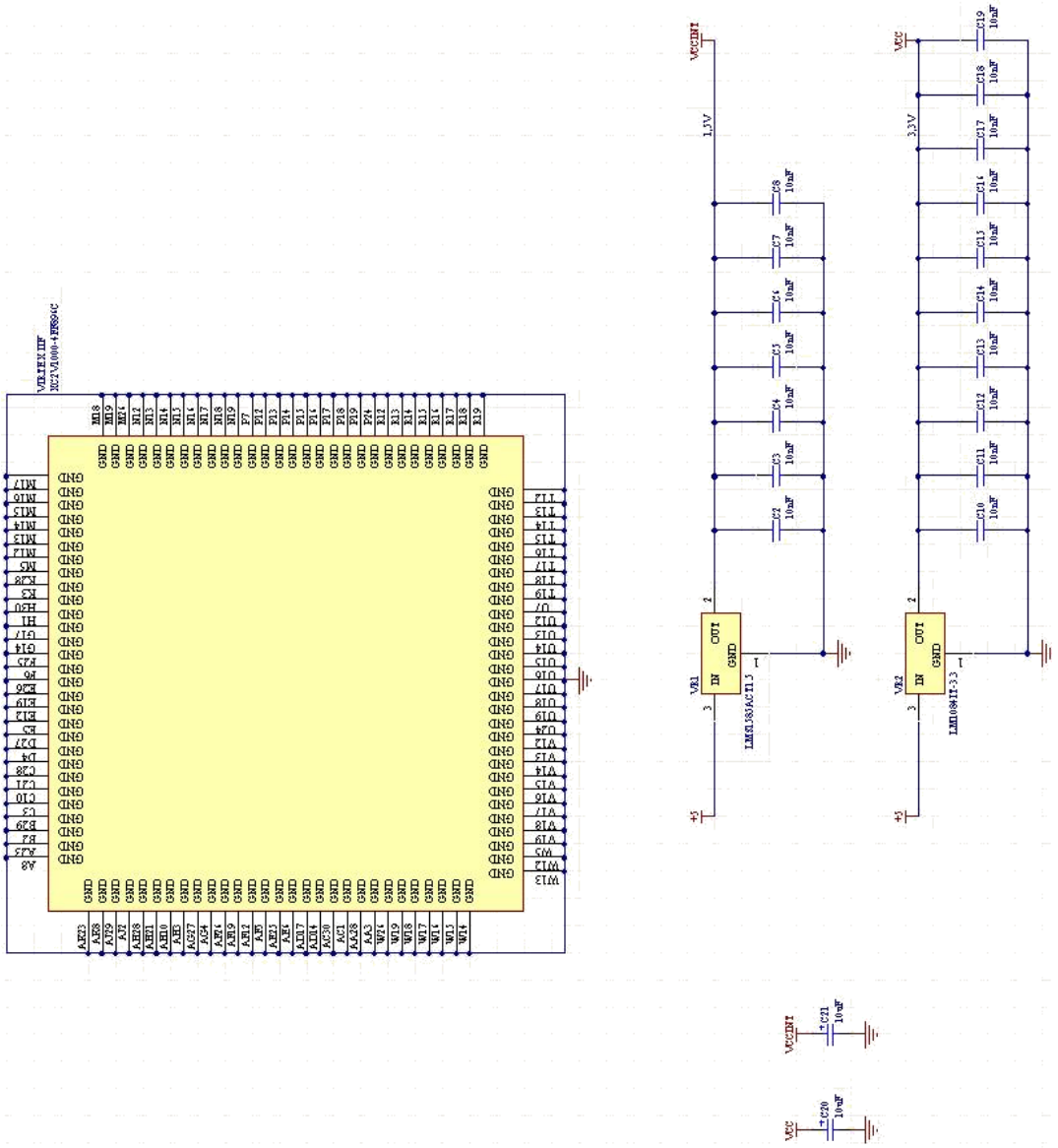

Figure B-6 : Régulateurs de tensions linéaires et retour de masse du FPGA

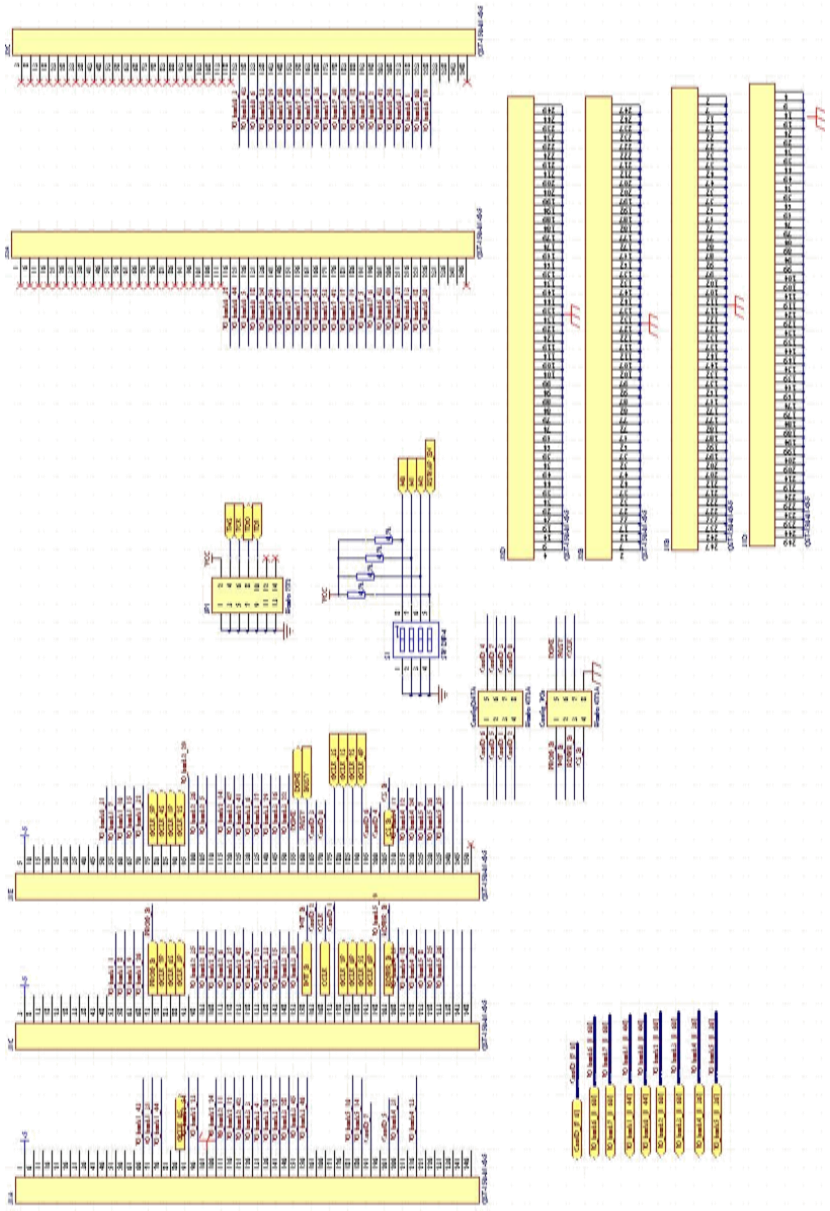

Figure B-7 : Connecteurs d'interfaçage au testeur THESIC+

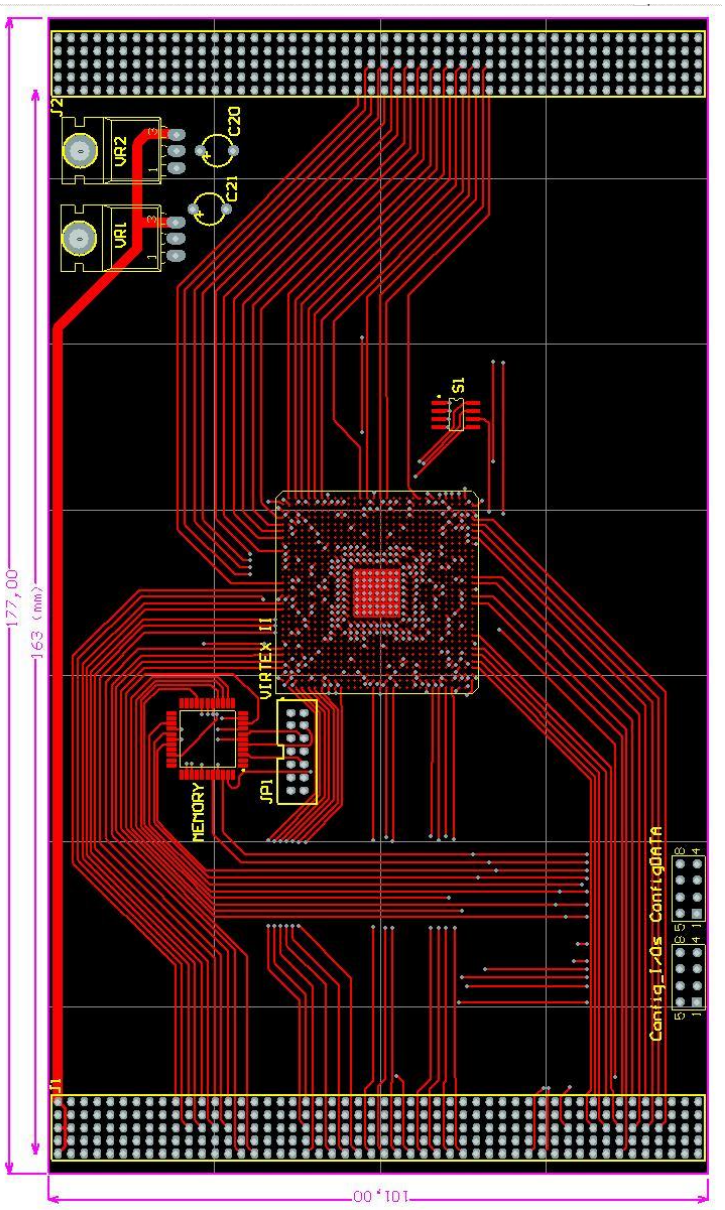

Figure B-8 : Circuit imprimé face composant

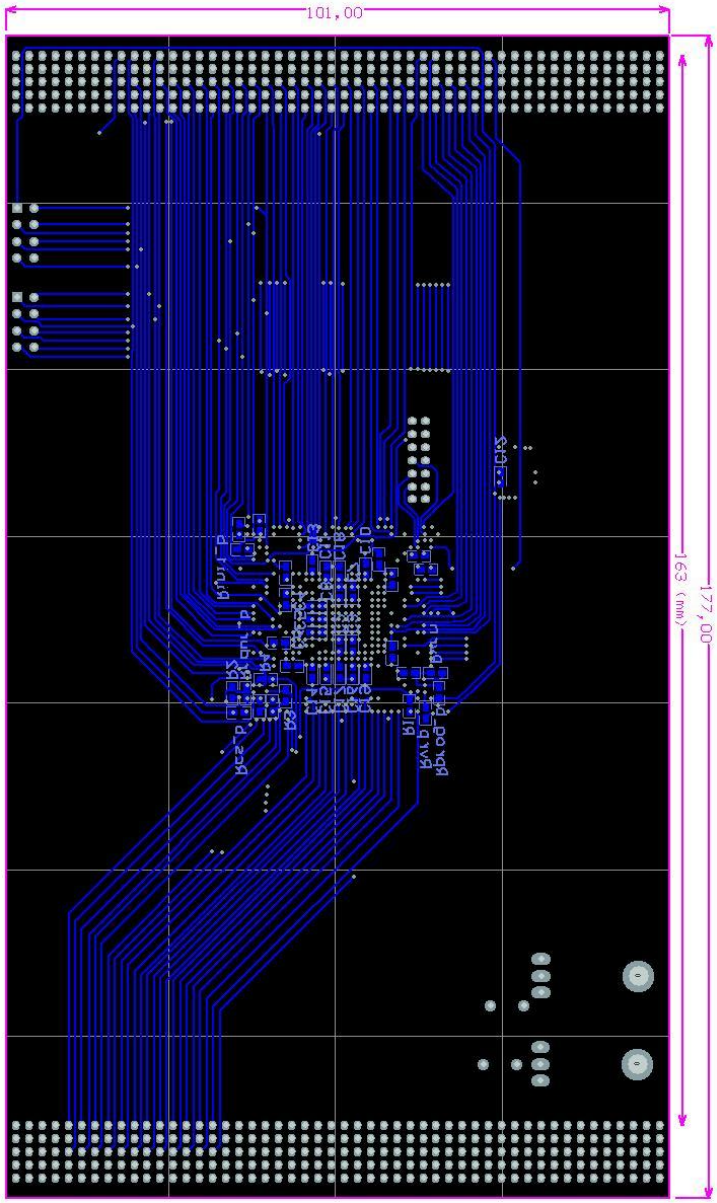

Figure B-9 : Circuit imprimé face soudure

Figure B-10 : Photo de la carte Virtex-II

Annexe C. Le programme spatial LWS-SET

C.1. Description de l'engin spatial

La première mission, nommée SET-1, vise à mettre en orbite l'engin spatial DSX (Demonstration & Science Experiments) construit par la société MicroSat Systems (Colorado, Etat-Unis) [Web MicroSat]. Il est prévu pour fonctionner durant une année à une altitude comprise entre 6.000 km et 12.000 km. Ses dimensions sont 3,6 m x 2 m x 1,1 m et il pèse 600 kg, dont 170 kg de charge utile. Le DSX est conçu pour être lancé soit comme charge principale depuis un petit lanceur, comme Minotaur, ou bien comme charge secondaire sur un gros lanceur comme les EELV (Evolved Expendable Launch Vehicle). Le programme EELV fut lancé au début des années 1990 par l'USAF (United States Air Force) dans le but de moderniser les lanceurs existant (ex : Delta II, Atlas II/Centaur, Titan IV, …). Ceci afin de proposer un lanceur moins couteux et plus fiable, donc plus abordable. Pour assurer cette flexibilité, le DSX est architecturé autour d'un anneau ESPA (EELV Secondary Payload Adapter). La Figure C-1 fournit une vue globale du satellite qui est constitué des éléments suivants :

- L'anneau ESPA permettant le montage sur le lanceur. Il fait office d'interface entre le lanceur et le satellite.
- Un module avionique qui incorpore le système de contrôle d'attitude, la partie puissance, le contrôle thermique, les communications, le stockage des commandes et des données, etc.
- Le module de charge utile qui contient les expériences, les outils d'observation des radiations, etc.
- Les deux antennes de communication avec le sol. Une antenne Y de 80 mètres et une antenne Z de 16 mètres.

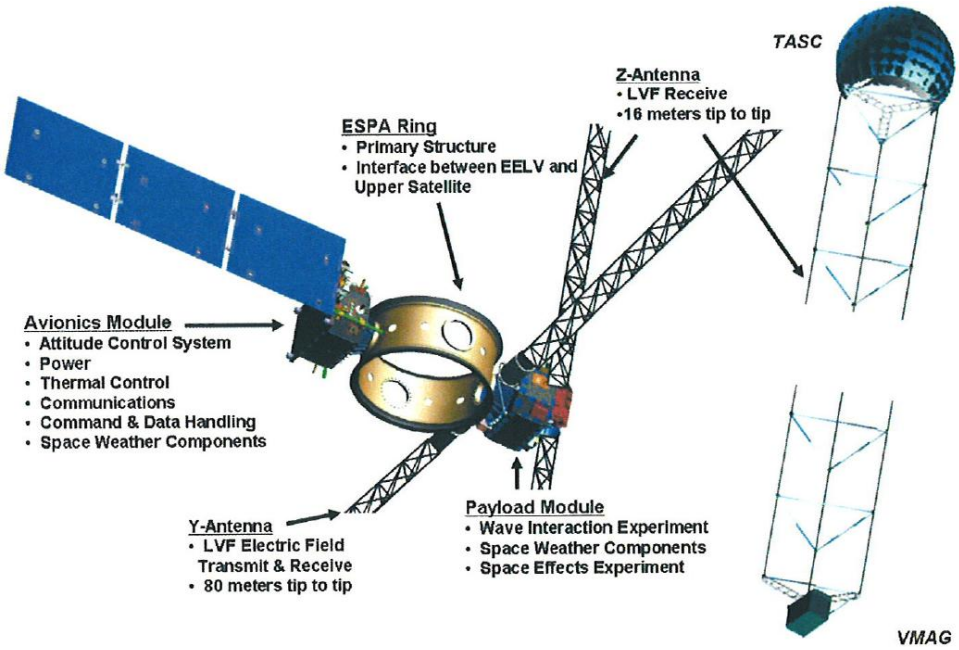

Figure C-1 : Vue globale du satellite DSX

La Figure C-2 illustre l'emplacement de la charge utile (Payload Module) sur le DSX. L'anneau ESPA permet le montage sur le lanceur. Le module en arrière plan et le module avionique.

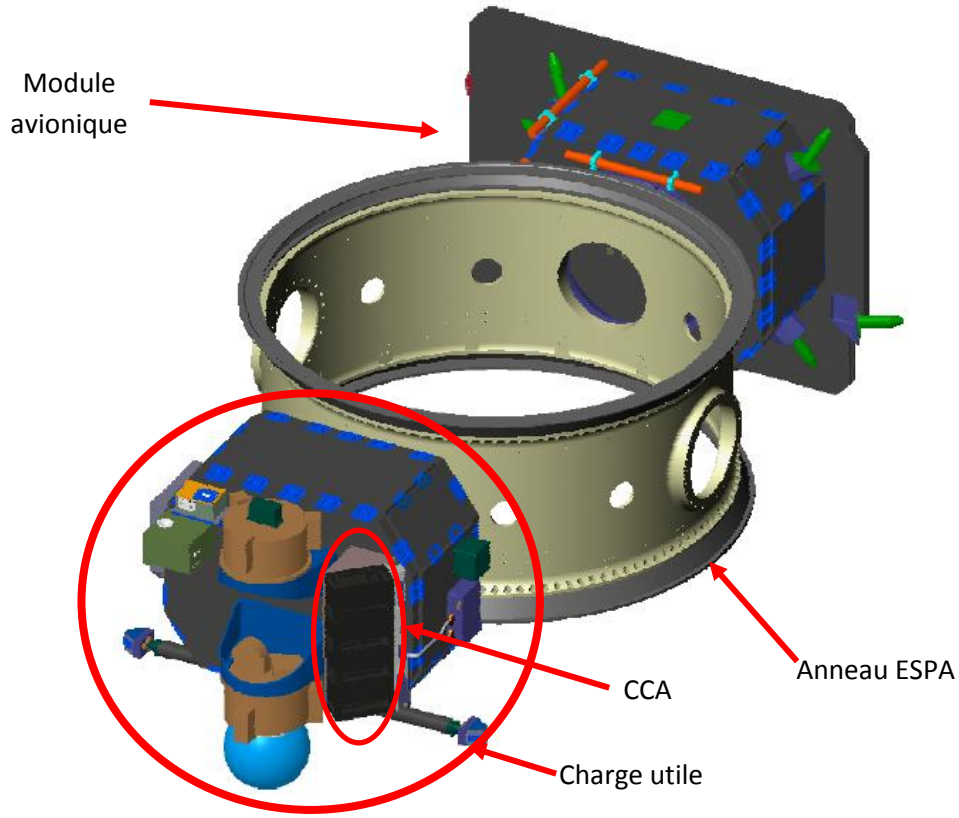

Figure C-2 : Plan mécanique 3D de l'engin spatial DSX

Sa charge utile est entre autre constituée de cinq expériences :

- CREDANCE (Cosmic Radiation Environment Dosimetry and Charging Experiment) est développé par la société QinetiQ en Angleterre. Son objectif est d'une part la caractérisation de l'environnement radiatif spatial et son interaction avec l'engin spatial dans le but d'améliorer les modèles environnementaux et les outils de conception. D'autre part CREDANCE doit fournir aux quatre autres expériences des données sur l'environnement.
- DIME 1 & 2 (Dosimetry Intercomparison and Miniaturization) est conçue à l'université de Clemson aux Etats-Unis. Elle a pour mission la caractérisation de la dose totale, le décalage des tensions de seuil et les SEEs sur différents composants COTS comme des RADFET (Radiation-Sensing Field-Effect Transistors), une EPROM, une SRAM, etc…
- ELDRS (Enhanced Low Dose Rate Sensitivity) provient de l'université d'Arizona aux Etats-Unis. Elle vise à caractériser les effets des protons et l'ELDRS sur des transistors bipolaires.
- COTS2 développée au laboratoire TIMA à Grenoble, dans le cadre de cette thèse, enregistre les perturbations causées par les particules sur la mémoire de configuration d'un FPGA COTS et sur son application embarquée.

Les quatre expériences DIME 1 & 2, ELDRS et COTS2 sont montées sur un châssis nommé CCA (Central Carrier Assembly). Celui-ci est composé d'un fond de panier sur lequel peuvent se connecter soit quatre cartes simples 3U ou bien deux cartes simples 3U et une carte double 6U. Le plan mécanique en trois dimensions donné en Figure C-3 illustre le CCA dans une configuration de deux cartes 3U présentes dans les emplacements 1 et 2 et d'une carte 6U disposée dans les emplacements 3 et 4.

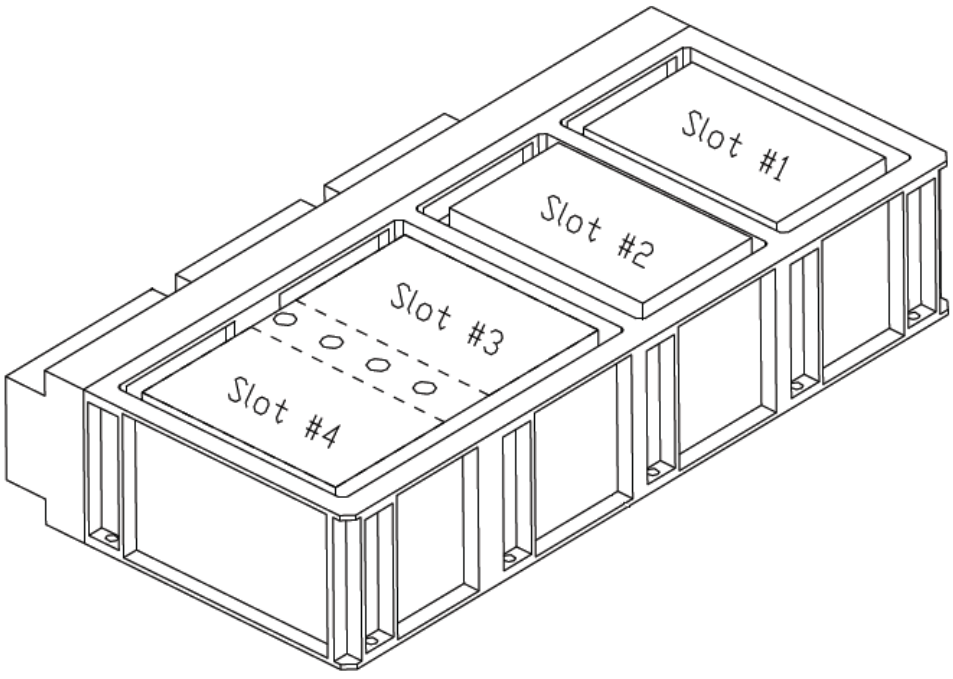

Figure C-3 : Plan mécanique 3D du Central Carrier Assemblies

CREDANCE et le CCA sont installés sur la partie extérieure du « Payload module », c'est-à-dire la charge utile du satellite (Figure C-4), afin d'être exposés directement aux particules.

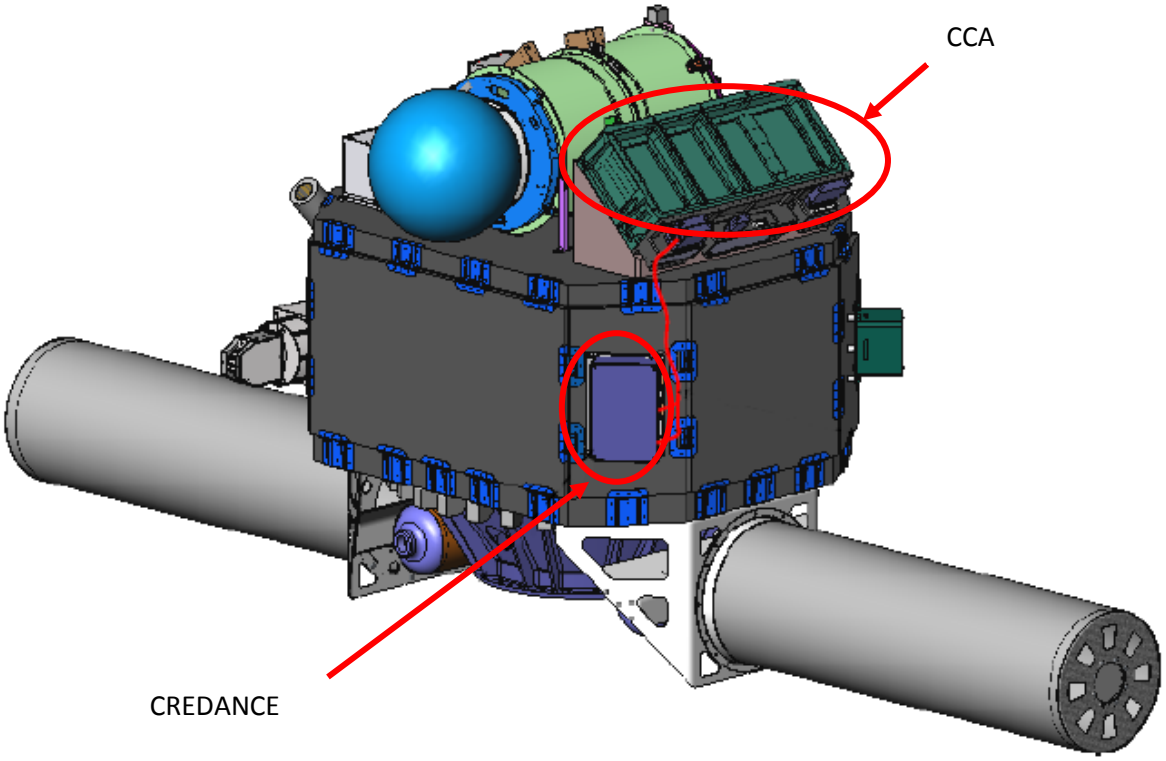

Figure C-4 Plan mécanique 3D de la charge utile

C.2. Etude du cahier des charges

Ce paragraphe présente quelques unes de contraintes et informations les plus importantes extraites du cahier des charges (document de 150 pages).

C.2.1. Contraintes mécaniques

Les contraintes mécaniques, illustrées en Figure C-5, sont les suivantes :

- Respecter le format simple 3U, soit 160 mm x 100 mm.
- Prévoir l'emplacement d'un connecteur 80 pins Airborn WG80PR monté sur la face soudure pour assurer la connexion avec le fond de panier du CCA.
- Prévoir un emplacement en face composant de dimensions 38 mm x 32 mm pour le montage de la carte fille de dosimétrie.
- Prévoir les trous de fixation sur le pourtour de la carte.

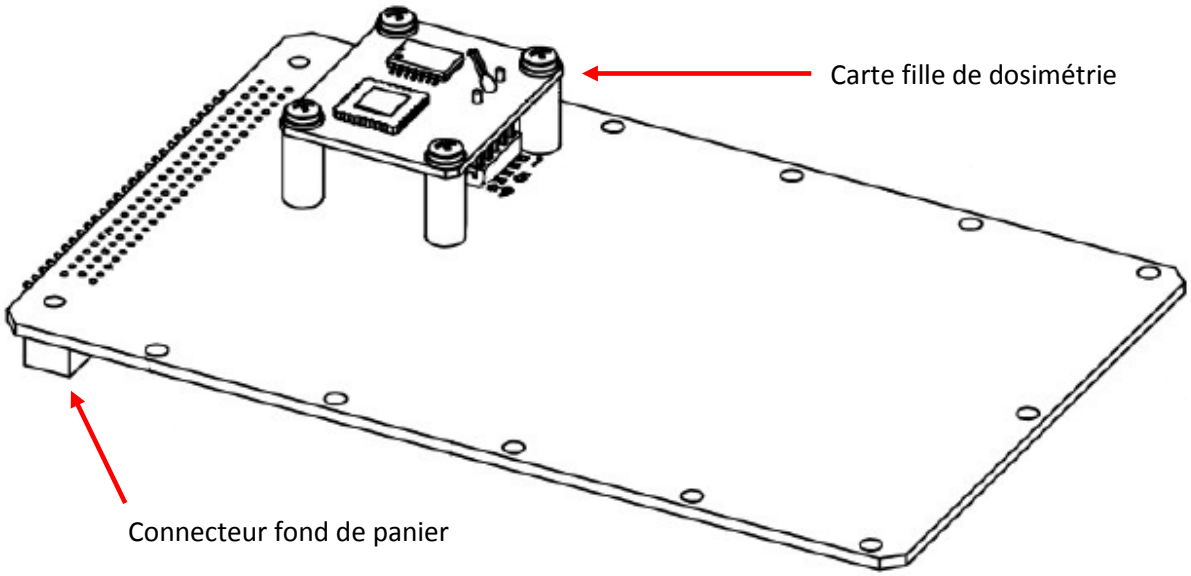

Figure C-5 : Contraintes mécaniques sur la carte COTS2

C.2.1. Contraintes électriques

Les tensions électriques fournies par le satellite sont détaillées dans le Tableau C-1. Elles sont séparées en deux groupes (primaire et secondaire). Les alimentations secondaires sont, soit de type analogique, soit numérique. La consommation de la carte ne doit pas dépasser les limites spécifiées dans le tableau pour chaque tension. De plus, une limite est fixée à 4W pour l'ensemble de la carte en fonctionnement normal. Les pics de courant, notamment au démarrage, sont tolérés tant que leur amplitude n'excède pas 1,5 fois la valeur nominale et que leur durée est inférieure à 50 ms.

Tableau C-1 : Tensions fournies par le satellite

Tensions	Type	Tolérance (%)	Courant nominal maximum (mA)
+3,3V numérique	secondaire	+/- 5%	1200
+5V numérique	secondaire	+/- 5%	800
+/- 5V analogique	secondaire	+/- 5%	+/- 640
+/- 15V analogique	secondaire	+/- 5%	+/- 210
+28V	primaire	+/- 5%	140

Les retours de masse des alimentations numériques et analogiques doivent être séparés. De même le retour de masse de l'alimentation primaire 28V doit être individuel. De manière générale chaque alimentation doit avoir un retour de masse qui lui est propre.

Une résistance d'isolation minimum de 1MΩ doit être assurée entre les alimentations et le châssis. L'isolation entre les alimentations numériques et analogiques doit être supérieure à 10kΩ.

C.2.2. L'interface avec le module de charge utile

Chaque carte d'expérience a ses propres signaux de télémétrie qui sont indépendants des autres. Ils sont de trois natures différentes suivant la fonction qu'ils doivent remplir.

A.2.2.1. Les signaux analogiques

Trois signaux analogiques permettent l'observation de l'environnement de carte grâce à un dosimètre et deux sondes de température :

- Un dosimètre disposé sur la carte fille de dosimétrie.
- Une sonde de température disposée sur la carte fille de dosimétrie.
- Une sonde de température installée directement sur le circuit imprimé de la carte COTS2.

A.2.2.2. Les signaux numériques de contrôle

Ensuite trois signaux numériques de contrôle servent à piloter la carte COTS2:

- Une horloge à 8 MHz.
- Un signal de RESET pour initialiser la carte.
- Un signal STANDBY qui permet de mettre la carte dans un mode de fonctionnement basse consommation. Ce mode ne peut pas être utilisé par COTS2 car les limites de courant imposées ne permettent pas le fonctionnement du processeur de la carte.

A.2.2.3. L'interface de communication série

Enfin une interface série est aussi disponible pour transmettre les données numériques. Une description en est donnée dans le paragraphe suivant.

C.2.3. Description de l'interface de communication

L'interface de communication avec le module de charge utile peut être divisée en trois couches. La couche électrique est la couche inférieure la plus proche du matériel de la carte. La couche intermédiaire, dite couche octet, explicite le mode de transmission des octets. Enfin la couche supérieure, dite couche trame, définit l'encapsulation des données dans les trames.

A.2.3.1. La couche électrique

Les spécifications électriques sont celles d'une liaison série différentielle au format RS-422. Le satellite est considéré comme le maître et la carte COTS2 comme l'esclave. Ceci implique que la carte ne peut pas initier la communication, elle ne peut que répondre à une commande envoyée par le maître. Le débit retenu est de 9,6 kbits/s.

A.2.3.2. La couche octet

La définition des deux couches supérieures (octet et trame) est régie par le protocole HDLC-UCC 12, 15.1 dont les spécifications sont fournies dans le standard ISO/IEC 13239 [200]. HDLC est le nom du protocole et signifie High-level Data Link Control. UCC définit la classe sélectionnée, soit Unbalanced Connectionless Class. Enfin 12 et 15.1 sont les fonctions optionnelles. L'option 12

permet un test basique de la communication grâce à la commande « TEST ». L'option 15.1 indique l'utilisation des bits de start et de stop ainsi que la fonction de transparence.

La Figure C-6 illustre la transmission d'un octet : elle doit débuter par l'envoi d'un *start bit,* puis viennent les 8 bits de données (en commençant par le bit de poids faible), et enfin un *stop bit*.

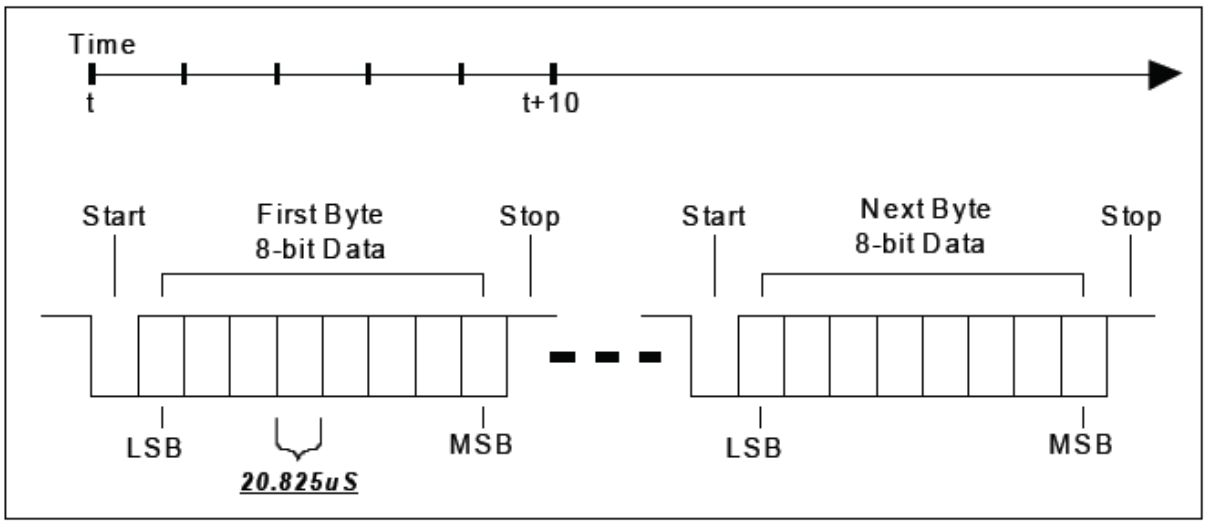

Figure C-6 : Format de transmission d'un octet par la liaison série

A.2.3.3. La couche trame

Le protocole HDLC permet d'encapsuler les données utilisateur dans une trame formatée. Le but est, d'une part, de garantir le bon fonctionnement des communications entre plusieurs terminaux et, d'autre part, de garantir l'intégrité des données transmisses. Une trame, dont le format est donnée en Figure C-7, doit être composé des éléments suivant :

- Un octet drapeau signalant le début d'une nouvelle trame.
- Un octet d'adresse. En effet la liaison RS-422 permet de connecter un maître et plusieurs esclaves sur une même ligne, par conséquent une adresse doit être assignée à chacun. Etant donné qu'ici seul le satellite et la carte sont sur la même ligne, cette valeur n'est pas prise en compte lors du décodage de la trame.
- Un octet de contrôle qui spécifie la nature de la trame (interrogation ou réponse) et la nature des données qui vont suivre. Cette valeur n'est pas non plus considérée comme significative.
- N octets d'information « utile ». Ici, N est limité à un maximum de 128 octets.
- 2 octets contenant le résultat du calcul de CRC (Contrôle de Redondance Cyclique) effectué par l'émetteur. Le récepteur applique le même calcul et compare la valeur résultante avec celle émise. Si elles divergent, alors la trame est considérée comme non valide et n'est pas prise en compte. Le polynôme utilisé est « $x^{16} + x^{12} + x^5 + 1$ ». Le calcul de CRC est effectué sur l'ensemble de la trame à l'exception des drapeaux, du CRC et des octets ajoutés par l'opération de transparence (expliquée ci-après).
- 1 octet drapeau indiquant la fin de la trame.

Drapeau	Adresse	Contrôle	Information	CRC	Drapeau
01111110	1 octet	1 octet	N octets	2 octets	01111110

Figure C-7 : Format d'une trame HDLC

L'opération de transparence permet de masquer les octets identiques aux drapeaux afin d'éviter les fausses détections de début ou de fin de trame. Elle est appliquée sur tous les octets de la trame à l'exception bien sur des drapeaux. L'opération consiste à remplacer toutes les occurrences des octets 01111110 (drapeau) et 10111110 (octet indiquant une opération de masquage) par deux octets. Le premier octet est 10111110 et le deuxième octet est égal à l'octet d'origine donc le 6^{eme} bit transmis est complémenté. Le récepteur effectue l'opération de transparence inverse. C'est-à-dire la recherche et la suppression de l'octet de transparence (10111110) puis l'inversion du 6^{eme} bit du mot suivant. La Figure C-8 illustre l'opération de transparence sur une trame HDLC. Si la donnée utilisateur s'avère être égale au drapeau, il faut par conséquent la masquer afin d'éviter une détection prématurée de la fin de trame par le récepteur. L'octet de transparence est inséré avant la donnée utilisateur dont le 6^{eme} bit est complémenté.

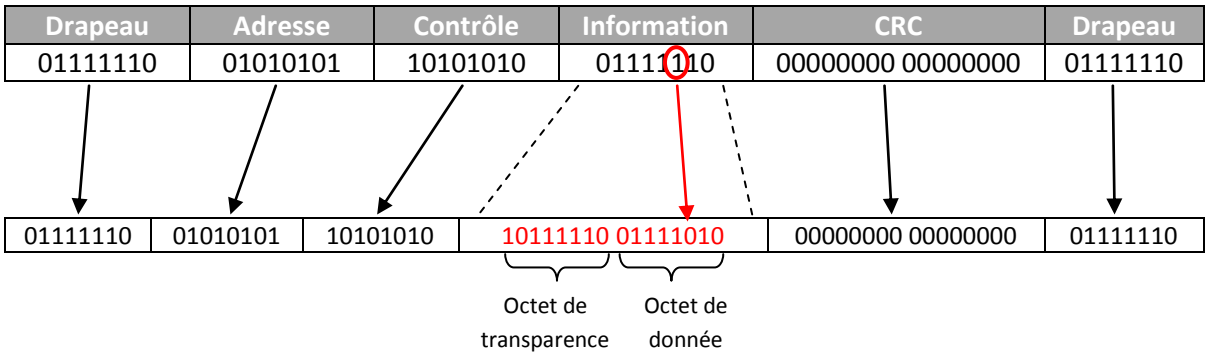

Figure C-8 : Exemple d'opération de transparence sur une trame HDLC

C.2.4. Les communications avec le sol

Le module de charge utile interroge régulièrement chaque expérience embarquée afin télécharger les données collectées. Ces données sont ensuite envoyées au module avionique qui les stocke. Elles sont par la suite transmises au sol lors du survol de la station terrestre. Enfin elles sont mises à disposition des propriétaires des cartes par l'intermédiaire d'un serveur FTP (File Transfer Protocol).

C.2.5. Gestion des plages de temps de fonctionnement de chaque expérience

Le module de charge utile gère les plages de temps de fonctionnement de chacune des quatre cartes expérimentales suivant plusieurs critères :

- Les conditions environnementales telles que la température, la densité de radiation et l'énergie des particules (en cas de forte activité solaire par exemple), etc.
- L'énergie disponible qui varie en fonction de l'exposition ou non au soleil des panneaux solaires.

- Les consignes données par les concepteurs des cartes.

C.2.6. Déroulement de l'intégration et des tests

Les tests fonctionnels, d'intégration et environnementaux se sont déroulés en quatre phases :

- La première phase concerne les tests de notre carte expérimentale. Ils ont été effectués dans les locaux des concepteurs des expériences. Il s'agit de vérifier les isolements, la continuité électrique, le bon fonctionnement de la carte, etc.
- La deuxième phase est l'intégration et le test de notre carte dans le module de charge utile. Nous les avons effectués dans les locaux de la NASA[11]. Dans un premier temps la fonctionnalité de la carte a été expertisée avec les ingénieurs de la NASA ; puis la carte branchée sur le module de charge utile. Les isolements et la continuité électrique ont de nouveau été contrôlés avant de mettre en route la carte et d'effectuer les essais fonctionnels. Ensuite la NASA a procédé aux tests environnementaux : vibration, compatibilité électromagnétique (CEM), température, vide, etc sur le module de charge utile.
- La troisième phase concerne l'intégration du module de charge utile avec le DSX. Cette opération a été effectuée par la NASA chez MicroSat Systems constructeur de l'engin spatial. Comme précédemment il s'agit de procéder aux tests d'isolation, puis aux tests fonctionnels et enfin aux tests environnementaux.
- La dernière phase est la mise en orbite et le démarrage consécutif des cartes expérimentales.

En tant que concepteurs de la carte COTS2 nous sommes intervenus dans la phase 1 et la phase 2[12]. Les essais environnementaux ont eu lieu par la suite et à chaque étape nous avons validé les résultats obtenus.

C.2.7. Définition des commandes envoyées par le satellite

La carte COTS2 n'est pas autorisée à initier une communication. Seul le satellite peut envoyer une telle commande. A chaque commande doit correspondre une réponse de la carte. Un jeu de trois commandes est suffisant pour piloter la carte COTS2 :

- La commande « PING », inspirée de la fonction du même nom utilisée dans l'informatique, permet de vérifier que la carte COTS2 est fonctionnelle. Elle est utilisée notamment lors de la phase de démarrage.
- La commande « GNR » (Get Number of Results) renvoie le nombre de rapports d'erreur en attente.
- La commande « GER » (Get Experiment Result) envoie au satellite un rapport d'erreur.

La séquence de démarrage de la carte est constituée d'une première salve de trois PINGs envoyée par le satellite. Si aucune réponse n'est reçue la carte est réinitialisée. Puis trois nouveaux PINGs sont émis. Si la carte n'a toujours pas répondu elle est coupée puis remise en route et un message d'erreur est transmis au sol. Enfin trois PINGs sont envoyés et si la carte ne répond toujours pas elle est désactivée en attendant l'analyse des données par les équipes au sol.

[11] Goddard Space Flight Center (GFSC) à Greenbelt dans le Maryland (Etat-Unis).
[12] Un voyage de deux semaines au GFSC a été effectué en janvier 2008 afin de procéder aux tests d'intégration de la carte dans le satellite.

Lors du fonctionnement normal, le satellite se contente de vider le buffer des messages d'erreurs. Pour cela il émet, à intervalles réguliers, la commande GNR. Si la carte COTS2 renvoie un résultat positif alors la commande GER est émise et COTS2 envoie un rapport d'erreur, en cas contraire le satellite ne fait rien et attend pour envoyer sa prochaine commande GNR.

La séquence d'extinction de la carte consiste à vider le buffer de rapport d'erreur. Pour cela le couple GNR/GER est utilisé jusqu'à ce que la carte COTS2 indique l'absence de rapport.

L'engin spatial oscillera entre une orbite base à 6000 km et une orbite haute a 12.000 km avec une inclinaison de 28°. Le lancement était initialement prévu pour fin 2007 / début 2008 mais dû à des retards, il n'aura pas lieu avant 2011.

Annexe D. Publications et Activités pendant la Thèse

Chapitres de Livres

R. Velazco, G. Foucard, P. Peronnard, "Integrated circuit qualification for Space and Ground-level Applications: Accelerated test and Error-Rate Prediction", Chapitre dans "Soft Errors in modern electronic systems", Springer Editions, parution fin 2009.

Conférences et Workshops

V. Pouget, A. Douin, D. Lewis, P. Fouillat, G. Foucard, P. Peronnard, V. Maingot, J. Ferron, L. Anghel, R. Leveugle, R. Velazco, "Tools and methodology development for pulsed laser fault injection in SRAM-based FPGAs", 8th Latin-American Test Workshop (LATW'07), Cuzco, Peru, March 12-14th, 2007.

A. Bocquillon, G. Foucard, F. Miller, N. Buard, R. Leveugle, C. Daniel, S. Rakers, T. Carriere, V. Pouget, R. Velazco, "Highlights of laser testing capabilities regarding the understanding of SEE in SRAM Based FPGAs", 9th European Conference on Radiation and its Effects on Components and Systems (RADECS'07), Deauville, France, September 10-14th, 2007.

V. Pouget, A. Douin, G. Foucard, P. Peronnard, D. Lewis, P. Fouillat, R. Velazco, "Dynamic Testing of an SRAM-Based FPGA by Time-Resolved Laser Fault Injection", 14th IEEE International Symposium On-Line Testing (IOLT'08), Rhodes, Greece, July 7-9th, 2008.

R. Velazco, P. Peronnard, G. Foucard, S. Fernandez, J. Pechiar, "A generic platform for SEE high altitude experiments", Electronique et rayonnements naturels au niveau du sol (RADSOL'08), CNRS, Paris, France, June 11-12th, 2008.

G. Foucard, P. Perronnard, R. Velazco, J. Ferron, A. Douin, V. Pouget, A. Bocquillon, F. Miller, G. Berger, F. Charlier, F. Boldrin, "Methodologies and Tools for the Evaluation of the Sensitivity to Radiation of SRAM-based FPGAs", 23rd International Conference on Design of Circuits and Integrated Systems (DCIS'08), Grenoble, France, November 12-14th, 2008.

P. Peronnard, R. Velazco, G. Foucard, "Impact of the Software optimization on the Soft Error Rate: a case study", 23rd International Conference on Design of Circuits and Integrated Systems (DCIS'08), Grenoble, France, November 12-14th, 2008.

P. Peronnard, R. Velazco, G. Foucard, V. Pouget, G. Berger, F. Charlier, F. Boldrin, "Remote SEE Testing Capabilities with Heavy Ions and Laser Beams at CYCLONE-HIF and ATLAS Facilities", Radiation Effects Data Workshop, Tucson, AZ, USA, July 14-18th, 2008.

P. Peronnard, R. Ecoffet, M. Pignol, D. Bellin, R. Velazco, G. Foucard, “Predicting the SEU Error Rate through Fault Injection for a Complex Microprocessor”, IEEE International Symposium on Industrial Electronics (ISIE'2008), Cambridge, UK, June 30th - July 2nd, 2008.

G. Foucard, P. Peronnard, R. Velazco, “Reliability Limits of TMR Implemented in a SRAM-based FPGA: Heavy ion measures vs. fault injection predictions”, 11th Latin-American Test Workshop (LATW’10), Punta del Este, Uruguay, March 28-31th, 2010.

Rapports Techniques et Projets

Gilles Foucard, “Rapport sur les mesures de sensibilité aux radiations du Virtex-II”, Rapport au CNES, December 5th, 2007.

Papiers acceptés

R. Velazco, G. Foucard, P. Peronnard, “Combining Results of Accelerated Radiation Tests and Fault Injection to Predict the Error Rates for SRAM-Based FPGAs”, 19th SEE Symposium, La Jolla, USA, April 12-14th, 2010.

R. Velazco, G. Foucard, P. Peronnard, “Combining Results of Accelerated Radiation Tests and Fault Injection to Predict the Error Rate of Applications Implemented in SRAM-Based FPGAs”, 31th Nuclear and Space Radiation Effects Conference (NSREC’10), Denver, USA, July 19-23th, 2010. Candidat pour parution dans IEEE Transaction on Nuclear Science fin 2010.

Bibliographie

[Adams 1993] A. Holmes-Siedle and L. Adams, “Handbook of Radiation Effects”, Oxford University Press, 1993.

[Atmel AT697E] Atmel AT697E processor, http://www.atmel.com/dyn/products/product_card.asp?part_id=3178

[Barak 2000] J. Barak, J.L. Barth, C.M. Seidleck, C.J. Marshall, P.W. Marshall, M.A. Carts, R.A. Reed, “Single event upsets in the dual-port-board SRAMs of the MPTB experiment”, IEEE Transaction on Nuclear Science, Vol. 47, pp. 712-712, 2000.

[Barth 1997] J. Barth, “Radiation Environments”, IEEE NSREC Short Course, Session I, July 21, 1997.

[Baumann 2005] R. C. Baumann, “Radiation-Induced Soft Errors in Advanced Semiconductor Technologies”, IEEE Transaction on device and materials reliability, Vol. 5, No. 3, September 2005.

[Bessot 1993-1] D. Bessot, “Radiation Hardening Technics facing Total Dose, “SEU and SEL in Space Environment”, rapport technique n°B465, Oxon, UK, Harwell Laboratory, 1993.

[Bessot 1993-2] D. Bessot, "Conception de deux points mémoire statiques CMOS durcis contre l’effet des aléas logiques provoqués par l’environnement radiatif spatial", Thèse de Doctorat, 26 Novembre 1993.

[Boudenot 2007] J.C. Boudenot, “Radiation Effects on Embedded Systems - Radiation Space Environment”, Springer, edition 2007.

[Bourrieau 1991] J. Bourrieau, “l’environnement spatial: flux, dose, blindage, effets des ions lourds”, Tutorial short course, RADECS’91, 1991.

[Buchner 1997] S. Buchner, M. Olmos, R. Velazco, P. Cheynet, J. Mellinger, “Pulsed Laser Validation of Recovery Mechanisms of Critical SEE’s in an Artificial Network System”, Proc. Of 4th European Conference on Radiation and its Effects on Components and Systems (RADECS’97), Cannes, France, pp. I10-I11, 1997.

[Calin 1996] T. Calin, M. Nicolaidis, R. Velazco, “Upset Hardened Memory Design for Submicron CMOS Technology”, IEEE Transaction on Nuclear Science, Vol. 43, pp. 2874-2878, 1996.

[Campbell 2002] A. Campbell, S. Buchner, E. Petersen, B. Blake, J. Mazur, C. Dyer, “SEU measurements and predictions on MPTB for a large energetic solar particle event”, IEEE Transaction on Nuclear Science, Vol. 49, pp. 1340-44, 2002.

[Canaris 1991] J. Canaris, “An SEU Immune Logic Family”, Proceedings of the 3rd NASA Symposium on VLSI Design, 2.3.1-2.3.11, 1991.

[Canivet 2008] G. Canivet, J. Clédière, J. B. Ferron, F. Valette, M. Renaudin, R. Leveugle, "Detailed analyses of single laser shot effects in the configuration of a Virtex-II FPGA", 14^th IEEE International On-Line Testing Symposium (IOLTS), Rhodes, Greece, July 7-9, 2008.

[Chau 2000] S .N. Chau, L. Alkalai, A. T. Tai, “Analysis of a multi-layer fault-tolerant COTS architecture for deep space missions”, Application-Specific Systems and Software Engineering Technology, 2000.

[Chee 2000] A. Chee, L. Braby, and T. Conroy, “Potential doses to passengers and crew of supersonic transports”, Health Phys., Vol. 79, pp. 547, 2000.

[Cheynet 1999] P. Cheynet, R.Valazco, R. Ecoffet, S. Duzellier, J.P. David, J.G. Loquet, “Comparison between ground tests and flight data for two static 32 KB memories”, 6th Radiation and Its Effects on Components and Systems (RADECS’99), pp. 554-557, 1999.

[Daly 1999] E.J. Daly, P. Buhler, M. Kruglanski, “Observations of the outer radiation belt with REM and comparisons with models”, IEEE Transaction on Nuclear Science, Vol. 46, pp. 1469-74, 1999.

[Diehl 1983] S.E. Diehl, J.E. Vinson, B. Shafer, T. Mnich, “Consideration for single event upset immune VLSI logic”, IEEE Transaction on Nuclear Science, Vol. 30, pp. 4501-4508, 1983.

[DiUbaldo 2000] J. A. DiUbaldo, “NASA Earth Observing System Mission Operations Center development using COTS products”, Aerospace Conference Proceedings, 2000.

[Dos Santos 1998] F. V. Dos Santos, “Techniques de conception pour le durcissement des circuits intégrés face aux rayonnements”, Thèse de doctorat de l’Université Joseph Fourier, 1998.

[Douin 2007] A. Douin, V. Pouget, D. Lewis, P. Fouillat, P. Perdu, “Jitter Improvement of Time-Resolved Photoelectric Laser Stimulation for Dynamic Imaging of Integrated Circuits”, Proc. of IEEE Instrumentation and Measurement Technology Conference (IMTC’07), Warsaw, Poland, May 1-3rd, 2007.

[Duzellier 1997] S. Duzellier, D. Falgubre, R. Ecoffet, I. Tsourilo, “EXEQ II and III: On-board experiments for the study of single events”, 4th Radiation and Its Effects on Components and Systems (RADECS’97), Cannes, France, pp. 504-509, 1997.

[Duzellier 2002] S. Duzellier, S. Bourdarie, R. Velazco, B. Nicolescu, R. Ecoffet, “SEE in-flight data for two static 32KB memories on high earth orbit”, Radiation Effects Data Workshop, pp. 1-6, 2002.

[Falguere 1994] D. Falguere, S. Duzellier, R. Ecoffet, "SEE In-Flight Measurement on the MIR Orbital Station", IEEE Transaction on Nuclear Science, Vol. 41, 1994.

[Faure 2002] F. Faure, P. Peronnard, R. Velazco, R. Ecoffet, “THESIC+, a flexible system for SEE testing”, Proc. of RADECS 2002 Workshop, pp. 231-234, 2002.

[Fucile 1997] P. Fucile, A. Gordon, F. Bahr, J. Dean, “Fiber optic control of an undulating platform”, OCEANS’97, MTS/IEEE, Vol. 1, pp. 105-108, 1997.

[Goldhagen 2000] P. Goldhagen, “Overview of aircraft radiation exposure and recent ER-2 measurements”, Health Phys., vol. 79, p. 526, 2000.

[Guenzer 1979] C. S. Guenzer, E. A. Wolicki, R. G. Allas, “Single event upset of dynamic ram's by neutrons and protons" IEEE Transactions on Nuclear Sciences, Vol. 26, December 1979.

[JEDEC] JEDEC, “Test procedures for the management of single-event effects in semiconductor devices from heavy ions irradiation”.

[JESD89] JESD89, “Measurement and reporting of alpha particle and terrestrial cosmic ray induced soft errors in semiconductor devices”.

[Kastensmidt 2005] F.L. Kastensmidt, L. Sterpone, L. Carro, M.S. Reorda, “On the optimal design of triple modular redundancy logic for SRAM-based FPGAs”, Proc. Of Design, Automation and Test in Europe (DATE’05), Vol. 2, pp. 1290-1295, 2005.

[Katz 1994] R. Katz, R. Barto, P. McKerracher, R. Koga, “SEU hardening of Field Programmable Gate Arrays (FPGAs) for Space Applications and Device Characterization”, IEEE Transaction on Nuclear Science, Vol. 41, pp. 2179-2186, 1994.

[Langenbacher 1996] H. Langenbacher, F. Zee, T. Daud, A. Thakoor, “Radiation behavior of analog neural network chips”, IEEE International Conference on Neural Networks, Vol. 2, pp. 943-950, 1996.

[Leray 2004] J.L. Leray, J. Baggio, V. Ferlet-Cavrois, O. Flanent, “Atmospheric Neutron Effects in Advanced Microelectronics, Standards and Applications”, ICICDT International Conference, 2004.

[Lesea 2005] A. Lesea, S. Drimer, J.J. Fabula, C. Carmichael, P. Alfke, “The rosetta experiment: Atmospheric soft error rate testing in differing technology FPGAs”, IEEE Trans. Device Mater. Rel., 5(3), Sept. 2005.

[Lopez 2005] C. Lopez-Ongil, M. Garcia-Valderas, M. Portela-Garcia, L. Entrena-Arrontes, "An autonomous FPGA-based emulation system for fast fault tolerant evaluation”, International Conference on Field Programmable Logic and Applications, 2005, Vol., pp. 397- 402, 24-26 Aug. 2005.

[Liu 1992] M.N. Liu, S. Whitaker, “Low Power SEU Immune CMOS Memory Circuits”, IEEE Transaction on Nuclear Science, Vol. 39, pp. 1679, 1992.

[LTspice4] http://www.linear.com/designtools/software/ltspice.jsp

[Ma 1989] T.P. Ma, P. Dressendorfer, “Ionizing Radiation Effects in MOS Devices and Circuits”, Wiley, New-York, 1989.

[MacKay 1997] G.F. MacKay, I. Thomson, I. Adams, R. Nickson, A. Ng, N. Sultan, “An instrument for monitoring proton-induced upsets in space electronics”, 4th Radiation and Its Effects on Components and Systems (RADECS’97), pp 495-498, 1997.

[Maingot 2007] V. Maingot, J. B. Ferron, R. Leveugle, V. Pouget, A. Douin "Configuration errors analysis in SRAM-based FPGAs: software tool and practical results", Microelectronics Reliability, Elsevier, Vol. 47, n°. 9-11, pp. 1836-1840, November, 2007.

[May 1979] T. C. May, M. W. Woods, “Alpha-particle-induced soft errors in dynamic memories" IEEE Transactions on Electronic Devices, Vol. 26, February 1979.

[Mc Lean 1984] F.B. Mc Lean, “A Framework of Understanding Radiation Induced Interface in SiO_2 MOS Structures”, IEEE Transaction on Nuclear Science, vol 31, pp. 1651-1657, 1984.

[Miller 1994] S.K. Miller, “Private communication”, Orbital Sciences Corporation, 1994.

[Moran 1995] A. Moran, K. LaBel, M. Gates, C. Seidleck, R. McGraw, M. Broida, J. Firer, S. Sprehn, “Single Event Effect Testing of the Intel 80386 Family and the 80486 Microprocessor”, Proc. Of 2nd European Conference on Radiation and its Effects on Components and Systems (RADECS’95), Arcachon, France, pp. 263-269, 1995.

[Musseau 1996] O. Musseau, “Single Event Effets in SOI technologies and devices", IEEE Transaction on Nuclear Science, Vol. 43, pp. 603-613, 1996.

[Normand 1998] E. Normand, “Single Event Effects in Avionics”, Boeing Radiation Effects Lab, December 1998.

[Normand 2001] E. Normand, “Correlation of inflight neutron dosimeter and SEU measurements with atmospheric neutron model”, IEEE Transaction on Nuclear Science, Vol. 48, pp. 1996-2003, 2001.

[Olsen 1993] J. Olsen, P.E. Becher, P.B. Fynboand, P. Raaby, J. Schultz, “Neutron-induced single event upsets in static rams observed at 10 km flight altitude”, IEEE Transaction on Nuclear Science, Vol. 40, pp 74-77, 1993.

[Peronnard 2008] P. Peronnard, R. Ecoffet, M. Pignol, D. Bellin, R. Velazco, “Predicting the SEU Error Rate through Fault Injection for a Complex Microprocessor”, 2008 IEEE International Symposium on Industrial Electronics (ISIE'08), Cambridge, UK, June 30th – July 2nd, 2008.

[Petersen 1983] E.L. Petersen, “Single Event Upsets in Space: Basic Concepts”, Tutorrial Short Course, IEEE NSREC’83, 1983.

[Pickel 1983] J.C. Pickel, “Single Event Upsets Mechanisms and Predictions”, Tutorial Short Course, IEEE NSREC’83, 1983.

[Pouget 2001] V.Pouget, H. Lapuyade, P. Fouillat, D. Lewis, S. Buchner, “Theorical Investigation o an Equivalent Laser LET”, Microelectronics Reliability, Vol. 41, pp. 1513-1518, 2001.

[Pouget 2007 SEEM] V. Pouget, P. Fouillat, D. Lewis, “Using the SEEM software for SET testing and analysis”, Radiation effects in embedded systems, Springer, edition 2007.

[Ritter 1997] J.C. Ritter, “Microelectronis and Photonics Test Bed”, 20th Annual ASS Guidance and Control Conference, Breckenridge, Colorado, USA, February 5-9th, 1997.

[Rockett 1988] L.R. Rockett, “An SEU-Hardened CMOS Data Latch Design”, IEEE Transaction on Nuclear Science, Vol. 35, pp. 1682, 1988.

[Salager 1999] L. Salager, “Conception en vue du Durcissement des circuits intégrés numériques aux effets radiatifs”, Thèse de Doctorat, 11 janvier 1999.

[Sexton 1989] F.W. Sexton, “SEU characterization of hardened CMOS 64K and 256K SRAM”, IEEE Transaction on Nuclear Science, Vol. 36, pp. 2311-2319, 1989.

[Sohn 2000] J. H. Sohn, “The effects of single event upsets in avionics systems”, M.S. thesis, Iowa State Univ., May 2000.

[Sterpone 2005] L. Sterpone, M. Violante, "Analysis of the robustness of the TMR architecture in SRAM-based FPGAs," IEEE Transaction on Nuclear Science, Vol.52, n°.5, pp. 1545- 1549, Oct. 2005.

[Taber 1993] A. Taber, E. Normand, “Single-event effects in avionics”, IEEE Transaction on Nuclear Science, Vol. 43, pp. 461-474, April 1993.

[Tylka 1997] A.J. Tylka, J.H. Adams, P.R. Boberg, B. Brownstein, W.F. Dietrich, E.O. Flueckiger, E.L. Petersen, M.A. Shea, D.F. Smart, E.C. Smith, “CREME96: A Revision of the Cosmic Ray Effects on Micro-Electronics Code”, IEEE Transaction on Nuclear Science, Vol. 44, pp. 2150-59, 1997.

[Velazco 1994] R. Velazco, D. Bessot, S. Duzellier, R. Ecoffet, S. Koga, “Two CMOS Memory Cells Suitable for the Design of SEU Tolerant VLSI Circuits”, IEEE Transaction on Nuclear Science, Vol. 41, pp. 2229, 1994.

[Velazco 1998] R. Velazco, Ph. Cheynet, R. Ecoffet, “Operation in space of artificial neural networks implemented by means of a dedicated architecture based on a transputer”, Integrated Circuit Design, pp 162-165, 1998.

[Velazco 1999] R.Velazco, P. Cheynet, A.Tissot, J. Haussy, L. Lambert, R. Ecoffet, “Evidences of SEU tolerance for digital implementations of artificial neural networks: one year MPTB flight results”, 6th Radiation and Its Effects on Components and Systems (RADECS’99), pp. 565-568, 1999.

[Velazco 2000] R. Velazco, S. Rezgui, R. Ecoffet, “Predicting Error Rate for Microprocessor-Based Digital Architectures through C.E.U. (Code Emulating Upsets) Injection”, IEEE Transaction of Nuclear Science, Vol. 47, pp. 2405-2411, 2000.

[Weatherford 1997] T.R. Weatherford, R. Radice, D. Eskins, J. Devers, D.J. Fouts, P.W. Marshall, C.J. Marshall, H. Dietrich, M. Twigg, R. Milano, “SEU Design Considerations for MESFETs on LT GaAs”, IEEE Transaction on Nuclear Science, Vol. 44, pp. 2282-2289, 1997.

[Web CREME96] https://creme96.nrl.navy.mil/

[Web Gaisler] Aeroflex Gaisler, http://www.gaisler.com/cms/

[Web LWS] http://lws.gsfc.nasa.gov/

[Web MicroSat] http://www.microsatsystems.com

[Web MPTB] http://www.nrl.navy.mil/content.php?P=04REVIEW209

[Web NASA] http://www.nasa.gov/

[Web NI] http://www.ni.com/lwcvi/

[Web NRL] http://www.nrl.navy.mil/

[Web Opencores] http://www.opencores.org

[Web PHP] http://www.php.net

[Web STRV] http://lasp.colorado.edu/strv/index.shtml

[Web TRAD] http://www.trad.fr

[Web TripleDES] http://www.opencores.org/project,des

[Whitaker 1991] S. Whitaker, J. Canaris, K. Liu, “SEU Hardened Memory Cells for a CCSDS Reed Solomon Encoder”, IEEE Transaction on Nuclear Science, Vol. 38, pp. 1471, 1991.

[Yui 2002] C. Yui, G. Swift, C. Carmichael, “Single Event Upset Susceptibility Testing of the Xilinx Virtex II FPGA”, MAPLD ’02 Laurel MD, 10-12 Sep 2002

TITRE

Taux d'erreurs dues aux radiations pour des applications implémentées dans des FPGAs à base de mémoire SRAM : prédictions versus mesures

La réduction des dimensions des transistors entraîne une augmentation de la sensibilité des circuits intégrés complexes face aux particules énergétiques présentes dans l'environnement naturel. Cela combiné à la tendance actuelle favorisant l'utilisation de circuits commerciaux pour les applications spatiales et aéronautiques impose de trouver des méthodologies permettant de durcir les applications face aux radiations. Les composants reprogrammables de type FPGA à base de mémoire de configuration SRAM sont des candidats appréciés pour les applications embarquées notamment grâce à leur capacité de se reconfigurer sur site. Cependant les particules peuvent engendrer une mutation de l'application implémentée en créant des erreurs dans la mémoire de configuration.

Les travaux réalisés au cours de cette thèse ont eu pour but principal l'étude d'une stratégie de prédiction du taux d'erreurs pour un système implémenté dans un FPGA à base de mémoire SRAM. Pour atteindre cet objectif une plateforme de test permettant la réalisation d'essais au sol en accélérateurs de particules et sous faisceau laser a été développée. Elle a également été utilisée pour mettre en œuvre une stratégie matérielle/logicielle d'injection de fautes. Le but final étant l'évaluation de la pertinence d'une telle approche, par confrontation des prédictions des taux d'erreurs avec les mesures issues des campagnes de test. Un second objectif fut consacré au développement d'une expérience embarquée, dans un satellite scientifique de la NASA, afin d'obtenir des informations sur le comportement du FPGA étudié et son application en environnement réel.

MOTS CLEFS

Environnement spatial, événements singuliers, FPGA à base de SRAM, Accélérateur de particules, faisceau laser, injection de fautes

TITLE

Radiation error rate for applications implemented in SRAM-based FPGA: prediction versus measures

ABSTRACT

Reducing the dimensions of transistors increases the sensitivity of complex integrated circuits to energetic particles present in natural environment. This combined with the current trend to use commercial components for space and aeronautical applications require finding appropriate methodologies to harden applications against radiations. Reprogrammable parts such as SRAM-based FPGAs are appreciated for embedded systems because of their on site reconfiguration capability. However, particles can cause a mutation of the implemented application by creating errors in their configuration memory.

The work achieved in this thesis focused on the study of an error rate prediction strategy for an application implemented in a SRAM-based FPGA. For this, a test platform was developed to perform accelerated ground tests in particle accelerators and with laser beams. Fault injection sessions were also conducted in order to evaluate the relevance of this approach by comparing the predictions with the measures obtained during radiation ground testing. A second aspect of the work concerned the development and the validation of an experiment devoted to be placed in orbit by NASA's LWS-SET scientific satellite in order to obtain data on the FPGA's behavior in real life environment.

KEYWORDS

Space environment, single effect events, SRAM-based FPGA, particle accelerators, laser beam, fault injection

INTITULE ET ADRESSE DU LABORATOIRE

Laboratoire TIMA, 46 avenue Félix Viallet, 38031 Grenoble, France.

ISBN : 978-2-84813-XXX-X