Analyse des effets d'attaques par fautes et conception sécurisée sur plate-forme reconfigurable

G. Canivet

INSTITUT POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

|__|__|__|__|__|__|__|__|__|__|

T H E S E

pour obtenir le grade de

DOCTEUR DE L'Institut polytechnique de Grenoble

Spécialité : « Micro et Nano Électronique »

préparée au laboratoire TIMA
dans le cadre de **l'École Doctorale *«Électronique, Électrotechnique, Automatique, Télécommunications, Signal»***

présentée et soutenue publiquement

par

Gaëtan CANIVET

le 23 septembre 2009

TITRE

Analyse des effets d'attaques par fautes et conception sécurisée sur plate-forme reconfigurable

DIRECTEUR DE THÈSE : Régis LEVEUGLE
CO- DIRECTEUR DE THÈSE : Marc RENAUDIN

JURY

M. Fabrice Monteiro,	Président
M. Lionel Torres,	Rapporteur
M. Dean Lewis,	Rapporteur
M. Régis Leveugle,	Directeur de thèse
M. Marc Renaudin,	Co-directeur de thèse
M. Frédéric Valette,	Examinateur
M. Jessy Clédière,	Invité

A la mémoire de mon grand-père Lucien et de ma grand-mère Suzanne

A mes parents et à ma sœur

Remerciements

Cette thèse a été financée par la Délégation Générale à l'Armement et je tiens à remercier en premier lieu les Délégués Généraux François Lureau et Laurent Collet-Billon pour ce financement.

Mes travaux de recherche ont été réalisés au sein du CESTI-LETI du service SASTI du CEA Grenoble et du groupe ARIS du laboratoire TIMA de Grenoble. Que François Vacherand, chef du service SASTI du CEA Grenoble, Jean-Yves Hervé, chef du laboratoire CESTI ainsi que Dominique Borrione, Directrice du laboratoire TIMA trouvent ici tous mes remerciements pour m'avoir accueilli et donné les moyens d'accomplir mes travaux de recherche dans leurs laboratoires.

Je voudrais remercier vivement les membres de mon jury de thèse :

Monsieur Fabrice Monteiro, Professeur à l'Université de Metz, pour m'avoir fait l'honneur d'être le Président de mon jury de thèse.

Messieurs Lionel Torres, Professeur à l'Université de Montpellier 2, et Dean Lewis, Professeur à l'Université de Bordeaux, pour m'avoir fait l'honneur de participer à mon jury de thèse en tant que rapporteurs.

Je remercie Jessy Clédière pour sa formation sur l'utilisation des bancs laser du laboratoire CESTI. A Frédéric Valette j'adresse mes remerciements pour ses conseils, ses pistes de recherche et ses commentaires sur mes rapports qui m'ont permis d'obtenir ces résultats.

Enfin, je tiens à exprimer toute ma profonde reconnaissance à mes directeurs de thèse, Messieurs Marc Renaudin, CTO de Tiempo et Régis Leveugle, Professeur à Grenoble INP. Je remercie dans un premier temps Marc pour m'avoir proposé ce sujet de thèse et m'avoir fait confiance dans différents domaines. Toute ma gratitude est adressée à Régis pour son encadrement, sa disponibilité, ses conseils, son soutien dans une période très difficile et pour la confiance qu'il m'a accordée durant ces trois années de thèse.

Je tiens aussi à remercier tous les membres du laboratoire CESTI pour leurs disponibilités et leurs conseils. Un grand merci à Axel Boness pour son soutien lors d'une période très difficile de ma thèse que lui seul connaît.

Je remercie également tous les membres du laboratoire TIMA que j'ai pu côtoyer durant ces trois années. Je pense à Anne-Laure Fourneret-Itié, Sophie Martineau et Corinne Durand-Viel pour leurs disponibilités et leurs joies de vivre. Je remercie également Nicolas Garnier pour ses compétences en informatique lors de mon arrivée au sein du laboratoire avec un ordinateur du CEA posant quelques soucis, ainsi que pour la création de comptes mails de l'association.

Pour leurs aides lors de la création de l'Association d'Accueil des Doctorants du Laboratoire TIMA (A2DT) mais également pour m'avoir supporté durant toutes les réunions que j'ai présidées, je remercie Hakim Zimouche, Taha Beyrouthy, David Rios et Gregory Lopin. Un grand merci aux autres membres du bureau qui nous ont aidés à faire vivre cette association.

Je remercie chaleureusement tous les membres des groupes d'ARIS. Les doctorants, stagiaires et ingénieurs : Gilles "l'expert" FPGA Xilinx, ma petite Juju, Gilles, Esteban et Pablo. Merci aux permanents des groupes Nacer, Mihail et Raoul que j'ai connus lors de ma première conférence internationale en Grèce.

Je ne saurais oublier mes collègues du groupe de Régis : Jean-Baptiste pour son super outil d'analyse, Salma pour sa bonne humeur et son apport féminin dans notre équipe, Pierre pour ses travaux manuels et les randonnées et enfin Vincent mon ex-collègue de bureau. Je pense aussi à Mohamed, Abdelhamid et Robert, pour les moments que nous avons passés dans notre si petit bureau initialement fait pour 2 alors qu'au final nous étions 3 ou 4.

Je voudrais également remercier Joël Bouvier du Laboratoire de Physique Subatomique et de Cosmologie de Grenoble pour son encadrement, ses formations et ses conseils qu'il a su me prodiguer durant mes différents stages et ma thèse.

Je remercie particulièrement Paolo plus connu sous le diminutif de "Papa" (merci Régis) et sa femme Anna pour tous les bons moments que nous avons passé ensemble. Je pense particulièrement à la Grèce mais aussi aux soirées que nous avons passées ensemble. Tu sais mon Papa tout ce que je pense de toi d'un point de vue professionnel et personnel….

Enfin, je remercie énormément ma famille, en particulier ma sœur et mes parents pour leur soutien et leurs encouragements qu'ils ont su m'apporter durant toutes ces années mais également pour leurs sacrifices. Je vous aime tous les trois, enfin quatre car j'ai oublié mon petit pioupiou Victor…
J'ai aussi une pensée pour mon grand-père Lucien et ma grand-mère Suzanne qui nous ont quittés trop tôt et qui doivent être fiers de mon cursus universitaire…

Table des matières

Liste des figures

Liste des tableaux

Introduction

Dans notre société actuelle, la sécurité est primordiale et touche tous les secteurs d'activité, tant industriels que tertiaires. Les systèmes mis en jeu, généralement à base de composants électroniques, peuvent contenir des informations confidentielles telles que des clés de chiffrement ou des codes de programmation dévoilant des savoir-faire industriels. Le principal objectif de la sécurité est d'assurer la confidentialité maximum aux données sensibles ainsi que leur intégrité. Le but est donc en particulier de garantir que les données protégées ne soient pas lisibles ou modifiables par tous mais uniquement par des personnes autorisées. Pour que ces données soient lues par les « bonnes » personnes et qu'elles puissent en vérifier l'authenticité et la provenance, il est possible d'utiliser des mécanismes de signature et d'authentification tels que les algorithmes de chiffrement de données à clés symétriques ou asymétriques. Dans certaines applications, les informations secrètes (telles que ces clés) sont stockées dans des composants et les fournisseurs de ces composants doivent garantir qu'elles ne peuvent pas être obtenues par des concurrents ou des personnes mal intentionnées.

Pour protéger efficacement les circuits, il est nécessaire de connaître les catégories d'attaquants. Ceux-ci peuvent être différenciés par leurs motivations (économique, curiosité ou recherche scientifique), leurs connaissances et leurs équipements techniques. La première catégorie d'attaquant est le *"pirate"* dit aussi *"hacker"*, lequel possède une faible connaissance du produit, peu de moyens techniques et financiers mais a de très bonnes idées et surtout du temps. Ensuite, les *"initiés"* possèdent une bonne connaissance des circuits et de leurs points faibles. Les *"criminels"* ont une faible connaissance technique mais une très forte motivation et surtout de l'argent. Il existe également des personnes possédant des moyens techniques et financiers et qui ont une grande connaissance des produits. Enfin, la dernière catégorie est la recherche scientifique composée des universités, des laboratoires de recherche avec leurs étudiants et leurs professeurs. Le but principal de cette catégorie est totalement l'opposé des précédents attaquants. La recherche scientifique est très utile dans le domaine de la sécurité et grâce aux attaques menées utilisant les moyens techniques des laboratoires, il est possible de proposer des contremesures efficaces contre les attaques des autres catégories.

En plus des catégories d'attaquants, il existe certaines normes de sécurité définissant les différents niveaux de sécurité des produits. En France, la Direction Centrale de la Sécurité des Systèmes d'Information

(DCSSI) assure la fonction d'autorité de régulation de la sécurité des systèmes d'information et est placée sous le contrôle du Secrétariat Général à la Défense Nationale (SGDN). La principale mission de la DCSSI est de délivrer aux industriels les certificats de sécurité selon la norme internationale des critères communs *"Common Criteria for Information Technology Security Evaluation"* définissant 7 niveaux d'assurance de l'évaluation (EAL1 à EAL7 pour *"Evaluation Assurance Level"*). Pour évaluer la sécurité des produits selon le niveau choisi par l'industriel et avant leur mise sur le marché, des centres sont chargés des expertises techniques et de l'évaluation : les Centres d'Évaluation de la Sécurité des Technologies de l'Information (CESTI). Les évaluations consistent dans un premier temps à vérifier la conformité du produit par rapport aux normes de sécurité internationales et de déceler les vulnérabilités potentielles. Dans un second temps, les évaluateurs matériels se mettent dans la peau d'un attaquant pour attaquer le produit par différentes techniques d'attaque générales et/ou spécifiques aux vulnérabilités du système.

Traditionnellement, les systèmes embarqués et/ou sécuritaires sont implantés dans des circuits **ASIC** (*Application Specific Integrated Circuit*) avec des fonctionnalités sur mesure pour une application donnée et possédant de hautes performances. Ces circuits doivent cependant être produits à volume très important (> 100 000 pièces par an) pour que les coûts de fabrication soient acceptables et la durée de développement de ces circuits peut être très longue. Les FPGA (*Field Programmable Gate Array*) permettent de concevoir des systèmes plus rapidement qu'avec des ASIC et à moindre coût. L'écart des caractéristiques entre les ASIC et les FPGA tend à diminuer avec les avancées technologiques et les performances et la taille des FPGA permettent aujourd'hui de remplacer les ASIC dans la plupart des applications. Les FPGA sont donc de plus en plus utilisés dans des systèmes embarqués.

Puisque les FPGA deviennent de plus en plus utilisés et qu'ils peuvent devenir une partie d'un système embarqué pouvant contenir des éléments confidentiels, il est nécessaire de caractériser leur sensibilité face aux attaques. De nombreux travaux font état d'attaques par observation (notamment analyse de la consommation). D'autres portent également sur les effets de perturbations naturelles sur la configuration du circuit dans le domaine spatial. Mais jusque là peu de recherches ont concerné l'effet des attaques par injections de fautes matérielles telles que les erreurs de configuration ou de données dues à des tirs laser ou à des surtensions. L'objectif de cette thèse est de caractériser les effets induits par ces deux types d'attaques sur la configuration et sur les bascules utilisateurs. Pour nos travaux de recherche, une famille ancienne de FPGA de type SRAM a été choisie : la famille Xilinx Virtex II. La technologie ayant une finesse de gravure moindre que les FPGA les plus récents, cette famille est a priori moins sensible aux perturbations que les familles plus récentes. Par ailleurs, nous avions sur cette famille une connaissance assez fine du rôle des différents bits de configuration, ce qui a pu être exploité lors des différentes analyses effectuées.

Le premier chapitre de ce manuscrit présente les différentes plates-formes reconfigurables et plus particulièrement les différentes technologies de FPGA. Dans ce chapitre, nous présenterons également le

circuit sur lequel les travaux de recherche ont été réalisés. Enfin, pour sécuriser leurs circuits contre différentes attaques, les fabricants de FPGA ont implanté des protections internes que nous résumerons.

Le second chapitre est un état de l'art des différentes techniques d'attaque de circuits électroniques touchant ou non à l'intégrité du circuit. Ces attaques peuvent ainsi être classées en trois catégories : les attaques invasives où le circuit est fortement (et souvent irrémédiablement) modifié, les attaques non invasives qui consistent principalement à observer les signaux et les attaques semi-invasives où quelques modifications sont apportées aux circuits. Nous présenterons également dans ce chapitre, de manière non exhaustive, quelques protections contre les attaques citées précédemment.

Les conditions expérimentales utilisées lors des ces travaux de recherche sont définies dans le troisième chapitre. Les différents bancs de surtensions et lasers seront introduits ainsi que la préparation des échantillons et les cartes de test conçues. Pour caractériser et comprendre les effets des attaques par injections de fautes sur des FPGA, nous avons aussi utilisé un outil développé au laboratoire TIMA de Grenoble. La fonctionnalité de cet outil et les différents circuits implantés dans le FPGA pour les caractérisations seront présentés.

Le chapitre 4 permet de caractériser les effets généraux des attaques par laser ou par application de surtensions. Lors des campagnes d'attaque laser, nous montrerons qu'un tir unique n'induit pas un nombre d'erreurs de configuration constant. Pour différents bancs laser, une classification de la sensibilité des éléments des tuiles CLB (logique ou interconnexion) sera introduite ainsi que les différentes modifications possibles des connexions en fonction de leurs états initiaux. En appliquant des surtensions, nous montrerons qu'un seul type d'élément est modifié dans la configuration du composant. Une analyse détaillée et une interprétation au niveau transistor des zones de sensibilité des bits de configuration seront présentées. Enfin, puisque les tirs laser n'induisent pas un nombre constant d'erreurs dans la configuration, une étude du cas particulier des fautes uniques sera menée selon le diamètre du spot laser.

Pour mener à bien une attaque laser, un attaquant contrôle plusieurs paramètres de son banc tels que le nombre de tirs par position, la longueur d'onde, la puissance, la taille du spot laser et la durée de l'impulsion. En utilisant un banc laser avec une longueur d'onde, une taille de spot et une puissance fixes, l'attaquant pourra reproduire plusieurs fois les tirs à une même position afin d'obtenir au moins une fois une perturbation efficace. La reproductibilité des effets des tirs laser sur la configuration des tuiles CLB et BRAM sera donc étudiée dans le chapitre 5. La reproductibilité du cas particulier des fautes uniques sera également étudiée. L'autre paramètre aisément modifiable est la durée de l'impulsion. Dans ce chapitre, nous montrerons que la durée de l'impulsion a une influence sur le nombre d'erreurs de configuration et plus particulièrement dans les tuiles CLB. Comme précédemment, nous étudierons le cas particulier des fautes uniques en fonction de ce paramètre.

Le dernier chapitre de cette thèse concerne l'étude d'un crypto-processeur sécurisé contre les attaques par fautes, implanté sur un FPGA de type SRAM. Dans ce chapitre, le fonctionnement du crypto-processeur

sera présenté avec la contremesure implantée. Les résultats des attaques par tirs laser et par application de surtensions seront donnés et une amélioration efficace de la contremesure sera proposée et validée.

Chapitre 1. Plates-formes reconfigurables

Dans ce chapitre, nous allons introduire les différentes technologies de FPGA présentes sur le marché. Nous présenterons plus particulièrement le circuit sur lequel les travaux de recherche ont été effectués. Dans une dernière partie, un état de l'art non exhaustif des protections intrinsèques implantées par les constructeurs afin de sécuriser les circuits implantés dans leurs FPGA sera établi.

1.1. Les différents types de réseaux programmables

1.1.1. Un peu d'histoire

Les premiers composants à logique programmable furent les mémoires PROM (*Programmable Read Only Memory*), qui sont des mémoires non-volatiles. Elles furent inventées en 1957 par Wen Tsing Chow travaillant pour *Arma Corporation*, une division d'*American Bosch Corporation* sur une demande de l'armée de l'air américaine afin de rendre plus flexibles et sécurisées les coordonnées des cibles des missiles *Atlas E/F*. Ce brevet et la technologie associée ont été tenus secrets durant plusieurs années ([Chow 1962]) alors que l'*Atlas E/F* était le missile opérationnel de l'armée américaine. Par la suite, les mémoires PROM pouvaient être configurées soit à la fabrication à partir de masques soit par l'utilisateur. Cependant, elles ne sont pas des architectures efficaces pour réaliser des circuits logiques et des évolutions technologiques ont été apportées pour pouvoir effacer le contenu de la mémoire soit par ultra-violet (*Erasable PROM* ou *UV-PROM*) soit électriquement (*Electrical Erasable PROM*). Au milieu des années 1980, une nouvelle technologie de PROM apparut combinant la programmabilité des EPROM et l'effacement des EEPROM, et cette nouvelle technologie s'appelle les *Flash-EEPROM* ([Sharma 2002]).

En 1970, Texas Instruments a développé un composant programmable par masques basés sur la *Read-Only Associative Memory (ROAM)* d'IBM. Ce composant était programmé en altérant une couche de métal lors de la fabrication du circuit intégré et s'appelait le TMS2000. Texas Instrument déposa alors le terme de *Programmable Logic Array (PLA)* pour ce composant. En 1973, National Instrument présenta un composant PLA programmable par masques, le DM7575 qui eut un succès plus important que le TMS2000.

Au milieu des années 1970, apparut le premier circuit développé par Philips pour implanter des circuits logiques les *Field-Programmable Logic Array (FPLA)* communément appelés PLA. Ces circuits étaient assez

cher à développer et n'offraient pas de grandes performances en vitesse. Pour combler ces faiblesses les *Programmable Array Logic (PAL)* ont été développés en 1978 par MMI qui a fusionné avec AMD par la suite. Les circuits PAL se composent d'un plan "ET" programmable connecté à des portes logiques "OU". L'étape suivante fut la création en 1985 des *Generic Array Logic (GAL)* par Lattice, qui sont des circuits en technologie EEPROM comprenant un nombre élevé de fonctions logiques "ET" avec la possibilité d'utiliser un registre. Ces circuits PLA, PAL, GAL sont des circuits à logique programmable (*Programmable Logic Devices : PLDs*). Pour réaliser de "gros" circuits logiques, les circuits PAL et GAL ne sont pas suffisants et apparurent les *Complex Programmable Logic Devices (CPLD)*. Les CPLD peuvent contenir plusieurs PAL connectés par des interconnexions programmables et remplacer plusieurs centaines de milliers de portes logiques. Les concepts de reprogrammation de portes ou de blocs logiques apparurent initialement dans les brevets déposés par David W. Page et LuVerne R. Peterson ([Page 1985-1], [Page 1985-2]).

En 1985, une nouvelle architecture de logique programmable a été présentée sous le terme de *Logic Cell Array (LCA)*. En parallèle, des réseaux de portes interconnectées de manière programmable grâce à des anti-fusibles ont été mis sur le marché par Actel sous l'appellation "Field Programmable Gate Array (FPGA)". Aujourd'hui, cette appellation a été généralisée aux successeurs des LCA. Dès 1985, les cofondateurs de Xilinx, Ross Freeman et Bernard Vonderschmitt ont mis sur le marché le premier LCA commercialement viable : le XC2064. Ce LCA avait des portes et des connexions programmables. Il embarquait un maximum de 1200 portes dont environ 800 utilisables, 58 entrées-sorties et des bascules pouvant fonctionner à 20Mhz. Au début de l'année 2009, Ross Freeman est entré au panthéon des inventeurs puisqu'il a été admis au *National Inventor's Hall of Fame* pour son invention ([Xilinx 2009-1]).

A la fin des années 1980, le département américain *Naval Surface Warfare Department* a mis en place une expérience proposée par Steve Casselman pour concevoir un système permettant d'implanter 600000 portes reconfigurables et ce projet fut breveté en 1992.

Les années 1990 furent une période très prospère pour les réseaux programmables de type FPGA à la fois dans la sophistication et dans la production. Au début de ces années, les FPGA furent principalement utilisés dans les télécommunications et les réseaux. Depuis, les réseaux programmables sont omniprésents dans tous les domaines y compris des domaines critiques tels que l'automobile, l'aéronautique, le spatial ou la sécurité.

Actuellement, il existe une multitude de FPGA qui se distinguent par plusieurs critères : le nombre de cellules équivalentes, les niveaux de tensions admissibles, les éléments additionnels tels que des convertisseurs, des DSPs, etc.…, mais également par le type de programmation. Actuellement, il existe les FPGA à configuration par anti-fusibles, à configuration SRAM ou à configuration Flash. Cependant des travaux de recherche portent également sur d'autres types de programmation comme sur les structures magnétiques.

1.1.2. Les anti-fusibles

Les réseaux programmables à anti-fusibles sont des composants programmables une seule fois. Les anti-fusibles sont des circuits ouverts jusqu'à ce qu'un courant soit appliqué à travers. Un exemple de structure d'anti-fusible est donné en Figure 1-1 et le fabricant Actel nomme cet anti-fusible le *"PLICE : Programmable Low Impedance Circuit Element"*. La Figure 1-1 montre un anti-fusible positionné entre 2 connexions et cet anti-fusible est constitué de 3 couches : 2 conducteurs et un isolant. Dans un état non programmé, l'isolant isole les deux couches conductrices, alors que lors d'une programmation l'isolant change d'état et devient une faible résistance permettant une connexion indirecte entre les deux conducteurs (Figure 1-2).

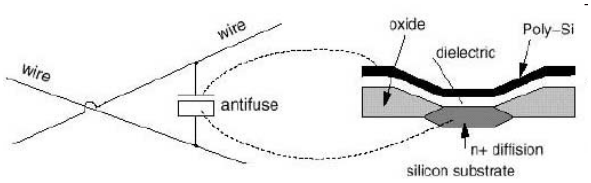


Figure 1-1 : Structure de programmation à anti-fusibles des composants Actel [Brown 1996]

En 1986, Actel a présenté le premier FPGA à anti-fusibles utilisant une structure composée de zones de diffusion n+, d'ONO (Oxyde-Nitride-Oxyde) et du poly-silicium. Le poly-silicium et les zones de diffusion sont utilisés comme conducteurs alors que l'isolant est basé sur l'ONO ([Hamdy 1988]). Cependant les anti-fusibles à base d'ONO réduisent la densité de portes logiques.

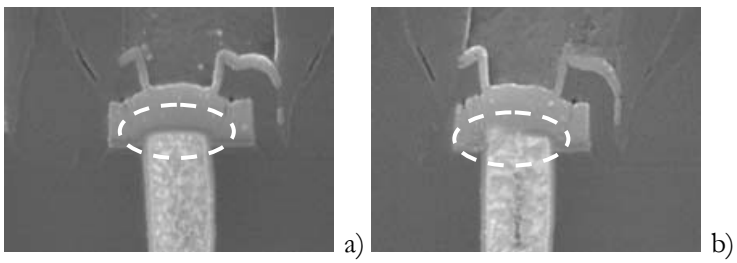


Figure 1-2 : Coupe d'un anti-fusible non-programmé (a) et programmé (b) [Actel 2002]

En 1991, QuickLogic a donc présenté une nouvelle technologie d'anti-fusibles basée sur du silicium amorphe ([Birkner 1992], [Marple 1994], [Liu 1998]). Cette nouvelle technologie permet d'obtenir une faible résistance et une petite taille pour l'anti-fusible permettant ainsi d'accroître les performances des FPGA. Une présentation plus détaillée des structures à anti-fusibles est donnée dans [Greene 1993].

Le principal avantage des anti-fusibles est leur petite taille, mais cet avantage est réduit par la taille importante des transistors de programmation. En effet, les transistors de programmation doivent être dimensionnés pour laisser passer de forts courants et permettre une isolation entre les fortes tensions de programmation et les faibles tensions de fonctionnement. D'autres avantages de cette technologie sont une confidentialité élevée du fait de la difficulté de lire l'état du fusible et une faible sensibilité aux radiations ou aux perturbations car il est impossible de changer l'état du fusible une fois programmé. Par

contre, les principaux inconvénients sont une programmation unique et une tension élevée de programmation.

1.1.3. Les technologies SRAM

Grâce à leur niveau de performance, les mémoires statiques à accès aléatoire (*Static-Random Access Memory : SRAM*) sont largement utilisées pour l'élaboration des circuits intégrés tels que les FPGA. La cellule SRAM la plus utilisée est une cellule à 6 transistors (6T-SRAM) constituée de 2 inverseurs montés tête-bêche et de 2 transistors d'accès (Figure 1-3).

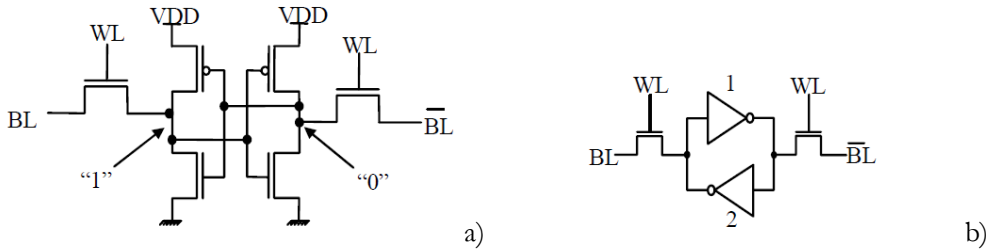


Figure 1-3 : Schéma d'une mémoire à 6T-SRAM (a) et vue schématique correspondante (b)

La lecture et l'écriture se font avec 2 signaux de commande : la paire de *"bit lines BL & BL/"* et la *"word line : WL"*. Pour écrire dans une cellule mémoire, on force la valeur dans la cellule sélectionnée : la *word line* est activée, la paire de *bit line* est forcée à des valeurs complémentaires et transfère la donnée à l'intérieur de la cellule. Pour la lecture, la paire de *bit line* est laissée flottante et l'ouverture de la *word line* transfert l'information de la cellule sur la paire de *bit line*.

Cependant, les FPGA utilisent classiquement des 5T-SRAM ([Tuan 2007]) et un transistor de remise à zéro lors de la mise sous tension (Figure 1-4). Le principe de fonctionnement de cette cellule à 5 transistors est similaire à celle à 6 transistors.

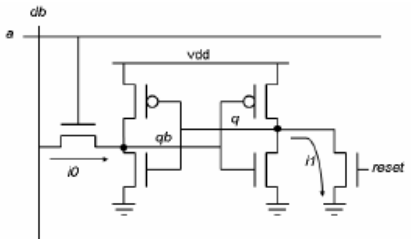


Figure 1-4 : Schéma du point mémoire SRAM ([Tuan 2007])

Un exemple d'utilisation de la structure de programmation à base de point mémoire SRAM est présenté dans la Figure 1-5 où ces cellules SRAM sont utilisées pour commander des grilles de transistors de connexions ou pour contrôler des multiplexeurs connectés à des blocs de logique. Ainsi pour réaliser des connexions, il suffit de rendre passant les transistors. Puisque les cellules SRAM sont volatiles, le FPGA doit être configuré à chaque mise sous tension et l'ajout d'éléments de stockage externes non volatiles est nécessaire tels que des EEPROM.

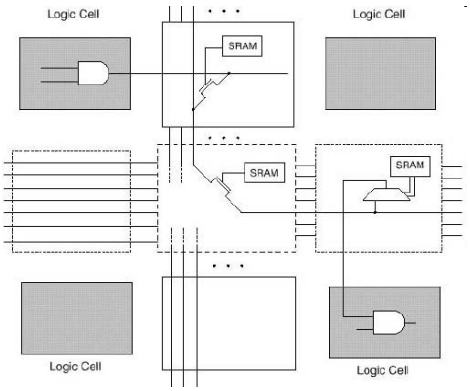


Figure 1-5 : Structure de programmation à base de point mémoire SRAM [Brown 1996]

Le principal inconvénient de cette technologie est sa surface à cause du nombre de transistors nécessaire par point mémoire. Malgré cet inconvénient, cette technologie a 2 principaux avantages que sont la reconfiguration partielle ou totale et l'utilisation des procédures de fabrication classiques des circuits intégrés.

1.1.4. Les technologies Flash

Les mémoires Flash sont des composants dont le contenu peut être effacé électriquement. Elles sont basées sur les technologies EPROM (*Erasable Programmable ROM*) ou EEPROM (*Electrically-Erasable Programmable ROM*) en fonction des applications nécessitant soit la forte densité des EPROM soit la flexibilité de programmation des EEPROM. La majorité des mémoires Flash sont programmées à partir des techniques des EPROM et effacées avec le mécanisme des EEPROM. La structure des points mémoires Flash est très similaire à celle des mémoires EPROM, d'une taille légèrement supérieure et avec une épaisseur d'oxyde très fine de l'ordre de quelques dizaines de nanomètres. Il existe trois architectures distinctes de mémoires Flash : les NOR-EPROM, les EEPROM basées sur des portes logiques NOR et les EEPROM basées sur des portes logiques NAND (Figure 1-6). Ces architectures diffèrent principalement par leur densité, les temps d'accès et la taille des blocs.

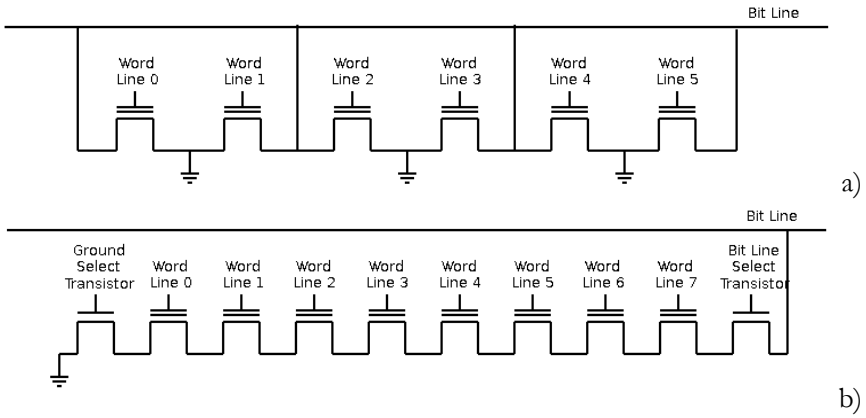


Figure 1-6 : Structures des cellules mémoires Flash à base de portes logiques NOR (a) et de portes logiques NAND (b)

Les principaux avantages de cette technologie Flash est quelle est normalement plus résistante aux particules ionisantes donc aux fautes que les technologies SRAM et que le temps de configuration est faible du fait de la configuration simultanée de plusieurs cellules. Par contre, cette technologie nécessite une alimentation supplémentaire élevée servant à la programmation et à l'effacement pouvant être intégrée au composant sous la forme d'un système de pompage de charge. Un autre inconvénient de cette technologie est une reconfiguration peu flexible.

1.1.5. Les technologies émergentes

Pour remplacer les mémoires SRAM ou Flash des FPGA, plusieurs travaux de recherche portent sur de nouvelles technologies de mémoires non-volatiles telles que les mémoires magnétiques ou les mémoires à nanotubes de carbone.

Le développement des mémoires magnétiques RAM (MRAM) a débuté dans les années 1990. Les informations sont stockées sous forme magnétique et non sous forme électrique (Figure 1-7). Des FPGA basés sur des mémoires magnétiques utilisant la charge de l'électron sont présentées dans [Zhao 2007] et [Guillemenet 2008]. Ce type de mémoires permet d'obtenir des performances élevées, une densité d'intégration importante et un stockage fiable.

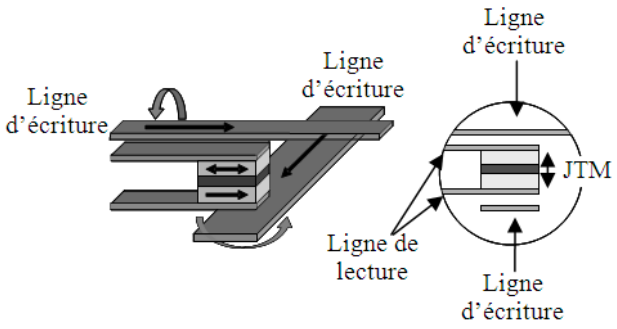


Figure 1-7 : Lecture-Écriture d'une jonction magnétique à tunnels [Guillemenet 2008]

D'autres travaux de recherche portent sur de nouvelles technologies de mémoires qui pourront probablement être implantées dans les FPGA. On peut citer par exemple les mémoires Nano-RAM dont le fonctionnement est fonction de la position des nanotubes de carbone, ou les mémoires PC-RAM (Phase-change RAM). Ce dernier type de mémoire utilise le comportement du verre de chalcogénure passant sous l'effet de la chaleur de la forme cristalline à la forme amorphe. Le principe de fonctionnement de ces mémoires est expliqué plus en détail dans [Postel-Pellerin 2008].

1.2. Le Virtex-II

Dans le cadre de nos études, nous nous sommes focalisé sur les FPGA configurables par mémoire SRAM. De plus, nous avons choisi de considérer une famille de composant un peu ancienne, a priori moins sensible aux perturbations que les FPGA récents du fait de la technologie de fabrication, des tensions de

seuils etc.…. Nous allons donc donner maintenant plus de détails sur la famille Xilinx Virtex II, sur laquelle s'appuie le travail effectué pendant cette thèse.

1.2.1. L'architecture du FPGA

1.2.1.1 Présentation générale

L'architecture du FPGA est optimisée pour des circuits à hautes densités et performances. Le composant programmable est composé de différents éléments configurables tels que les blocs d'entrées-sorties (communément appelés *Input/Output Blocks : IOBs*) et de la logique interne configurable (Figure 1-8).

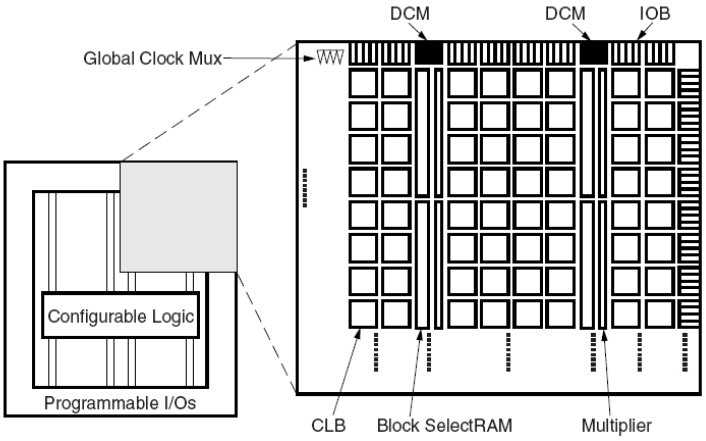


Figure 1-8 : Architecture du Virtex-II [Xilinx 2007-1]

Les blocs d'entrées-sorties permettent de faire l'interface entre les broches du composant et la logique interne de configuration. Les entrées-sorties programmables supportent la plupart des niveaux de tension présents actuellement (TTL, LVTTL, LVDS, etc…).

La logique interne configurable est constituée de 4 éléments majeurs organisés dans une matrice régulière (Figure 1-9). Ces éléments sont les tuiles de blocs de logique configurable (*Configurable Logic Block : CLB*), les blocs de mémoire RAM (*Block SelectRAM : BRAM*), les blocs multiplieurs et les générateurs d'horloge numérique (*Digital Clock Manager : DCM*) :

- ✓ Les tuiles CLBs fournissent des éléments fonctionnels utiles pour implanter la logique combinatoire et la logique séquentielle en incluant des éléments de stockage de base;
- ✓ Les modules BRAMs fournissent des éléments de stockage de type RAM de 18Kbit avec un accès double port;
- ✓ Les blocs Multiplieurs sont des multiplieurs dédiés de 18 bits par 18 bits;
- ✓ Et les blocs DCMs permettent une auto-calibration des horloges afin d'obtenir des horloges en phase et sont utilisés par exemple pour faire des boucles à verrouillage de phases (*Phase Locked Loop : PLL*).

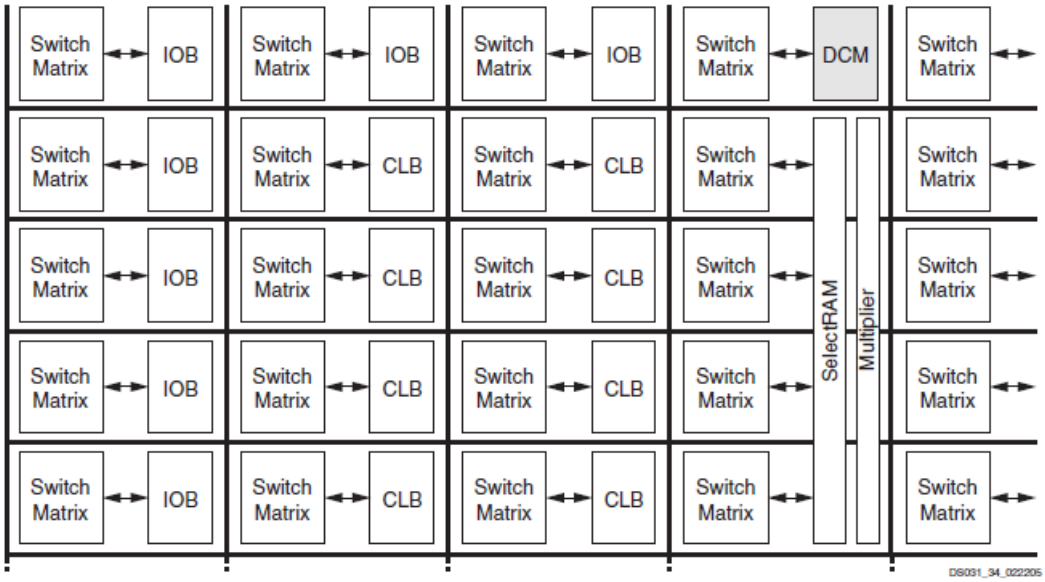


Figure 1-9 : Organisation des éléments fonctionnels et des interconnexions [Xilinx 2007-1]

Les différents blocs cités précédemment utilisent tous le même schéma d'interconnexion. A chaque bloc correspond une matrice d'interconnexions qui lui permet de communiquer avec l'extérieur via des ressources de routage globales et locales.

1.2.1.2 Blocs de logique configurables

Les blocs de logique configurables sont tous constitués de 4 slices, 2 buffers 3 états et une matrice d'interconnexions appelée *"Switch Matrix"* (Figure 1-10-a). Chaque slice présente dans les CLBs du FPGA est constituée de la même manière et possède les mêmes ressources (Figure 1-10-b) : 2 générateurs de fonctions à 4 entrées (F & G), 1 chaîne de retenue, des portes logiques pour les calculs arithmétiques, 2 éléments de mémorisation et des multiplieurs.

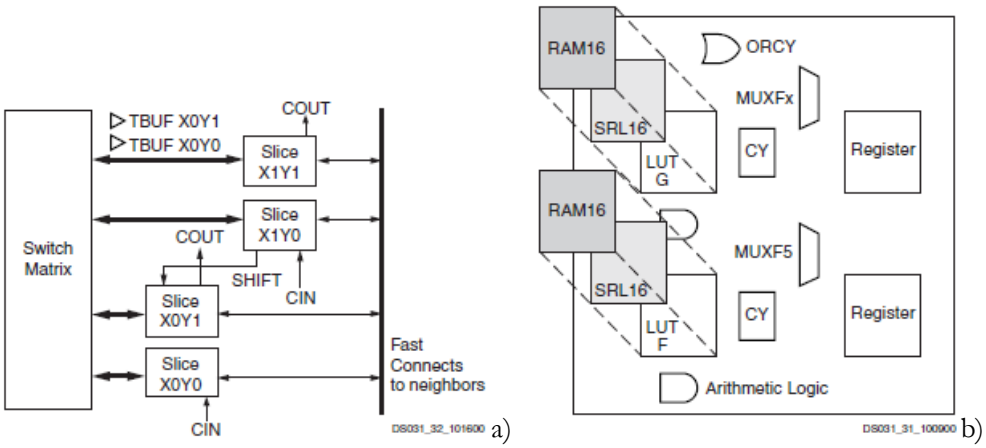


Figure 1-10 : Éléments constituant les CLBs (a) et les slices (b) [Xilinx 2007-1]

Les générateurs de fonctions F & G peuvent être utilisés en différentes configurations tels qu'en LUTs à 4 entrées servant aux équations logiques, en registres à décalages de taille 16 bits ou en mémoire distribuée

de type SelectRAM de taille 16 bits. Les éléments de mémorisation sont soit des bascules "D" actives sur le front montant de horloge soit des verrous sensibles à un niveau.

1.2.1.3 Blocs de mémoire Select-RAM

Les blocs de mémoires SelectRAM sont des blocs de 18 kilobits en technologie Dual-Port. Ces blocs sont programmables avec des profondeurs et des largeurs de configuration différentes, allant de 16K x 1 bit à 512 x 36 bits. Chaque port est totalement synchrone et indépendant. Afin d'obtenir un élément de stockage plus important, il est possible de cascader les blocs. Enfin, à chaque bloc de mémoire est associé un bloc de multiplieur 29 x 18 bits.

1.2.1.4 Horloges globales

Le Virtex-II comporte 12 blocs de DCMs. Chaque DCM peut être utilisé pour supprimer les délais dus aux distributions d'horloge. Ils servent également à concevoir un synthétiseur de fréquences ou une boucle à verrouillage de phase (PLL).

1.2.1.5 Ressources de routage

Les ressources de routage des FPGA Virtex-II sont optimisées pour être rapides et prévisibles temporellement en utilisant une matrice d'interconnexions configurable. Cette matrice d'interconnexions se situe au niveau des bus interconnectables et est reliée aux entrées de la tuile concernée. A chaque point d'interconnexion se trouve un PIP (*Programmable Interconnect Point*) lequel connecte les deux fils qu'il contrôle lorsqu'un ou plusieurs bits de routage sont activés. La configuration des connexions peut se faire par un ou plusieurs bits et sera présentée plus en détail dans la section suivante.

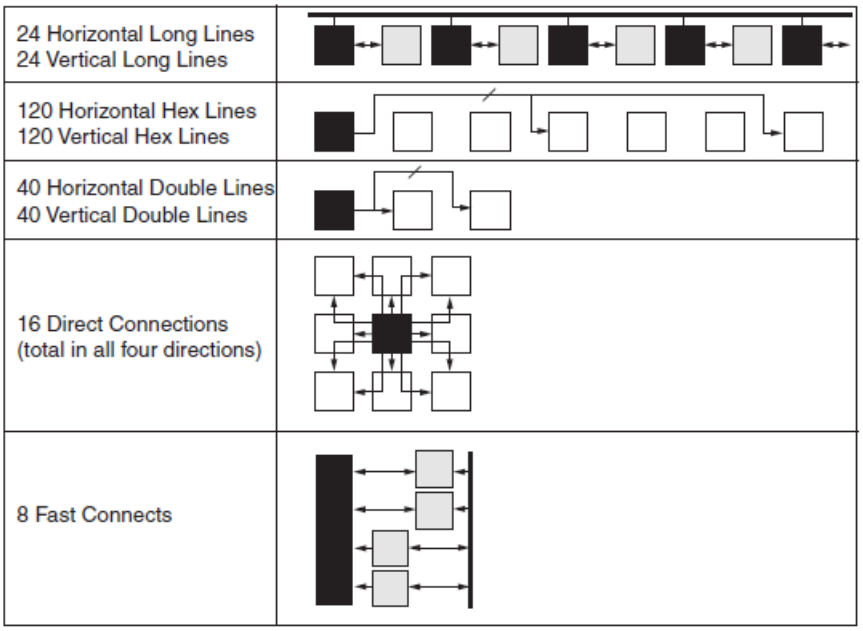


Figure 1-11 : Différents types de ressources accessibles par une tuile [Xilinx 2007-1]

Les ressources de routage connectées à la matrice d'interconnexions peuvent être définies en deux grands groupes : les ressources de routage globales et les ressources de routage locales (Figure 1-11).

Les ressources globales sont appelées ***"long lines"*** et sont des bus bidirectionnels qui parcourent le FPGA de haut en bas (long lines verticales) et de gauche à droite (long lines horizontales). Entre chaque tuile CLB passe un bus de type *"long line"* contenant chacun 20 fils.

Dans les ressources de routage locales, il existe trois types de liaisons : les "*double lines",* les "*hex lines*" et les *connexions directes*.

Les liaisons de type ***"double lines"*** mettent en relation une tuile CLB avec une tuile CLB mitoyenne et/ou la suivante. Ce type de connexion est séparé en 4 groupes de 10 fils dépendant de leur orientation cardinale (nord, sud, est ou ouest).

Les ***"hex lines"*** permettent de communiquer avec une tuile CLB éloignée de 3 ou 6 blocs horizontalement ou verticalement, et fonctionnent de la même façon que les liaisons de type "*double*".

Les ***connexions directes*** permettent de communiquer avec toutes les tuiles voisines, y compris les tuiles diagonales. Ce type d'élément correspond aux entrées-sorties des slices et sont connectés aux matrices d'interconnexions des CLB voisins.

En plus de ces ressources, des liaisons rapides existent à l'intérieur même d'une tuile CLB afin de pouvoir reboucler certains signaux. Elles permettent principalement de relier les sorties du LUT aux entrées du slice.

1.2.1.6 Structures de configuration des interconnexions

A l'intérieur d'une tuile CLB et comme présenté précédemment, il existe des interconnexions dont la structure de configuration est inhomogène et dont le nombre de bits nécessaire pour les configurer dépend de l'interconnexion considérée. Selon [Maingot 2007], la grande majorité des interconnexions nécessite 2 bits (90,3%) tandis que d'autres n'ont besoin que d'un bit (9,5%) ou de trois bits (0,2%).

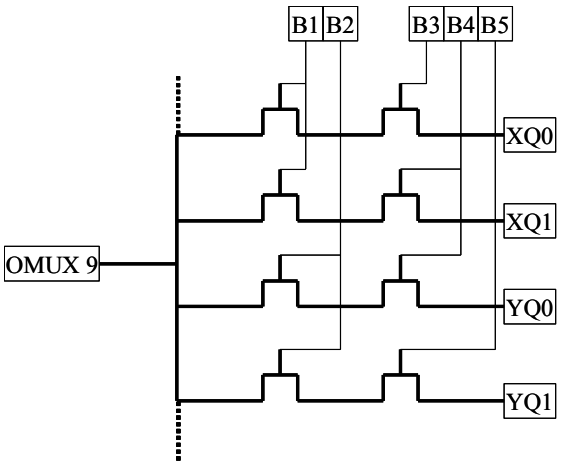


Figure 1-12 : Exemple de connexion définie par 2 bits ([Maingot 2007])

Le principe de fonctionnement des connexions gérées par un seul bit de configuration est assez simple puisque le bit de configuration agit comme un interrupteur : soit il y a une connexion soit il n'y en a pas.

Pour les autres interconnexions, la situation est plus compliquée puisque les bits ne sont pas affectés à des ressources de connexion mais à des listes de connexions possibles entre une ressource et plusieurs sources connectables. La Figure 1-12 présente un exemple de structure de configuration de la ressource OMUX9, une sortie de multiplexeur, configurable par 2 bits. Si l'on désire connecter la ressource OMUX9 avec la ressource XQ1, il faut trouver une intersection de listes de connexions possibles. Dans notre exemple, le bit B1 définit une liste permettant de connecter OMUX9 à XQ0 et à XQ1. Le bit B2 peut lier OMUX9 à YQ0 et à YQ1, etc…. Ainsi, en activant les bits B1 et B4, il sera possible de connecter la ressource OMUX9 à la ressource XQ1.

1.2.2. La configuration

La mémoire de configuration du composant Virtex-II est arrangée en frames verticales. Les frames sont les plus petits éléments adressables du plan mémoire du composant et leur longueur dépend de la taille du composant ([Xilinx 2007-1]). Le composant utilisé lors de nos travaux possède 1104 frames et chaque frame vaut 106 mots de 32 bits. Les frames de configuration sont regroupées en 6 types de colonnes correspondant aux ressources physiques du composant que sont les blocs d'entrées-sorties et leurs interconnexions, les blocs de logique configurable, les horloges globales et les blocs de mémoire avec leurs interconnexions. La Figure 1-13 représente la relation entre les ressources physiques du composant et la mémoire de configuration.

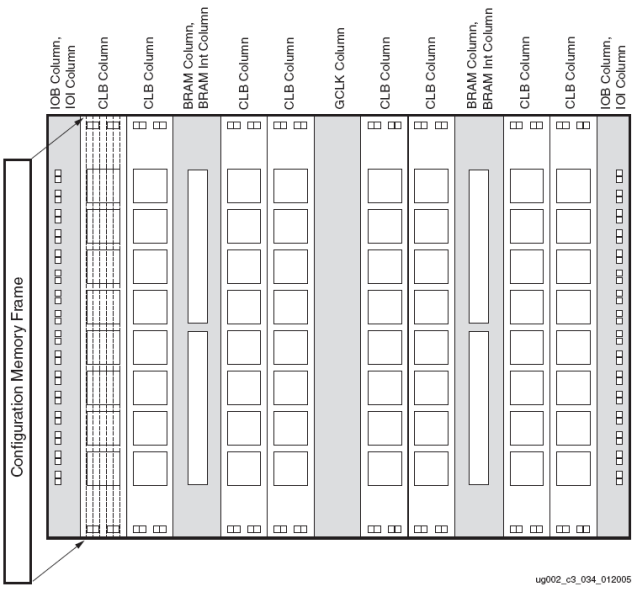


Figure 1-13 : Organisation des colonnes dans les composant Virtex [Xilinx 2007-1]

Tous les Virtex-II possèdent le même nombre de colonnes pour les blocs d'entrées-sorties et leurs interconnexions et pour les horloges globales. Cependant, le nombre de colonnes de logique configurable et de mémoire RAM avec leurs interconnexions diffère selon le composant ([Xilinx 2007-1]).

Pour configurer le composant en envoyant la configuration des différentes colonnes, il existe différents modes de programmation tels que le mode série, le mode Select Map ou le JTAG ([Xilinx 2007-1]). Ce

mode de programmation JTAG a été utilisé pour configurer notre circuit et relire sa configuration lors de nos travaux de recherche. Le mode série permet de configurer le FPGA bit par bit à partir d'une mémoire externe (Figure 1-14-a). Le mode Select Map est le mode de configuration le plus rapide car la programmation du circuit se fait octet par octet (Figure 1-14-b). Ces deux modes utilisent des broches ayant une double fonctionnalité car elles peuvent être utilisées pour la configuration mais également comme des entrées-sorties classiques. Enfin, le dernier mode est le JTAG qui permet à partir de broches du composant initialement dédiées au test de charger une configuration de manière série bit à bit (Figure 1-14-c).

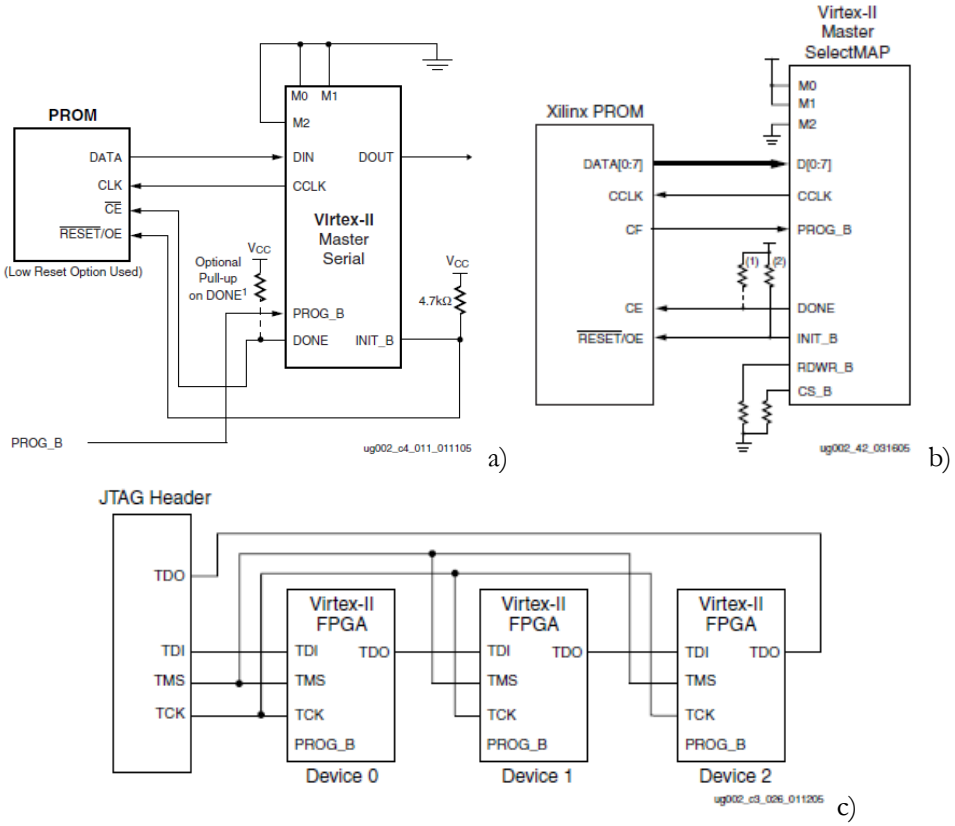


Figure 1-14 : La configuration série (a), Select Map (b) ou JTAG du Virtex-II [Xilinx 2007-2]

1.3. Les protections intrinsèques

Pour assurer une bonne intégrité de fonctionnement et une sécurité élevée, les fabricants de FPGA ont implanté certaines protections pour se protéger contre différents types d'attaques tels que le clonage du fichier de configuration, le changement de l'environnement de fonctionnement ou l'injection de fautes. Le chapitre 2 présentera plus en détail les différentes catégories d'attaques. Nous ne nous limiterons pas ici à la famille Virtex II.

1.3.1. Les verrous de relecture

Pour vérifier le bon fonctionnement des circuits implantés dans les FPGA, les fabricants de FPGA SRAMs ou Flash permettent la relecture de la configuration implantée. Cependant face à certaines attaques, une relecture peut être utilisée pour cloner par exemple un circuit. Pour contrecarrer cette attaque, les concepteurs peuvent bloquer la relecture de manière permanente ou non.

Le fabricant Actel a conçu 2 types de sécurités pour protéger les composants à mémoire Flash : le *FlashLock* et le *Permanent FlashLock*. Le système *FlashLock* est un mécanisme permettant à l'utilisateur de verrouiller ou de déverrouiller le composant avec une clef qu'il a défini dont la taille dépend du composant ([Actel 2003]). Lorsque le composant est verrouillé par ce mécanisme, les modes de lecture, d'écriture, de vérification et d'effacement sont désactivés. Le mécanisme *Permanent FlashLock* utilise le même principe que précédemment sauf que la clef est cassée rendant ainsi l'accès aux ressources internes du composant impossible même en utilisant la bonne clef. Ce dernier mécanisme permet de rendre un FPGA de type Flash aussi sécuritaire qu'un FPGA programmé par anti-fusibles.

Altera n'autorise pas la relecture de la configuration pour ces composants. Pour vérifier que la bonne configuration est implantée dans le composant, il utilise le contrôle de redondance cyclique (CRC) lors du chargement du bitstream mais également de façon périodique. Le principe du CRC sera présenté dans la section 1.3.3 de ce chapitre.

Le fabricant Lattice a également le même principe de protection afin d'empêcher la lecture des données de configuration en utilisant un bit de sécurité. Ce bit protège la relecture des mémoires SRAMs ou Flash ([Lattice 2006-1], [Lattice 2006-2]). Cependant, lorsque le bit est activé, les opérations d'écriture et d'effacement sont autorisées et lors d'opérations de lecture uniquement des zéros seront retournés.

Enfin Xilinx possède également le même principe de blocage de la relecture de la configuration. Cette option se fait à partir de l'outil de compilation, dans lequel il est possible d'interdire la relecture ou d'interdire la relecture et la reconfiguration. Ces options seront directement générées dans le fichier de configuration qui sera envoyé au FPGA en même temps que la configuration des différents éléments présentés dans la section 1.2.

1.3.2. Le chiffrement du bitstream

Comme présenté précédemment, les FPGA de type SRAM nécessitent tous l'ajout de mémoires externes contenant la configuration qui sera transmise à la mise sous tension du composant. Le fait d'utiliser des mémoires externes augmente le risque de clonage des circuits. Pour se prémunir contre ce type d'attaque, il est possible de chiffrer les fichiers de configuration et des modules de déchiffrement ont été implantés. En fonction de la famille de composant choisie, les algorithmes de chiffrement utilisés ne sont pas les mêmes. Par exemple, Xilinx utilise l'algorithme de chiffrement DES ou triple DES ([FIPS PUB 46-3]) pour les composants Virtex-II ([Xilinx 2005]) alors que leurs composants plus récents à savoir les Virtex-4 ou

Virtex-5 ([Xilinx 2006-1], [Xilinx 2006-2]) utilisent le chiffrement de type AES-256 ([FIPS PUB 197]). Le chiffrement est réalisé par l'outil de compilation et le déchiffrement par le module implanté dans le composant. Pour déchiffrer la configuration, le composant a besoin d'une clef, laquelle est stockée dans le FPGA et maintenue par une batterie additionnelle (Figure 1-15).

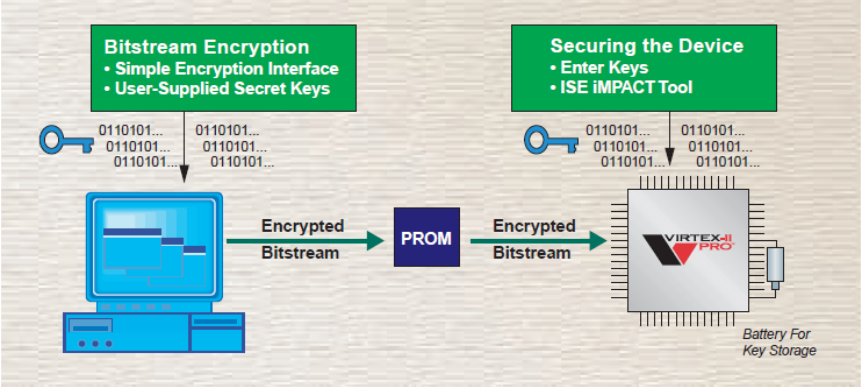


Figure 1-15 : Sécurisation de la configuration des composants Xilinx [Telikepali 2003]

Altera a implanté dans ses composants Stratix-II un module de déchiffrement AES avec une clef de 128 bits laquelle est présente dans le FPGA (Figure 1-16). Il en est de même pour le fabricant Lattice avec ses composants LatticeECP2/M ([Lattice 2006-3]).

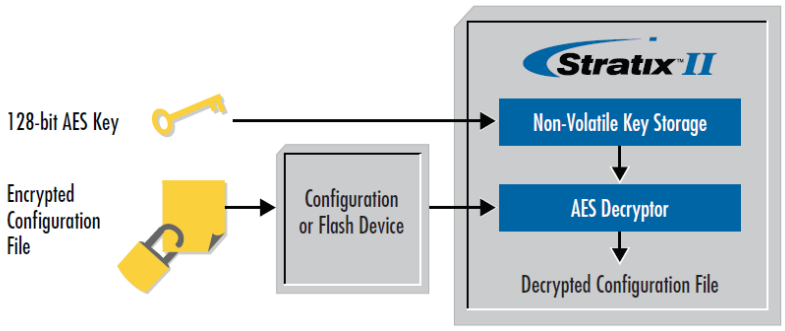


Figure 1-16 : Sécurisation de la configuration des Stratix-II d'Altera [Altera 2006]

Les composants Actel possèdent également la possibilité d'utiliser des configurations chiffrées. Comme pour les composants d'Altera, les composants Actel utilisent un module de déchiffrement AES avec une clef de 128 bits laquelle est directement stockée dans la mémoire Flash de configuration (Figure 1-17).

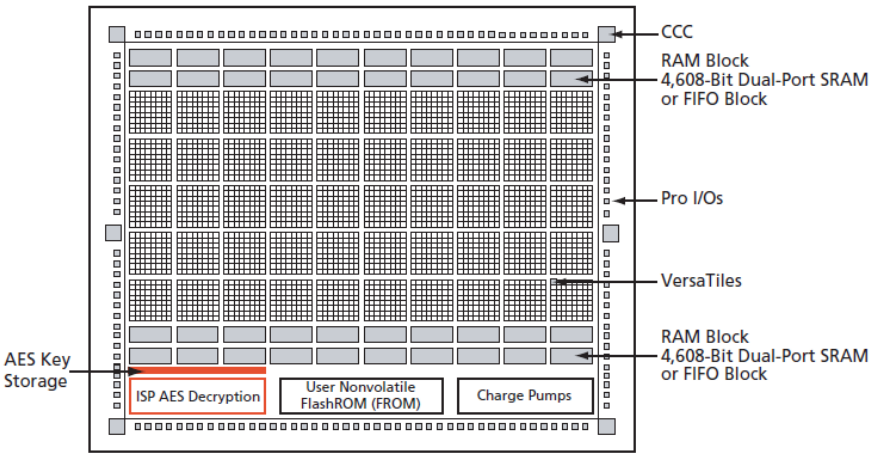


Figure 1-17 : Stockage de la clef de chiffrement pour les composants Actel [Actel 2009]

1.3.3. Le contrôle de redondance cyclique

Lors de la transmission de la configuration entre le module externe de stockage et le FPGA, il est possible que certains bits soient modifiés. La majorité des FPGA utilise donc le contrôle de redondance cyclique (CRC). Le CRC est une somme de vérification utilisée pour détecter des erreurs lors de la transmission ou de la réception de données. Lors du chargement de la configuration, les FPGA peuvent vérifier directement l'intégrité de la configuration. Les CRC sont calculés et comparés à des valeurs stockées dans le fichier de configuration envoyé. Si les deux valeurs calculées et stockées correspondent alors la configuration peut se poursuivre et dans le cas contraire, il faut resynchroniser et reconfigurer le circuit. Les vérifications par CRC peuvent également être désactivées par les outils de compilation, mais il existe un risque de charger une configuration incorrecte pouvant causer un mauvais fonctionnement du circuit implanté ou d'endommager la puce ([Xilinx 2009-2]).

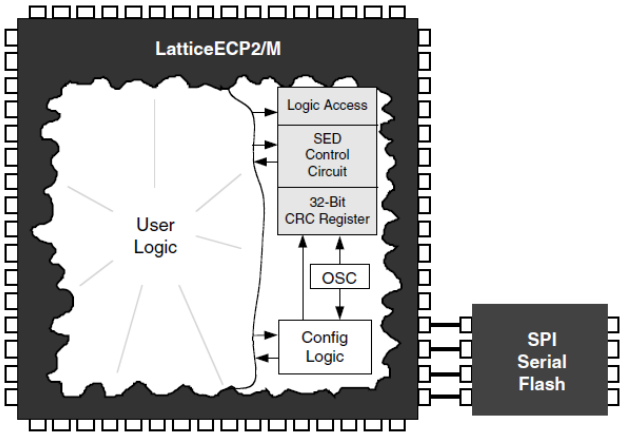


Figure 1-18 : Système de détection d'erreurs [Lattice 2009]

Certains fabricants proposent également de vérifier l'intégrité de la mémoire en réalisant un CRC sur cette dernière. Cette vérification se nomme *"Configuration RAM : CRAM"* chez Altera et peut être rendue

automatique ([Altera 2008-1]) et *"Soft Error Detection : SED"* chez Lattice ([Lattice 2009]) et ne modifie en rien les performances de la logique utilisateur (Figure 1-18).

1.3.4. La reconfiguration partielle

D'autres systèmes de protection sont également présents dans les FPGA et concerne leur reprogrammation. Les fabricants Altera et Lattice permettent de stocker plusieurs configurations et en cas d'erreur de charger une configuration correcte.

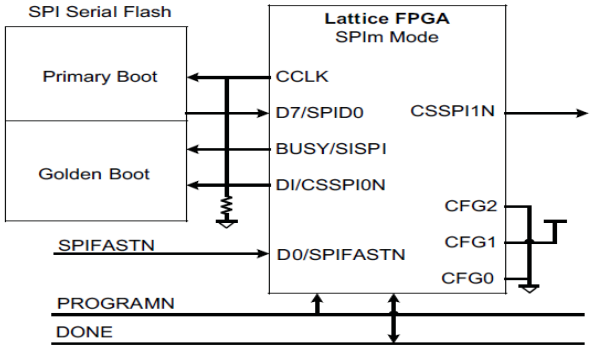


Figure 1-19 : Configuration à doubles images [Lattice 2006-2]

Ainsi, le Stratix d'Altera autorise la configuration dynamique et le stockage jusqu'à un maximum de 8 pages de configuration si l'utilisateur choisit la configuration dynamique ([Altera 2008-2]). Les composants Lattice permettent uniquement de stocker une double configuration comprenant une image principale et une image de sauvegarde (Figure 1-19). Grâce à l'utilisation du CRC, si une erreur est détectée alors une nouvelle configuration parmi les 8 pages pour Altera ou la configuration de secours est chargée dans le FPGA. Si des erreurs sont également détectées dans l'image de secours alors le composant Lattice arrêtera la configuration.

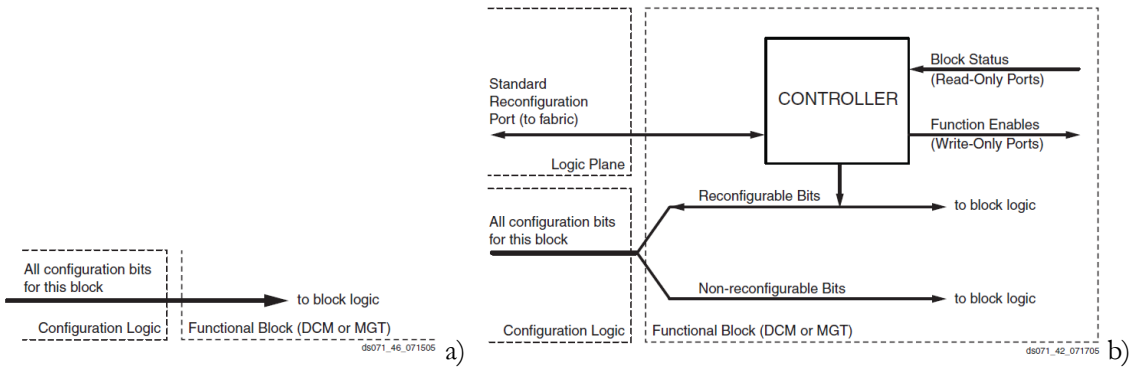


Figure 1-20 : Reconfiguration classique (a) et partielle (b) des composants Xilinx ([Xilinx 2006-1])

Xilinx propose une autre solution en cas de problèmes détectés sur la configuration du composant : la reconfiguration partielle ([Xilinx 2006-1]). Grâce à cette technique, il est possible de reconfigurer une seule partie du composant sans avoir besoin de refaire une configuration complète et d'arrêter le fonctionnement du circuit (Figure 1-20).

1.3.5. Les capteurs intégrés

Pour se prémunir des attaques par changement de l'environnement telles que les variations de la température ou des niveaux de tensions, les fabricants peuvent implanter différents capteurs. Ces capteurs ne sont pas à l'origine implantés pour de la sécurité mais pour être utilisés dans des applications.

Xilinx a implanté une diode sensible aux variations de température. En ajoutant une sonde de température sur laquelle l'utilisateur définit la gamme de température de fonctionnement et si la température de fonctionnement dépasse les gammes alors une interruption peut être générée et une action prise en conséquence. De plus, certaines puces de Xilinx telles que la famille Virtex-5 peuvent vérifier la température du silicium ([Xilinx 2009-3]). Il en est de même pour les tensions d'alimentation.

Par ailleurs, certains composants Actel sont principalement conçus pour des applications militaires et spatiales et ont été durcis en conséquence pour être les moins sensibles aux particules ionisantes.

Pour éviter le clonage des composants de la famille Spartan-3A, Xilinx ajoute lors du processus de fabrication un mot d'identification unique de 57 bits qui peut être assimilé à de l'ADN. Chaque FPGA a un numéro d'identification unique permettant d'associer un circuit à une puce spécifique. Un algorithme de sécurité est utilisé pour vérifier que les numéros d'identification de la puce et de la configuration sont bons. Le principe de fonctionnement est très similaire à une transaction bancaire ([Xilinx 2008]). Pour obtenir de l'argent à un distributeur de monnaie, on doit insérer une carte bancaire et entrer un code PIN. Si la carte et le code PIN sont bien associés à un compte alors l'argent pourra être disponible (Figure 1-21). En ce qui concerne le FPGA, le numéro d'identification et l'algorithme de sécurité génèrent une valeur qui sera comparée avec une autre valeur stockée soit dans la mémoire de configuration soit dans une mémoire externe, permettant ainsi la programmation du FPGA avec la configuration testée.

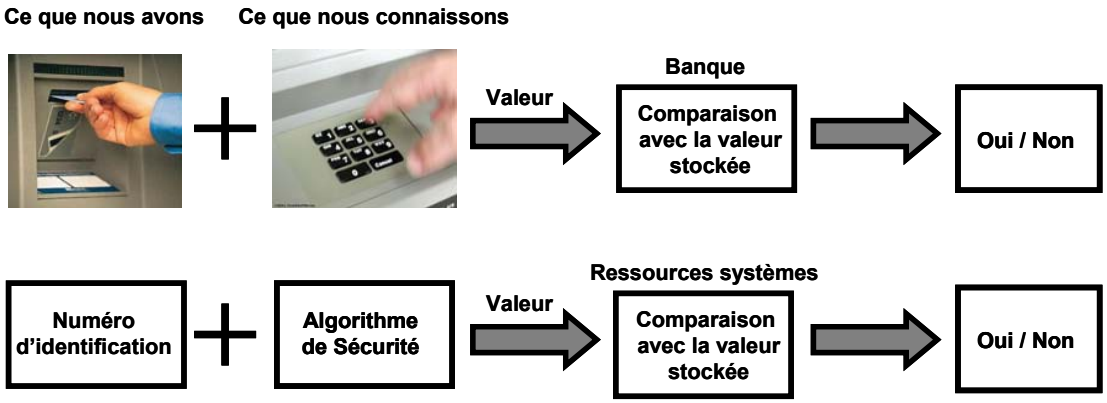


Figure 1-21 : Flux de sécurité

1.4. Conclusion

Aujourd'hui les réseaux programmables sont de plus en plus présents dans nos applications. Avec les évolutions des IPs, on tend à tout intégrer sur une et une seule puce : CNA, CAN, co-processeurs, microprocesseurs, etc.…. Le choix entre les différentes plates-formes n'est pas quelque chose de simple. Plusieurs questions sont à se poser lors de la définition du cahier des charges. Quel type de réseaux programmables employer ? Quelle technologie de configuration utiliser pour mon application entre les anti-fusibles, les SRAMs et les Flash ? Bientôt on aura également le choix avec les nouvelles technologies émergentes basées sur la nanotechnologie. Quelle que soit la technologie utilisée, exceptés les FPGA anti-fusibles, la structure interne des circuits est similaire avec les deux principaux éléments : la logique configurable et les interconnexions. Quelle technologie offre un niveau de robustesse asez élevé pour des applications avioniques, spatiales ou sécuritaires…? On sait que les réseaux à mémoire de configuration SRAM sont nettement plus sensibles aux perturbations que les technologies à anti-fusibles ou Flash, mais offrent des performances très élevées et des coûts plus faibles. Si notre choix se porte sur la technologie SRAM et que des niveaux de sécurité et/ou de sûreté sont nécessaires, est-ce que les protections mentionnées sont suffisantes face à toutes les catégories d'attaques ?

Dans la suite du document et dans le contexte d'utilisation de FPGA dans des systèmes sécuritaires, nous allons présenter un état de l'art des attaques de composants électroniques. Nous présenterons les trois grands types d'attaques : invasives, semi-invasives et non-invasives, et pour contrecarrer ces attaques quelques exemples de protections à implanter dans le circuit seront également introduits.

Chapitre 2. État de l'art des attaques

Ce chapitre a pour objectif de présenter les différentes méthodes d'attaque pour récupérer des informations confidentielles présentes dans des circuits. Ces attaques peuvent être classées en 3 catégories : les attaques invasives, non-invasives et semi-invasives. Nous aborderons également quelques contremesures pour sécuriser les circuits, hors des solutions proposées par les fabricants de FPGA, et enfin les différents types d'erreurs générées par les attaques seront introduits.

2.1. Les attaques invasives

Les attaques invasives sont directement réalisées sur le silicium du composant et peuvent être destructives. Dans un premier temps, il faut ôter le boîtier en le chauffant par exemple ou en utilisant des procédés chimiques (Figure 2-1-a). En ayant des accès physiques à la puce, il est possible de reconstruire le layout de la puce en effectuant de l'ingénierie inverse ou de réaliser des analyses sous pointes pour extraire des informations confidentielles (Figure 2-1-b).

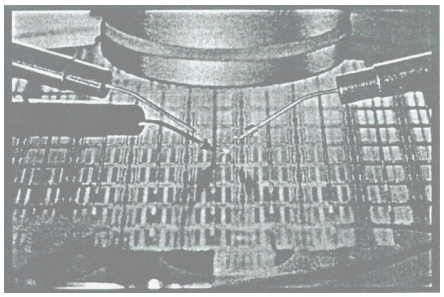

a)

b)

Figure 2-1 : Préparation chimique pour réaliser de l'ingénierie inverse (a) et des analyses sous pointes (b)

2.1.1. L'ingénierie inverse

L'ingénierie inverse a pour but de comprendre la structure du circuit en identifiant les éléments d'une puce en reconstruisant son layout. Pour reconstruire la structure de la puce, il faut réaliser en sens inverse les différentes étapes des procédés de fabrication des circuits intégrés ([Anderson 1996]). Ainsi, les différentes couches de résines, de métaux… sont enlevées les unes après les autres en utilisant des solutions chimiques et/ou mécaniques. A partir des images de chaque couche, il est possible de reconstruire le

circuit complet avec les différentes connexions des différents niveaux (Figure 2-2). Cette attaque peut être utilisée pour lire le contenu d'une mémoire comme présentée dans [Samyde 2002], mais également pour avoir une connaissance plus détaillée des mécanismes de sécurité implantés, pour orienter des attaques non-invasives ou semi-invasives.

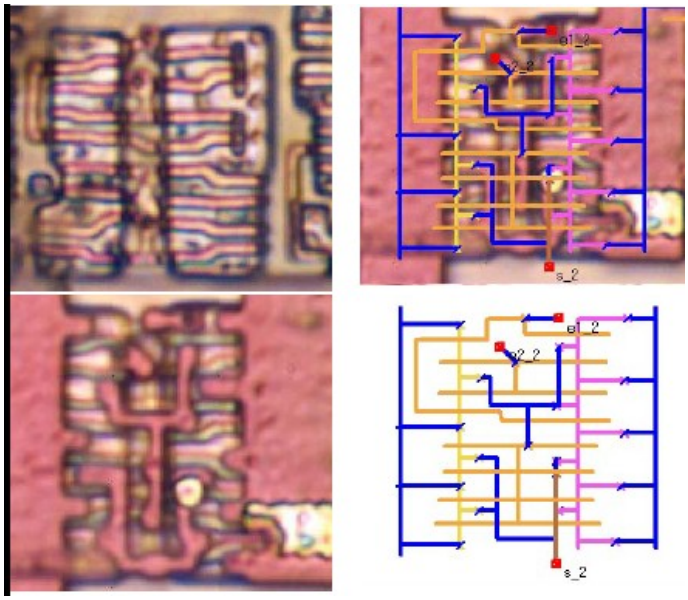


Figure 2-2 : Images d'ingénierie inverse d'un composant [Merle 2006]

Il est également possible d'utiliser un microscope à balayage électronique (Microscope Electronic Beam : MEB) pour connaître les valeurs d'un élément en analysant des photographies d'activité de cet élément à des instants différents. Par exemple, la Figure 2-3 représente l'activité d'un bus à différents instants de fonctionnement et les zones les plus foncées représentent des bits à '1'. Il est possible de connaître toutes les instructions passant sur des bus grâce à des techniques de reconnaissance d'images.

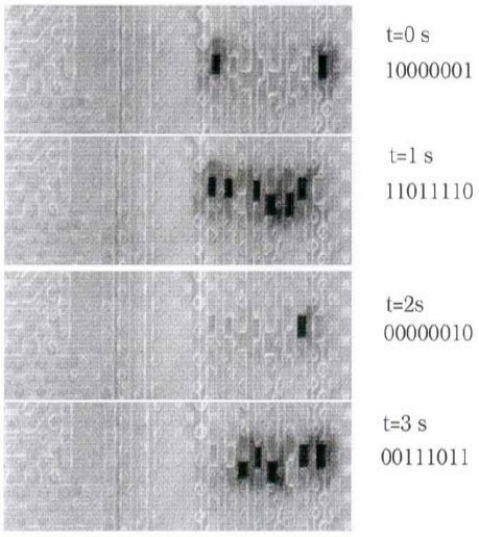


Figure 2-3 : Image sous faisceau d'électrons en contraste de potentiel des états électriques des lignes du bus de données en fonction du temps [Merle 2006]

2.1.2. Les analyses sous pointes

Cette attaque communément appelée *"probing"* utilise des sondes ou des pointes initialement prévues pour le test de wafer ([More 2000]), et permet de lire des valeurs transitant sur des bus (données ou adresses), des entrées ou des sorties de cellules ou blocs etc.… En utilisant les pointes (Figure 2-4), il est possible de forcer certains nœuds de la puce à des valeurs désirées pour rétablir des fusibles de programmation ou des bits interdisant une relecture ([Kömmerling 1999], [Skorobogatov 2005]).

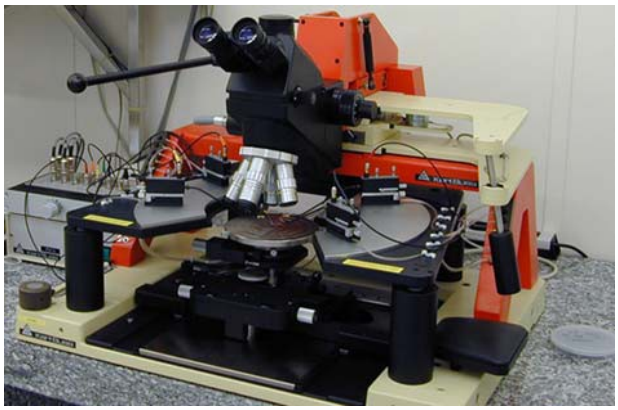
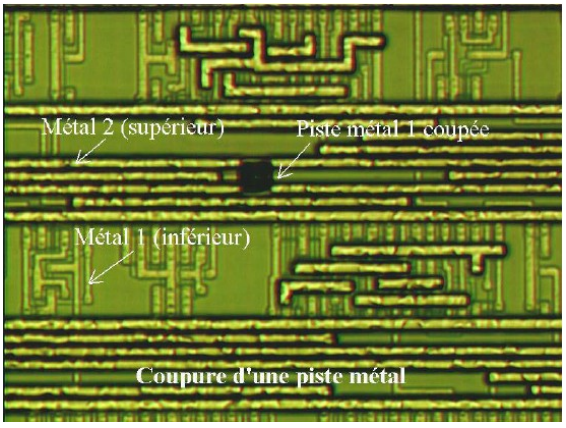
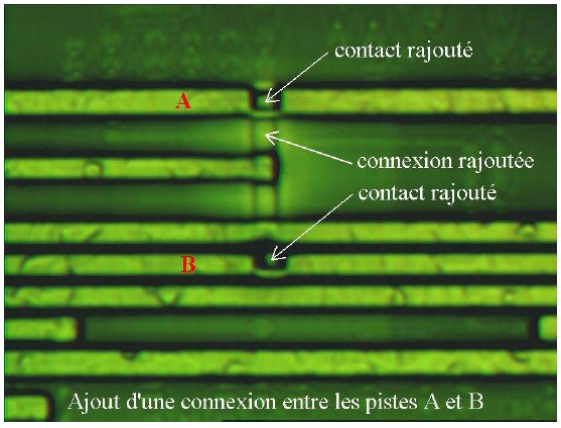


Figure 2-4 : Station de mesure et d'attaque sous pointes

En complément et pour des circuits de très petite taille (technologie récente), il est possible d'utiliser un appareil très sophistiqué à savoir le microscope à faisceau d'ions focalisé (*Focused Ion Beam* : FIB) pour remonter à des niveaux accessibles certains signaux. Le FIB peut créer des incisions au niveau nanométrique et déposer des contacts métalliques pour remonter à la couche la plus haute certains signaux internes, mais également pour dérouter certains de ces signaux : ajout ou suppression de connexions (Figure 2-5). Dans [Skorobogatov 2005] est décrite en détail une procédure à suivre pour conduire ce type d'attaque.



a)



b)

Figure 2-5 : Modifications de circuits par FIB : coupure (a) et ajout (b) d'une connexion [Merle 2006]

2.2. Les attaques non-invasives

2.2.1. Les attaques en boîte noire

Une méthode classique pour connaître le contenu d'un "petit" circuit consiste à tester toutes les combinaisons possibles d'entrées, tout en notant les résultats de sortie. Grâce à cette méthode, l'attaquant peut déterminer toute la logique interne du circuit en utilisant un tableau de Karnaugh ou des algorithmes de simplification des résultats obtenus. Le coût de ce type d'attaque augmente avec l'usage des machines d'état, de la bidirectionnalité des broches etc.… [Dipert 2000].

2.2.2. Les attaques par clonage de la mémoire

Les FPGA actuels sont des composants génériques, impliquant qu'un fichier de configuration d'un FPGA peut être utilisé dans un autre composant de la même famille et de même taille. Les FPGA de type SRAM se configurent à chaque mise sous tension à partir du contenu d'une mémoire externe non-volatile (EEPROM, Flash). Ainsi, l'attaquant peut cloner le bitstream en interceptant les signaux transitant entre la mémoire de configuration et le composant. Dès qu'il est en possession du bitstream, l'attaquant peut l'utiliser sur un système ou produit équivalent. Cette attaque est assez facile à mettre en place si les protections mentionnées au Chapitre 1 ne sont pas employées puisqu'elle ne requiert qu'un analyseur logique et certaines compétences en électronique.

2.2.3. Les attaques en relecture et/ou ingénierie inverse du bitstream

L'ingénierie inverse du bitstream est une transformation permettant à partir d'un fichier de configuration de remonter à une description fonctionnelle du circuit implanté dans le composant. La relecture consiste à lire le contenu de la configuration ainsi que le contenu des bascules utilisateurs du FPGA. Certains éléments du composant peuvent être relus directement sans avoir besoin de remonter à une description fonctionnelle, tels que les mémoires RAM, le contenu des LUTs ou des bascules utilisateurs. Pour effectuer cette attaque, un attaquant utilise un des modes de programmation et de relecture du circuit tels que le bus JTAG ou un autre mode de programmation, mais également le clonage de la mémoire présenté dans la section précédente. Une fois que l'attaquant a récupéré la configuration, il doit essayer de retrouver le circuit implanté. La révélation complète de la configuration d'un circuit est quelque chose de très cher en termes de ressources financières et humaines du fait de la présence de certaines protections intrinsèques proposées par les fondeurs, lesquelles ont été développées dans le Chapitre 1.

Au début des années 1990, la société *NeoCAD* a développé un outil permettant d'implanter des circuits dans des FPGA de différents fabricants sans avoir besoin d'utiliser les outils fournis par ces fabricants. Contrairement à ce qui avait été annoncé à cette époque et selon Xilinx ([Lesea 2005], [Trimberger 2007]), *NeoCAD* n'a pas réalisé de la rétro-conception de la configuration du FPGA, mais un générateur de fichier de configuration compatible avec les FPGA de Xilinx. A la fin des années 1990, la société *Clear Logic* a été capable d'utiliser les outils de génération de fichier de configuration du fondeur Altera afin de produire des

ASICs compatibles broche à broche, plus petits, moins chers et plus sécurisés car ne nécessitant pas de mémoires externes. La société Altera a attaqué en justice *Clear Logic* pour des dommages et intérêts ([Altera 2002], [Altera 2005]). Ces deux sociétés, *NeoCAD* et *Clear Logic*, ont permis de montrer qu'il était possible de générer des configurations compatibles avec d'autres circuits, mais aucune n'a pu obtenir la description fonctionnelle du circuit implanté.

Pour éviter de relire directement le bitstream, les fabricants de FPGA permettent au concepteur de circuit de chiffrer les fichiers de configuration. Dans le Chapitre 1 ont été présentés les différents processeurs de chiffrement ou déchiffrement utilisés, et avec le peu d'informations disponibles, la complexité et la taille du fichier de configuration souvent de plusieurs mégabits rendent très difficile l'ingénierie inverse du fichier de configuration. Actuellement aucun article sur le sujet ne dit si des personnes, des laboratoires ou des sociétés sont arrivés à déchiffrer la configuration complète du composant. Certains projets tels que "ULogic" qui est un graticiel permet à partir d'un fichier de configuration de retrouver une netlist du circuit implanté ([Note 2007], [Ulogic 2007]). Actuellement, seules les configurations des FPGA de Xilinx et plus particulièrement des familles Virtex 2, Virtex 4-VLX, Virtex 5-VLX et Spartan 3 peuvent être analysées par ce graticiel à cause de leur architecture très régulière. La netlist est la liste des connexions entre les divers éléments du circuit et une connaissance de cette dernière peut permettre de reconstruire le schéma logique du circuit implanté.

L'extraction du contenu d'une RAM ou d'une LUT à partir de la configuration est possible à partir des manuels utilisateurs des composants, ou des outils des fabricants. En effet, les outils de synthèse peuvent générer des fichiers indiquant la position des bits configurant le contenu des points mémoires. Ainsi, il suffit de lire l'état de ces bits dans le fichier de relecture de la configuration pour extraire le contenu de l'élément analysé.

2.2.4. Les attaques par observation

Les attaques par observation ou canaux auxiliaires consistent à partir de signaux externes, qui sont une image de l'activité interne du circuit, de déduire certains secrets en exploitant l'implantation du circuit et/ou les différentes opérations effectuées par l'algorithme. Le but des concepteurs contre de telles attaques est de brouiller les signaux pouvant donner des informations tels que la consommation, le rayonnement électromagnétique et les variations de température.

2.2.4.1 Analyse des temps d'exécution

Les attaques temporelles ou analyses des temps d'exécution consistent pour un attaquant à analyser le temps mis pour réaliser des opérations. Chaque opération logique est réalisée en un certain temps, lequel dépend des calculs à effectuer. En analysant précisément les durées de chaque opération, un attaquant peut remonter aux signaux d'entrée grâce à une connaissance de la fonction ou de la structure du circuit implanté.

En 1996, Kocher publia un article théorique ([Kocher 1996]) sur l'utilisation de la mesure du temps, pour essayer de retrouver des clés cryptographiques. En 1998, Dhem mis en œuvre cette analyse ([Dhem 1998]) ce qui permit d'attaquer l'algorithme "*Square* & *Multiply*" implanté sur un microcontrôleur. L'algorithme DES a également été attaqué par cette méthode et [Hevia 1999] montre la possibilité de retrouver le poids de Hamming de la clef de chiffrement. [Schindler 2000] présente une attaque en temps d'exécution sur une implantation d'une exponentiation d'un RSA utilisant le Théorème des Restes Chinois (*Chinese Remainder Theorem : CRT*).

2.2.4.2 Analyse de la consommation

La consommation des composants électroniques peut être divisée en 2 types de consommations : la consommation dynamique et la consommation statique. La consommation dynamique est due aux changements d'état des transistors CMOS chargeant ou déchargeant les capacités de charges ou parasites permettant ainsi les transitions de '0' à '1' ou de '1' à '0'. Un modèle simple de la consommation dynamique d'un transistor CMOS est : $\mathbf{P = C_{charge} \cdot V_{alim}^2 \cdot f \cdot A}$ avec $\mathbf{C_{charge}}$ la capacité de charge incluant toutes les capacités parasites (fils, sortie, parasite….), $\mathbf{V_{alim}}$ la tension d'alimentation du circuit, **f** la fréquence de fonctionnement et **A** la probabilité de transition de '0' à '1' ou de '1' à '0'. Ce modèle est confirmé par [Standaert 2004] pour des FPGA. Pour des FPGA fabriqués en technologie 150nm, selon [Shang 2002] la consommation dynamique est due pour 60% aux capacités des connexions, mais également à 14% à la logique et à 16% aux arbres d'horloges. La consommation statique est la consommation des transistors du circuit lorsque ces derniers ne changent pas d'état. La consommation statique représente entre 5% et 20% de la consommation totale selon [Shang 2002], et est principalement due aux courants de fuite. Ces courants de fuites ont grandement augmenté avec les nouvelles technologies de fabrication (90nm et 65nm). Une diminution de 15% de Vth à chaque nouvelle technologie aura pour effet d'augmenter d'un facteur 5 les courants de fuites, et les paramètres technologiques conduisant à ces augmentations sont présentés dans [Anis 2005].

Pour obtenir des courbes de consommation, la méthode la plus connue dans la littérature est de mesurer la consommation à travers une résistance de faible valeur placée entre la masse du circuit étudié et la masse du reste de la carte. L'analyse des courbes de consommation de circuits intégrés permet d'obtenir des informations sur les opérations en cours, et peut conduire à retrouver des clefs de chiffrement lors d'opérations de chiffrement par exemple. En 1999, Kocher présente 2 types d'analyse de la consommation ([Kocher 1999]) : une analyse simple (en anglais *Simple Power Analysis :* SPA) et une analyse statistique (en anglais *Differential Power Analysis :* DPA).

La SPA permet à un attaquant de rechercher directement sur les courbes de consommation des motifs reproductibles pouvant être corrélés avec les opérations de chiffrement (Figure 2-6) telles que des branchements conditionnels, des multiplications, des exponentiations, etc.…. Sur la Figure 2-6, on peut observer les 16 rondes utilisées lors du chiffrement d'une donnée par un crypto-processeur DES.

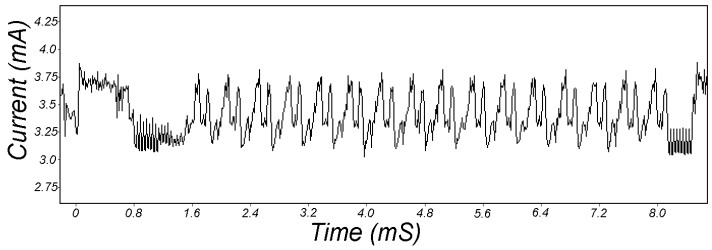


Figure 2-6 : Courbe de consommation d'un chiffrement DES ([Kocher 1999])

La DPA est un peu plus complexe puisqu'il faut utiliser un modèle statistique de consommation et réaliser un traitement avec des courbes acquises dans plusieurs conditions. Le modèle décrit la consommation instantanée du composant lorsqu'il exécute un algorithme de cryptographie ([Kocher 1999], [Örs 2003]). A partir d'un programme informatique exécutant l'algorithme du modèle, il est possible de calculer la consommation pour toutes les clés possibles. En corrélant la consommation du modèle avec celui de la cible, il est possible de déterminer si la clé supposée est correcte (Figure 2-7). En effet, si aucun pic n'est présent sur le signal corrélé alors la clé est incorrecte, et dans le cas d'une clé correcte il y aura la présence d'un pic.

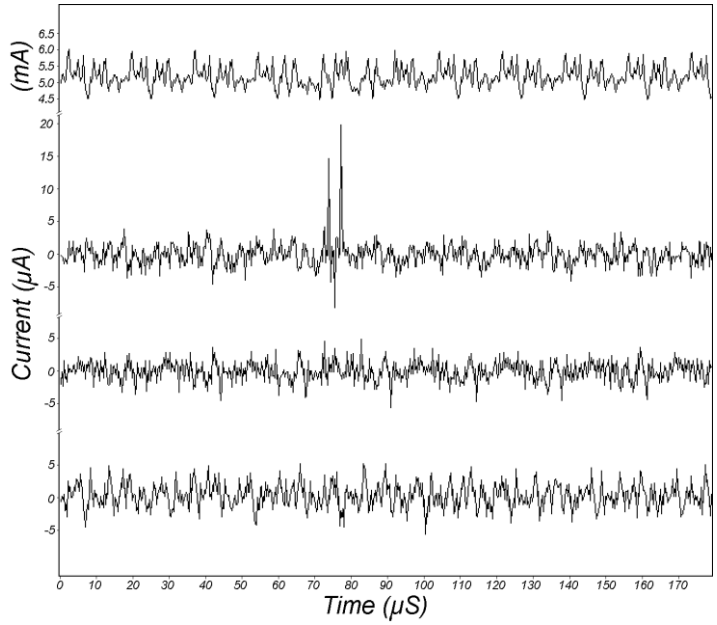
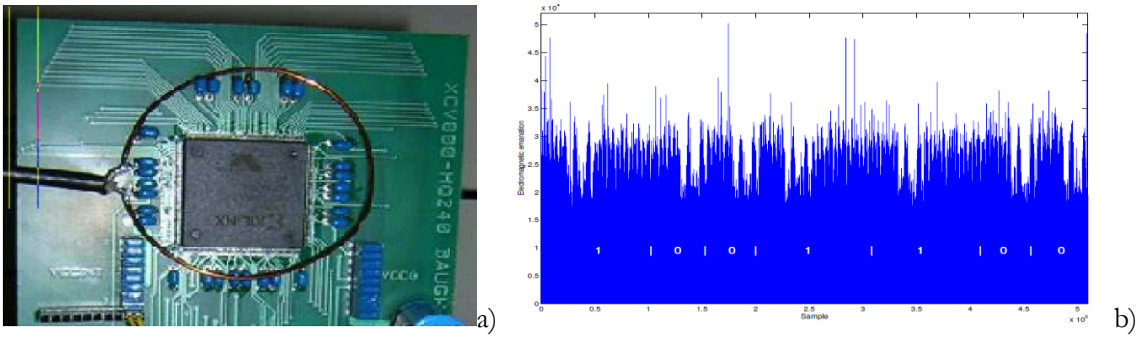


Figure 2-7 : Courbes DPA d'un chiffrement DES obtenues avec 1000 échantillons avec la courbe de consommation de référence, une clef correcte et 2 clefs incorrectes ([Kocher 1999])

Il existe également de la DPA d'ordre n (*High Order Differential Power Analysis : HO-DPA*) également basée sur une étude statistique de la consommation en courant du circuit. Cette HO-DPA utilise des corrélations entre la consommation et n variables intermédiaires dépendant uniquement d'un fragment de la clef et du message d'entrée. Une réalisation de cette attaque peut être trouvée dans [Messerges 2000]. D'autres types d'attaques basées sur la consommation ont été développés: CPA ([Brier 2004]), attaques par gabarit ([Chari 2002]), ….

2.2.4.3 Analyse des fuites électromagnétiques

Comme pour les attques en consommation, il est possible d'obtenir des secrets à partir d'une lecture directe des fuites électromagnétiques et cette attaque s'appelle en anglais "*Simple ElectroMagnetic Attack* (SEMA)". Ainsi les SEMA exploitent la relation entre la radiation électromagnétique et les instructions exécutées. Chaque instruction a sa propre signature en radiation électromagnétique et les mesures sont effectuées à partir d'antennes ou de sondes électromagnétiques (Figure 2-8-a). En effet, le champ entourant la puce est principalement un champ magnétique, alors une antenne ou une sonde mesure les variations de ce champ et à partir de ces mesures il est possible de déterminer la clef. La Figure 2-8-b montre une mesure électromagnétique d'une clef et cette dernière vaut en base binaire 0b11001100.



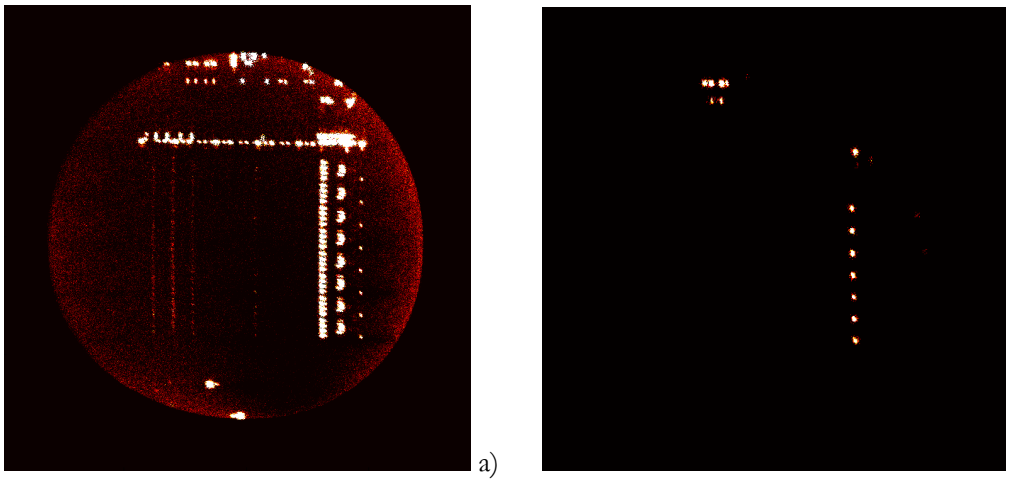
a) b)

Figure 2-8 : Mesure du rayonnement électromagnétique d'un FPGA Xilinx (a) et exemple de courbe de rayonnement électromagnétique obtenue par la mesure (b) [Mulder 2005]

Il existe également des attaques différentielles sur les signaux électromagnétiques (DEMA), et le principe de ces attaques est similaire à la DPA.

2.2.4.4 Analyse des rayonnements thermiques et/ou acoustiques

Ferrigno présente dans [Ferrigno 2008] les premiers résultats d'attaques par observation du rayonnement thermique d'un microcontrôleur PIC16F84A. L'idée de base est que lors de modifications d'état, de la lumière est émise sous forme de photons et en utilisant un appareil capable de détecter ces photons, il est



a) b)

Figure 2-9 : Ensemble des photons observés sur une seule image (a) et lors d'un coup d'horloge (b) [Ferrigno 2008]

possible de récupérer ces modifications (Figure 2-9). Dans cet article, il est montré la faisabilité de retrouver directement la clef de chiffrement de l'AES lors de l'opération d'ajout de la sous-clé.

Shamir à également montré l'efficacité en pratique de la cryptanalyse acoustique sur le bruit d'un processeur ([Shamir 2004]), mais cette technique n'est pas simple à mettre en œuvre.

2.2.5. Les attaques par changement de l'environnement

Les composants électroniques sont tous conçus pour fonctionner dans des environnements définis par les fabricants : gamme de tensions, gamme de température, etc.… Ces gammes de fonctionnement dépendent des applications : commerciales, militaires ou spatiales. Ainsi, si un attaquant met un composant en dehors de ces gammes, il est fort possible que le fonctionnement du circuit ne soit pas celui attendu par le concepteur.

2.2.5.1 Variation de la température de fonctionnement

Lorsqu'un attaquant fait varier la température pour dépasser les limites définies par le fabricant, plusieurs effets peuvent être obtenus selon [Bar-El 2006] : la modification aléatoire des points mémoire SRAM, et l'exploitation du fait que les températures limites d'écritures et de lectures ne coïncident pas. Ces effets sont dus au changement de la vitesse de propagation des signaux

2.2.5.2 Variation de la tension d'alimentation

L'autre paramètre sur lequel un attaquant peut agir de manière non-invasive est la tension d'alimentation du circuit. Comme pour les attaques en température, l'attaquant fait varier l'alimentation pour sortir des limites en appliquant soit une tension hors des limites durant toutes les opérations soit des impulsions de surtensions. Ces attaques par impulsions, communément appelées Glitches, peuvent dans le cas d'un chiffrement d'une donnée réduire le nombre de cycles nécessaire au bon déroulement du calcul en mettant par exemple dans un état faux certaines bascules ([Anderson 1997], [Kömmerling 1999], [Bar-El 2006]).

2.2.5.3 Variation des horloges externes

Il existe aussi des glitches d'horloge créant de nouveaux fronts d'horloges qui seront interprétés par la partie logique comme une nouvelle horloge [Pacalet 2005]. Suite à ces nouveaux fronts, de mauvaises valeurs seront stockées dans les points mémoire car des différences seront observées entre les chemins critiques et la nouvelle horloge. Cette méthode est utilisée pour faire sauter un accès mémoire ou pour faire commencer une instruction suivante avant la fin de la précédente.

2.3. Les attaques semi-invasives

Les attaques semi-invasives nécessitent une préparation des circuits telle que l'ouverture du boitier pour avoir accès à la puce, et également une réduction de l'épaisseur du silicium pour avoir une bonne pénétration des éléments perturbateurs tels que la lumière, les ions lourds ou le laser.

2.3.1. La lumière blanche

Les circuits électroniques sont très sensibles à la lumière blanche à cause des effets photoélectriques. Le courant induit par les photons peut être utilisé pour générer des fautes lorsque le circuit est exposé à une lumière intense durant une durée brève. Cette attaque est assez simple à mettre en œuvre puisqu'elle ne nécessite qu'un flash d'appareil photo par exemple. Cette technique d'attaque par lumière blanche est l'ancêtre des attaques laser et a été présentée dans [Skorobogatov 2002] sur une mémoire SRAM d'un microcontrôleur. Dans [Schmidt 2007], une amélioration est introduite en utilisant une fibre optique afin de guider spatialement le faisceau lumineux sur la puce de silicium (Figure 2-10).

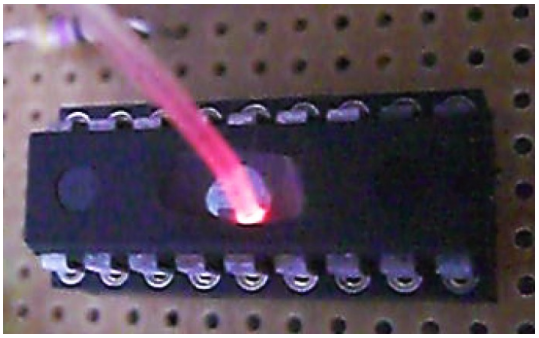


Figure 2-10 : Attaque d'un microcontrôleur en utilisant un guide de lumière à base de fibre optique [Schmidt 2007].

2.3.2. Les particules

Des fautes peuvent également être injectées en utilisant les ions lourds ou d'autres particules ionisantes telles que les neutrons ou les particules alpha. Cette technique est utilisée pour reproduire les effets des particules présentes dans l'espace sur des composants susceptibles de fonctionner dans cet environnement. Cependant, il est difficile d'avoir un bon contrôle spatial et temporel, à cause des contraintes de fabrication des diamètres des canons à particules utilisés et des problèmes de synchronisation. Malgré ces inconvénients, des travaux ont été réalisés par exemple sur des mécanismes de détection d'erreur pour un microcontrôleur MC6809E ([Gunneflo 1989]). Ces injections de fautes sont réalisées en utilisant des accélérateurs de particules tels que ceux de l'*Institut Laue-Langevin (ILL*) de Grenoble, de *Los Alamos Neutron Science Center (LANSCE)* aux États-Unis d'Amérique ou encore de *Paul Scherrer Institute (PSI)* de Villigen en Suisse. Cependant, il est également possible d'utiliser des équipements plus légers basés sur des sources radioactives.

2.3.3. Le laser

En 1965 apparurent les premiers résultats des simulations des effets des radiations ionisantes en utilisant un laser [Habling 1965]. Cependant, l'utilisation du laser pour injecter des fautes dans les circuits électroniques est assez récente. Cet écart dans l'utilisation de cette méthode est principalement dû au temps mis pour valider les premiers résultats sur des circuits électroniques envoyés dans l'espace

[Binder 1975]. Depuis 1975 et la validation des résultats, de très nombreuses recherches ont été conduites dans le domaine des injections de fautes par laser. De nombreux circuits ont été testés par cette méthode tels que des microprocesseurs durcis [Samson 1998], de la mémoire RAM [Duzellier 2000], mais également des circuits analogiques [Moss 2003].

L'utilisation des lasers présente plusieurs avantages par rapport aux particules. Premièrement avec un laser, il est possible d'avoir une bonne maîtrise temporelle et spatiale pour injecter la faute. Mais des contraintes technologiques ne permettent pas d'avoir des faisceaux lumineux très fins, ce qui est un inconvénient avec les avancées technologiques actuelles qui tendent à diminuer la taille des circuits. Dans un second temps, le coût d'utilisation des lasers est nettement plus faible que celui des accélérateurs de particules, ce qui rend les lasers plus accessibles. Les lasers sont donc souvent utilisés lors des certifications de composants tels que les cartes à puce afin de forcer des erreurs dans le circuit.

2.4. Les différents types d'erreurs

Suite aux différentes attaques présentées précédemment, des fautes peuvent apparaître dans le fonctionnement du circuit à cause de conditions anormales : variation de la tension d'alimentation, de l'horloge, de la température de fonctionnement, d'exposition à des particules ionisantes ou à des faisceaux lumineux. Toutes ces conditions peuvent conduire à différents types de fautes dont la classification dépend de leur durée : faute permanente ou faute temporaire.

Une **faute permanente** est une modification permanente et irréversible du circuit. Dans ce cas, la probabilité que le composant ne fonctionne plus est importante et la seule solution est en général le remplacement du circuit. Pour un attaquant, il faut être sûr que la faute provoque bien l'effet voulu, sans quoi il faut pouvoir continuer l'attaque avec un autre circuit. Dans le cadre de cette thèse, nous n'allons pas nous focaliser sur ce type de fautes, mais plutôt sur les fautes transitoires.

Lorsqu'une modification temporaire d'un signal logique apparaît, on parle de **faute transitoire** laquelle disparaît après un certain temps. Ce type de fautes est généralement de courte durée et n'endommage pas le circuit mais elles provoquent un ou plusieurs bits erronés dans les registres internes du circuit. Dans le cas de FPGA de type SRAM, les modifications sont présentes jusqu'à une nouvelle reconfiguration du composant que nous appellerons des **fautes rémanentes**. Les fautes temporaires peuvent être classées en 2 grandes catégories : les fautes simples (Single Event Upsets : SEU) et les fautes multiples (Multiple Cell Upsets : MCU).

Les SEUs sont des modifications d'une seule cellule logique ou point mémoire communément appelées "bit flip" en anglais. Traditionnellement, ces effets sont observés dans le domaine spatial où les ions lourds sont principalement considérés comme la cause de ces SEUs ([Normand 1996]), mais peuvent également être observées lors d'utilisation de laser. Avec les évolutions technologiques qui tendent à diminuer la taille des transistors, les circuits sont nettement plus sensibles même au niveau de la mer ([Ziegler 1981],

[Normand 1996]). Pour expliquer le principe des SEUs, prenons l'exemple suivant. Soit le point mémoire de la Figure 2-11 constitué de 2 inverseurs têtes-bêches avec 2 transistors chacun (Pmos et Nmos). Lorsqu'une particule ou un faisceau laser touche le drain du transistor N2, initialement bloqué, alors le potentiel du nœud B change et se propage dans l'autre cellule. Si la variation de potentiel est suffisante pour modifier l'état du transistor P1 alors le nœud A changera également d'état. Si ces modifications sont maintenues assez longtemps pour que la mémorisation intervienne alors un basculement logique se produira.

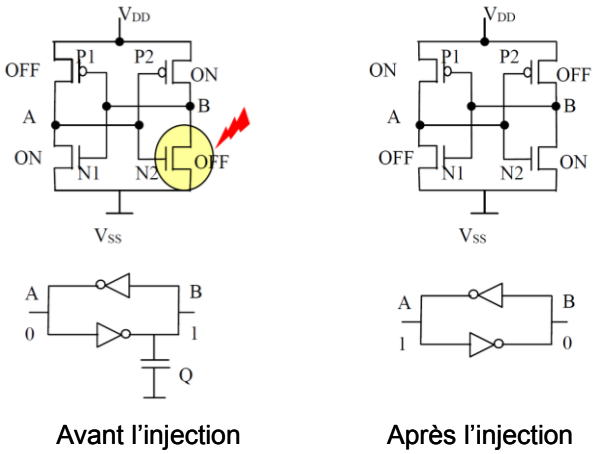


Figure 2-11 : Mécanismes de "bit flips" pour un point mémoire

Les MCUs sont une généralisation des SEUs. Ce type de faute correspond à plusieurs SEUs simultanés. Ceci peut être dû à un effet simultané sur plusieurs cellules mémoires (cas par exemple d'un faisceau laser à faible focus) ou bien à cause de la propagation d'un signal erroné sur plusieurs sorties d'un bloc combinatoire.

2.5. Les protections ou contremesures

Afin de protéger les circuits électroniques contre les attaques invasives, les fabricants de puces électroniques peuvent implanter des éléments spécifiques contre les attaques par analyse sous pointes. Pour les autres types d'attaques (observation ou injection de fautes), il est nécessaire au concepteur du circuit (ASICs ou FPGA) d'implanter des mécanismes de protection. Toutes les protections permettent uniquement de rendre plus difficile les attaques, mais en aucun cas elles ne sécurisent à 100% un circuit. Ces mécanismes de protection sont conçus pour protéger contre un type d'attaque et peuvent rendre le circuit plus vulnérable aux autres attaques. Dans cette partie, nous présenterons quelques protections contre les attaques par sondage, par observation et par perturbation.

2.5.1. Contre les attaques par analyse sous pointes

Précédemment, nous avons introduit les attaques invasives avec les attaques par ingénierie inverse et par analyse sous pointes. Une grande majorité des contremesures contre les attaques invasives sont utilisées

sur les cartes à puces mais peuvent être également utilisées sur d'autres circuits. Pour se prémunir de ce type d'attaque, la solution consiste à détecter l'intrusion pour bloquer dans un premier temps l'activité de la puce et dans un deuxième temps envisager le suicide logique de la puce afin qu'aucune donnée ne soit transmise. La Figure 2-12 montre une grille de protection contre les attaques par sondage, communément appelée *"layer anti-probing"* empêchant l'utilisation des micros-sondes pour récupérer des signaux internes. Il est également possible de tracer des pistes à la place de la grille et de faire passer un signal qui sera comparé à différents endroits du circuit. Si des sondes sont placées sur ces pistes, des retards peuvent être induits ou les signaux peuvent être changés et l'intrusion sera détectée lors des comparaisons. Lorsque des agressions sont détectées sur le circuit, des contremesures logicielles et matérielles peuvent être ajoutées pour "tuer" le composant en effaçant par exemple son contenu ([Kömmerling 1999]).

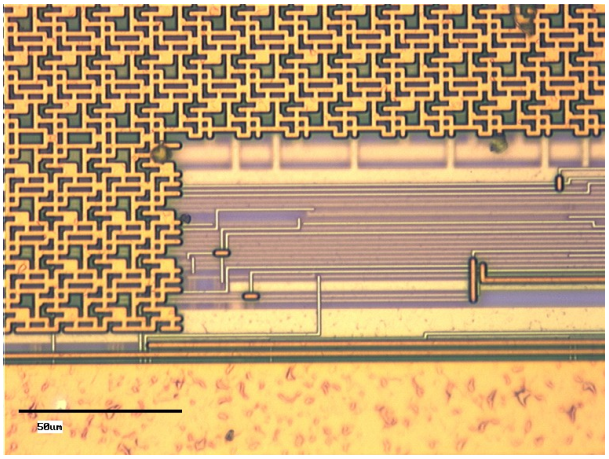


Figure 2-12 : Grille de protection contre les attaques par sondage [Sauveron 2005]

Une autre technique consiste à réaliser un placement routage aléatoire du circuit ([Samyde 2002]). Cette technique, également appelée *"brouillage de conception"*, consiste à répartir aléatoirement les éléments de mémorisation du circuit. Ainsi, l'identification des registres et des bus de données est rendue plus difficile pour des attaques par sondage et d'ingénierie inverse.

2.5.2. Contre les attaques par observation

Les attaques non-invasives consistent essentiellement à déterminer l'activité de la puce pour en déduire des informations secrètes et cela passe par l'analyse de paramètres physiques tels que le temps d'exécution, le courant, le champ magnétique.... Les contre-mesures à ce type d'attaques vont consister donc soit à brouiller l'information qui passe par ces canaux soit à atténuer ou éliminer le signal émis. Pour brouiller les informations, plusieurs solutions techniques sont possibles :

- le chiffrement des signaux transitant sur les mémoires et les bus,
- la génération d'activité aléatoire telle que des calculs supplémentaires.

Lorsque les signaux sont chiffrés à l'intérieur de la puce ceci signifie qu'aucune donnée n'est compréhensible sans les clefs de déchiffrement. De ce fait, tous les résultats obtenus par ces attaques seront nettement plus difficiles à analyser.

Il existe une technique similaire qui consiste à appliquer pseudo-aléatoirement un masque binaire permettant de chiffrer ces mêmes données. Générer de l'activité aléatoire, c'est introduire des opérations leurres et aléatoires que la puce effectuera en parallèle pour induire l'attaquant en erreur. Le circuit effectue des calculs aléatoires en parallèle et cache donc les opérations réellement effectuées par la puce électronique dans le cadre cryptographique.

Enfin certaines techniques consistent à atténuer le signal émis par le circuit. Ces techniques sont principalement matérielles et le concepteur de circuit va essayer d'équilibrer la consommation de la puce, de sorte que les attaques par observation de la consommation ou du rayonnement électromagnétique soient rendues plus difficiles. Le concepteur peut également diminuer la taille des composants et ajouter des couches de métal pour protéger son circuit en utilisant les principes de la compatibilité électromagnétique (CEM) et des blindages.

2.5.3. Contre les attaques par fautes

Pour se prémunir des attaques par injections de fautes, une des contre-mesures les plus efficaces est la redondance qui peut être de trois types : matérielle, temporelle ou d'information. En plus de ces techniques, il est possible de réaliser des calculs de vérification tels que le chiffrement d'une donnée puis du déchiffrement du résultat pour détecter les injections de fautes ([Karri 2002]).

2.5.3.1 Redondance matérielle

Pour réaliser de la redondance matérielle, il suffit de faire la même opération sur plusieurs copies d'un même bloc de calcul et de comparer les résultats obtenus. La Figure 2-13 montre quelques structures de redondance matérielle.

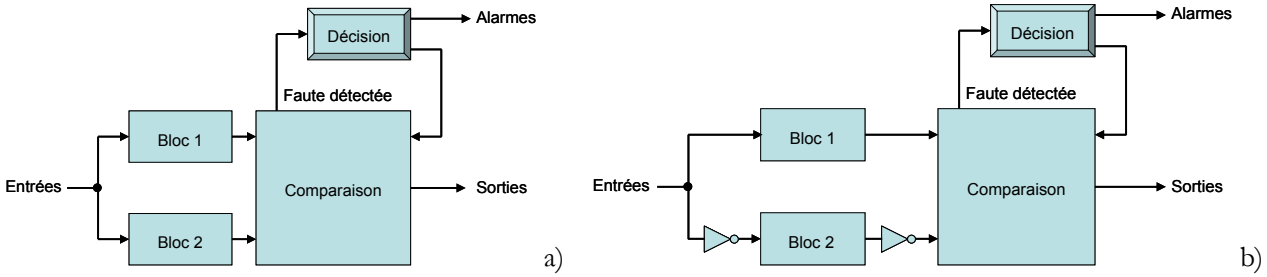


Figure 2-13 : Duplication simple (a) et avec redondance complémentaire (b)

La structure de base en redondance matérielle est la duplication simple avec comparaison (en anglais *Simple Duplication with Comparison :* SDC) représentée en Figure 2-13-a. En comparant les résultats issus des deux blocs et si des différences sont présentes alors un signal d'alerte est envoyé vers un bloc décisionnel. C'est le bloc décisionnel qui permet d'agir sur la suite des opérations telles que le déclenchement d'alarmes ou des remises à zéro. Ce type de redondance permet juste de détecter des erreurs "simples". Une

optimisation de cette redondance, utilisant le même principe que la duplication simple, est la duplication simple avec une redondance complémentaire (Figure 2-13-b). Dans cette structure, les blocs sont dupliqués, les entrées de ces blocs sont complémentaires et que la sortie du bloc 2 est également inversé pour comparer les résultats. Cette technique est efficace pour détecter des erreurs "multiples" puisqu'il est assez difficile d'injecter deux fautes différentes avec des effets complémentaires.

La duplication multiple avec comparaison (Figure 2-14) est une structure où chaque bloc est dupliqué plusieurs fois comme pour la structure à duplication simple. Comme précédemment, le comparateur détecte les différences entre les résultats et déclenche des alarmes. Dans ce cas, les mêmes réactions que celles de la duplication simple sont possibles mais également un vote à la majorité pour assurer un signal de sortie correct. Ainsi, si des erreurs sont présentes, elles seront directement corrigées.

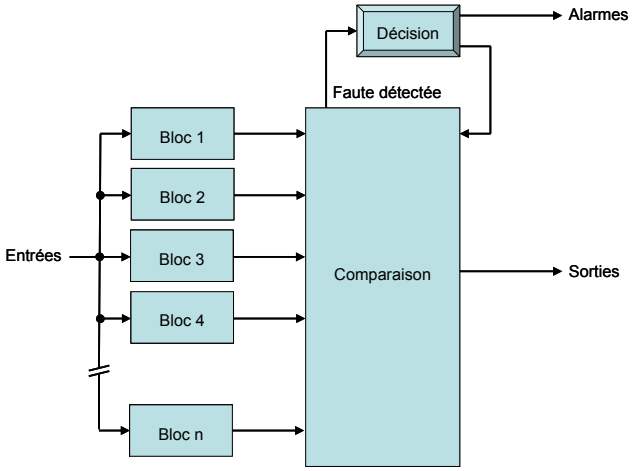


Figure 2-14 : Duplication multiple avec comparaison

La duplication dynamique, présentée en Figure 2-15-a est une optimisation de la duplication multiple avec comparaison. En cas d'erreur détectée, le bloc décisionnel réalise un vote majoritaire et le résultat fauté ne sera plus utilisé jusqu'à une réinitialisation du système. La duplication hybride (Figure 2-15-b) regroupant

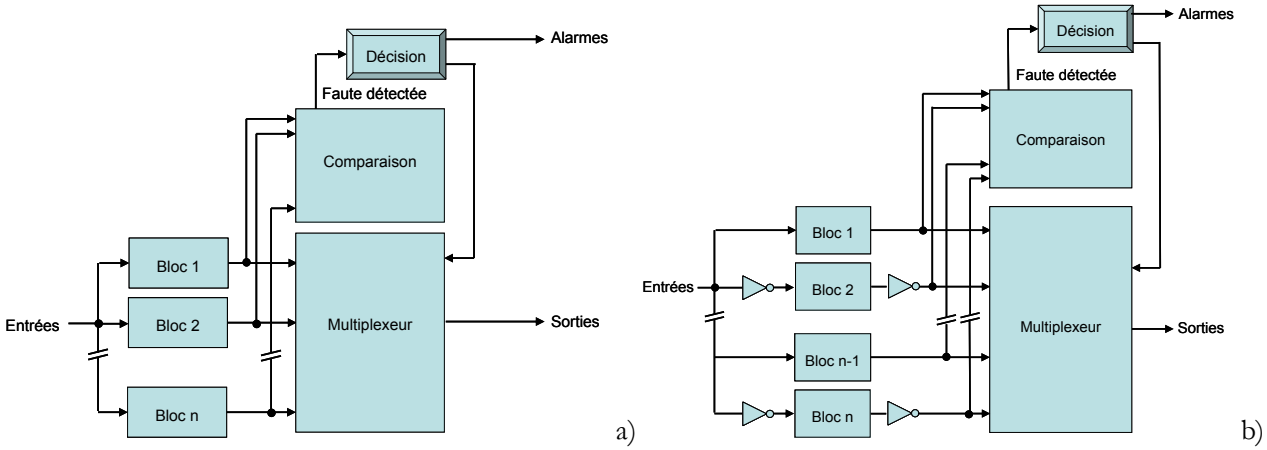


Figure 2-15 : Duplication dynamique (a) et Duplication hybride (b)

toutes les implantations précédentes (multiple, complémentaire et dynamique) est très efficace contre l'injection de fautes "simples" ou "multiples".

Les structures de redondance matérielle sont très efficaces contre les fautes, et certaines de ces structures détectent et corrigent les erreurs. Cependant, elles ont un coût en surface qui n'est pas négligeable, et il faut trouver un compris entre la surface et le niveau de sécurité nécessaire.

2.5.3.2 Redondance temporelle

La redondance temporelle consiste à effectuer le même calcul avec le même bloc matériel mais à des instants différents. Ce type d'architecture est efficace dans le cas de fautes non permanentes. Les figures ci-après montrent quelques structures de redondance temporelle.

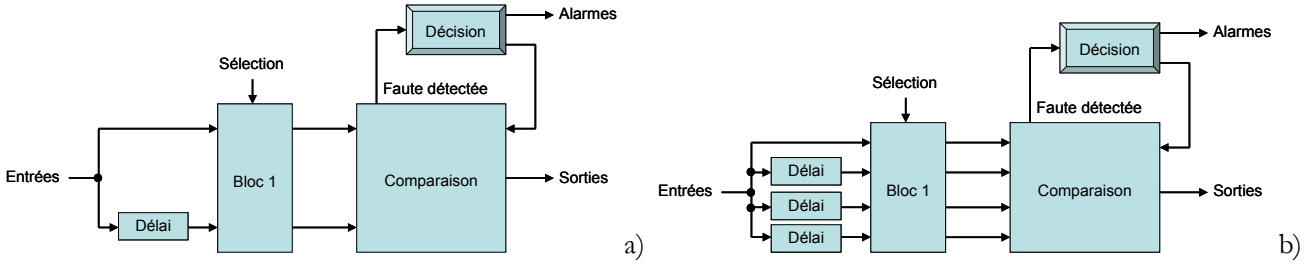


Figure 2-16 : Redondance temporelle simple (a) et multiple (b)

La structure représentée dans la Figure 2-16-a est la structure de base de la redondance temporelle et se nomme redondance simple (en anglais *Simple Time Redundancy with Comparison : STRC*). Cette structure consiste à effectuer deux fois les calculs à des instants différents et à comparer les résultats. Comme pour les architectures à redondance matérielle, si une différence est observée entre les résultats alors une action sera prise par le bloc décisionnel. Cette structure est efficace pour détecter les erreurs non permanentes simples ou multiples.

La structure de redondance temporelle multiple présentée dans la Figure 2-17-b utilise le même principe que la redondance temporelle simple sauf que les calculs sont réalisés plus de 2 fois. Cette structure peut détecter les erreurs mais également faire des corrections en implantant un système de vote majoritaire.

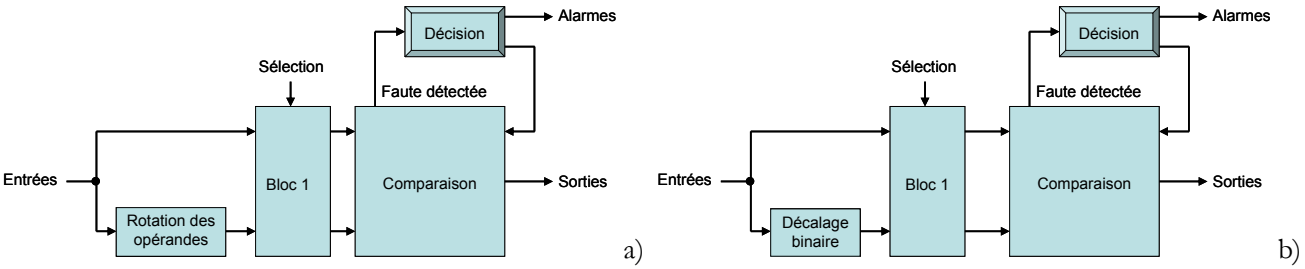


Figure 2-17 : Redondance temporelle simple avec rotation des opérandes (a) et avec décalage binaire (b)

D'autres structures de redondance temporelle consistent à agir sur les opérandes. Par exemple, il est possible d'inverser l'ordre des octets des opérandes ou de décaler les opérandes d'un nombre de bits donnés. La structure réalisant des rotations des opérandes, représentée dans la Figure 2-17-a, est réalisable

uniquement pour des fonctions mathématiques permettant cette inversion, et a l'avantage de réaliser une désynchronisation des deux opérations différentes rendant les attaques par fautes nettement plus difficiles. La structure avec le décalage binaire de la Figure 2-17-b calcule le résultat à partir de l'opérande décalé d'un nombre de bits donnés.

2.5.3.3 Redondance d'information

Pour sécuriser les circuits, il est également possible d'utiliser de la redondance d'information. Cette technique utilise par exemple de la parité, des checksums ou les codes de Hamming….

La redondance d'information par parité est une des techniques les plus utilisées et consiste à ajouter un bit en plus à la donnée. La valeur de ce bit dépend de la donnée mais également du type de parité désirée : paire ou impaire. Par exemple, dans le cas de la parité paire, le bit de parité vaut '0' si la somme des bits à '1' de la donnée est paire afin que la somme des bits de la donnée et de la parité soit paire. En pratique, cette technique est assez simple car uniquement un Ou Exclusif est nécessaire.

Les Check sum consistent à faire la somme de tous les bits d'une donnée et à comparer le résultat obtenu avec une valeur stockée. Dans le cas où les résultats diffèrent alors une faute est présente. Ce type de redondance ne permet pas de corriger les fautes, mais juste de les détecter. Cependant, il est possible de corriger des erreurs si 3 Checksums sont implantés grâce au principe de vote majoritaire.

Les codes de Hamming permettent de corriger une et une seule erreur dans une donnée. Le *code de Hamming (7,4)* transfère 4 bits de données et 3 bits de parité pour un message de 7 bits, et si l'un de ces 7 bits est modifié alors il existe un algorithme permettant de corriger l'erreur. Si on ajoute un bit de parité à la fin des blocs de Hamming, ce code s'appelle *code de Hamming étendu*, et permet toujours de corriger une seule erreur mais est capable d'en détecter deux. Il existe également d'autres codes de Hamming tel que le *code de Hamming cyclique*….. Un exemple d'implantation de code de Hamming peut être trouvé dans [Lima 2000].

2.6. Conclusion

Tous les jours, nous sommes soucieux de notre sécurité et pour assurer cette dernière, un grand nombre d'applications utilisent des systèmes sur puces. Ces systèmes sur puces sont de plus en plus utilisés afin de simplifier la vie des gens tels que les cartes à puces qui contiennent des informations confidentielles.

Pour récupérer les informations confidentielles ou pour détourner le système de son bon fonctionnement, les attaquants ont plusieurs méthodes nécessitant plus ou moins de connaissance du produit, de temps et surtout d'argent. Nous avons présenté les différentes catégories d'attaques possibles ainsi que quelques méthodes de protection. Mais il ne faut pas oublier le fait qu'une protection ne protège pas un système à 100%, elle permet juste de complexifier l'intrusion. Il existe une course entre les attaquants et les fabricants de systèmes. Tous les jours de nouvelles attaques sont mises en place et les fabricants doivent réagir en conséquence en anticipant les protections possibles.

Dans la suite du document, nous allons chercher à évaluer l'effet d'attaques par perturbation sur des FPGA configurables par SRAM. Mais avant cela, nous allons présenter les conditions expérimentales utilisées pour mener à bien nos attaques.

Chapitre 3. Conditions expérimentales

Ce chapitre présente les conditions expérimentales mises en œuvre pour les différentes campagnes réalisées. Pour injecter des fautes dans les composants nous avons utilisé des bancs laser et de surtensions. Des cartes de test dédiées à chaque catégorie de campagne ont été développées. Une préparation du circuit FPGA a été nécessaire afin d'avoir une bonne pénétration du faisceau laser. Les différentes étapes de préparation seront introduites dans ce chapitre. Nous présenterons également l'outil d'analyse utilisé pour caractériser les effets des injections de fautes, ainsi que les différents circuits implantés dans le FPGA.

3.1. Le banc de surtensions

Une autre façon de générer des fautes dans un circuit est d'utiliser un banc de surtensions. Ce banc permet de générer des surtensions sur les alimentations (Vcc ou Gnd) du composant lors du fonctionnement de ce dernier.

Le banc de surtensions est constitué d'un générateur permettant d'obtenir des impulsions allant jusqu'à une tension de 100V sous 2A et d'une carte dédiée sur laquelle les différentes alimentations du composant sont séparées. Les caractéristiques détaillées de cette carte seront présentées dans la section 3.4.2.

Contrairement aux bancs laser, uniquement des campagnes dynamiques ont été réalisées. Le schéma de principe de fonctionnement du banc de surtensions est donné dans la Figure 3-1 et le principe est très similaire aux bancs laser.

Pour injecter des fautes par surtensions, différents paramètres peuvent être modifiés tels que la largeur et l'amplitude de l'impulsion ainsi que le retard par rapport au signal de synchronisation. Ces paramètres sont réglables sur le générateur d'impulsions de manière manuelle ou de manière automatique par le bus GPIB (Figure 3-1). Pour ce type de campagne, nous avons opté pour un réglage automatique des paramètres par le bus GPIB. Ainsi, l'ordinateur règle les différents paramètres du générateur d'impulsions (amplitude, durée, délai) en utilisant le bus GPIB et communique via une liaison RS232 avec le composant testé à travers une carte mère à base de FPGA identique à celle utilisée dans le banc laser. Pour configurer le composant testé, comme pour les bancs laser, nous utilisons la carte DIO générant les signaux JTAG de configuration et de relecture de la configuration.

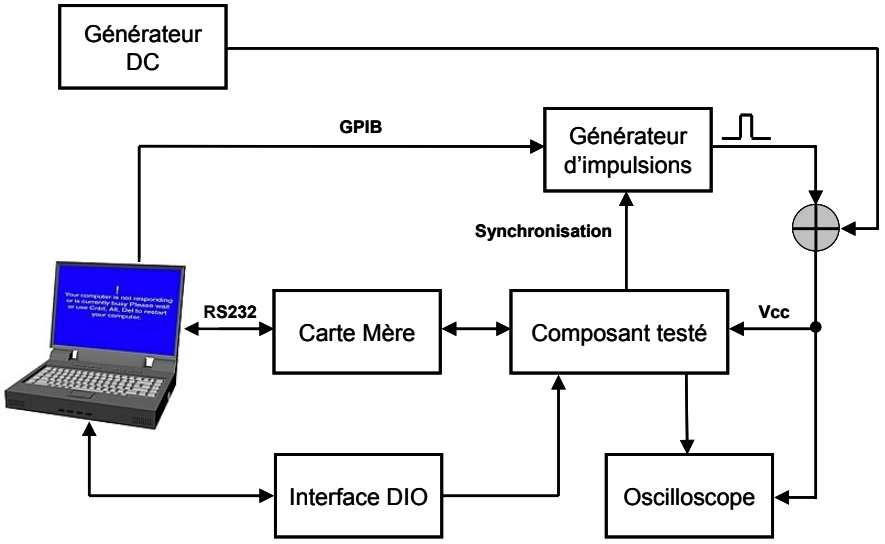
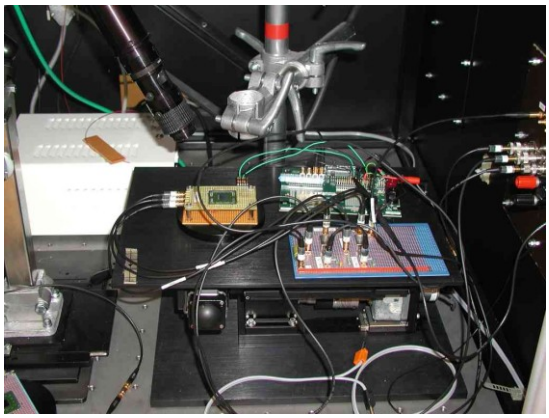
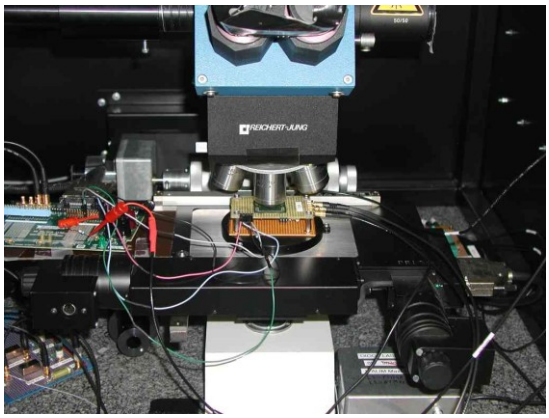


Figure 3-1 : Schéma de principe du fonctionnement du banc de surtensions

Lorsque l'ordre de déclenchement est activé, le générateur génère une impulsion de surtension dont les caractéristiques ont été réglées précédemment. Cette surtension est ajoutée à une tension continue égale à la tension nominale de fonctionnement. Pour réaliser la somme des deux tensions (surtension et tension nominale), nous avons réalisé un simple montage constitué d'une diode et d'un condensateur. La diode est mise en série sur la tension nominale afin de protéger le générateur de tension continue contre la surtension. En série avec la sortie du générateur d'impulsions, nous avons ajouté un condensateur afin de laisser passer uniquement l'impulsion tout en ôtant la composante continue pouvant provenir du générateur d'impulsions. Le choix du condensateur dépend des caractéristiques des impulsions, plus l'impulsion est courte plus la capacité doit avoir une faible valeur.

3.2. Les bancs laser

Pour réaliser les différentes campagnes d'injection de fautes par tir laser, nous avons utilisé les bancs laser du CESTI-LETI du CEA Grenoble (Figure 3-2). Ces bancs laser permettent d'obtenir des tailles de spot laser différentes allant de 4 µm à environ 100 µm, soit à partir d'une focalisation réalisée en interne soit à partir d'objectifs de microscope. Le banc laser représenté dans la Figure 3-2-a, que nous appellerons dans la suite du document *"banc laser fibré"*, permet d'obtenir une taille de spot de l'ordre du diamètre extérieur de la fibre c'est-à-dire de l'ordre de 100 µm de diamètre. Le banc microscope (Figure 3-2-b) permet d'obtenir plusieurs tailles de spot puisque le banc possède plusieurs grossissements. Avec les grossissements de X100, X50, X20 et X10, il est possible d'obtenir respectivement des tailles de spot d'environ 4 µm, 8 µm, 20 µm et 40 µm.


a)


b)

Figure 3-2 : Vues internes des bancs laser utilisés – banc laser fibré (a) et banc microscope (b)

Ces deux bancs laser sont constitués des mêmes équipements tels que des tables XYZ motorisées afin d'automatiser les déplacements lors d'une étude d'une zone du composant ou du composant entier, d'amplificateurs de puissance pour les diodes laser, de caméra, etc.….. Les tables XYZ peuvent être pilotées soit par liaison série RS232 soit par un joystick et la précision de ces dernières est nettement inférieure au micromètre. Les amplificateurs de puissance permettent de fournir la puissance nécessaire au déclenchement du faisceau laser afin d'injecter des fautes. Plusieurs paramètres peuvent avoir une influence sur les effets des tirs laser tels que la durée d'éclairement du composant, mais également la puissance du faisceau laser. Dans le Chapitre 5, une étude montrera l'influence de l'énergie sur le nombre de fautes et plus particulièrement l'influence de la durée d'éclairement. La caméra sert principalement au positionnement du laser sur le composant car les bancs laser sont dans des boîtiers fermés pour des raisons de sécurité.

Par ailleurs, selon l'effet désiré dans les zones actives du composant, il est possible de modifier différents paramètres des lasers tels que la taille du spot (grossissement, fibre, etc…), la longueur d'onde de la diode laser, la focalisation du faisceau dont le réglage n'est pas quelque chose de simple ou la valeur de l'énergie émise. Tous les changements pouvant être fait sur le banc conduisent à de nouvelles conditions expérimentales.

Au sein du laboratoire plusieurs diodes laser sont disponibles, possédant des longueurs d'onde différentes dans l'infrarouge comprises entre 745 nm et 1000 nm. Le choix des longueurs d'ondes dépend de l'environnement que le faisceau laser doit traverser. En effet, pour perturber le fonctionnement d'un composant deux types d'approches sont possibles, à savoir injecter une perturbation par la face avant ou par la face arrière du circuit. Selon la conception du composant les zones actives se trouvent sur la face avant, ou sur la face arrière comme pour les composants Flip-Chip. Pour éclairer les zones actives présentes sur la face avant du composant, la longueur d'onde pourra être plus faible que celle utilisée lorsque les zones actives sont sur la face arrière puisqu'il y aura moins d'éléments à traverser. La Figure 3-3, nous montre que le faisceau ne traverse pas les mêmes éléments lorsque nous injectons par la face avant du composant ou par la face arrière. Dans le cas d'une perturbation par la face avant (Figure 3-3 a),

le faisceau lumineux traverse uniquement l'air ce qui implique que le faisceau n'est pas modifié. Par contre lors d'une perturbation par la face arrière (Figure 3-3 b) le faisceau traverse à la fois l'air et le silicium ce qui modifie la réfraction du laser. Typiquement, lors d'une perturbation d'un circuit par la face avant, il faut utiliser une longueur d'onde faible par exemple 745 nm alors que par la face arrière une longueur d'onde proche de 1000 nm est suffisante.

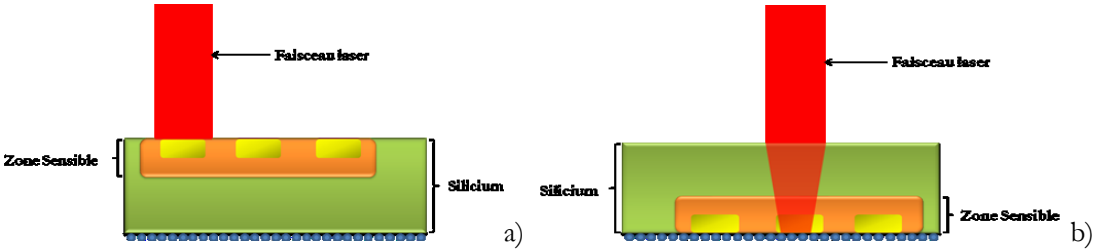


Figure 3-3 : Activation des zones actives par un laser à travers la face avant (a) et la face arrière (b) du composant

L'énergie émise par la diode laser est également à prendre en compte lors des perturbations. Les diodes utilisées au sein du CESTI-LETI possèdent des puissances de plusieurs watts. En agissant sur la tension ou le courant d'alimentation de la diode en fonction des caractéristiques des amplificateurs de puissance, il est possible de faire varier la valeur efficace de la puissance émise par la diode laser. En effet, toutes les diodes suivent une caractéristique linéaire de la puissance en fonction du courant laquelle est donnée par le constructeur de la diode lors de sa phase de caractérisation. Un autre paramètre peut également être pris en compte pour faire varier l'énergie : la durée de l'impulsion (l'énergie est le produit entre la puissance et le temps). Dans la pratique et dans la mesure où cela est possible (à cause des cycles d'horloge), on agira plus facilement sur la durée de l'impulsion et on fixera la puissance de la diode à sa valeur nominale.

Le principe de fonctionnement des bancs laser est le même, que l'on utilise le banc fibré ou le banc microscope (Figure 3-4). Un ordinateur commande par liaison série RS232 la table XYZ et permet de réaliser un balayage spatial de la zone d'étude définie dans le scénario de test avec des pas plus ou moins importants. Cet ordinateur possède une carte DIO (*Digital Input Output*) PCI-Express NI6537 du fabricant National Instrument permettant de commander le composant testé en réalisant l'interfaçage JTAG. Cette carte DIO est une carte numérique avec 32 entrées-sorties configurables séparément et possède une horloge interne de 50 Mhz. Il est également possible de choisir les niveaux de tension de la carte, lesquels sont compatibles 2,5 volts, 3,3 volts ou 5 volts TTL. Cette carte DIO permet de concevoir de façon numérique par exemple un analyseur logique, un générateur de signaux numériques ou un système regroupant l'analyseur et le générateur. Pour nos travaux de recherche, nous utilisons la carte DIO comme un système regroupant à la fois l'analyseur logique et le générateur de signaux. En effet, pour programmer le composant en utilisant le protocole JTAG, il est nécessaire de générer tous les signaux utiles à savoir les signaux TMS, TCK, TDI qui sont des entrées vu du composant et de pouvoir analyser la sortie TDO.

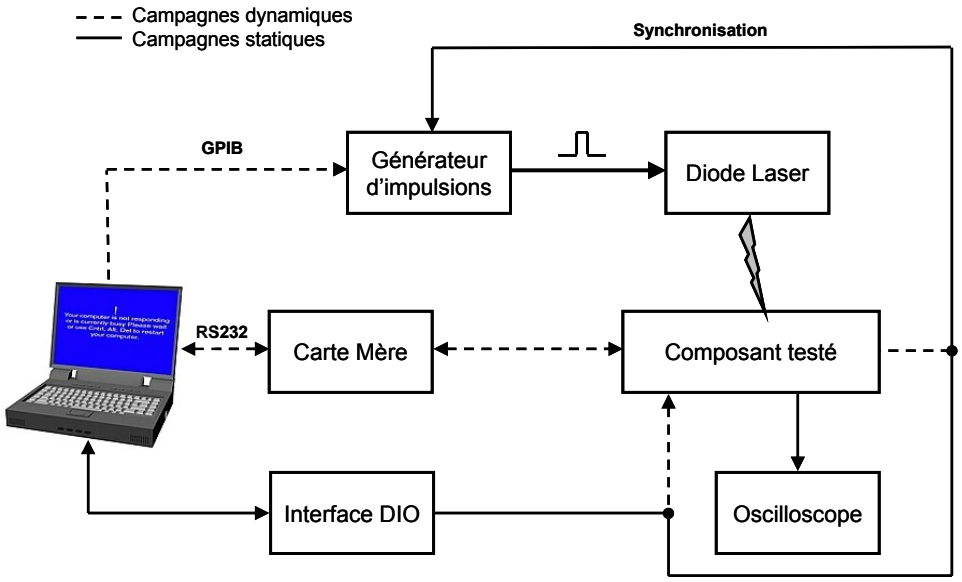


Figure 3-4 : Schéma de principe du fonctionnement du banc laser

Lors des campagnes de caractérisation statique, la carte DIO sera également utilisée pour générer un signal de synchronisation vers un générateur d'impulsion. Ce générateur d'impulsion servira au déclenchement du tir laser et la durée d'éclairement sera directement réglée sur le générateur.

Lors des campagnes de caractérisation dynamique, la carte DIO est uniquement utilisée pour configurer le FPGA. Une carte mère à base de FPGA est ajoutée aux éléments précédemment présentés. Cette carte mère sert d'interface entre l'ordinateur et le composant testé, et génère différents signaux tels qu'une horloge, une remise à zéro et des données utiles pour le bon fonctionnement du circuit. Lors de ces campagnes, la commande de synchronisation vers le générateur d'impulsion est générée par le composant testé. On peut par exemple utiliser le signal de début de chiffrement comme signal de synchronisation. Plusieurs paramètres pourront être réglés directement sur le générateur d'impulsions tels que la durée de l'impulsion, le retard vis-à-vis de la synchronisation, etc.…afin d'injecter la faute à l'instant désiré.

3.3. La préparation des échantillons

Le FPGA étudié est un composant Virtex-II de Xilinx, fabriqué dans une technologie CMOS 0,15 µm avec 8 couches de métal. Il est mis dans un boîtier flip-chip fine-pitch avec 896 broches. Puisque le composant est flip-chip et pour obtenir une bonne pénétration du faisceau lumineux dans les zones actives, une préparation de l'échantillon est nécessaire. Les préparations sont communes lors d'attaques laser sur des composants tels que les cartes à puces ou les FPGA. Deux types de préparations peuvent être effectués sur le composant : une préparation chimique et une préparation mécanique.

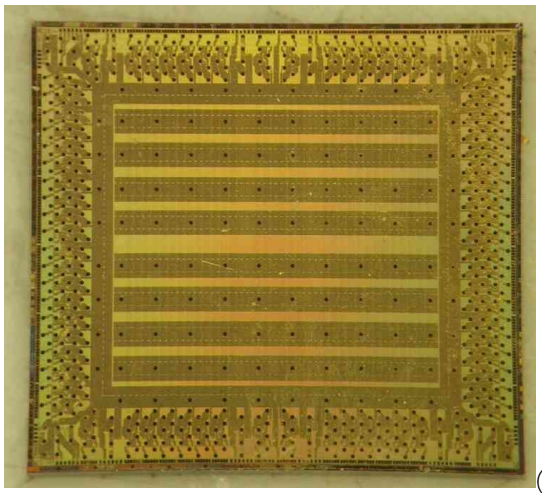
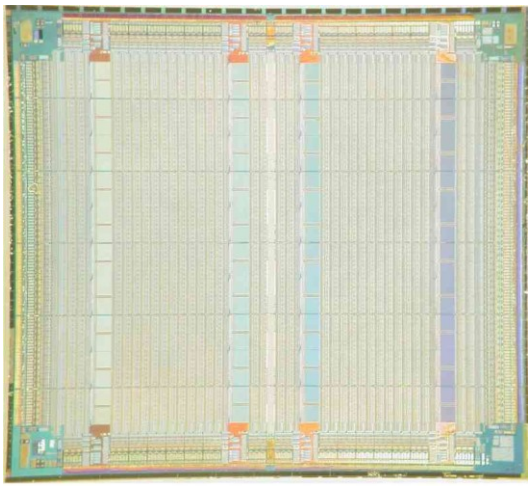

(a)

(b)

Figure 3-5 : Vues du niveau métal avant l'attaque chimique (a) et du niveau poly-silicium après l'attaque chimique (b) du FPGA

La préparation chimique est réalisée par exemple pour obtenir une photographie des différents éléments du circuit et/ou sera utilisée pour le positionnement géographique des différentes campagnes. Ce type de préparations est destructif pour le composant et est réalisé par des personnes spécialisées en chimie. Dans un premier temps, il faut ôter la puce de silicium de son support en époxy en utilisant de l'acide nitrique (Figure 3-5-a). Une fois que la puce est ôtée de son support, il faut dans un deuxième temps enlever les différentes couches d'oxyde. Cette étape est réalisée à partir de fluorure d'hydrogène connu sous le sigle

Figure 3-6 : Préparation chimique de la puce FPGA avec le mélange "eau oxygénée – ammoniaque" à 80°C

"HF". Enfin, la dernière étape a pour but d'enlever les différentes couches de métal à partir d'un mélange *"eau oxygénée – ammoniaque"* à 60°C (Figure 3-6). Dès que toutes ces étapes sont effectuées, nous obtenons la Figure 3-5-b à partir de laquelle nous pouvons identifier les différentes zones ou tuiles telles que les CLBs, les BRAMs, les IOBs, etc. (Figure 3-7)… ce qui permettra de se positionner précisément sur les tuiles désirées.

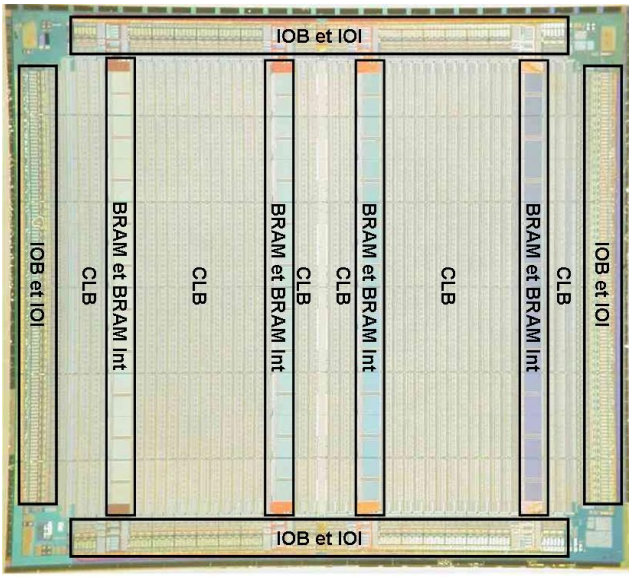
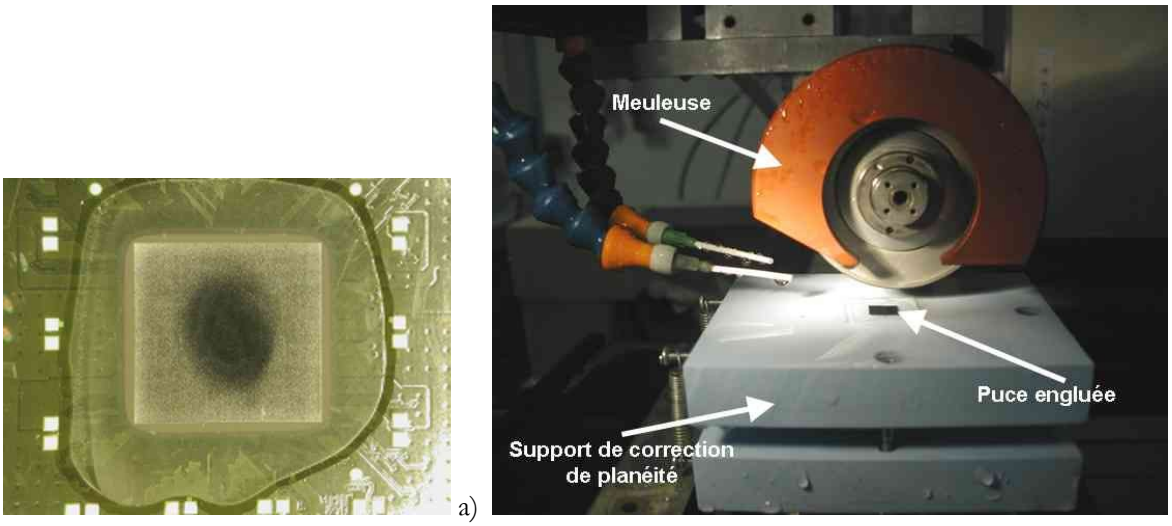


Figure 3-7 : Localisation des différents éléments constituant le FPGA Virtex-II

La préparation mécanique consiste à amincir le silicium afin que le faisceau lumineux puisse toucher les zones actives en perdant le moins d'énergie possible. Cette préparation est très importante pour les attaques. Pour réaliser un amincissement des puces, il faut les conserver sur leur support en époxy. Lorsque les puces sont collées sur leur support en époxy, il arrive que ces dernières ne soient pas planes et qu'il existe une légère inclinaison. En effet, l'inclinaison a peu d'importance dans le fonctionnement classique de la puce, mais peut être quelque chose à prendre en compte lors de la préparation mécanique. Ainsi, un système a été réalisé pour rectifier cette inclinaison et permet une correction inférieure au micromètre (Figure 3-8). Dès que l'inclinaison de la puce est rectifiée, il faut amincir cette dernière en utilisant soit une micro-fraiseuse soit un usinage mécanique. La micro-fraiseuse permet de faire principalement des amincissements localisés alors que la machine réalisant l'usinage mécanique permet d'amincir des surfaces nettement plus importantes de l'ordre de plusieurs centimètres carrés. La puce du FPGA étudié possède une surface de l'ordre du centimètre carré (10,6 mm de largeur et 9,7 mm de longueur) et la solution utilisant l'usinage mécanique a été retenue.



a) b)

Figure 3-8 : Puce du FPGA engluée (a) et en cours d'usinage mécanique (b)

L'amincissement mécanique se fait en répétant plusieurs fois les mêmes opérations communément appelées *"des passes"*. Initialement la puce possède une épaisseur de 790 µm et nous avons choisi de l'amincir à 30 µm car pour cette épaisseur résiduelle, nous avons une très bonne pénétration du faisceau laser.

La première étape de l'amincissement est d'engluer la puce dans de la résine pour éviter d'abimer les arrêtes de la puce. Dès que la puce est protégée, on réalise un amincissement de l'ordre de 10 µm par passe (Figure 3-8). Lorsque nous somme proche de l'épaisseur résiduelle désirée, des passes de finition sont réalisées. Ces passes de finition enlèvent moins de matière et sont effectuées plus lentement. Au final pour préparer la puce FPGA, nous avons ôté près de 760 µm de silicium en plus de 6 heures.

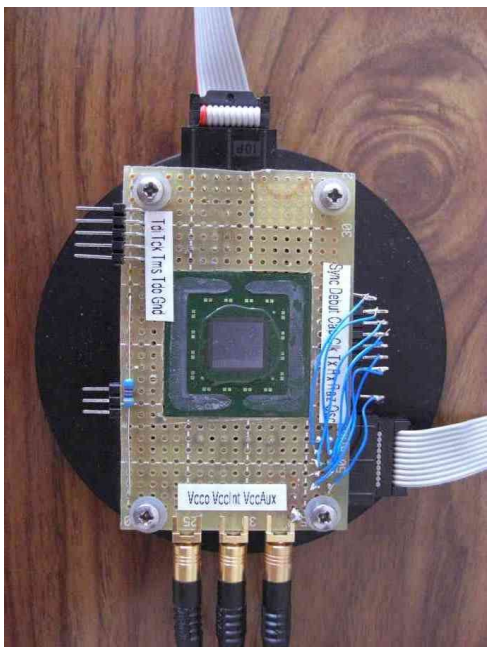
3.4. Les cartes de test

Pour réaliser les campagnes d'injection de fautes par tir laser ou par application de surtensions, j'ai développé des cartes propres à chaque campagne. Les caractéristiques principales de ces cartes de test sont présentées dans les sections suivantes. On montrera également que pour les campagnes d'injection de fautes par tirs laser, il n'est pas nécessaire de câbler toutes les broches du circuit. Enfin, nous montrerons ce qu'il est nécessaire de réaliser pour appliquer des surtensions sur les alimentations du circuit.

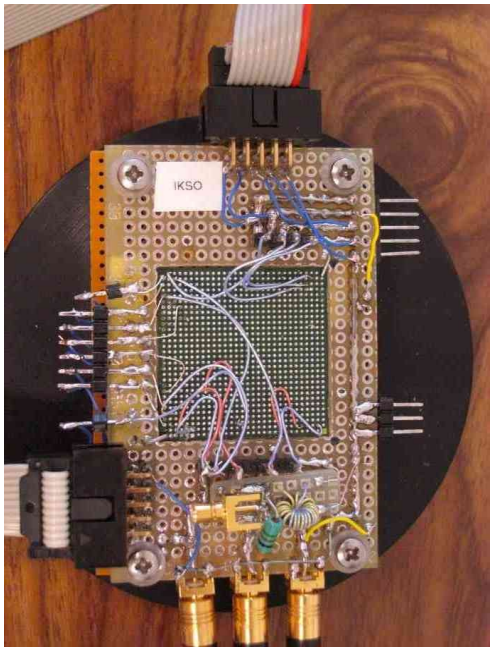
3.4.1. Les campagnes laser

Une carte de test dédiée aux injections de fautes par tir laser a été développée sur laquelle uniquement les broches du composant utiles ont été soudées (Figure 3-9). Les broches utiles pour le bon fonctionnement du composant lors des campagnes statiques sont les connexions JTAG (TMS, TCK, TDI et TDO), l'alimentation de la banque 4 (Vcc_I04), l'alimentation auxiliaire (Vcc_Aux) nécessaire pour le JTAG, l'alimentation du cœur (Vcc_Int) et la masse du circuit. Uniquement l'alimentation de la banque 4 est utile pour les tests statiques selon [Xilinx 2007-1] car aucune entrée-sortie du circuit n'est utilisée lors des tests

statiques. Cette alimentation est vérifiée lors de la remise à zéro avant toute configuration et si aucune tension n'est présente sur les broches dédiées à cette banque alors il sera impossible de configurer le circuit FPGA. Dans le cas des campagnes dynamiques, nous avons soudé les broches supplémentaires nécessaires telles que l'horloge, la remise à zéro et plusieurs entrées sorties.



a)



b)

Figure 3-9 : Vues de dessus (a) et de dessous (b) de la carte de test pour les attaques laser

On note sur la Figure 3-9 qu'une ouverture a été faite dans la carte de test pour avoir accès direct au silicium aminci mais également pour simplifier le brasage des différentes connexions (Figure 3-9-b). Cette ouverture a permis de ne pas faire fabriquer spécifiquement un support BGA avec une ouverture sur le dessus, car ce support aurait été fabriqué sur mesure entraînant un coût de fabrication élevé. De plus l'utilisation du support aurait entraîné des contraintes supplémentaires pour le réglage de la focalisation des bancs laser et plus particulièrement avec le banc microscope.

3.4.2. Les campagnes de surtensions

Une carte a également été développée pour les campagnes d'injections de fautes par surtensions. La caractéristique principale de cette carte dédiée est la séparation des alimentations. Sur la Figure 3-10 nous pouvons voir les différentes alimentations séparées du FPGA telles que l'alimentation du cœur du composant (Vcc_Int), l'alimentation des modules auxiliaire internes (Vcc_Aux) et les alimentations pour le réglage des niveaux de tensions admissibles par le composant pour chaque banque (Vcc_IOx avec x le numéro de la banque). Cette séparation des alimentations permet de maîtriser l'alimentation sur laquelle les perturbations seront injectées sans perturber les autres alimentations du circuit. Cette carte ne permet pas de générer des surtensions sur la masse du circuit puisque la masse du circuit doit être découplée du reste de la carte, ce qui n'a pas été défini initialement dans le cahier des charges.

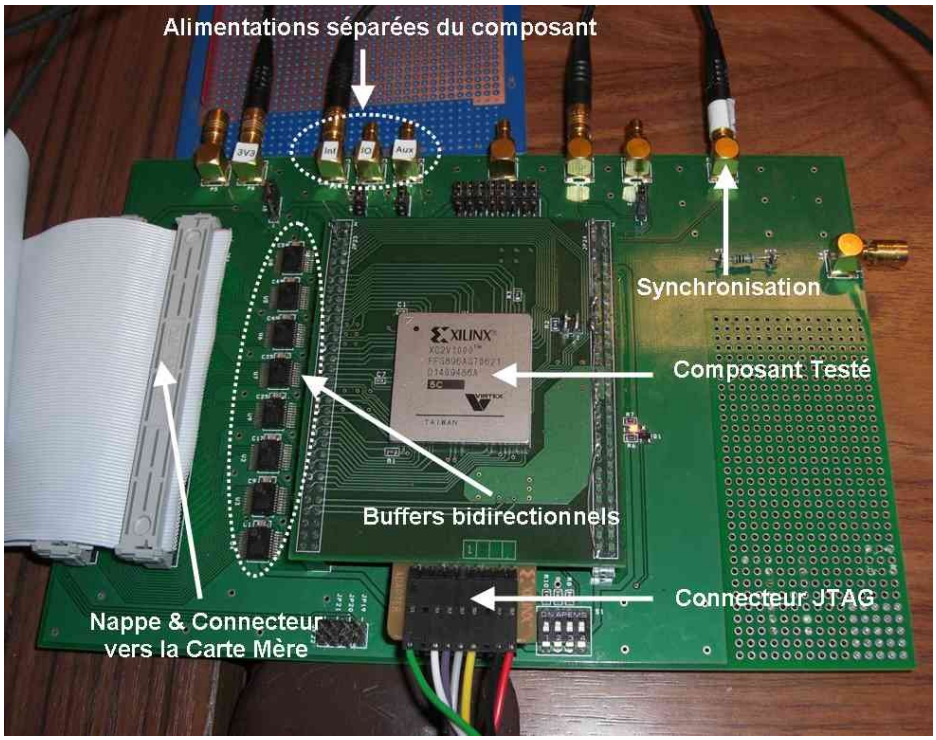


Figure 3-10 : Photo de la carte de test avec les alimentations séparées

Pour assurer la communication entre la carte mère et le composant testé, nous utilisons une nappe dans laquelle les signaux (commandes, JTAG, etc.…), les alimentations et la masse transitent. Sur la carte fille, des buffers bidirectionnels ont été implantés entre le circuit testé et la nappe de connexion. Ces buffers ont pour but premier de protéger la carte mère mais également de remettre en forme les signaux. Enfin, une autre spécificité de la carte est la possibilité de fonctionner sans avoir besoin de la connecter à la carte mère. En effet, nous avons implanté un connecteur JTAG permettant de configurer le composant testé et un connecteur pour connecter une horloge externe. Cependant, si la carte est dans la configuration "autonome", aucune communication avec l'environnement extérieur ne sera possible tels que l'écriture de registre et la lecture de ces derniers par un ordinateur.

3.5. L'outil d'analyse SefeaProD

Pour analyser les effets des injections de fautes dans la configuration du circuit, un outil a été développé au laboratoire TIMA. Cet outil utilise des classes Java fournies par Xilinx, nommées JBits 3.0 qui permettent d'accéder à la configuration des composants de la famille Virtex II. Ces classes Java ne sont plus actualisées pour les composants récents de la famille Virtex et plus particulièrement pour les familles conçues après la famille Virtex II.

Initialement, ces classes Java permettent de créer ou de modifier des fichiers de configuration. L'outil ainsi développé ne modifie pas les fichiers de configuration mais compare un fichier de configuration ou de relecture de référence avec un fichier dans lequel des modifications ont pu être apportées par des

injections de fautes. En comparant les différents fichiers bits à bits, une analyse des modifications est faite en fonction des différents éléments des tuiles CLB et de la valeur initiale du bit. La classification des différents éléments pouvant être analysés dans les tuiles CLB est donnée dans la Figure 3-11 et reprend les différents éléments des CLB présentés précédemment dans la section 1.2 du Chapitre 1. On remarque qu'il existe une catégorie nommée "Inconnue" correspondant à des bits dont la fonctionnalité est actuellement indéfinie par l'outil développé.

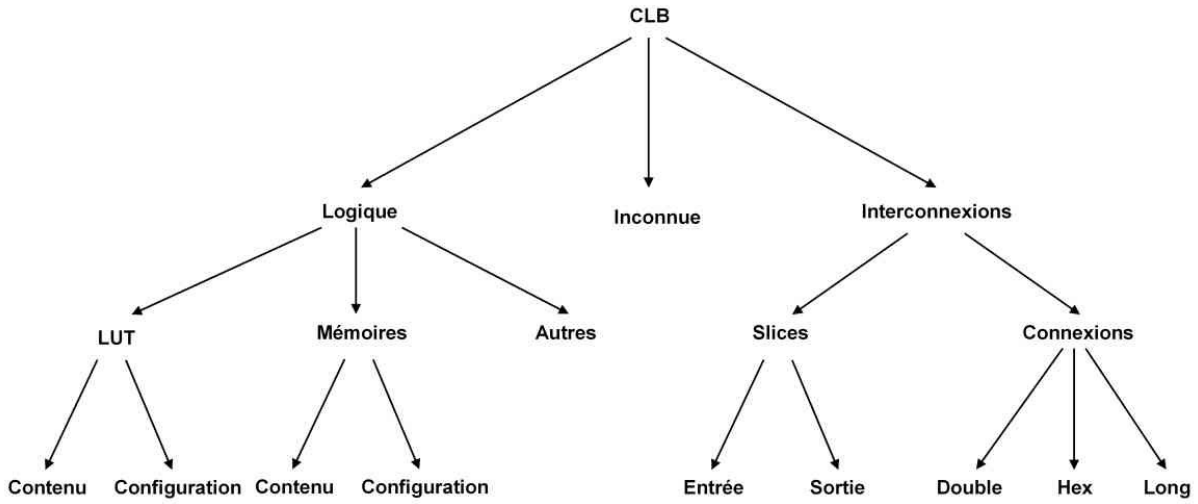


Figure 3-11: Différentes ressources des tuiles CLB pouvant être analysées par l'outil SefeaProD

Une autre spécificité de l'outil est de connaître les effets des modifications sur les connexions en fonction des états initiaux de ces dernières (Figure 3-12). Lorsqu'il existe un segment de connexion dans la configuration initiale, 4 effets sont possibles : la *modification* de la connexion correspondant à la suppression et à l'ajout d'une ou plusieurs nouvelles connexions, la *suppression* de la connexion initiale, *l'ajout* d'une ou plusieurs nouvelles connexions à celle initialement présente et la possibilité de n'avoir *aucun effet* sur la connexion. Ce dernier cas peut sembler un peu bizarre mais est tout à fait possible. En effet comme expliqué dans la section 1.2.1 du Chapitre 1, les bits configurant les connexions définissent une liste de sensibilité et toute modification de bits ne conduit pas obligatoirement à des effets sur les connexions. Enfin, lorsqu'il n'existe pas de connexion à l'état initial, uniquement la *création* d'une ou plusieurs connexions et la possibilité de n'avoir *aucun effet* pour les mêmes raisons que précédemment peuvent être obtenus lors des injections de fautes.

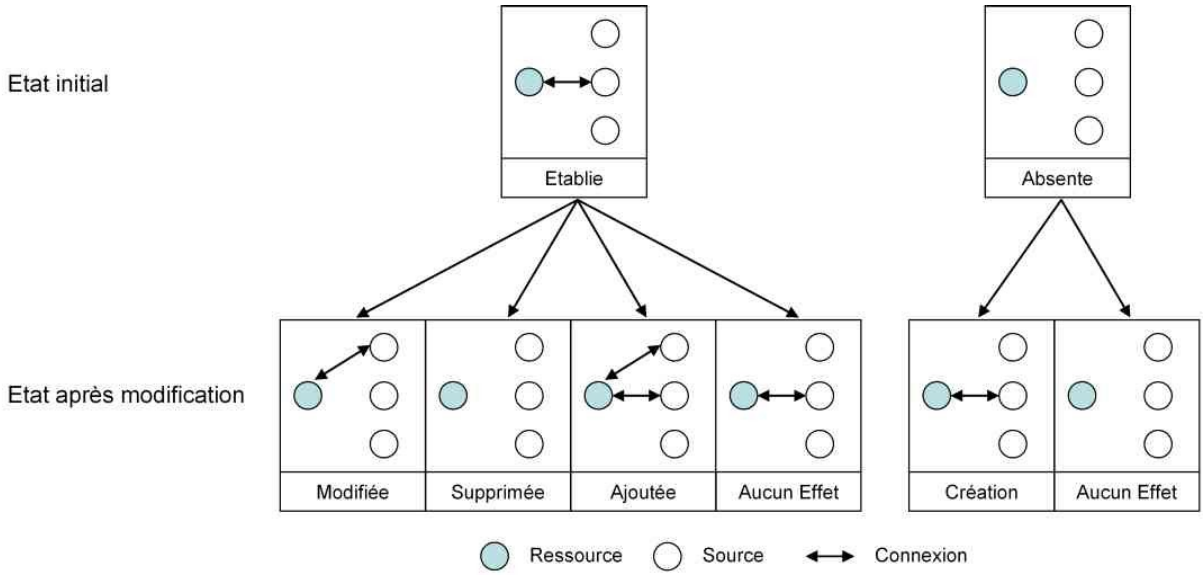


Figure 3-12: Différents types de modifications des connexions

D'autres fonctionnalités sont également présentes dans l'outil comme les visualisations au niveau des frames, de l'architecture et des tuiles CLB. Pour une présentation plus détaillée de l'outil se référer à [Pouget 2007] et [Maingot 2007].

3.6. Les circuits implantés pour les caractérisations

Pour la caractérisation des effets des tirs laser sur la configuration du FPGA, j'ai implanté deux circuits, l'un composé uniquement de logique combinatoire et l'autre constitué à la fois de logique combinatoire et de logique séquentielle. Ces circuits ont été utilisés lors de campagnes statiques.

3.6.1. Le circuit à base de logique combinatoire

Le circuit à base de logique combinatoire est constitué uniquement de portes logiques **'ET'** et **'OU'**. L'architecture du circuit est composée de 3 types d'étages : entrée, intermédiaire et sortie et est représentée sur la Figure 3-13. Chaque étage possède deux entrées (***A*** et ***B***), un sélecteur de fonctions (***'ET'*** ou ***'OU'***) et une sortie.

L'étage d'entrée est configuré en fonction 'ET' et possède les entrées A et B du circuit. L'étage intermédiaire utilise la fonctionnalité 'OU' et est reproduit plus de 4500 fois grâce à de la généricité afin de remplir au maximum la surface du composant. Cet étage intermédiaire possède l'entrée A du circuit et la sortie de l'étage précédent (n-1). En ce qui concerne l'étage de sortie, celui-ci est configuré en 'ET' logique ayant pour entrées celle du circuit et la sortie de l'étage précédent (n) et la sortie de cet étage correspond à celle du circuit.

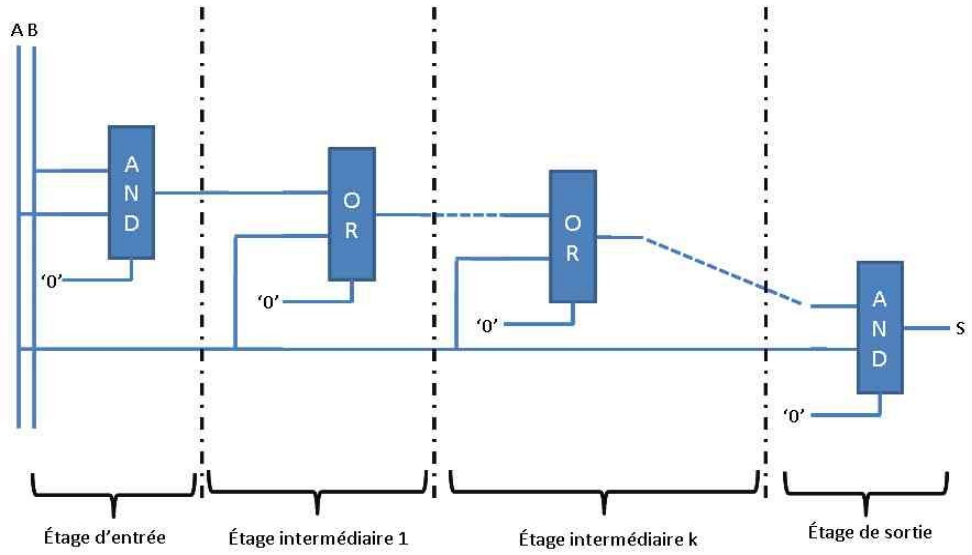


Figure 3-13 : Schéma logique du circuit à base de logique combinatoire implanté

A partir des rapports de synthèse du logiciel de Xilinx ISE-9.2i, nous notons que 4509 slices sont occupées par ce circuit sur les 5120 slices disponibles soit près de 88% de la surface totale.

Tableau 3.1: Répartition des bits de configuration du circuit à base de logique combinatoire en fonction du type de tuile et de leur valeur initiale.

Catégorie de tuiles / Bits de la configuration		Total	GCLK	IOB	IOI	CLB		BRAM	BRAM Int
						CLB	IOI		
Total		3.748.160	13.568	23.744	152.640	2.275.328	112.640	868.352	298.496
'0'	**Nombre**	3.625.648	13.568	23.725	152.623	2.155.667	112.517	865.792	298.364
	Pourcentage	96,7%	100%	99,9%	99,9%	94,7%	99,9%	99,7%	99,9%
'1'	**Nombre**	122.512	0	19	17	119.661	123	2.560	132
	Pourcentage	3,3%	0%	0,1%	0%	5,3%	0,1%	0,3%	0,1%

La répartition des bits présents dans le fichier de configuration en fonction des différentes tuiles du FPGA est donnée dans le Tableau 3.1. On remarque que principalement des bits initialement à '0' sont présents dans la configuration du circuit. Ceci s'explique par le fait que le circuit implanté utilise uniquement des LUT donc des CLB, pour réaliser les portes logiques. Lorsque les LUT sont configurées en mode "*équation*" alors les bits de configuration sont à leur valeur par défaut '0'. Des bits initialement à '1' dans les tuiles BRAM et leurs interconnexions associées sont présents pour initialiser ces éléments.

3.6.2. Le circuit à base de logique combinatoire et séquentielle

Le circuit à base de logique combinatoire et séquentielle est constitué d'une UART (*Universal Asynchronous Receiver Transmitter*) et d'un crypto-processeur. L'UART permet de faire dialoguer un ordinateur avec une carte électronique à travers une liaison série RS232. Pour analyser les commandes envoyées par la liaison série, une machine d'état est utilisée laquelle est active sur les fronts montants. Des bascules sensibles au front montant sont utilisées ainsi que des registres pour mémoriser des données. Nous ne donnerons pas plus d'informations sur le fonctionnement de l'UART car il est connu des électroniciens et des informaticiens mais également parce que le circuit implanté a été utilisé uniquement pour les campagnes de tests statiques. Le crypto-processeur est issu de [OpenCores 2007] et utilise l'algorithme de chiffrement DES ([FIPS PUB 46-3]). Le circuit ainsi implanté, regroupant l'UART et le crypto-processeur, occupe 925 slices sur les 5120 slices disponibles soit 18% des ressources du composant.

Tableau 3.2: Répartition des bits de configuration du circuit à base de logique combinatoire et séquentielle en fonction du type de tuile et de leur valeur initiale.

Catégorie de tuiles / Bits de la configuration		Total	GCLK	IOB	IOI	CLB		BRAM	BRAM Int
						CLB	IOI		
Total		3.748.160	13.568	23.744	152.640	2.275.328	112.640	868.352	298.496
'0'	**Nombre**	3.694.842	13.564	23.725	152.585	2.225.490	112.456	865.792	297.838
	Pourcentage	98,6%	100%	99,9%	100%	97,8%	99,8%	99,7%	99,8%
'1'	**Nombre**	53.318	4	19	55	49.838	184	2.560	658
	Pourcentage	1,4%	0%	0,1%	0%	2,2%	0,2%	0,3%	0,2%

La répartition des bits présents dans la configuration du FPGA est donnée dans le Tableau 3.2. Comme pour le circuit précédent, une majorité de bits est à '0' qui est la valeur par défaut. La majorité des bits initialement à '1' est présent dans les tuiles CLB qui sont les éléments les plus importants du FPGA. On note que le nombre de bits à '1' configurant les BRAM est constant quel que soit le circuit implanté et la valeur initiale. Ceci confirme que les bits initialement à '1' dans les tuiles BRAM sont utilisés pour l'initialisation.

3.7. Conclusion

Ce chapitre a permis de définir les conditions expérimentales dans lesquelles doivent se dérouler les différentes campagnes d'injection de fautes, notamment statiques. Les campagnes d'injection dynamiques seront présentées dans le dernier chapitre.

Plusieurs cartes électroniques ont été développées en fonction des spécificités des bancs utilisés et des campagnes. Pour injecter des fautes par tir laser, une préparation des échantillons a été nécessaire. En effet, les premières campagnes ont montré que lorsque le circuit FPGA n'est pas aminci, aucune faute ne peut être injectée dans la configuration. La préparation est donc quelque chose d'essentiel pour les injections de fautes par tir laser.

Dans ce chapitre, nous avons introduit les différents circuits implantés dans le FPGA sous test mais également l'outil d'analyse développé au laboratoire TIMA. Cet outil a été utilisé pour caractériser les effets des injections de fautes sur les différents éléments du FPGA et plus particulièrement sur ceux des tuiles CLB.

Maintenant que nous venons de présenter les conditions expérimentales, nous allons dans les prochains chapitres présenter les résultats de caractérisation des effets des injections de fautes par tirs laser et par surtensions sur les éléments du FPGA.

Chapitre 4. Caractérisation des effets des injections de fautes

Ce chapitre présente une caractérisation des effets des injections de fautes par tir laser et par application de surtensions sur le circuit FPGA testé. Lors des campagnes d'injection de fautes par tirs laser, la configuration des différents éléments des tuiles CLB (logique et interconnexion) peut être modifiée. Une analyse de la sensibilité de ces éléments et des motifs de modification des interconnexions sera réalisée selon différentes caractéristiques des bancs laser. Une analyse détaillée des zones de sensibilité des bits en fonction du type de tuile (CLB et BRAM) et une interprétation au niveau transistor permettant d'expliquer certains résultats obtenus seront introduits. Nous montrerons également qu'il est possible quelles que soient les caractéristiques du banc laser et plus particulièrement la taille du spot laser utilisée de modifier un seul bit de la configuration. Enfin dans une dernière partie, nous présenterons le seul type d'élément modifiable dans la configuration du composant lors de l'application de surtensions.

4.1. Les effets généraux des attaques laser

Dans cette section, nous allons présenter les principaux effets des injections de fautes par tirs laser. Une classification des éléments modifiés dans les tuiles CLB en fonction de la taille du spot laser sera réalisée. Enfin, nous analyserons également les différents motifs de modification possibles des interconnexions en fonction de la taille du spot laser.

4.1.1. Répartition du nombre de fautes par tir laser

Pour caractériser les effets des tirs laser sur la configuration du composant, nous avons implanté le circuit composé de logique combinatoire et séquentielle présenté dans la section 3.6.2 du Chapitre 3. Ce circuit a été implanté afin d'avoir la possibilité de modifier les différents types d'éléments constituant les tuiles CLB. Des fautes ont été injectées en utilisant les différents bancs laser (section 3.2 du Chapitre 3) avec différentes valeurs de diamètres de spot laser : 100 µm, 40 µm, 20 µm et 8 µm. Différentes zones du composant ont été étudiées selon les valeurs du diamètre du spot laser. Les tuiles CLB étant toutes composées de la même manière, je n'ai pas jugé utile de sélectionner soigneusement la zone d'étude.

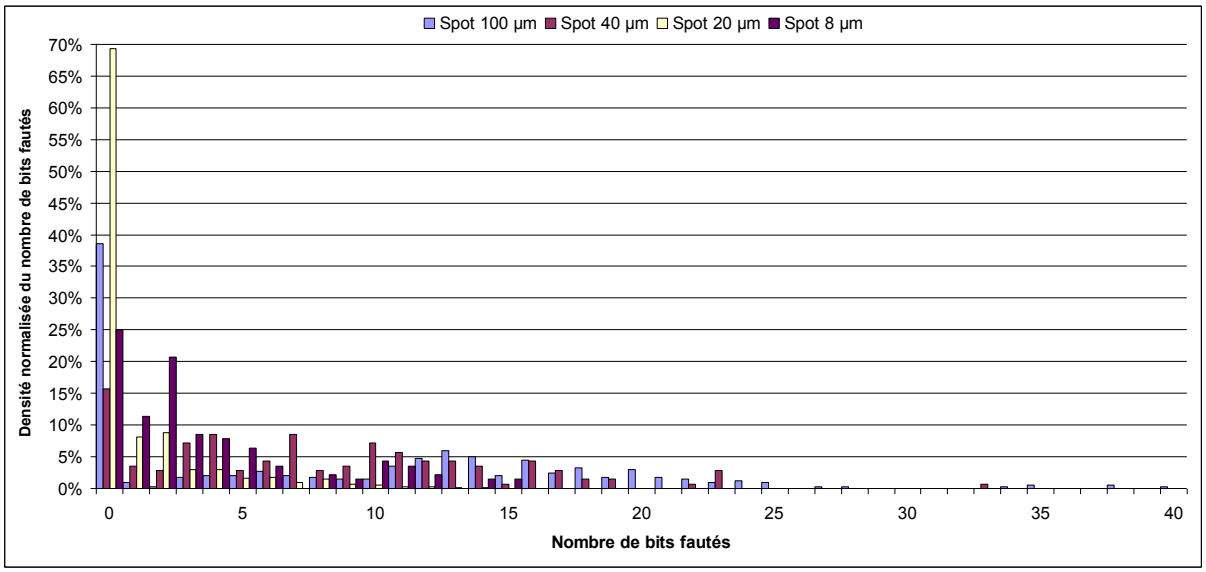


Figure 4-1 : Répartition du nombre de bits fautés par tir laser dans les tuiles CLB en fonction du diamètre du spot.

La Figure 4-1 représente les répartitions normalisées du nombre de bits fautés par tir laser en fonction de la taille du spot. Une majorité de tirs laser n'ont aucun effet sur la configuration du FPGA et un tir laser unique peut cependant induire un nombre variable de bits fautés quelle que soit la taille du spot laser. Ainsi avec un spot de 100 µm de diamètre, un tir laser peut générer de 0 à 40 erreurs dans la configuration du composant lors d'une même campagne selon la position de tir. Cette variabilité du nombre d'erreurs par tir peut être expliquée par le positionnement du faisceau laser sur le point mémoire, lequel peut différer d'une position à une autre puisque le point mémoire mesure quelques micromètres. En effet, selon l'emplacement du faisceau il est possible qu'une faible surface du point mémoire soit éclairée sans pour autant faire commuter la valeur mémorisée ou alors que plusieurs points mémoires soient illuminés en même temps induisant plusieurs bit-flips. La multiplicité peut être observée pour les autres valeurs de taille de spot mais avec des valeurs maximales de bits modifiés moindres. Toutefois avec des spots de 40 µm, 20 µm et de 8 µm, le nombre de fautes peut atteindre respectivement jusqu'à 33, 17 et 16 erreurs par tir laser.

4.1.2. Classification des éléments modifiés

Nous venons de montrer que des tirs laser sur le composant peuvent induire des erreurs de configuration. Pour caractériser leurs effets et pour dresser une classification des éléments modifiés dans les tuiles CLBs, une analyse fine a été réalisée en utilisant l'outil développé au laboratoire TIMA (section 3.5 du Chapitre 3) . Avec cet outil d'analyse, il est possible de déterminer quels sont les éléments des tuiles CLB modifiés : logique et/ou interconnexions et/ou éventuellement fonctionnalité inconnue.

La Figure 4-2 présente la répartition des bits fautés dans les tuiles CLB et plus particulièrement celle de la logique interne et des interconnexions. Nous n'avons pas représenté les bits dont la fonctionnalité est

actuellement inconnue car ils ne représentent que moins de 1% des bits modifiés. Sur la Figure 4-2, on note que les bits les plus sensibles des tuiles CLB sont les interconnexions qui représentent plus de 60% des modifications indépendamment de la taille du spot laser utilisée, excepté pour le spot de 20 µm. Le cas particulier du spot de 20 µm de diamètre peut être expliqué par la zone d'étude positionnée sur une zone regroupant une majorité d'éléments configurant la logique. Les bits configurant les interconnexions représentent plus de 80% des bits de configuration d'une tuile CLB, ce qui permet d'expliquer leur sensibilité plus importante.

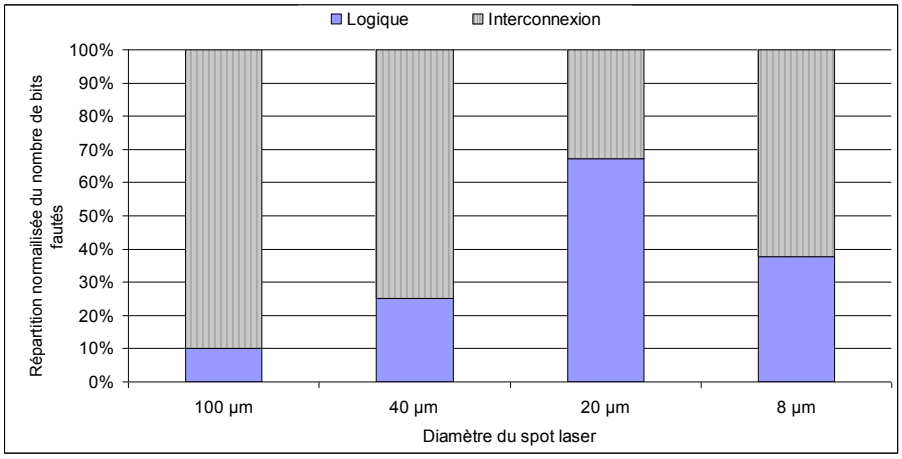


Figure 4-2 : Répartition des fautes dans les tuiles CLB

Tableau 4.1 : Répartition des bits configurant les interconnexions des tuiles CLB

Fonctionnalité	Nombre de bits par tuile	Pourcentage des interconnexions
Entrées des slices	658	45,1%
Sorties des slices	160	11,0%
Double	320	21,9%
Hex	280	19,2%
Long	40	2,7%
Total	1458	100%

Les interconnexions étant les éléments les plus sensibles de la tuile CLB, il faut identifier parmi les différents éléments les constituant ceux possédant une sensibilité plus importante. Les interconnexions peuvent être divisées en 5 éléments : les entrées et sorties des slices et les connexions de types Double, Hex et Long (section 1.2 du Chapitre 1). La répartition du nombre de bits configurant ces éléments par tuile CLB est donnée dans le Tableau 4.1. Nous notons que la majorité des bits configurant les interconnexions permettent de définir les entrées des slices (45%) et les connexions de type Double (22%) et Hex (19%). Lors des différentes campagnes, nous avons remarqués que les entrées des slices et les connexions de type Hex sont les éléments les plus sensibles avec près de 70% des modifications impactant

ces types de connexions (Figure 4-3). Les entrées des slices sont les éléments dont le nombre de bits de configuration est le plus important (Tableau 4.1). Par contre, bien que les connexions de type Hex soient les plus sensibles, rien ne permet d'expliquer actuellement pourquoi elles le sont plus que les connexions de type Double. En analysant la configuration du circuit implanté, la proportion de bits de bits à '1' configurant les Double est près de 6 fois plus importante que celle des Hex ce qui aurait pu laisser supposer que les Double sont plus sensibles, alors que ce rapport est beaucoup plus faible dans le Tableau 4.1. Une autre explication possible serait la position de la zone d'étude du composant se situant dans une zone configurant majoritairement les connexions de type Hex.

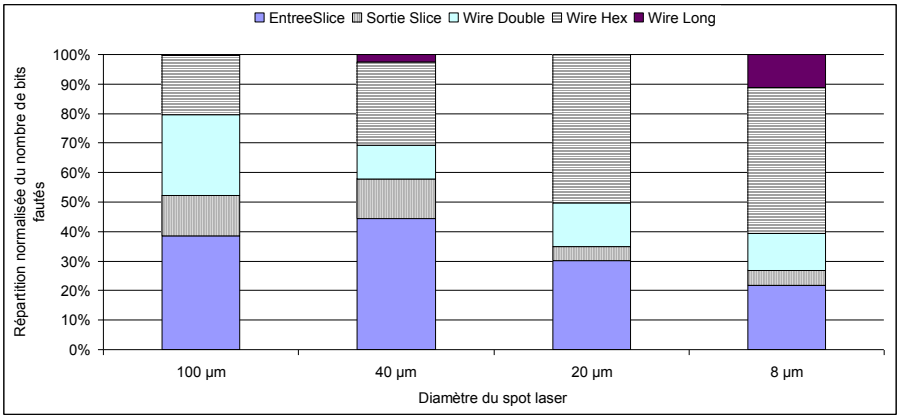


Figure 4-3 : Répartition des bits fautés configurant les différents éléments des interconnexions

La sensibilité de ces deux éléments, i.e. les entrées des slices et les connexions de type Hex, est dépendante de la taille du spot laser. La probabilité de modifier les bits configurant les entrées des slices est plus importante (38% des modifications) que celle des connexions de type Double (27%) et Hex (20%) en utilisant un spot laser de 100 µm. Pour le spot laser de 40 µm, la probabilité de modification de bit configurant l'entrée d'un slice est également plus importante (43%) que celle des bits configurant les connexions Hex (26%) et Double (11%). Une inversion de la sensibilité des connexions de type Double et Hex est observée lors de la diminution de la taille de spot. En effet, lors des études avec des diamètres de spot de 20 µm et de 8 µm, les tendances sont inverses puisque les tirs modifient majoritairement les connexions de type Hex et dans une moindre mesure les entrées des slices. Cependant avec des spots laser de 40 µm, 20 µm et 8 µm, la proportion de connexions de type Double modifiées reste quasi-constante autour des 13 %. En diminuant le diamètre du spot laser d'un rapport 5, le nombre de bits fautés configurant les entrées des slices est également divisé d'un rapport proche de 5 alors que celui des bits de configuration des connexions de type Hex l'est d'un rapport proche de 1,4 et de 2,5 pour les Double. Cependant, l'interprétation de ces résultats est assez difficile.

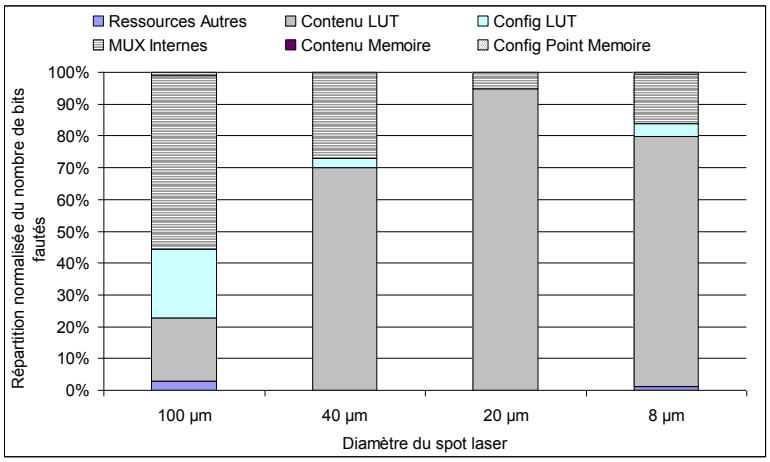


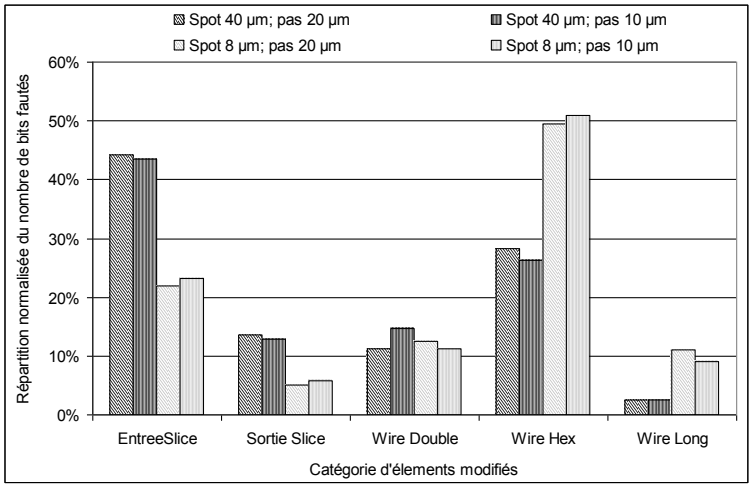
Figure 4-4 : Répartition des bits fautés configurant les différents éléments de la logique

Les bits de logique peuvent configurer des LUT (contenu et configuration), des multiplexeurs internes, des mémoires (contenu et configuration) et d'autres ressources (section 1.2 du Chapitre 1). Lors des campagnes, deux catégories d'éléments sont nettement plus sensibles que les autres : le contenu des LUT et les multiplexeurs internes (Figure 4-4). Ces deux catégories d'éléments représentant plus de 85% des bits de la configuration de la logique (Tableau 4.2), regroupent au total 95% des éléments modifiés. Nous notons également sur la Figure 4-4 qu'une diminution du diamètre du spot laser à pour effet d'augmenter la proportion de bits modifiés configurant le contenu des LUT et de diminuer ceux configurant les multiplexeurs internes.

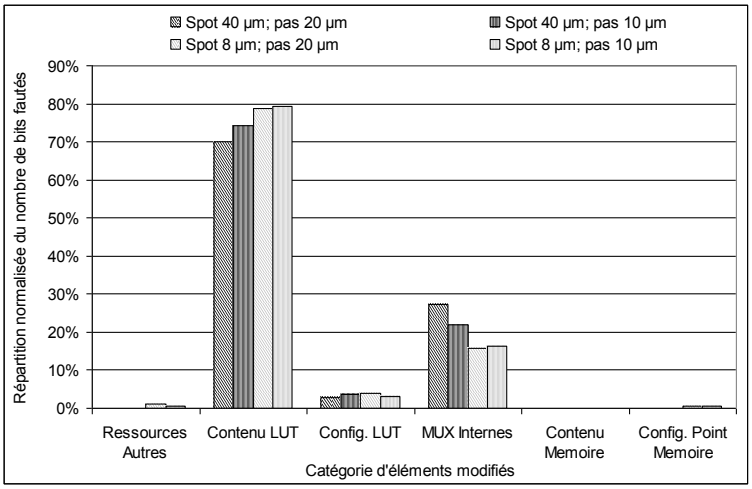
Tableau 4.2 : Répartition des bits configurant les éléments de logique des tuiles CLB

Fonctionnalité	Nombre de bits par tuile	Pourcentage de la configuration du CLB
Contenu des bascules	8	2,8%
Config. des bascules	16	5,7%
Contenu des LUT	128	45,4%
Config. des LUT	8	2,8%
Multiplexeurs internes	122	43,3%
Total	282	100%

Nous venons de montrer que la sensibilité des différents éléments constituant la logique interne et les interconnexions des tuiles CLB dépend de la taille du spot. La Figure 4-5 représente pour 2 diamètres de spot laser (40 µm et 8 µm) et pour 2 valeurs d'incrément spatial, les variations de la répartition des bits fautés. On note que la variation d'incrément spatial à très peu d'influence (inférieure à 4%) sur la répartition des bits fautés configurant soit les interconnexions (Figure 4-5-a) soit la logique interne (Figure 4-5-b). A contrario, la taille du spot laser est nettement plus influante et les variations de la répartition est quasi-proportionnelle au changement de taille de spot. Cependant la répartition des différents éléments de logique et de connexions n'évoluent pas de la même façon lors des changements de taille de spot montrant une différence de sensibilité entre les différentes catégories d'éléments.



a)



b)

Figure 4-5 : Variation de la répartition des bits fautés lors des changements de pas incrémental (hachures obliques correspondant à un pas de 20 µm et les verticales à un pas de 10 µm) et de spot laser (couleur noire correspondant à une taille de 40 µm et la grise à une taille de 8 µm) pour les éléments configurant les interconnexions (a) et la logique (b).

4.1.3. Motifs de modification des interconnexions

Dans le Chapitre 1, les différentes catégories d'interconnexions pouvant être définies par 1, 2 ou 3 bits ont été introduites. Plus de 90% des connexions sont configurées par plusieurs bits selon [Maingot 2007] et nous nous intéresserons principalement à cette catégorie. Dans nos résultats, aucune distinction n'est faite pour les connexions définies par plusieurs bits (2 ou 3 bits) car cette fonctionnalité n'est pas intégrée dans l'outil d'analyse utilisé. La répartition des bits fautés configurant les interconnexions définies par un ou plusieurs bits est donnée dans la Figure 4-6. Les connexions modifiées définies par un seul bit représentent 1% et près de 4% des modifications respectivement pour les spots laser de 40 µm et 8 µm confirmant les résultats présentés dans [Maingot 2007] lors de tirs multiples. Une diminution de la taille du

spot laser peut permettre d'augmenter la probabilité de modification des connexions définies par un seul bit mais les connexions définies par plusieurs sont toujours majoritaires.

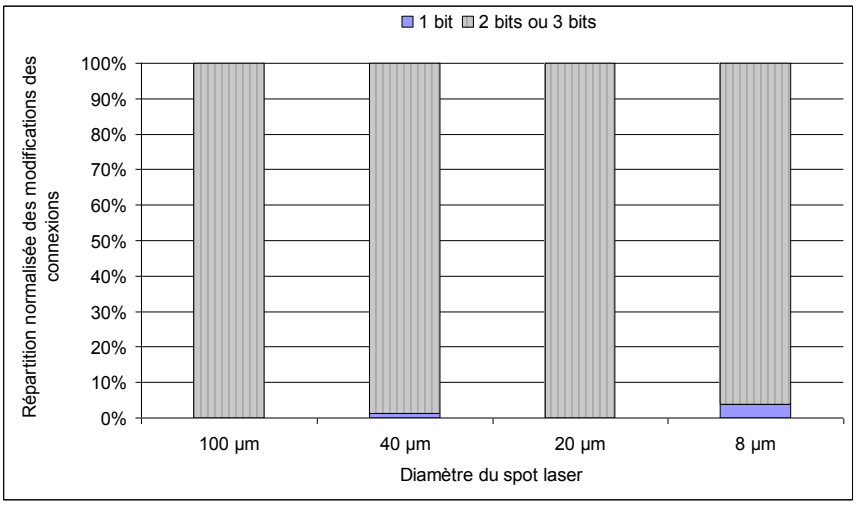


Figure 4-6 : Classification des motifs de modification des connexions en fonction du nombre de bits de configuration

Les effets des tirs laser sur les connexions dépendent principalement de leur état initial : connectée ou non comme représenté sur la Figure 4-7. Lorsqu'une connexion est présente à l'état initial, 4 motifs de modification différents sont possibles : la modification, la suppression, l'ajout ou aucun effet sur la connexion. Les deux premiers motifs correspondent à la suppression de la connexion initiale et potentiellement à l'ajout d'une ou plusieurs nouvelles connexions. Les deux autres motifs, "ajout" et "aucun effet" consistent à conserver la connexion initiale et à ajouter ou non une ou plusieurs nouvelles connexions. Par contre, lorsque la connexion est initialement non-connectée, uniquement deux motifs sont possibles : la création ou aucun effet avec aucune création.

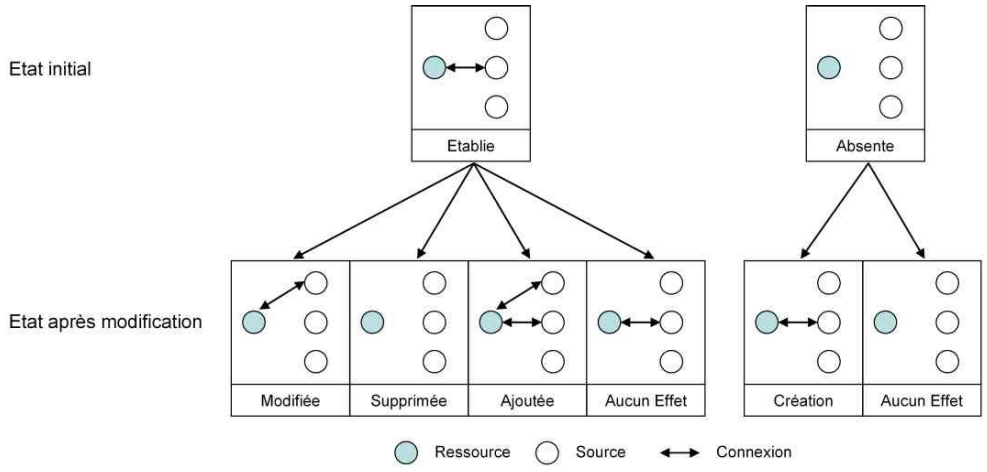


Figure 4-7 : Différents types de modifications des connexions

Selon [Maingot 2007], plus de 90% des modifications des connexions concernent celles qui sont initialement non connectées lors de tirs multiples sur la configuration. Des résultats similaires (supérieurs à 90%) ont été obtenus indépendamment de la taille de spot utilisée lors de tirs uniques. Près de 80% des modifications sur ce type de connexions initiales n'ont "aucun effet" puisqu'il est nécessaire de modifier au moins deux bits configurant la même ressource pour créer une connexion.

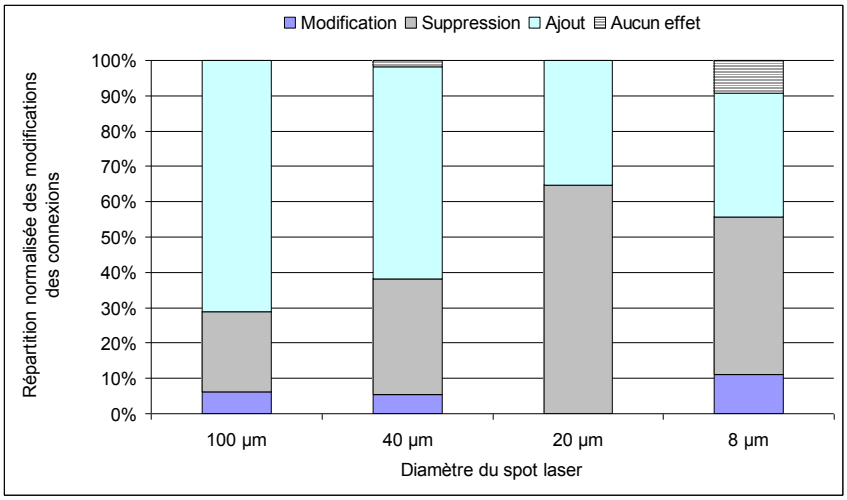


Figure 4-8 : Classification des motifs de modification des connexions définies par 2 bits initialement connectées

Lorsqu'une connexion est initialement connectée (Figure 4-8), les modifications induites par le ou les tirs laser ajoutent principalement des connexions à celle existante (50%) et la supprime (modification et suppression) dans près de 30% et de 40% des cas respectivement pour les spots laser de 100 µm et 40 µm. Ces résultats confirment ceux obtenus dans [Maingot 2007]. Cependant avec les spots laser de 20 µm et de 8 µm, les résultats sont inverses puisque nous supprimons principalement la connexion existante. Comme pour la sensibilité des interconnexions, les effets tendent à s'inverser lors de l'utilisation d'un spot laser de 20 µm ou de 8 µm montrant ainsi une dépendance de la taille du spot laser sur les résultats obtenus.

4.2. L'analyse détaillée des zones de sensibilité

4.2.1. Quelques définitions

Afin d'analyser les zones de sensibilité des bits, nous allons définir les termes de "zone de sensibilité" et de "diamètre équivalent", lesquels seront utilisés dans la suite de cette section.

Définition 1 : La **zone de sensibilité** correspond pour un bit donné au nombre de positions XY permettant de modifier la valeur du dit bit. La surface calculée de la zone de sensibilité est dépendante de la valeur d'incrément spatial utilisée.

Définition 2 : Le ***diamètre équivalent*** de la zone de sensibilité est calculé à partir de la surface totale de la zone quelle que soit sa forme. On suppose pour ce calcul que cette zone de sensibilité est assimilée à un disque.

4.2.2. Les zones de sensibilité des tuiles CLB et BRAM

Pour caractériser les zones de sensibilité des tuiles CLB et BRAM, nous avons utilisé deux tailles de spot : le 20 µm et le 100 µm. Trois zones ont été étudiées avec la taille de spot de 20 µm de diamètre, deux dans les tuiles CLB afin de montrer l'indépendance de la localisation géographique des zones de sensibilité et une dans les tuiles BRAM. Avec le banc laser permettant d'obtenir une taille de spot de 100 µm, uniquement une zone de tuiles CLB et une de BRAM ont été balayées. La valeur d'incrément spatial vaut 1/10 de la valeur théorique du diamètre du spot laser. Ainsi, pour le spot laser de 20 µm nous avons utilisé un pas d'incrément spatial de 2 µm et de 10 µm pour le spot laser de 100 µm de diamètre. Dans les statistiques issues des différentes sous-campagnes qui suivent, les diamètres équivalents sur le bord des zones d'études n'ont pas été éliminés ce qui peut modifier les résultats ainsi obtenus. L'analyse de la taille moyenne des spots laser consiste pour chaque bit de la configuration à regarder l'ensemble des positions XY du tir laser ayant conduit à une modification de ce bit. La taille observée est mesurée en nombre de points en fonction de l'état initial du bit et donc de son forçage. La surface de chaque point dépend de la valeur du pas d'incrément spatial. Chaque point est représenté pas un carré de côté valant la valeur du pas de déplacement. Ainsi, pour le spot laser de 20 µm, le point à une surface de 4 µm² et de 100 µm² pour le spot de 100 µm.

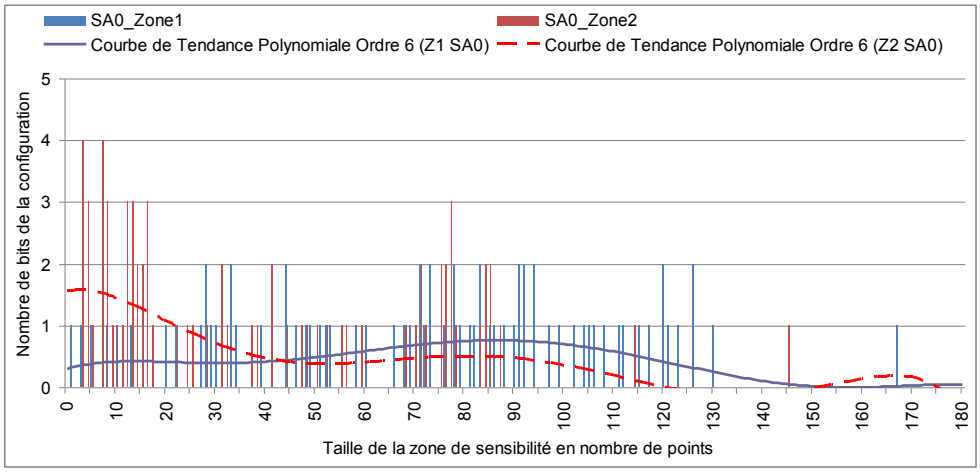


Figure 4-9 : Répartition des bits de la configuration en fonction de la taille de la zone de sensibilité observée, mesurée en nombre de points, lors des forçages à '0' dans la zone d'étude 1 et la zone d'étude 2 des tuiles CLB en utilisant le spot laser de 20 µm.

Sur la Figure 4-9 nous remarquons que lors des transitions de '1' vers '0', la surface de la zone de sensibilité varie entre 4 µm² et 668 µm² pour la zone d'étude 1 et entre 12 µm² et 580 µm² pour la zone 2. A cause des variations des surfaces des zones de sensibilité, nous décidons de tracer une courbe de tendance

polynomiale afin de représenter ces fluctuations. Nous avons choisi d'utiliser l'ordre du polynôme le plus important disponible du tableur utilisé.

A partir des courbes de tendance polynomiales à l'ordre 6, deux tangentes nulles sont obtenues pour les zones 1 et 2. Pour la zone 1, les tangentes sont nulles autour de 10 points et 90 points respectivement 7,14 µm et 21,41 µm alors que pour la zone 2 elles le sont autour de 5 points et 85 points (5,05 µm et 20,81 µm). La localisation spatiale a donc peu d'influence sur les résultats du fait de la régularité de la matrice de CLB.

Pour des transitions de '0' vers '1', nous remarquons sur la Figure 4-10 que les valeurs des surfaces efficaces sont nettement inférieures à celles des transitions de '1' vers '0'. A partir des courbes de tendance des zones 1 et 2, nous notons que le maximum de points se situe autour de 5 points (5,05 µm de diamètre) et que les deux courbes ont des tendances similaires. Cependant, il est difficile de conclure sur la surface exacte de la zone de sensibilité dans le cas des transitions de '0' vers '1'.

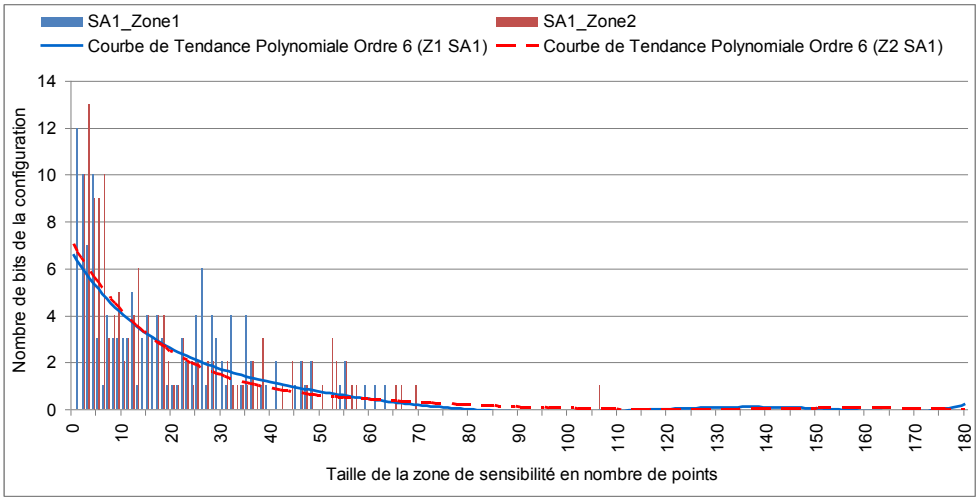


Figure 4-10 : Répartition des bits de la configuration en fonction de la taille de la zone de sensibilité observée, mesurée en nombre de points, lors des forçages à '1' dans la zone d'étude 1 et la zone d'étude 2 des tuiles CLB en utilisant le spot laser de 20 µm.

La Figure 4-11 représente pour le spot laser de 100 µm, la répartition des bits de la configuration modifiés initialement à '1'. Cette répartition couvre un spectre important de taille avec des zones de sensibilité variant entre 1 point et 193 points soit entre 11,28 µm et 156,76 µm. Sur cette figure, trois tendances se dégagent : 2 points, 55 points et 160 points soit des diamètres équivalents de 15,96 µm, 83,68 µm et 142,73 µm. Cependant la tendance la plus probable est 83,68 µm car c'est pour cette dernière qu'il apparaît un nombre maximal de points.

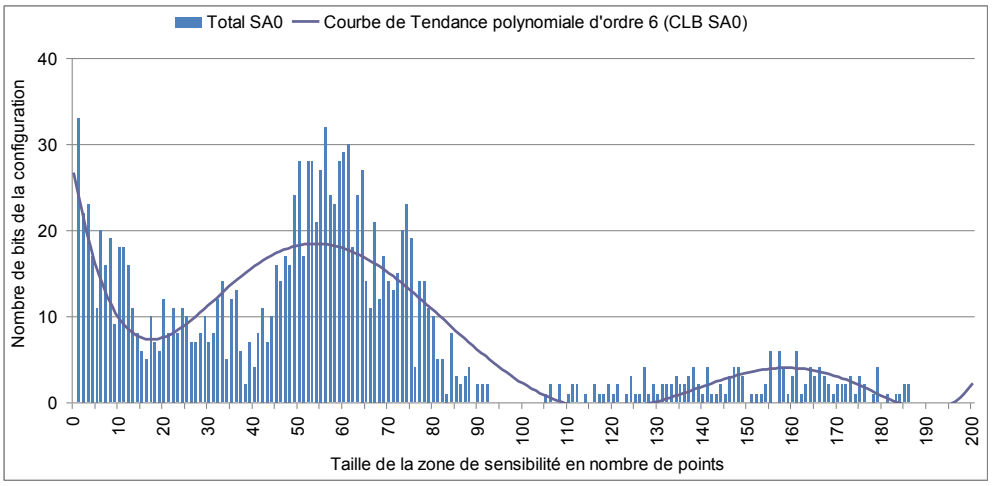


Figure 4-11 : Répartition des bits de la configuration en fonction de la taille de la zone de sensibilité observée, mesurée en nombre de points pour des forçages à '0' dans la zone d'étude des tuiles CLB en utilisant le spot laser de 100 µm.

La zone de sensibilité des bits passant de '0' à '1' est nettement plus faible que pour des bits passant de '1' vers '0' (Figure 4-12) et il est plutôt difficile de définir une valeur précise de la taille de la zone de sensibilité comme lors de l'utilisation du spot de 20 µm.

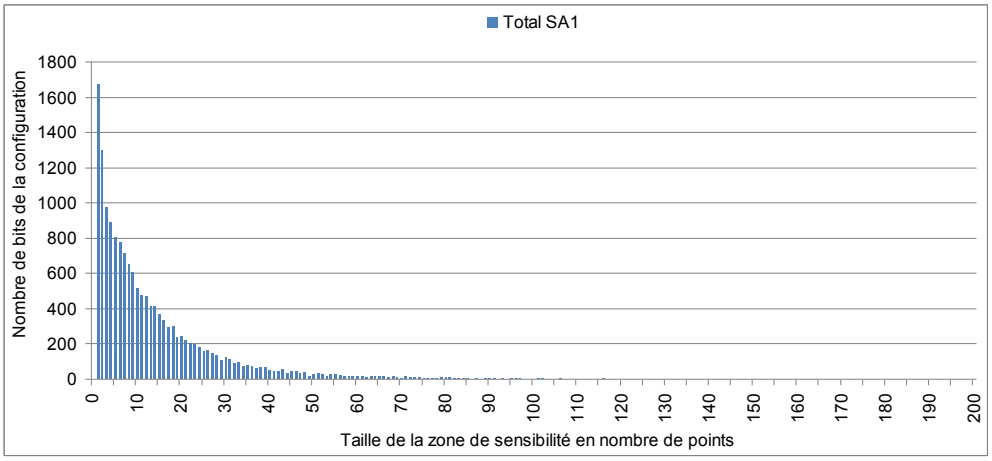


Figure 4-12 : Répartition des bits de la configuration en fonction de la taille de la zone de sensibilité observée, mesurée en nombre de points pour des forçages à '1' dans la zone d'étude des tuiles CLB en utilisant le spot laser de 100 µm.

En définitive, quelle que soit la taille du spot laser il existe bien des différences de taille des zones de sensibilité selon la valeur initiale du bit mémorisé. Pour expliquer ces différences, nous avons tracé sur la zone d'étude les positions de tirs permettant de modifier chaque bit individuellement. Les résultats obtenus sont illustrés sur la Figure 4-13 pour des exemples de bits passant de '1' à '0' (Figure 4-13-a) et de '0' à '1' (Figure 4-13-b) lors de l'utilisation d'un spot laser de 20 µm. Pour modifier des bits initialement à '1', il faut éclairer le point mémoire dans une surface proche d'un disque. Alors que la zone de sensibilité est similaire à un croissant pour des bits initialement à '0'. Pour toutes les catégories de bits des tuiles CLB

(logique et interconnexions) et des tuiles BRAM, des résultats similaires ont été obtenus : un disque pour les transitions de '1' vers '0' et un croissant pour les transitions inverses. Des formes similaires ont été obtenues pour un spot de 100 µm de diamètre. La forme de la zone de sensibilité est indépendante de la fonctionnalité du bit, mais dépendante de la valeur initiale du bit.

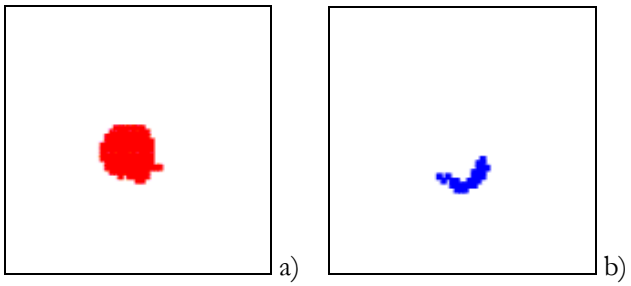
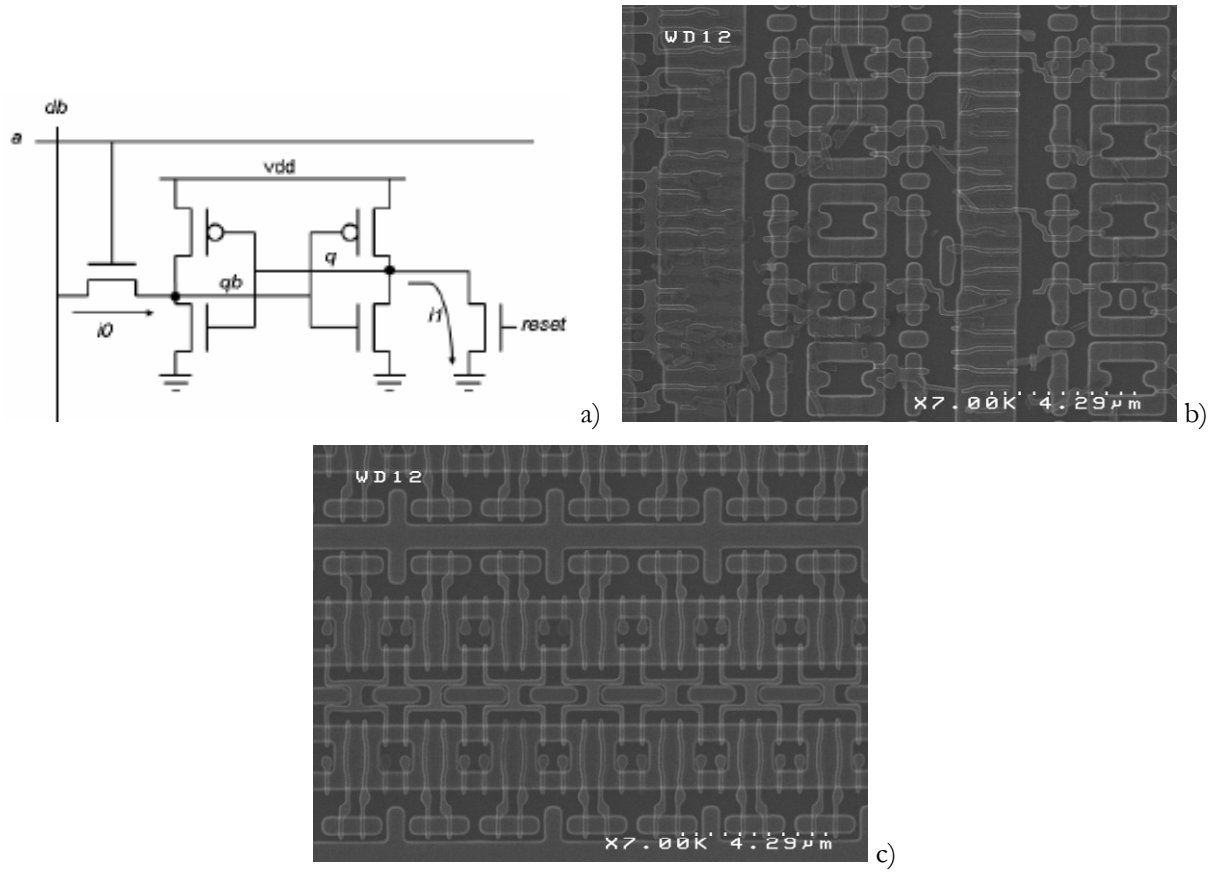


Figure 4-13 : Exemples de forme de zone de sensibilité pour des bits passant de '1' à '0' (a) et de '0' à '1' (b) dans une zone CLB en utilisant le spot de 20 µm

Afin d'affiner cette étude et pour savoir s'il existe une corrélation entre la durée d'éclairement et la forme de la zone de sensibilité, une campagne de tests supplémentaires a été réalisée en agissant sur la valeur de la largeur de l'impulsion (500ns, 250ns, 100ns et 50ns) et donc de l'énergie émise par la diode laser. Les résultats détaillés de l'influence de la durée d'éclairement sur les formes des zones de sensibilité seront présentés dans la section 5.2 du Chapitre 5. Toutefois, il faut noter que la modification de ce paramètre ne change pas la forme de croissant observée. La différence de forme n'est donc pas explicable ainsi.

4.2.3. Interprétation au niveau transistor

Il est possible de supposer que la forme de la zone sensible est due à la conception du FPGA et plus particulièrement au dessin des transistors. En effet, le composant FPGA utilisé est un composant de type SRAM c'est-à-dire que la configuration est stockée dans des points mémoires SRAM. Un point SRAM est constitué dans la plupart des cas de six transistors (SRAM-6T), or Xilinx utilise une structure à base de 5 transistors (SRAM-5T) plus un transistor de remise à zéro [Tuan 2007]. Ce point mémoire possède ainsi 2 inverseurs (2 transistors chacun : un PMOS et un NMOS) ainsi qu'un transistor servant d'interrupteur pour le stockage de la donnée (Figure 4-14-a).



a) b) c)

Figure 4-14 : Schéma électrique du point mémoire SRAM de configuration [Tuan 2007] (a) et dessin des tuiles CLB (b) et des tuiles BRAM (c) obtenu au CEA Grenoble en utilisant un Microscope Électronique à Balayage

Une première hypothèse possible concernant la différence de sensibilité entre le niveau '0' et le niveau '1' serait une dissymétrie de dimensionnement entre les deux inverseurs. Une étude du dessin des masques par microscope électronique à balayage a donc été réalisée (Figure 4-14-b). Il est possible de noter que le dessin des tuiles CLB est nettement moins régulier que le dessin des tuiles BRAM (Figure 4-14-c). Toutefois, la largeur et la longueur des canaux des transistors N ou P sont identiques pour les divers inverseurs identifiés. La dissymétrie ne provient donc pas du dessin des inverseurs.

La structure utilisée est symétrique du point de vue du schéma transistors puisque nous avons deux inverseurs tête-bêche et deux transistors de part et d'autre de ces derniers. Par contre, cette structure est asymétrique du point de vue fonctionnel car nous n'utilisons que 5 transistors pour faire fonctionner ce point mémoire SRAM, le transistor de remise à zéro n'étant utilisé que lors de la mise sous tension du composant.

En mesurant les dimensions des transistors issus des cellules SRAM de configuration (Figure 4-15), nous avons de plus obtenu pour le transistor NMOS une largeur Wn de 0,27 µm et une longueur Ln de 0,13 µm, et pour le transistor PMOS une largeur Wp de 0,45 µm et une longueur Lp de 0,15 µm. Nous avons donc des rapports W/L d'environ 2 pour le NMOS et 3 pour le PMOS. Le courant Ion du transistor

NMOS est donc de 30% à 100% plus élevé que le courant Ion du transistor PMOS, selon que l'on considère un rapport des mobilités électrons/trous égal à 2 ou 3. Ces inverseurs ont donc une zone de commutation décentrée, avec une commutation plus rapide de '1' vers '0'.

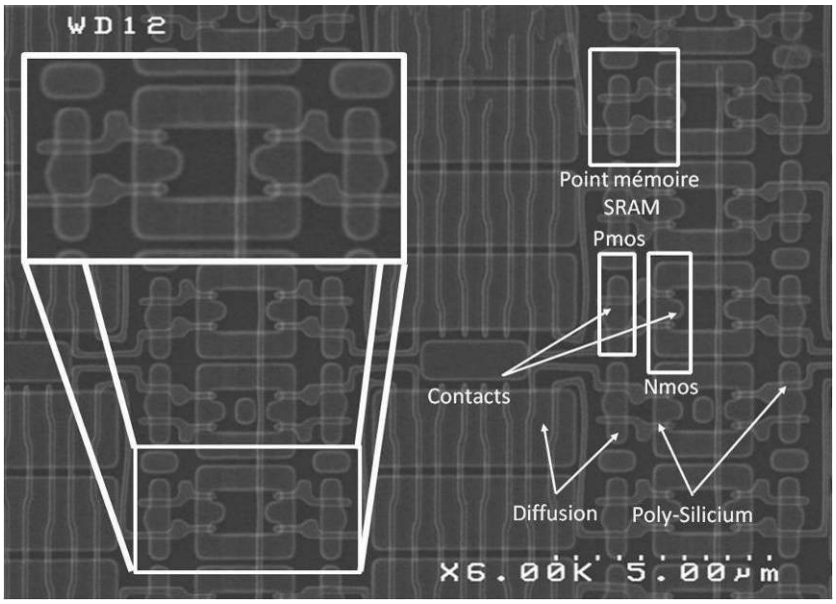


Figure 4-15 : Localisation des points mémoire SRAM et des transistors NMOS et PMOS dans les tuiles CLB

Un autre point important entraîne une dissymétrie supplémentaire du point mémoire. Du côté de la sortie (information de configuration mémorisée), le transistor de remise à zéro génère des courants de fuite vers la masse, donc ayant tendance à décharger la sortie (Figure 4-14-a). Au contraire, les lignes de bit étant préchargées au niveau haut [Tuan 2007], les courants de fuite du transistor d'entrée tendent à charger l'entrée, ce qui revient à renforcer un niveau '0' mémorisé. On voit donc ici que l'aggravation des courants de fuite dans ces deux transistors suite à une attaque tend à forcer un niveau '0' dans le point mémoire de configuration.

Si l'on considère maintenant les deux inverseurs rebouclés, lorsqu'un niveau '1' est mémorisé, le déclenchement du transistor N de l'inverseur direct (celui générant q) aura pour conséquence probable de forcer un '0' sur q et donc un basculement du point mémoire. Ce basculement sera rendu possible puisque le courant de décharge du transistor N sera plus élevé que le courant de charge fourni par le transistor P. Pour la même raison, le déclenchement du transistor P de l'inverseur de re-bouclage aura une plus faible probabilité de se traduire par une commutation du point mémoire.

Lorsqu'un niveau '0' est mémorisé, un basculement probable sera obtenu en mettant en conduction le transistor N de l'inverseur de re-bouclage. Le déclenchement du transistor P de l'inverseur direct aura a priori moins d'impact.

Toutes les informations nécessaires pour une quantification précise ne sont pas disponibles, mais cette analyse montre que le basculement de ‘1’ vers ‘0’ du point mémoire de configuration est beaucoup plus probable que le basculement inverse, ce qui est cohérent avec les résultats obtenus.

Par ailleurs, un basculement de ‘0’ vers ‘1’ nécessite de centrer la perturbation sur le transistor N de l'inverseur de re-bouclage ou éventuellement sur le nœud de mémorisation (perturbation directe de la charge stockée), sans perturber d'autres composants pouvant avoir un effet inverse.

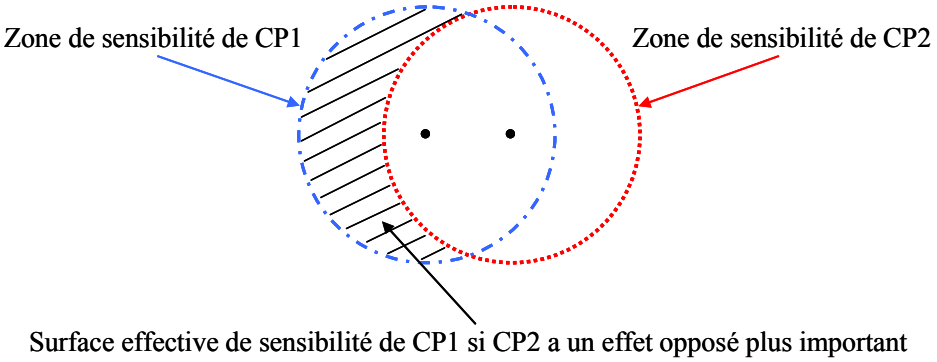


Figure 4-16 : Interprétation géométrique de la forme de la zone de sensibilité de bits passant de ‘0’ à ‘1’- CP1 étant un point d'attaque permettant un basculement de '0' vers '1' et CP2 étant un point d'attaque permettant un basculement de '1' vers '0'

En regardant plus précisément les masques de conception, une interprétation géométrique peut être proposée concernant les différences entre les formes des zones de sensibilité. Une transition de '0' à '1' est obtenue si le spot laser active le "bon" transistor NMOS, ce que nous appellerons "Point Critique 1" ou "CP1", mais qu'aucun des autres éléments du point mémoire qui tend à avoir un effet inverse ne soit activé. Nous appellerons "Point Critique 2" ou "CP2", un des autres éléments pouvant être activé par le laser et conduisant à un effet inverse. La zone de sensibilité CP1 ou CP2 peut être représentée par un disque centré sur le point critique (Figure 4-16). Si l'attaque laser a lieu dans la partie excentrée des disques alors un seul point critique sera activé. Par contre, si l'attaque apparaît dans la portion commune aux deux disques alors les deux points critiques sont activés mais suite à la discussion précédente, un seul effet sera prépondérant.

4.3. Le cas spécial des fautes uniques

Dans les sections précédentes comme dans [Maingot 2007], nous avons montré que des tirs laser peuvent induire des modifications multiples de bits permettant à un attaquant de contourner certaines contremesures. Par exemple, lors des tirs laser il serait possible de créer un nombre d'erreurs nettement supérieur à celui pouvant être détecté par une des contremesures présentées dans la section 2.5 du Chapitre 2. Mais un autre type d'attaque consiste à induire des fautes uniques dans la configuration du composant. Si un attaquant a la possibilité de modifier un seul bit de configuration choisi soigneusement,

il est capable de modifier une fonction particulière du circuit afin que toute corruption de donnée ne soit pas détectée. Par exemple, il sera possible de contourner des contremesures basées sur la redondance temporelle ou de forcer la sortie d'un compteur pour supprimer certaines rondes lors de calculs cryptographiques.

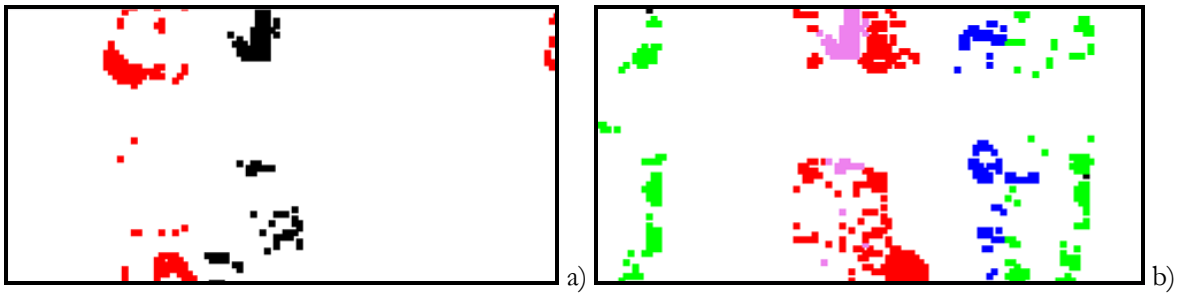


Figure 4-17 : Cartographie des positions de tirs laser conduisant à des fautes uniques avec un spot de 20 µm dans les tuiles CLB pour les éléments configurant la logique (a) et les interconnexions (b)

La Figure 4-17 montre les positions de tir laser avec un spot de 20 µm conduisant à des fautes uniques dans la configuration du FPGA et plus particulièrement dans les tuiles CLB. Dans la zone d'étude des tuiles CLB, 738 positions de tirs laser conduisent à des fautes uniques. Ce résultat correspond à 10% du nombre total de tirs laser réalisés lors de la campagne et 25% des configurations modifiées. Lors du balayage de cette zone, seuls 93 bits différents ont été modifiés à cause de la taille de la zone de sensibilité expliquée précédemment mais des fautes uniques ont été observées pour les différentes catégories de bits configurant la logique (Figure 4-17). Le contenu des LUT peut être en particulier modifié de manière sélective ainsi que les multiplexeurs internes. Des fautes uniques ont également été obtenues pour les interconnexions (Figure 4-17-b). Ce résultat est très intéressant dans l'optique de préparation de nouvelles attaques et/ou de validation de nouvelles contremesures.

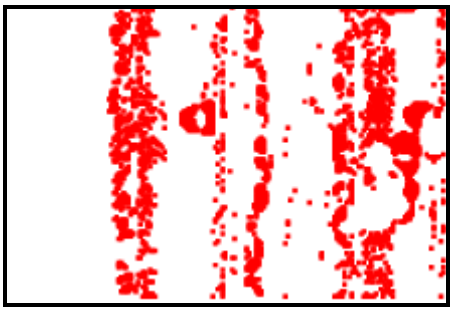


Figure 4-18 : Cartographie des positions de tirs laser conduisant à des fautes uniques avec un spot de 20 µm dans les tuiles BRAM

Les mêmes types de résultats ont été observés pour les tuiles BRAM comme représenté sur la Figure 4-18. Dans la zone d'étude des tuiles BRAM, 304 bits différents ont été modifiés à partir des 2350 positions de tirs laser conduisant à des fautes uniques. Cette proportion représente un peu plus de 11% du nombre total des tirs laser et 29% du nombre de configurations modifiées.

Enfin, en utilisant le banc laser permettant d'obtenir un spot de 100 µm, des fautes uniques ont également été obtenues (43 positions différentes pour 33 bits dans les tuiles CLBs et 24 cas pour 21 bits différents dans les BRAM). Bien que l'augmentation de la taille du spot laser fasse croître la multiplicité du nombre de bits fautés, il est cependant toujours possible d'obtenir des fautes uniques. Ainsi, pour induire des fautes uniques, il n'est pas nécessaire d'utiliser un banc laser très sophistiqué avec une taille de spot très fine.

4.4. Les effets généraux des attaques par surtensions

Pour caractériser les effets d'application de surtensions sur le composant, différentes campagnes ont été réalisées : statiques (circuit inactif) et dynamiques (circuit en fonctionnement). Lorsque le circuit ne fonctionne pas (mode statique), alors quelles que soient les surtensions appliquées aucune modification de la configuration (configuration et/ou bascules utilisateurs) n'est observée contrairement aux campagnes dynamiques. Pour les campagnes, nous avons implanté un circuit constitué de logique combinatoire et séquentielle dont les caractéristiques seront détaillées dans le Chapitre 6. Les ressources utilisées par ce circuit ont peu d'importance pour ce type de campagne car nous allons le montrer uniquement une seule catégorie de bits est modifiée. Différentes valeurs de surtensions (amplitude et durée) ont été appliquées à la fois sur les fronts montant et descendant de l'horloge principale de fonctionnement.

Une première sous-campagne notée S1-1 consiste à étudier les effets de surtensions d'amplitude comprise entre 5 volts et 45 volts, alors qu'une seconde sous-campagne (S1-2) permettra de caractériser les effets d'amplitudes supérieures comprises entre 45 volts et 80 volts. Pour ces deux sous-campagnes, les surtensions ont été appliquées sur le front montant de l'horloge Une seconde série de campagnes consiste à appliquer les surtensions sur les fronts descendant de l'horloge avec des amplitudes allant de 45 volts à 80 volts (S2). Toutes les surtensions seront appliquées durant différents cycles d'horloge de la machine d'état afin d'observer une influence des surtensions en fonction de l'instant d'injection.

Pour connaître les éléments modifiés de la configuration du composant à savoir des bits de la configuration et/ou les bascules utilisateurs, il est nécessaire d'utiliser un fichier de masquage généré par l'outil de compilation ISE-9.1 de Xilinx. Ce fichier de masquage possède la même taille que les fichiers de relecture permettant ainsi une comparaison caractère par caractère. Selon [Xilinx 2007-2], si le caractère du fichier de masquage est '0' alors le bit doit être vérifié avec le fichier de référence. Dans le cas contraire, il n'est pas nécessaire de le vérifier ce qui signifie que les bits masqués correspondent à des bits ne servant pas à la configuration du circuit implanté mais plutôt à des données utilisateurs. Ainsi en comparant un fichier de relecture fauté avec un fichier de référence en utilisant le fichier de masquage, il est possible de déterminer si les surtensions modifient la configuration et/ou les bascules utilisateurs.

Lors des différentes comparaisons, nous notons que les surtensions ne modifient pas la configuration du composant mais uniquement les bascules utilisateurs puisque seul l'état de bits masqués a changé. Ce

résultat est vérifié pour les deux fronts d'horloge étudiés ainsi que pour tous les cycles d'injections. En faisant varier les paramètres de la surtension : durée d'application, amplitude et délai utilisés lors des différentes campagnes, les mêmes résultats ont été observés. Contrairement aux tirs laser qui modifient la fonctionnalité des éléments des tuiles CLB (configuration et/ou utilisateur), les surtensions ne modifient pas la configuration du circuit mais uniquement les bascules utilisateurs.

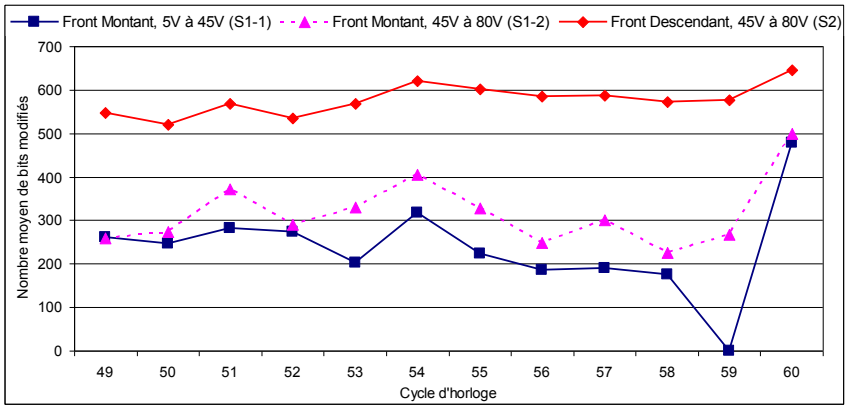


Figure 4-19 : Nombre moyen de bits de la configuration modifiés par les surtensions en fonction du cycle d'injection lors des campagnes S1 et S2.

La Figure 4-19 représente le nombre moyen de bits modifiés configurant le contenu des bascules utilisateurs selon le type de front d'horloge d'injection et la gamme d'amplitude de la surtension. Les nombres moyens sont calculés à partir du nombre total de bits modifiés par cycle de campagne et normalisés avec le nombre de configurations fautées. On remarque les mêmes tendances de résultats lors d'application de surtensions sur le front montant de l'horloge pour des amplitudes comprises entre 5 volts et 45 volts (S1-1) et entre 45 volts et 80 volts (S1-2). Ce nombre moyen de bits fautés dépend de l'amplitude de la surtension, plus la gamme de surtensions est élevée plus il y a d'erreurs dans le contenu des bascules. On note par ailleurs que le circuit est nettement plus sensible lors de surtensions sur le front descendant de l'horloge. En effet, en comparant les tendances d'une même gamme de tension (45 volts à 80 volts) en fonction du type de front d'horloge, on note que le nombre moyen de bits fautés dans les bascules utilisateurs est nettement plus important d'un rapport supérieur à 2 lors du front descendant. Dans [Djellid-Ouar 2006], une explication de la sensibilité des bascules suite à des glitches "raisonnables", c'est-à-dire n'excédant pas la tension nominale d'alimentation, concerne des violations de temps. Cependant aucune explication n'est donnée sur la sensibilité en fonction du type de front d'horloge. Une explication possible des différences observées de la sensibilité des bascules utilisateurs selon le front d'injection est une différence de dimensionnement des verrous.

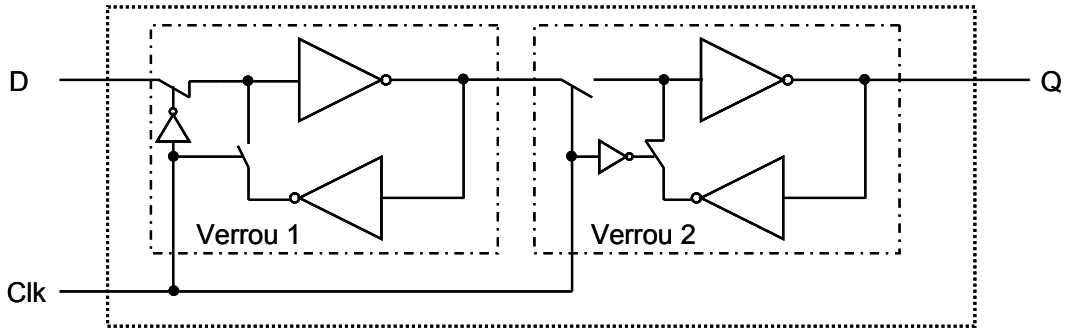


Figure 4-20 : Schéma d'une bascule à base de verrous

En effet, les bascules actives sur front montant sont conçues à partir de 2 verrous (Figure 4-20) : un verrou d'entrée transparent au niveau bas (*Verrou 1*) et un verrou de sortie transparent au niveau haut (*Verrou 2*). Le premier verrou est uniquement connecté au second alors que ce second verrou doit être dimensionné de sorte à connecter plusieurs autres éléments. Une différence de dimensionnement pourrait expliquer les sensibilités disparates.

4.5. Conclusion

Dans ce chapitre, nous avons présenté les principaux effets sur la configuration du composant lors d'injections de fautes par tir laser ou par application de surtensions.

Les tirs laser sur la configuration du FPGA ne permettent pas d'obtenir un nombre d'erreurs constant. Quelles que soient les zones étudiées et les tailles de spot laser employées, des positions de tirs permettent d'induire soit aucune erreur soit des erreurs simples ou multiples dans la configuration. Cependant, malgré ce nombre variable de bits fautés, il a été possible de réaliser une classification des éléments les plus sensibles dans les tuiles CLB (logique et interconnexion). En définitive, les interconnexions sont les éléments dont la probabilité de modification est la plus élevée ; ces éléments sont aussi ceux dont le nombre de bits de configuration est le plus important.

Une analyse plus fine a permis de répertorier pour chaque élément constituant la logique et les interconnexions ceux qui sont les plus sensibles. Si des bits configurant la logique sont modifiés alors dans la majorité des cas ces bits configurent des multiplexeurs ou le contenu des LUT. Tandis que pour des bits configurant les interconnexions, les entrées des slices et les connexions de type Hex sont les plus sensibles.

Dans un souci de sécurisation maximale d'un circuit implanté dans un FPGA de type SRAM, il est nécessaire d'avoir une bonne connaissance de la sensibilité des différents éléments mais également des effets que peuvent avoir les tirs laser sur ces éléments. Puisque les interconnexions sont les plus sensibles, nous avons analysé ce qui pouvait advenir des connexions lors des diverses modifications. Les modifications dépendent de l'état initial de la connexion (connectée ou non). Lorsqu'une connexion est initialement présente alors les tirs laser permettent principalement d'ajouter ou de supprimer la connexion et ce résultat dépend de la taille du spot laser. Par contre, si aucune connexion n'est présente alors dans la

majorité des cas, les modifications de la configuration n'ont aucun effet sur la connexion souhaitée. Dans plus de 90% des cas les connexions sont définies par 2 ou 3 bits. Pour créer une connexion il est nécessaire de modifier l'état d'au moins 2 bits définissant des listes de connexions possibles communes. Du fait de la définition des connexions, il est nettement plus aisé de supprimer ou d'ajouter une connexion en modifiant uniquement un seul bit.

Pour expliquer la sensibilité des bits, les zones de sensibilité des bits lors des tirs laser ont été détaillées. Quelle que soit la fonctionnalité du bit, les mêmes formes de zones ont été observées. Cependant, une différence majeure permet d'expliquer pourquoi certaines valeurs de bits sont plus sensibles que d'autres. Nous avons noté que pour des bits initialement à '1', la zone de sensibilité est proche d'un cercle alors que pour un bit à '0' elle est plutôt proche d'un croissant. Une interprétation de cette différence a été donnée dans ce chapitre et est basée sur une dissymétrie du point mémoire tendant à induire plus facilement un '0' qu'un '1' logique.

Nous avons également introduit un cas particulier d'attaque ne nécessitant pas l'utilisation de bancs laser très sophistiqués avec une taille de spot très fine pour induire des fautes uniques dans la configuration. Près de 10% des positions de tirs avec des focalisations moyennes conduisent à des fautes uniques ce qui peut être intéressant pour un attaquant pour contrecarrer certaines contremesures.

Pour les campagnes de surtensions, nous avons montré qu'uniquement des bits configurant le contenu des bascules utilisateurs ont été modifiés. En effet, quels que soient les paramètres de la surtension appliquée aucun bit de configuration n'a été modifié. Des tests en sous-tension consistant à ôter une impulsion de la tension nominale n'ont pas permis de générer des erreurs dans les bascules utilisateurs. Le composant étant alimenté en 1,5 volt, il faut générer une sous-tension maximale de la même valeur afin de ne pas déprogrammer le circuit FPGA. Les surtensions sont donc plus dangereuses dans l'optique d'une attaque.

Maintenant que nous connaissons les effets pouvant être obtenus par des injections de fautes par tirs laser et par application de surtensions, nous allons étudier la contrôlabilité des attaques laser. En particulier, nous allons présenter la reproductibilité des effets de ce type d'attaque sur la configuration, l'influence de l'énergie sur le nombre de fautes induites ainsi que l'influence de la plate-forme de test sur les injections.

Chapitre 5. Contrôlabilité des attaques laser

Précédemment nous avons vu que les attaques laser ont des effets sur la configuration du FPGA et une caractérisation de ces effets a été faite en fonction de plusieurs valeurs de taille de spot. Cependant, un attaquant peut modifier d'autres paramètres d'attaque afin de produire un effet désiré. L'objectif principal d'un attaquant est d'arriver au moins une fois à perturber le circuit. Si les effets ne sont pas reproductibles, il a la possibilité de réaliser plusieurs fois le même tir laser au même endroit pour détourner le fonctionnement du circuit ou contourner une protection. Ainsi dans ce chapitre nous présenterons la reproductibilité des effets des tirs laser.

Pour arriver à ses fins, l'attaquant peut aussi modifier la longueur d'onde, la puissance, la taille du spot et/ou la durée d'éclairement. Le fait de changer de longueur d'onde ou de puissance revient à changer de banc laser. Ainsi, pour éviter d'acquérir un nouveau banc laser qui est onéreux, il peut simplement modifier la durée d'éclairement du silicium. Dans ce chapitre, l'influence de ce paramètre sera étudiée afin de savoir s'il existe réellement une dépendance entre le nombre de fautes et la durée.

5.1. La reproductibilité

Pour sécuriser un circuit, il est utile de savoir si les effets des tirs laser sont reproductibles. Dans cette section, la reproductibilité sera étudiée pour des injections de fautes dans des tuiles CLB et BRAM. Pour cette étude, nous avons utilisé uniquement le banc permettant d'obtenir une taille de spot laser de 20 µm. Une zone mesurant 120 µm par 120 µm a été balayée lors de l'étude des tuiles CLB et une zone de 60 µm par 60 µm de côtés pour les tuiles BRAM. Le nombre de tirs laser par position a été fixé à 10 et 20 tirs laser par position XY respectivement pour les tuiles CLB et BRAM. Pour ces deux types de tuiles, nous avons utilisé la même valeur d'incrément spatial de 2 µm pour les déplacements sur les axes X et Y.

5.1.1. Définitions

Dans la suite de cette section, plusieurs termes concernant la reproductibilité seront utilisés. Nous allons tout d'abord définir ces termes.

Définition 1 Nous appellerons **"Reproductibles en nombre de tirs"** l'ensemble des positions ayant conduit systématiquement à des modifications de bits de la configuration.

Définition 2 Les positions **"Reproductibles en nombre de bits"** correspondent aux positions ayant conduit systématiquement à la même liste de bits erronés.
Il est possible d'avoir des positions reproductibles en nombre de tirs mais non reproductibles en nombre de bits. Le cas particulier où aucune faute n'est induite dans la configuration lors des différents tirs laser sera assimilé à des tirs reproductibles en nombre de tirs.

5.1.2. Reproductibilité dans les tuiles CLB

5.1.2.1 Reproductibilité des tirs laser

Pour connaître le taux de reproductibilité des tirs laser sur la configuration, il faut regarder la proportion de relectures modifiées par position XY lors de plusieurs tirs laser. Lors de cette campagne, plus de 37210 tirs laser ont été réalisés sur le FPGA étudié et les résultats sont présentés dans le Tableau 5.1. Plus de 96% des positions permettent de modifier à chaque tir la configuration du composant et près de 2% n'induisent aucune fautes, i.e. "*0 relecture fautée par position*". Au total, il est donc possible de dire que 98,04% des positions sont reproductibles en nombre de tirs.

Tableau 5.1 : Proportion de la reproductibilité en nombre de tirs exprimée en pourcentage pour la zone d'étude des tuiles CLB.

Nombre de relectures fautées par position	Nombre de coordonnées différentes	Pourcentage
0	74	1,99%
1	12	0,32%
2	9	0,24%
3	3	0,08%
4	9	0,24%
5	5	0,13%
6	9	0,24%
7	5	0,13%
8	12	0,32%
9	9	0,24%
10	3574	96,05%

5.1.2.2 Reproductibilité en nombre de bits

A - Cas des tirs laser conduisant systématiquement à des erreurs

Lorsque la configuration est systématiquement modifiée (ligne "*10 relectures fautées par position* " du Tableau 5.1), il est intéressant de savoir si les mêmes listes de bits sont modifiées pour les différents tirs. Les résultats obtenus sont regroupés dans le Tableau 5.2. Un peu plus d'un tiers des positions reproductibles en nombre de tirs permettent d'obtenir la même liste de bits modifiés à chaque tir laser. Ainsi, lorsque les positions sont reproductibles en nombre de tirs, leurs effets ne sont pas majoritairement reproductibles.

Tableau 5.2 : Répartition de la reproductibilité en nombre de bits pour des coordonnées XY reproductibles en nombre de tirs lors de l'étude de tuiles CLB.

Nombre de coordonnées XY de la zone d'étude		3721
Nombre de coordonnées XY différentes fautées		3574
Nombre de positions avec les mêmes effets pour tous les tirs	Total	1278
	Pourcentage	35,76%
Nombre de positions avec des listes de bits modifiés variables	Total	2296
	Pourcentage	64,24%

B - Cas des positions non-reproductibles en nombre de tirs

Lorsque les tirs laser ne provoquent pas systématiquement d'erreur, quel est le niveau de reproductibilité des effets ? La Figure 5-1 représente la répartition du nombre de listes différentes de bits modifiés selon le taux de reproductibilité en nombre de tirs. Sur cette figure, uniquement les taux compris entre 20% et 90% sont représentés. En effet, nous ne prenons pas en compte le cas où une seule configuration a été modifiée sur les 10 tirs réalisés (10% de reproductibilité en nombre de tirs) puisqu'une seule liste de bits modifiés est possible. De plus ce cas particulier ne représente que 0,32% des positions de tirs (12 positions de tirs sur plus de 3600).

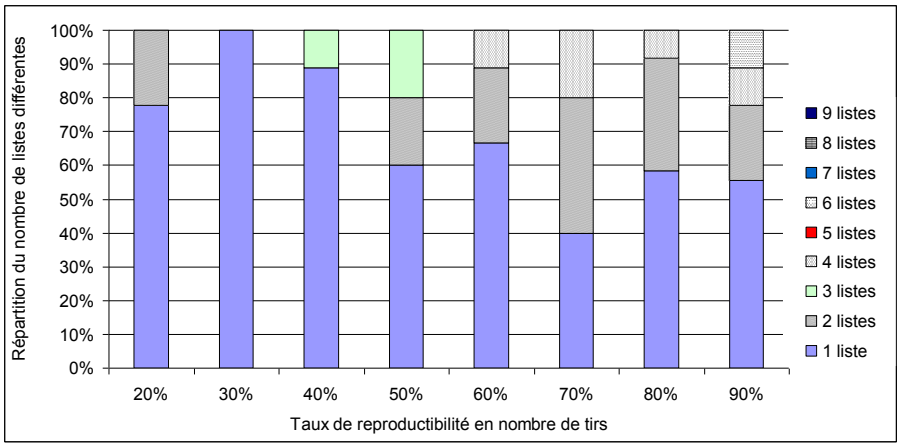


Figure 5-1 : Répartition du nombre de listes différentes en fonction du taux de reproductibilité en nombre de tirs.

Par exemple pour des positions de tir conduisant à des taux de reproductibilité en nombre de tirs de 50% (5 tirs générant des fautes sur les 10 tirs réalisés), on note que près de 60% de ces positions permettent de modifier la même liste de bits, 20% conduisent à 2 listes différentes et 20% conduisent à 3 listes. Par ailleurs, plus le taux de reproductibilité en nombre de tirs est élevé, plus il est difficile d'obtenir les mêmes ensembles de bits modifiés. Toutefois, dans la majorité des cas les listes de modifications sont identiques lors de plusieurs tirs à une même position.

5.1.3. Reproductibilité dans les tuiles BRAM

5.1.3.1 Reproductibilité des tirs laser

Les BRAM et leurs interconnexions sont d'autres éléments importants des FPGA lorsqu'elles sont utilisées par le circuit implanté. Lors de cette campagne, plus de 19220 tirs laser ont été réalisés. Plus de 30% des tirs laser sont sans effet lors des tirs sur ces catégories de tuiles (Tableau 5.3). Ce résultat diffère de celui obtenu pour l'étude des tuiles CLB et peut être expliqué par le positionnement du faisceau lumineux par rapport aux lignes-colonnes de la mémoire de configuration ou par la structure du point de mémorisation. En effet, des différences entre les dessins des tuiles CLB et BRAM ont été observées telles qu'une régularité moindre pour les tuiles CLB ou une taille de point mémoire nettement plus faible pour les tuiles BRAM (Chapitre 4). Ces différences technologiques peuvent expliquer les divergences de résultats obtenus vis-à-vis des tuiles CLB. Comme pour l'étude des tuiles CLB, il est cependant possible de dire que plus de 97% des positions sont reproductibles en nombre de tirs (i.e. lignes "0" et "20" configurations fautées).

Tableau 5.3 : Proportion de la reproductibilité en nombre de tirs exprimée en pourcentage pour la zone d'étude des tuiles BRAM.

Nombre de relectures fautées par position	Nombre de coordonnées différentes	Pourcentage
0	289	30,07%
1	2	0,21%
2	0	0,00%
3	0	0,00%
4	0	0,00%
5	0	0,00%
6	0	0,00%
7	2	0,21%
8	1	0,10%
9	1	0,10%
10	0	0,00%
11	0	0,00%
12	1	0,10%
13	1	0,10%
14	0	0,00%
15	0	0,00%
16	0	0,00%
17	2	0,21%
18	2	0,21%
19	14	1,46%
20	646	67,22%

5.1.3.2 Reproductibilité en nombre de bits

La reproductibilité en nombre de tirs sur des tuiles BRAM est du même ordre de grandeur que celle des tuiles CLB. En analysant les résultats présentés dans le Tableau 5.4, nous notons toutefois une différence sur la reproductibilité en nombre de bits. En effet, plus de 60% des positions reproductibles en nombre de

tirs permettent d'induire des listes d'erreurs identiques pour tous les tirs, alors que le taux de reproductibilité est de l'ordre de 35% lors de l'étude des tuiles CLB. Ces différences de résultats peuvent être expliquées par les différences de dessins observées entre les points mémoires des tuiles CLB et BRAM (section 4.2 du Chapitre 4).

Tableau 5.4 : Nombre de coordonnées XY en fonction de la reproductibilité dans le cas des tuiles BRAM.

Nombre de coordonnées XY de la zone d'étude		961
Nombre de coordonnées XY différentes fautées		646
Nombre de positions avec les mêmes effets pour tous les tirs	Total	400
	Pourcentage	61,92%
Nombre de positions avec des listes de bits modifiés variables	Total	246
	Pourcentage	38,08%

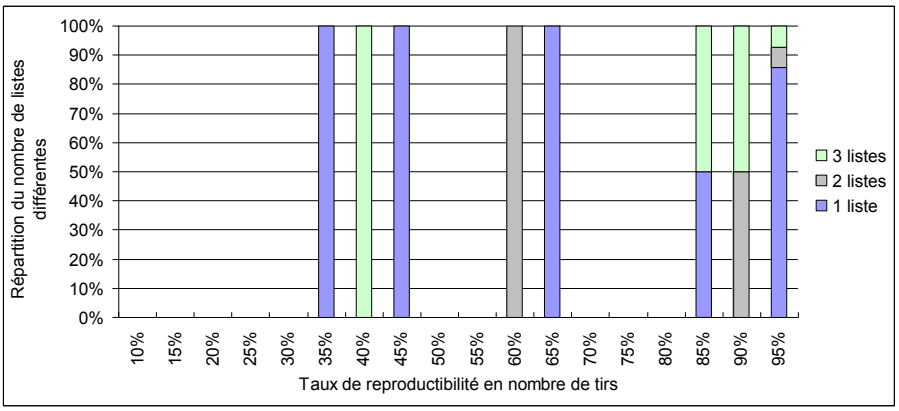


Figure 5-2 : Répartition du nombre de listes différentes en fonction du taux de reproductibilité en nombre de tirs (ne sont représentés que les taux obtenus).

En analysant le nombre de listes différentes obtenues pour des tirs laser possédant des taux de reproductibilité compris entre 10% et 95%, nous observons qu'il varie entre 1 et 3 listes différentes, comme représenté sur la Figure 5-2. Il faut noter sur cette figure que seules certaines valeurs de taux de reproductibilité en nombre de tirs ont été obtenues pendant les expériences, ceci explique la forme de l'histogramme. Contrairement aux résultats présentés pour les tuiles CLB, on observe une disparité du nombre de listes différentes en fonction du taux de reproductibilité en nombre de tirs. Pour certaines valeurs de reproductibilité, il est possible d'obtenir une seule liste d'erreurs alors que pour d'autres, on peut avoir jusqu'à 3 listes différentes. Cependant, la majorité des résultats montre que la même liste de modifications est principalement obtenue. Par exemple lorsque le taux de reproductibilité est de 95% (19 tirs conduisant à des erreurs sur les 20 tirs réalisés), plus de 85% de ces positions conduisent à la même liste de modifications.

5.1.4. Reproductibilité des tirs laser conduisant à des fautes uniques dans des tuiles CLBs

Dans le Chapitre 4, le cas particulier des fautes uniques induites par un tir laser en utilisant différentes tailles de spot laser a été présenté. Nous venons également d'introduire la reproductibilité des tirs laser, mais qu'en est-il pour ce cas particulier des fautes uniques ?

Lors de la campagne d'injection de fautes dans les tuiles CLB, 3% des positions de tir (116 sur 3721) ont conduit à des fautes uniques lors d'au moins un des tirs laser. Pour ces positions, les taux de reproductibilité sont très variables en nombre de tirs puisqu'ils varient entre 10% et 100% (Tableau 5.5). Cependant, plus de 64% de ces positions sont reproductibles en nombre de bits puisqu'elles ont permis de modifier la même liste (modification d'un seul bit de la configuration) lors des 10 tirs. Le taux de reproductibilité des tirs laser conduisant à des fautes uniques et donc à la même liste de bit modifié est nettement supérieur au cas général de l'étude (tirs reproductibles en nombre de tirs et en nombre de bits). En effet, un peu plus de 35% des positions de tirs permettent de modifier la même liste de bits fautés lors des 10 tirs (section 5.1.2) contre plus de 64% pour le cas particulier des fautes uniques. Ce résultat est à prendre en compte lors d'attaques ciblées cherchant à modifier un bit précis de la configuration.

Tableau 5.5 : Taux de reproductibilité en nombre de tirs pour des tirs laser conduisant à des fautes uniques.

Nombre de relectures fautées par position	Nombre de coordonnées différentes	Pourcentage
1	8	7,21%
2	6	5,41%
3	2	1,80%
4	7	6,31%
5	2	1,80%
6	4	3,60%
7	2	1,80%
8	4	3,60%
9	4	3,60%
10	72	64,86%

5.2. L'influence de l'énergie

Dans la section 2.5 du Chapitre 2, nous avons présenté certaines protections qui permettent de détecter un nombre défini de bits fautés. Si un attaquant arrive à induire un nombre de fautes plus important que la valeur maximale détectable alors il sera en mesure de contourner la contremesure implantée. Pour ce faire, il peut agir sur différents paramètres du banc laser : la longueur d'onde, la taille du spot, la puissance émise et la durée de l'impulsion. La longueur d'onde est propre à la diode laser utilisée et la taille du spot dépend du matériel à disposition. Le paramètre sur lequel il est plus aisé de réaliser des modifications est l'énergie du laser qui dépend de la puissance émise par la source laser et de la durée de l'impulsion. Cependant, la

plupart des bancs sont utilisés à leur puissance nominale pour obtenir un rendement maximal. Ainsi, l'attaquant pourra modifier la durée de l'impulsion, laquelle fait varier proportionnellement l'énergie à partir d'une durée supérieure à 10 ns (Figure 5-3).

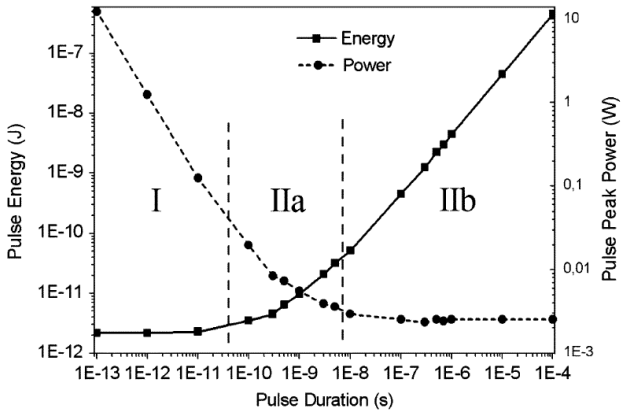


Figure 5-3 : Influence de la durée de l'impulsion sur la valeur de l'énergie émise par le laser [Douin 2006].

Pour nos campagnes, nous avons réalisé des tests statiques sur le circuit à base de logique combinatoire (Chapitre 3) afin de caractériser les effets de l'énergie sur le nombre d'erreurs de configuration. Plusieurs valeurs de durée d'éclairement allant de 500 ns à 2000 ns par pas de 500 ns ont été choisies. Une même zone mesurant 120 µm de hauteur et 240 µm de largeur, se situant dans les tuiles CLB a été étudiée en utilisant un pas d'incrément spatial de 2 µm. Ainsi, pour chaque durée d'impulsion étudiée plus de 7300 tirs laser ont été effectués.

5.2.1. Nombre de configurations modifiées

Pour connaître l'influence de l'énergie sur le nombre de fautes induites lors du changement de durée d'éclairement du FPGA testé, nous avons analysé le nombre de configurations modifiées lors de l'étude de la même zone pour différentes valeurs de durée. Les résultats ainsi obtenus sont regroupés dans le Tableau 5.6.

Tableau 5.6 : Nombre de configurations et de bits modifiés en fonction de la durée d'éclairement.

Durée de l'impulsion laser (ns)	500	1000	1500	2000
Nombre de tirs	7381			
Pourcentage de tirs sans effet	26,99%	24,16%	21,14%	14,01%
Nombre de bits différents modifiés	855	801	807	1059
Pourcentage d'augmentation	0%	-6,32%	-5,61%	23,86%

Le pourcentage de tirs sans effet diminue lorsque la durée de l'éclairement augmente montrant une influence de cette dernière. Cependant, le nombre total de bits modifiés diminue légèrement pour des valeurs comprises entre 500 ns et 1000 ns et reste quasi-constant pour des durées allant de 1000 ns à 1500 ns. Cette diminution peut être expliquée par le taux de reproductibilité des tirs laser de l'ordre de 95% et/ou par la multiplicité des fautes montrée précédemment dans le Chapitre 4. De plus, lorsque la

durée d'éclairement est supérieure à 1500 ns, le nombre total de bits modifiés augmente de près de 25% par rapport à 500 ns. Donc, à partir d'une certaine durée (ou énergie), toute augmentation de la durée a une influence assez importante sur le nombre de bits fautés.

Ainsi, la valeur de l'énergie influe sur le nombre de fautes générées dans la configuration du FPGA. En analysant les valeurs maximales du nombre d'erreurs induites par tir laser, nous remarquons des tendances différentes en fonction de la durée d'éclairement. Pour des valeurs d'impulsion laser de 500 ns et 1000 ns, une grande partie des tirs laser permettent d'induire un nombre de fautes allant jusqu'à 18 fautes par tir (Figure 5-4). Ce nombre tend vers des valeurs plus élevées de l'ordre de 26 fautes par tir pour des énergies plus fortes (1500 ns et 2000 ns). On note cependant que dans la majorité des cas un tir laser génère entre 1 et 10 fautes.

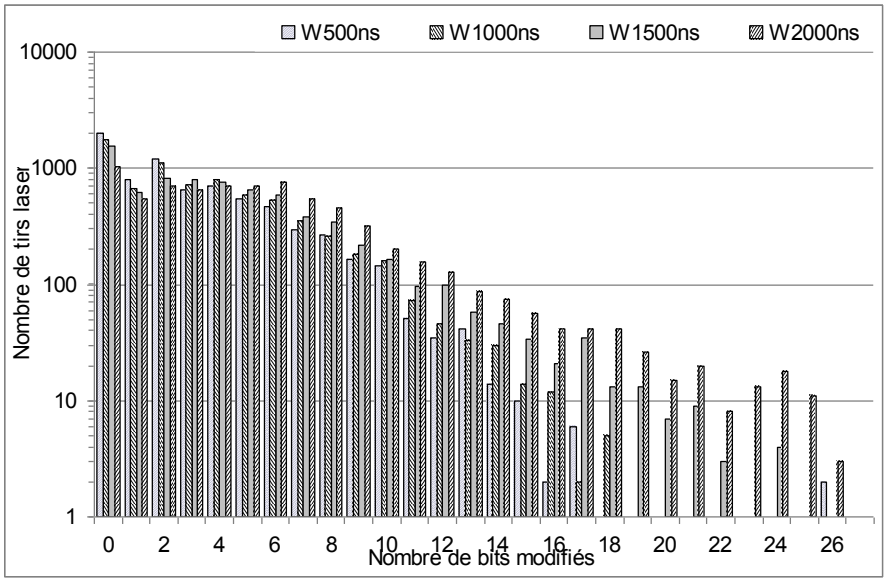


Figure 5-4 : Répartition du nombre de bits modifiés par tir en fonction de la durée de l'impulsion laser.

En représentant le nombre de fautes générées par position de tir en niveau de gris proportionnellement au nombre maximal d'erreurs, on observe sur la Figure 5-5 des différences de contraste nettement plus importantes pour les valeurs d'énergie élevées. Comme expliqué précédemment, on note que certains éléments sont plus sensibles que d'autres vis-à-vis de la valeur de l'énergie. En effet certaines zones possèdent des bits modifiés uniquement lors de tirs avec une énergie importante. Un exemple de ce résultat est présenté et entouré sur la Figure 5-5 montrant ainsi la nécessité d'appliquer une énergie plus élevée pour changer l'état de certains éléments configurant les CLB.

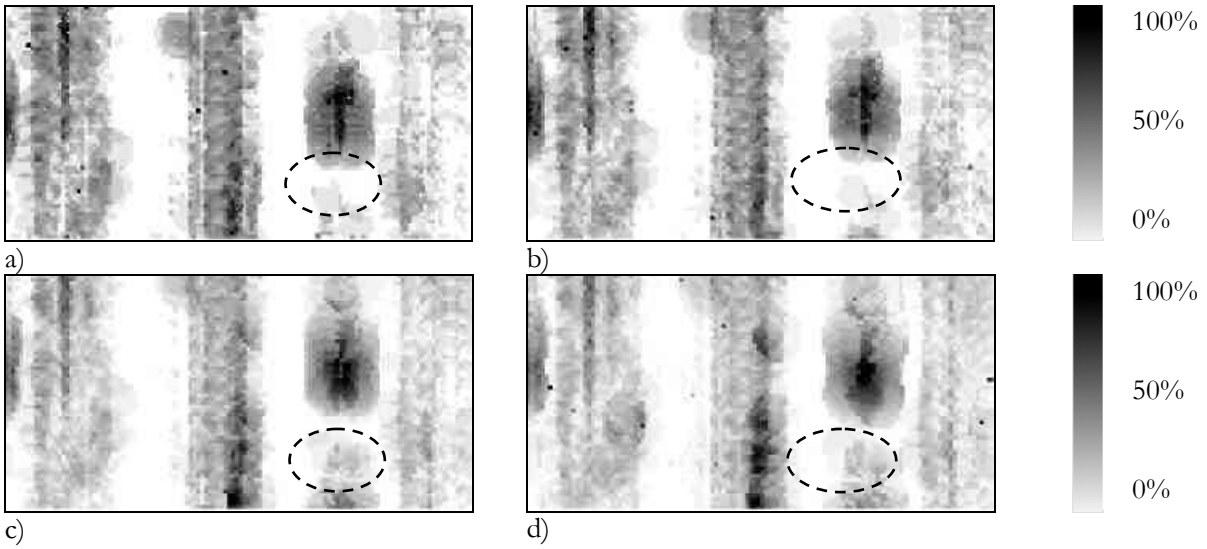


Figure 5-5 : Nombre de bits fautés représenté en niveau de gris pour des largeurs d'impulsion de 500ns (a), 1000ns (b), 1500ns (c) et 2000ns (d).

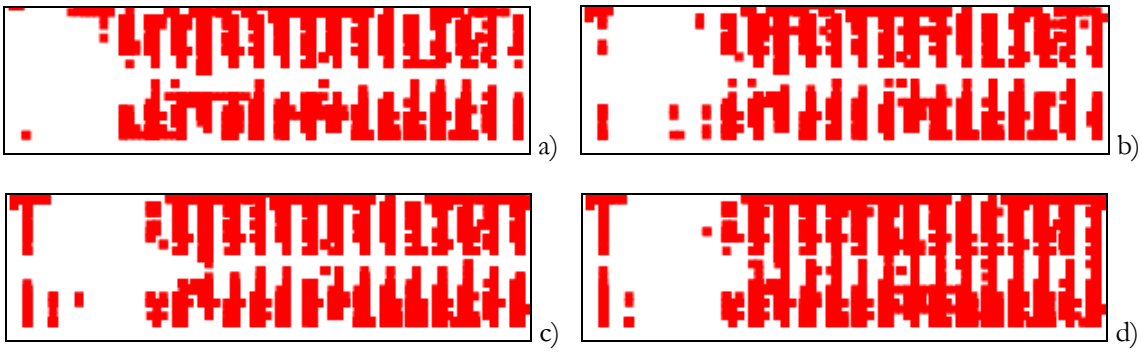


Figure 5-6 : Localisation des bits modifiés dans les CLB pour des largeurs d'impulsion laser de 500 ns (a), 1000 ns (b), 1500 ns (c) et 2000 ns (d).

En regroupant sur une même tuile CLB toutes les catégories de bits modifiés (Figure 5-6) où tous les points de couleur représentent des bits modifiés, nous notons que nous n'avons pas modifié les mêmes bits ni tous les bits de la tuile lors du changement de durée d'éclairement. Comme précédemment, certains bits ont uniquement été modifiés pour des durées d'impulsion importantes. Il existe par exemple une bande horizontale blanche au centre de la tuile CLB présente sur les Figure 5-6-a, b, c. Cette bande horizontale où aucun bit n'a été modifié disparaît sur la Figure 5-6-d signifiant que des bits ont vu leur état changé pour des valeurs d'énergie importante. Ce résultat confirme qu'il existe une différence de sensibilité des bits vis-à vis de l'énergie.

Tableau 5.7 : Tuiles CLB modifiées en fonction de la durée de l'impulsion.

500 ns		1000 ns		1500 ns		2000 ns	
Tuiles Modifiées	Nb de bits modifiés	Tuiles Modifiées	Nb de bits modifiés	Tuiles Modifiées	Nb de bits modifiés	Tuiles Modifiées	Nb de bits modifiés
2-15	6	3-15	8	23-14	654	3-15	6
9-15	10	12-15	11	**23-15**	**10878**	9-15	14
13-15	13	14-15	15	24-13	62	10-15	14
23-13	27	23-14	103	**24-14**	**464**	**23-15**	**589**
23-14	6	**23-15**	**8158**	**24-15**	**17671**	**24-14**	**14282**
23-15	**7218**	24-13	22			**24-15**	**590**
24-13	82	**24-14**	**64**				
24-14	**2**	**24-15**	**16648**				
24-15	**15399**						

Par ailleurs, en analysant les numéros les différentes tuiles modifiées ainsi que le nombre d'occurrence, nous remarquons que les numéros et le nombre de tuiles modifiées ne sont pas les mêmes malgré l'étude de la même zone (Tableau 5.7). En effet, pour une durée d'éclairement de 500 ns, le nombre de tuiles modifiées est plus important que pour une durée de 2000 ns. Ce résultat peut être dû au phénomène de reproductibilité des tirs laser, au positionnement du faisceau lumineux par rapport aux lignes/colonnes de la mémoire de configuration et/ou aux formes des zones de sensibilité des bits qui changent légèrement avec la durée. La taille et la forme de la zone de sensibilité en fonction de l'énergie seront étudiées dans la suite de ce chapitre.

Du fait de l'étude de la même zone avec plusieurs durées d'éclairement, il est cohérant d'obtenir des tuiles communes lors des différentes campagnes. Ces tuiles communes correspondent à celles ayant le plus de bits de configuration modifiés telles que les tuiles 23-15, 24-14, 24-15. Cependant, des tuiles non proches de la zone d'étude (i.e. tuiles 2-15, 3-15 ou 9-15) ont également été illuminées mais avec un nombre de fautes nettement plus faible. Ce résultat peut provenir de tirs laser sur des éléments communs aux différentes tuiles tels que des rails d'alimentation.

5.2.2. Nombre de fautes induites dans les éléments configurant les CLB

Maintenant que nous venons de discuter de l'influence de l'énergie sur le nombre de fautes induites dans les tuiles CLB, intéressons nous aux effets sur les différents éléments des tuiles CLB. Pour des durées d'impulsion laser de 500 ns à 1500 ns, aucun bit configurant la fonctionnalité des bascules n'a été modifié alors que ces bits sont présents dans la zone illuminée. Pour preuve, lors de l'utilisation de la durée de 2000 ns, 4 bits configurant ce type d'élément ont été modifiés. Ainsi pour modifier ce type de bit, une certaine énergie est a priori nécessaire et/ou dépend de la position exacte d'illumination.

De plus, une augmentation linéaire de la durée de l'impulsion laser ne permet pas d'obtenir un nombre de bits fautés proportionnel. Les résultats obtenus lors des différentes campagnes sont représentés dans le Tableau 5.8. Par exemple lors de tirs laser sur des bits configurant la fonctionnalité des LUT, un changement de durée d'impulsion laser de 500 ns à 1000 ns augmente de 25% le nombre d'erreurs

induites. Une augmentation de la durée d'éclairement de 500 ns à 1500 ns et de 500 ns à 2000 ns fait croître ce nombre de bits fautés respectivement de 175% et de 525%. La même valeur de seuil à partir de laquelle la durée est nettement plus influente sur le nombre de bits fautés a été observée pour les bits configurant le contenu des LUT. Cependant pour les multiplexeurs internes le seuil est différent et se situe pour une énergie plus importante (200 ns).

Les mêmes tendances de résultats ont été obtenues pour des bits configurant les interconnexions mais avec des tendances des pourcentages d'augmentation dépendant de la catégorie d'élément. Comme pour la logique, il existe une différence de sensibilité des bits vis-à-vis de l'énergie.

Tableau 5.8 : Nombre de bits modifiés dans le cas de bits configurant de la logique et des interconnexions en fonction de la durée d'éclairement.

Durée de l'impulsion laser		500 ns	1000 ns	1500 ns	2000 ns
Configuration des LUT	Nombre total de bits modifiés	28	35	77	175
	Pourcentage d'augmentation	0%	25%	175%	525%
Contenu des LUT	Nombre total de bits modifiés	4995	5727	7998	10591
	Pourcentage d'augmentation	0%	15%	60%	112%
Multiplexeurs internes	Nombre total de bits modifiés	1670	1670	1693	1988
	Pourcentage d'augmentation	0%	0%	1%	19%
Entrées des slices	Nombre total de bits modifiés	6248	7008	7465	9966
	Pourcentage d'augmentation	0%	12%	19%	60%
Sorties des slices	Nombre total de bits modifiés	2428	2751	2973	3850
	Pourcentage d'augmentation	0%	13%	22%	29%
Connexions de type "Double"	Nombre total de bits modifiés	2537	2700	2760	3279
	Pourcentage d'augmentation	0%	6%	9%	29%
Connexions de type "Hex"	Nombre total de bits modifiés	4430	4744	6238	7508
	Pourcentage d'augmentation	0%	7%	41%	69%
Connexions de type "Long"	Nombre total de bits modifiés	208	200	335	562
	Pourcentage d'augmentation	0%	-4%	65%	170%

Cette augmentation non-linéaire du nombre de bits fautés peut être expliquée par la répartition de l'énergie lumineuse sur la surface effective du spot laser. L'intensité lumineuse émise par le laser suit une gaussienne où l'énergie est maximale au centre du faisceau lumineux. Ainsi, si l'énergie augmente il est fortement possible que cette énergie se répartisse sur une plus grande surface.

En analysant la forme des zones de sensibilité, nous obtenons toujours une forme de croissant ou partie du disque pour des bits passant de '0' à '1' (Figure 5-7) lors de l'augmentation de la durée. Pour des bits passant de '1' à '0' une forme de disque est observée sur la Figure 5-8. Ces résultats sont cohérents avec ceux présentés dans le Chapitre 4. Par ailleurs, lors de l'augmentation de l'énergie les zones de sensibilité se déplacent selon l'axe des ordonnées croissantes de la zone d'étude. Ce déplacement peut être expliqué par le repositionnement de la table XY après chaque campagne qui entraîne une erreur de position de quelques micromètres malgré une régulation de la table en boucle fermée.

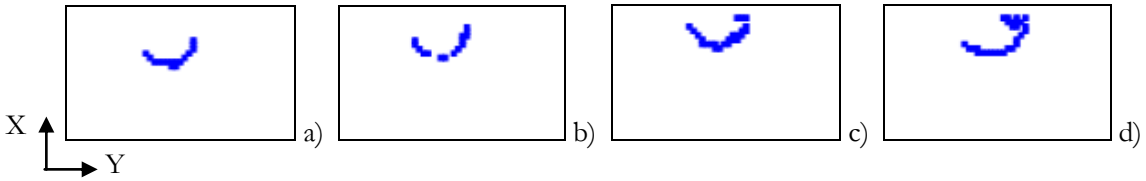


Figure 5-7 : Exemple de formes de la zone de sensibilité pour des largeurs d'impulsions de 500 ns (a), 1000 ns (b), 1500 ns (c) et 2000 ns (d) pour des bits passant de '0' à '1' configurant le contenu des LUT

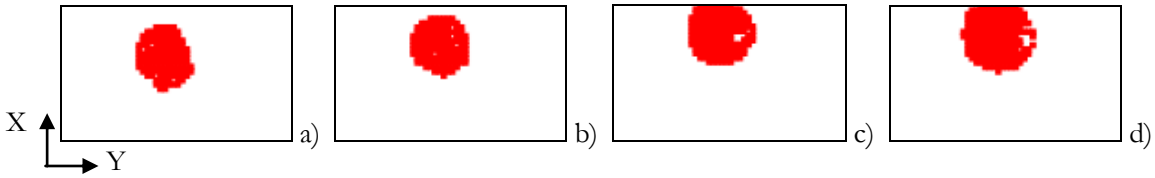


Figure 5-8 : Exemple de formes de la zone de sensibilité pour des largeurs d'impulsions de 500 ns (a), 1000 ns (b), 1500 ns (c) et 2000 ns (d) pour des bits passant de '1' à '0' configurant le contenu des LUT

Puisque les formes des zones de sensibilité sont similaires lors de l'augmentation de la durée d'éclairement d'après les formes observées sur la Figure 5-8, nous pouvons dire que ce ne sont pas les formes qui varient mais les surfaces effectives équivalentes qui augmentent avec l'énergie. En effet, la surface en nombre de points de la zone de sensibilité du bit représentée sur la Figure 5-7 (i.e. bit passant de '0' à '1') vaut 16, 15, 24 et 28 points pour des durées d'impulsion de 500 ns, 1000 ns, 1500 ns et 2000 ns. Ainsi lorsque la durée de l'impulsion augmente, la surface en nombre de points augmente également. Cependant, en mesurant le diamètre réel de la zone de sensibilité pour les différentes énergies utilisées, nous obtenons des valeurs de diamètres de 10, 22, 24 et 24 µm. Le diamètre réel de la zone de sensibilité varie donc très légèrement ce qui peut être dû aux taux de reproductibilité mais également aux repositionnement lors des différentes campagnes.

Pour des bits initialement à '1', nous obtenons les mêmes types de résultats que pour ceux passant de '0' à '1' avec un déplacement de la zone de sensibilité selon l'axe des ordonnées et une augmentation de la surface effective. La zone de sensibilité représentée sur la Figure 5-8 mesure 137, 137, 167 et 197 points respectivement pour des durées de 500 ns, 1000 ns, 1500 ns et 2000 ns. En conséquence, la surface effective de la zone de sensibilité augmente avec l'énergie. Par contre, ces augmentations de la durée d'éclairement font varier les diamètres des disques de quelques micromètres (2 à 6 µm).

En définitive l'augmentation de la durée d'éclairement ne fait pas varier le diamètre réel du spot laser mais répartit l'énergie sur une surface plus importante.

5.2.3. Cas particulier des fautes uniques

Dans les chapitres précédents, nous avons montré la possibilité d'obtenir des fautes uniques lors de tirs laser. Puisque la durée d'éclairement influe sur le nombre d'erreurs de configuration, il est fortement possible que le nombre de fautes uniques diminue lors de l'augmentation de la durée.

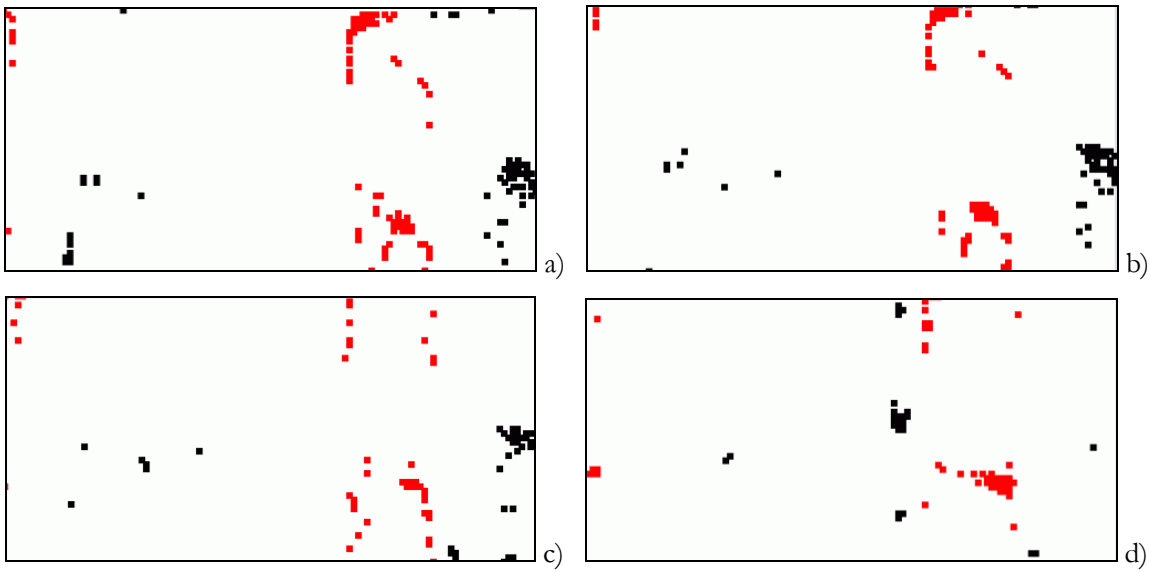


Figure 5-9 : Localisation des tirs laser ayant conduit à des fautes uniques pour des largeurs d'impulsion de 500 ns (a), 1000 ns (b), 1500 ns (c) et 2000 ns (d) dans le cas de bits configurant des éléments de logique (Contenu des LUT et Multiplexeurs internes)

La Figure 5-9 représente les différentes positions de tirs laser ayant conduit à des fautes uniques dans des éléments configurant de la logique. Tout changement de la durée d'éclairement a une influence sur le nombre de positions menant à des fautes uniques. En effet, lorsque nous augmentons la valeur de l'énergie, la proportion de fautes uniques diminue. Par exemple, lors de tirs laser sur des bits configurant le contenu des LUT, le nombre de positions induisant des fautes uniques diminue de 82 positions à 51 positions pour des durées d'éclairement comprises entre 500 ns et 2000 ns. Il en est de même pour les bits configurant les multiplexeurs internes qui est l'autre catégorie de bits pour laquelle des fautes uniques ont été obtenues. Par ailleurs, ce n'est pas parce que le nombre de positions de tir varie que le nombre de bits différents change. En effet dans la section 4.2 du Chapitre 4, nous avons montré que pour modifier un bit donné il faut éclairer une zone de sensibilité pouvant être illuminée à plusieurs positions de tir. Ces résultats peuvent être expliqués par la répartition du nombre de fautes qui augmente avec l'énergie comme présenté dans la section 5.2.1. De plus, comme pour le cas général de l'étude des fautes induites on note que des bits sont modifiés uniquement pour certaines énergies. Pour corroborer ce résultat de la sensibilité de certains bits face à l'énergie, nous allons regarder s'il existe des bits communs aux différentes campagnes.

Définition 3 Le nombre de bits présents lors des différentes campagnes à une même position de tir est appelé "*nombre de bits communs*".

Tableau 5.9 : Nombre de bits différents et nombre de bits communs modifiés lors de tirs laser conduisant à des fautes uniques pour de bits configurant de la logique en fonction des durées d'impulsion

Types d'éléments		Contenu des LUT	Multiplexeurs internes
Nombre de bits différents modifiés	Durée de 500 ns	16	8
	Durée de 1000 ns	10	8
	Durée de 1500 ns	19	7
	Durée de 2000 ns	11	7
Nombre de bits communs	Durée de 500 ns à 1500 ns	9	4
	Durée de 500 ns à 2000 ns	6	2

Le Tableau 5.9 regroupe les nombres de bits différents et communs modifiés par des tirs laser conduisant aux fautes uniques. Le nombre de bits différents configurant les multiplexeurs internes est constant à 7 ou 8 bits, indépendamment de la valeur de l'énergie étudiée. Bien que ce nombre de bits différents reste constant, la moitié des fautes uniques sont communes aux campagnes utilisant des durées comprises entre 500 ns et 1500 ns alors qu'un quart le sont pour l'ensemble des durées étudiées. Pour les bits configurant le contenu des LUT, le nombre de bits différents varie lors du changement d'énergie. La proportion de bits communs configurant cette catégorie de bits est semblable à celui des bits configurant les

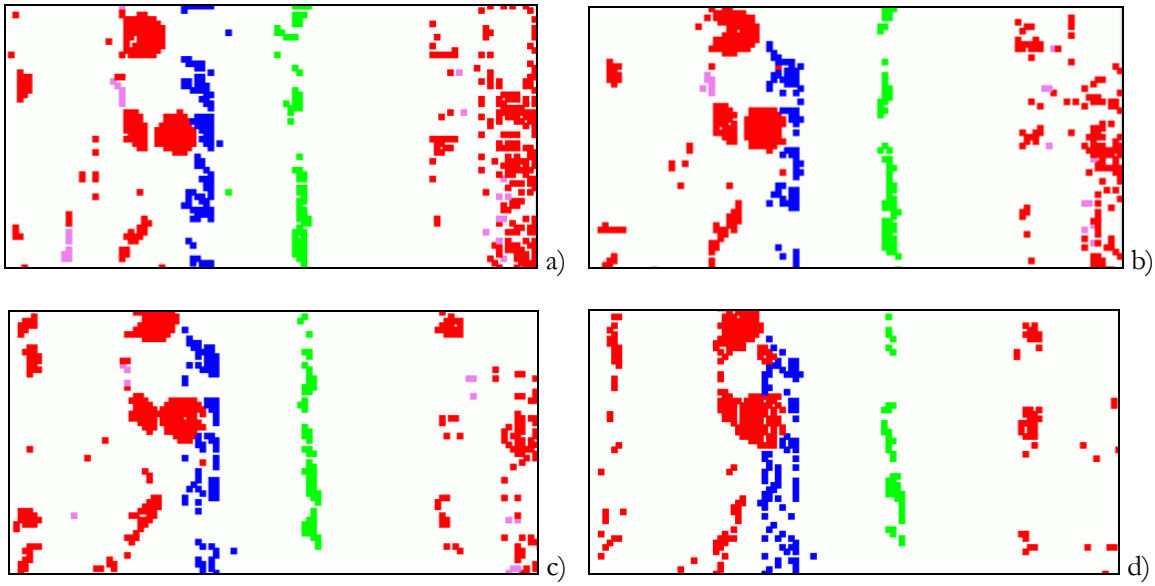


Figure 5-10 : Localisation des tirs laser ayant conduit à des fautes uniques pour des largeurs d'impulsion de 500 ns (a), 1000 ns (b), 1500 ns (c) et 2000 ns (d) dans le cas de bits configurant des éléments d'interconnexions

multiplexeurs internes avec environ la moitié des fautes communes aux 3 durées premières durées étudiées et un tiers à la campagne d'étude complète. Ainsi lors du changement de la durée d'éclairement, les bits modifiés ne sont pas tous modifiés pour ces différentes valeurs d'énergie. Ce résultat peut être expliqué

par le taux de reproductibilité et/ou par le positionnement du faisceau laser sur les points mémoires. Cette dernière hypothèse du positionnement est très probable puisque nous avons montré dans la section précédente qu'une augmentation de la durée a pour effet de mieux répartir l'énergie émise sur la surface du spot laser, rendant probable le fait de toucher d'autres bits de la configuration.

Des résultats similaires ont été obtenus dans le cas de bits configurant les interconnexions puisque le nombre de positions de tirs conduisant à des fautes uniques diminue lorsque la valeur de l'énergie augmente. Par exemple avec une durée d'impulsion de l'ordre de 2000ns, il n'existe plus de positions permettant de générer des fautes uniques dans les sorties des slices alors qu'il y en avait pour des valeurs de durée plus faibles (Tableau 5.10). Lors de tirs laser sur les éléments configurant les connexions Hex, le nombre de bits différents modifiés diminue avec une augmentation de la durée d'éclairement, excepté pour les connexions de type Double dont la proportion fluctue.

Tableau 5.10 : Nombre de bits différents et nombre de bits communs modifiés lors de tirs laser conduisant à des fautes uniques pour des bits configurant les interconnexions en fonction des durées d'impulsion

Types d'éléments		Slices		Connexions de type	
		Entrées	Sorties	Double	Hex
Nombre de bits différents modifiés	Durée de 500ns	41	12	18	16
	Durée de 1000ns	39	10	15	14
	Durée de 1500ns	38	8	19	12
	Durée de 2000ns	36	0	19	10
Nombre de bits communs	Durée de 500ns à 1500ns	25	4	12	11
	Durée de 500ns à 2000ns	15	0	11	9

Même si le nombre de bits différents configurant les connexions de type Double fluctue lors des variations de la durée de l'impulsion, on note cependant que le nombre de bits communs reste quasi-constant à 11 bits. Ainsi, les mêmes bits sont modifiés lors des différentes campagnes et certains sont modifiés uniquement de temps en temps à cause de la position du faisceau sur les points mémoires et/ou du taux de reproductibilité. Il en est de même pour les bits configurant les connexions de type Hex avec près de 10 bits communs quelle que soit la durée d'éclairement. Contrairement à ces deux catégories de connexions, les autres catégories (i.e. les entrées et les sorties des slices) sont nettement plus dépendantes de l'énergie. En effet, le nombre de bits communs varie lors de l'augmentation de l'énergie. Comme pour les bits configurant la logique, environ la moitié des fautes uniques sont des bits communs pour des durées comprises entre 500 ns et 1500 ns alors qu'environ un tiers le sont pour la campagne complète.

5.3. Conclusion

Dans ce chapitre, nous avons présenté la contrôlabilité des attaques laser. En particulier, nous nous sommes focalisé sur la reproductibilité des injections de fautes laser et de leurs effets, mais également sur l'influence de la durée d'éclairement sur le nombre erreurs induites dans la configuration du FPGA.

Nous avons montré que les effets de ces tirs n'étaient pas reproductibles à 100% tant en nombre de tirs qu'en nombre de bits modifiés. En effet, plusieurs tirs laser sur une même position XY du composant ne permettent pas d'obtenir forcément un nombre d'erreurs de configuration constant. En analysant la reproductibilité des différents types de bits configurant les tuiles CLB, nous avons toutefois remarqué que les tirs laser sur des bits configurant de la logique ou des interconnexions permettaient d'obtenir des taux de reproductibilité en nombre de tirs de l'ordre de 95%. Des taux similaires ont été obtenus lors de l'étude des tuiles BRAM.
Pour des positions de tir ayant conduit à des fautes uniques, le taux de reproductibilité en nombre de tirs et en nombre de bits, de l'ordre de 64%, est nettement supérieur au cas général de l'étude (35% pour les zones CLB).
En définitive, le taux de reproductibilité en nombre de tirs est très peu dépendant de la fonctionnalité des bits considérés puisqu'il est supérieur à 95%. Grâce à ce taux de reproductibilité élevé nous pouvons dire qu'il ne sera pas nécessaire de réaliser un grand nombre de tirs laser par coordonnée si on cherche seulement à provoquer des erreurs dans la configuration. Deux ou trois tirs laser peuvent être nécessaires par coordonnée pour toutefois obtenir des erreurs sur un maximum de positions. Si l'objectif est en plus de reproduire un motif d'erreurs donné, le nombre de tirs requis peut être nettement plus élevé, la reproductibilité en nombre de bits étant nettement plus faible.

Nous avons aussi montré dans ce chapitre que la durée de l'impulsion laser a une influence sur le nombre de fautes. En effet, lors de l'étude d'une même zone et pour 4 valeurs de durée différentes, le nombre total de bits fautés augmente assez régulièrement et à partir d'une certaine valeur d'énergie cette augmentation est nettement plus importante. De plus, des sensibilités différentes vis-à-vis de l'énergie ont été observées en fonction de la catégorie du bit. Pour preuve, les bits configurant la fonctionnalité des bascules ont uniquement été modifiés pour une durée d'impulsion de 2000 ns. Ainsi, il est assez difficile de définir la "meilleure valeur" permettant de modifier le nombre de bits désiré dans la configuration, ceci à cause des seuils mais également du fait de la multiplicité des fautes par tir.

Un autre résultat important de cette étude est que l'énergie n'influe pas ou très peu sur le diamètre réel du spot lumineux. La surface effective de la zone de sensibilité des bits varie lors de l'augmentation de l'énergie du fait que cette dernière se répartit sur une surface plus importante.
Enfin, le cas des fautes uniques a également été étudié dans ce chapitre afin de savoir si l'énergie a également une influence sur ce cas particulier. Le nombre de positions conduisant à cette catégorie de fautes diminue lorsque la durée de l'impulsion augmente du fait de la répartition de l'énergie plus étendue sur la surface du spot. Cependant pour des énergies plus importantes, de nouvelles zones générant ce type de fautes ont également été touchées. L'augmentation de l'énergie peut donc permettre la génération de fautes uniques nouvelles, même si le nombre de positions conduisant à des fautes uniques diminue.

Dans le prochain chapitre, nous allons montrer l'effet des erreurs générées lors des campagnes d'attaques dynamiques sur un crypto-processeur sécurisé contre les attaques par fautes.

Chapitre 6. Cas d'étude : un crypto-processeur sécurisé

Précédemment, nous avons caractérisé les effets des tirs laser et des surtensions sur la configuration du FPGA et sur les données stockées dans les bascules utilisateurs. Nous avons montré la sensibilité des différents éléments constituant les tuiles CLB face aux injections de fautes ainsi que les modifications possibles des connexions. Toutes les campagnes de caractérisation ont été réalisées en statique. Dans ce chapitre, nous présentons les résultats de campagnes dynamiques sur un crypto-processeur sécurisé contre les attaques par injection de fautes. Dans une première partie, nous présenterons le crypto-processeur et le mécanisme de sécurisation. L'implantation de ce circuit dans le FPGA sera introduite dans une seconde partie avec les contraintes de placement routage et les différentes ressources utilisées. Enfin, dans la dernière partie, les résultats des campagnes seront présentés ainsi qu'une nouvelle contremesure suite aux résultats obtenus.

6.1. Le fonctionnement du crypto-processeur AES

Au sein du laboratoire TIMA a été développé un crypto-processeur sécurisé contre les attaques par fautes. Ce crypto-processeur AES dont la structure détaillée est présentée dans [Maistri 2007], utilise les 2 fronts d'horloge pour effectuer les calculs, comme les mémoires DDR (Double-Data-Rate). Dans les parties suivantes, nous allons présenter brièvement l'algorithme de chiffrement AES, l'architecture du crypto-processeur ainsi que le principe de la redondance DDR.

6.1.1. Le chiffrement AES

L'algorithme de chiffrement AES (*Advanced Encryption Standard*) aussi connu sous le nom de *Rijndael*, du nom de ses deux concepteurs Vincent Rijmen et Joan Daemen, est un algorithme de chiffrement symétrique par bloc. Un chiffrement symétrique signifie que la même clef est utilisée pour les opérations de chiffrement et de déchiffrement. Les algorithmes de chiffrement par bloc utilisent une clef pour transformer des blocs de texte clair de longueur fixe (typiquement 8 ou 16 octets) en des blocs de texte

chiffré de même longueur. L'AES, adopté en 2001, remplace le DES (*Data Encryption Standard*) qui était le standard dans les années 1970 et qui devient obsolète.

Cet algorithme de chiffrement permet de chiffrer des données de 128 bits (16 octets) avec plusieurs tailles de clefs : 128, 192 ou 256 bits. Les données sont représentées sous forme d'une matrice 4x4 octets nommée *"État"*. Les opérations de chiffrement et de déchiffrement sont découpées en rondes (ensemble d'opérations à répéter) dont le nombre dépend de la taille de la clef. Ce nombre de rondes vaut 10, 12 ou 14 respectivement pour des clefs de 128, 192 et 256 bits.

Les opérations présentes dans les rondes sont au nombre de 4 : ***SubBytes, ShiftRows, MixColumns*** et ***AddRoundKey***. En fonction du numéro de ronde certaines opérations ne sont pas réalisées telles que l'opération **MixColumns** dans la dernière ronde (Figure 6-1).

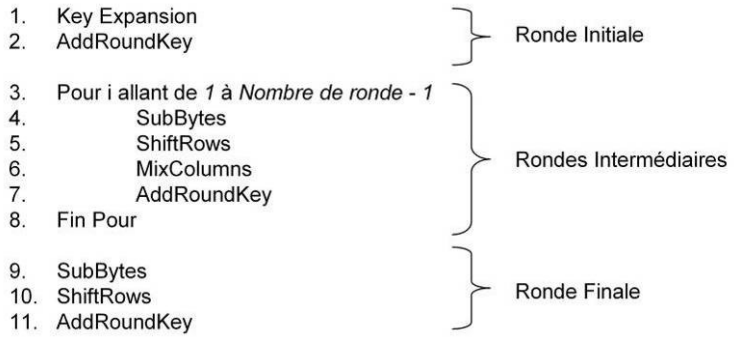


Figure 6-1 : Pseudo code du chiffrement AES

La fonction ***KeyExpansion*** permet à partir de la clef de chiffrement initiale de générer des sous-clés qui seront utilisées lors des différentes rondes.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 6-2 : Valeurs de substitution de l'octet xy codé en hexadécimal

La fonction ***SubBytes*** est une transformation non linéaire définie dans la Figure 6-2, modifiant chaque octet de l'état courant en utilisant des tables de substitution appelées ***S-box***. Par exemple si l'octet en

entrée de la S-box vaut *0x53* alors la valeur de substitution correspond à l'intersection de la ligne 5 et de la colonne 3 et la valeur en sortie de la S-box vaut *0xED*.

Dans l'opération ***ShiftRows***, les octets des trois dernières lignes de la matrice d'état sont permutés circulairement avec différentes valeurs de décalage. Le décalage de l'octet dépend de l'indice de la ligne. Ainsi, les lignes 2, 3 et 4 de l'état sont permutées respectivement de 1, 2 et 3 octets.

MixColumns est une transformation multipliant chaque colonne de l'état par un polynôme donné.

Dans la transformation ***AddRoundKey***, la sous-clé de ronde calculée lors de l'opération ***KeyExpansion*** est ajoutée à l'état en utilisant un *"Ou Exclusif"*.

6.1.2. L'architecture du crypto-processeur implanté

Le crypto-processeur implanté dont la structure est présentée en détail dans [Mangard 2003], est décomposé en trois blocs principaux : le bloc de contrôle, le bloc de génération des clefs de ronde et le chemin de chiffrement. Le chemin de chiffrement est l'élément le plus gros et le plus important de l'architecture que nous allons présenter brièvement.

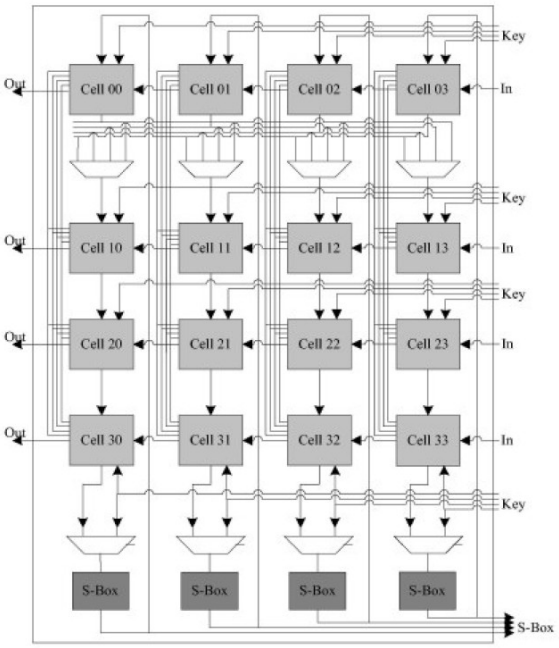


Figure 6-3 : Architecture du chemin de données [Mangard 2003]

La Figure 6-3 présente la structure du chemin de chiffrement, laquelle est régulière et très proche de la matrice d'état. Le chemin de chiffrement contient 16 cellules de données contenant chacune un des octets de l'état, ainsi que 4 S-Box.

Les S-Box du crypto-processeur implanté dans le composant testé, sont conçues sur une architecture pipeline à deux étages utilisant des opérations dans $GF((2^4)^2)$ [Mangard 2003]. La réduction du nombre de S-Box permet une optimisation du coût d'implantation du circuit; la réduction à 4 S-box permet d'obtenir un bon compromis entre le coût et la vitesse de chiffrement. Les entrées des S-Box sont connectées aux

cellules de la dernière ligne. Lors d'une ronde, les lignes se décalent successivement vers le bas pour entrer dans les S-Box. Après 5 cycles d'horloge la dernière ligne a été substituée ; au 6ème cycle l'état se retrouve à nouveau entièrement dans les cellules de données et a subi les transformations "*ShiftRows*" et "*MixColumns*".

6.1.3. La redondance temporelle et le DDR

Dans la littérature, on trouve des solutions pour contrecarrer les injections de fautes en implantant différents types de redondance (Chapitre 2). La répétition des calculs sous différentes formes est l'une des solutions communes et nécessite uniquement au niveau matériel l'ajout de quelques registres pour la sauvegarde des résultats et l'implantation de comparateurs. Cependant, la redondance réduit la vitesse de calcul d'une donnée car il est nécessaire d'effectuer plusieurs fois l'opération avant d'obtenir le résultat.

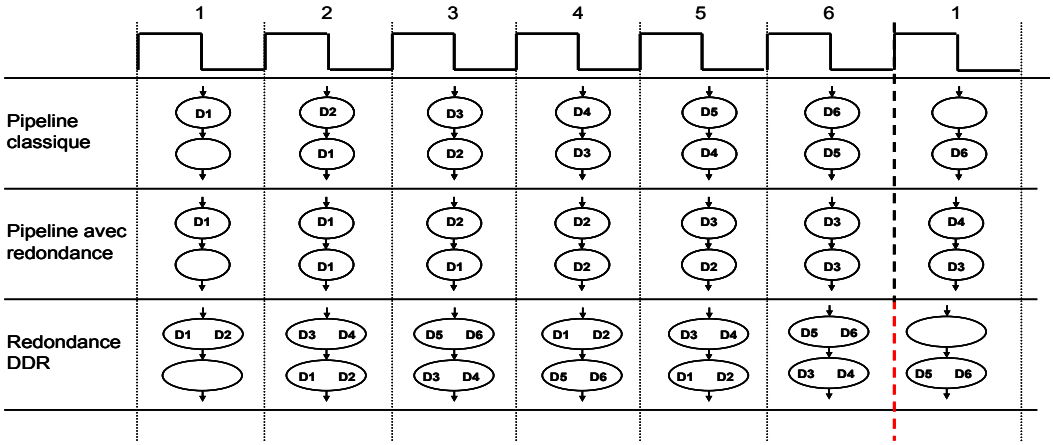


Figure 6-4 : Schéma temporel d'un pipeline régulier, d'un pipeline avec redondance, et de la redondance DDR [Maistri 2008]

La Figure 6-4 compare les temps d'exécution de deux solutions de redondance temporelle dans une implantation pipeline : le pipeline avec redondance et la redondance DDR (*Double Data-Rate* [Maistri 2008]). Un pipeline avec redondance temporelle consiste à exécuter deux fois de suite la même opération pour vérifier la bonne exécution [Wu 2001]. Avec l'approche DDR, les deux fronts d'horloge sont employés, ce qui permet deux opérations par cycle. De plus, à l'intérieur d'une ronde, les opérations similaires sont éloignées le plus possible afin de réduire la probabilité d'un effet identique d'une même perturbation sur les deux opérations.

En comparant la structure pipeline classique avec la structure DDR, on note que le pipeline est deux fois moins "rapide" que la structure DDR pour une même fréquence d'horloge en l'absence de doublement des calculs. Il en est de même pour le pipeline avec redondance qui est deux fois moins "rapide" que l'approche DDR avec redondance temporelle. Ainsi, en appliquant ce principe DDR sur un algorithme AES, il est possible de réaliser une ronde en 3 cycles d'horloge (6 fronts : 3 montants et 3 descendants) au lieu des 6 cycles typiques. De ce fait, une ronde de vérification peut être réalisée pour arriver aux 6 cycles classiques. L'objectif principal de cette structure employant la redondance DDR pour l'algorithme de

chiffrement n'est pas de réduire le temps de calcul, mais de proposer une contremesure efficace contre les injections de fautes. A la fin des 6 cycles d'horloge (fin de la ronde), une comparaison est effectuée entre les résultats de la ronde "normale" et de la ronde de vérification et une alarme est activée si les résultats diffèrent.

6.2. L'implantation du circuit dans le FPGA

6.2.1. Le circuit implanté

Le circuit implanté dans le FPGA Virtex-II xc2v1000 est constitué d'une UART (Universal Asynchronous Receiver/Transmitter) afin de dialoguer par une liaison série avec la carte mère, du crypto-processeur sécurisé AES (section 6.1) et d'une primitive Capture_Virtex2 permettant de lire le contenu des bascules. Le fonctionnement du circuit est cadencé par une horloge de fréquence 14,7456Mhz, générée par la carte mère. Grâce à cette gestion d'horloge par la carte mère, il est possible d'agir sur le fonctionnement du circuit en arrêtant l'horloge. Une liaison de type série a été choisie pour réduire le nombre de connexions sur le composant. Les instructions envoyées par cette liaison permettent d'écrire et de lire différents registres internes tels que les registres de clefs, de messages, de chiffrés (résultats du chiffrement) et d'erreur mais également de remettre à zéro ces différents registres. De plus, des commandes de chargement de la clef et du message ou de lancement du chiffrement ont été implantées pour le fonctionnement du crypto-processeur.

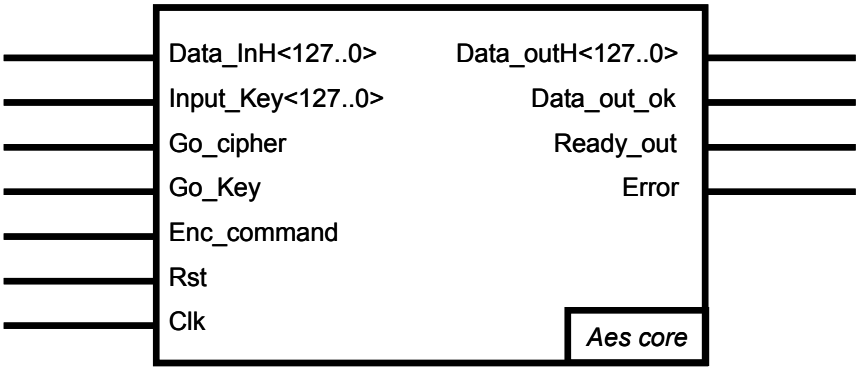


Figure 6-5 : Interface du crypto-processeur sécurisé

Le crypto-processeur AES développé au laboratoire TIMA possède différentes entrées-sorties (Figure 6-5) dont les fonctionnalités sont définies dans le Tableau 6.1. Pour lancer le chiffrement, il faut activer au préalable les entrées *Go_Key* puis *Go_Cipher* indiquant respectivement que la clef et le message sont chargés. Dès que le crypto-processeur est en cours d'utilisation, un signal d'occupation *Ready_out* passe du niveau haut au niveau bas et ce signal est utilisé comme signal de synchronisation. Une fois le chiffrement terminé, le signal *Ready_out* change d'état (passage de '0' à '1') ainsi que la sortie *Data_out_ok* indiquant qu'il est alors possible de lire le résultat du chiffrement (*Data_outH<127..0>*) et le signal d'erreur (*Error*). Le signal d'erreur est activé (passage de '0' à '1') lorsqu'une faute de chiffrement est détectée lors d'une des

rondes. Si des fautes sont induites dans le contrôleur, le signal d'erreur est également activé et une remise à zéro du crypto-processeur est effectuée. De plus pour remettre à zéro le signal d'erreur, il est nécessaire d'activer une remise à zéro du circuit complet.

Tableau 6.1 : Signaux utiles pour le fonctionnement du crypto-processeur

Nom du signal	Entrée / Sortie	Taille en bits	Fonctionnalité
Data_InH	Entrée	128	Donnée à chiffrer
Input_Key		128	Clef de chiffrement
Go_cipher		1	A activer lorsque l'on veut lancer un chiffrement
Go_Key		1	A activer dès que la clef est chargée
Enc_command		1	Permet de choisir entre un chiffrement et un déchiffrement
Rst		1	Remise à zéro du crypto-processeur
Clk		1	Horloge de fonctionnement du circuit
Data_outH	Sortie	128	Résultat du chiffrement
Data_Out_ok		1	Actif dès que la lecture du résultat est possible
Ready_out		1	Indique l'occupation du crypto-processeur
Error		1	Indique si une faute a été détectée

Pour lire le contenu des bascules utilisateurs, il faut instancier la primitive Capture_Virtex2 et générer les signaux utiles à cette primitive. Le processus de relecture du contenu des bascules est activé en passant de '0' à '1' l'entrée "CAP" de la primitive. Au prochain front montant de l'entrée "CLK", toutes les valeurs des registres des tuiles CLB et IOB seront stockées dans la mémoire de configuration. Ces valeurs pourront ensuite être relues en effectuant une relecture classique de la configuration. Si le signal "CAP" est actif durant plusieurs cycles d'horloge "CLK", alors les valeurs des registres stockées dans la configuration seront mises à jours à chaque front montant d'horloge. Dès que la relecture de la configuration est effectuée, la lecture des valeurs des bascules utilisateurs se fait en utilisant le fichier d'allocation logique.

Ce circuit constitué de l'UART, du crypto-processeur sécurisé et de la primitive Capture_Virtex2, lorsqu'il est implanté dans le FPGA testé, utilise 1740 slices sur les 5120 disponibles (33%) pour cette version de composant ainsi qu'un total de 19% de bascules utilisateurs (1969 sur 10240 disponibles).

6.2.2. Les contraintes de placement routage

Pour conduire efficacement les campagnes d'attaque par tirs laser en boîte noire, il faut réaliser un balayage spatial de toute la puce du composant testé. Comme indiqué précédemment dans le Chapitre 3, la surface de la puce représente près d'un centimètre carré. Ainsi, en utilisant un pas d'incrément spatial d'un dixième de la taille du spot, soit par exemple 2 µm pour le spot de 20 µm de diamètre théorique, il faut plusieurs mois (6 mois) pour balayer complètement la puce. Par ailleurs, il ne suffit pas d'injecter la faute de manière spatiale mais également de manière temporelle. Ainsi, il sera nécessaire de balayer la puce spatialement et temporellement augmentant d'autant la durée de la campagne.

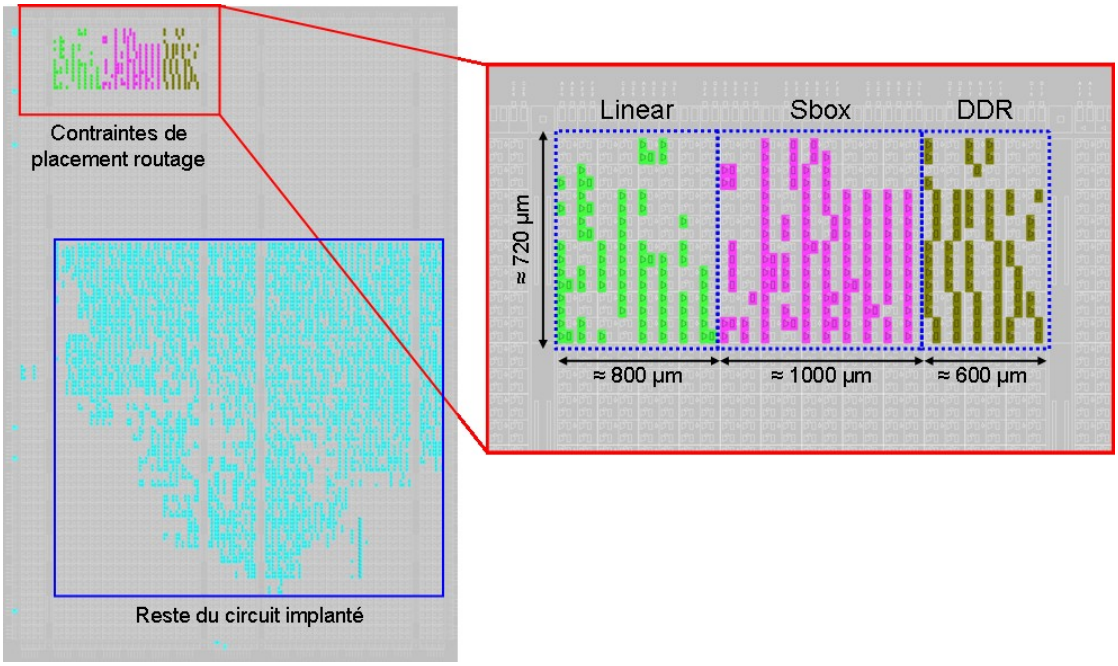


Figure 6-6 : Implantation du crypto-processeur dans le FPGA et localisation des partitions utiles pour les attaques

L'architecture du crypto-processeur implanté est symétrique et utilise plusieurs fois les mêmes éléments tels que les S-box, la partie linéaire et la contremesure DDR. Pour réduire la durée des campagnes, nous avons décidé de contraindre le placement-routage de ces trois éléments de la dernière colonne de l'état. Nous nous intéressons uniquement à une colonne de la matrice d'état du fait de la symétrie de la structure. Grâce aux contraintes, la zone d'étude se réduit à une zone de largeur 2400 µm et de hauteur 720 µm (Figure 6-6) au lieu des 10,6 mm de largeur et 9,7 mm de hauteur. La partie linéaire mesure 800 µm par 720 µm, la zone contenant la S-Box a pour dimensions : 1000 µm de largeur et 720 µm de hauteur, et la protection DDR de la dernière colonne de la matrice mesure 600 µm par 720 µm.

6.3. Les chemins d'attaque

6.3.1. Attaque de Piret-Quisquater

Suite à la publication [Skorobogatov 2002], un grand nombre de recherches sont faites sur des attaques théoriques de l'AES en exploitant les fautes induites par des tirs laser. Une des attaques la plus connue est celle de Piret-Quisquater [Piret 2003] que nous allons présenter brièvement.

Cette attaque consiste à analyser les chiffrés fautés lorsque la faute est injectée dans un octet juste avant opération "*MixColumns*" de la ronde 8 dans le cas d'un AES-128. La Figure 6-7 présente la propagation de l'erreur lors des différentes opérations. L'opération "*MixColumns*" va propager l'erreur dans une colonne entière de la matrice d'état, modifiant ainsi 4 octets. Par la suite, l'opération non-linéaire "*SubBytes*" est réalisée et les octets sont déplacés par l'opération "*ShiftRows*". Suite à ces différentes opérations, nous

avons toujours 4 octets erronés mais dans différentes colonnes de la matrice. La dernière opération "*MixColumns*" aura pour effet de propager les erreurs dans la matrice complète et de modifier les 16 octets.

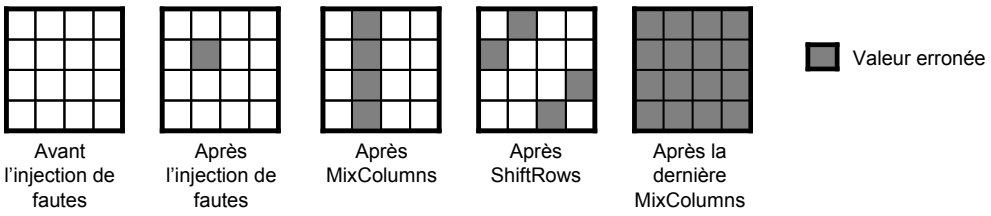


Figure 6-7 : Propagation de l'erreur lors de l'attaque de Piret-Quisquater

En exploitant cette attaque, il est possible d'extraire 16 candidats par faute injectée, à savoir une pour chaque octet de clef. Sous certaines conditions favorables telles qu'une faute reproductible, uniquement deux résultats erronés sont nécessaires pour la clef entière de chiffrement.

6.3.2. Conditions expérimentales et méthodologie utilisée

6.3.2.1 Généralités

Pour attaquer le crypto-processeur implanté dans le FPGA testé, nous allons utiliser les techniques d'injections de fautes par surtensions et par tirs laser. Pour ces campagnes, nous utilisons les vecteurs de test (clef, message) définis par [FIPS PUB 197] :

- Clef = 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
- Message = 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
- Chiffré = 69 C4 E0 D8 6A 7B 04 30 D8 CD B7 80 70 B4 C5 5A

Ainsi en utilisant ces vecteurs, nous avons une bonne connaissance des résultats intermédiaires pour réaliser de la cryptanalyse. Cependant les résultats obtenus dans la suite de ce chapitre sont propres aux vecteurs utilisés et pourront donner juste une indication sur les faiblesses potentielles.

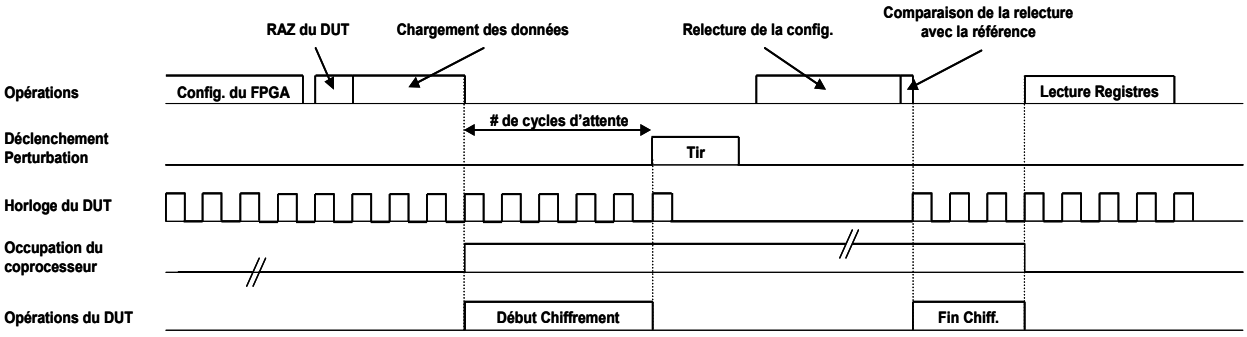


Figure 6-8 : Diagramme temporel de principe d'une séquence d'attaque réalisée – attaques par surtensions et par tirs laser

La Figure 6-8 décrit le principe de la séquence de test d'un point de vue temporel. L'ordinateur envoie la configuration du FPGA sous test à travers l'interface DIO, ainsi que les signaux utiles à la carte mère pour initialiser le circuit implanté (remise à zéro du circuit et chargement des données: instant d'injection de la

faute, clef, message, lancement du chiffrement, etc.). Dès que le chiffrement est lancé, la carte mère compte le nombre de fronts montants pour stopper l'horloge au moment désiré (nombre de cycles d'attente) contrairement à [Pouget 2008] qui ne gère pas l'horloge du circuit testé. Cet instant d'arrêt de l'horloge est également envoyé au générateur d'impulsion par un bus GPIB afin de déclencher l'injection de fautes sur le dernier front montant (attaques par surtensions et par tirs laser) ou descendant (attaques par surtensions). Dès que l'injection de fautes est effectuée, l'envoi des commandes de relecture de la configuration est possible et la comparaison avec le fichier de référence effectuée. Dans [Pouget 2008], la relecture de la configuration est réalisée une fois le chiffrement terminé et il est quasiment impossible de connaître précisément l'effet de l'injection de fautes dans les bascules utilisateurs. Lorsque la comparaison est réalisée, l'ordinateur envoie l'ordre de remise en fonctionnement du circuit en redémarrant l'horloge pour terminer le calcul de chiffrement. La lecture du contenu des différents registres tels que le résultat du chiffrement, les registres de clefs, de message ainsi que le signal de détection d'erreurs sont possibles une fois le chiffrement terminé.

6.3.2.2 Campagnes par surtensions

Lors des campagnes d'injections de fautes par application de surtensions, plusieurs valeurs de durée d'impulsions et d'amplitude ont été utilisées. La première série de campagnes consiste à appliquer des surtensions comprises entre 5 volts et 80 volts au moment du front montant de l'horloge. Dans un souci d'optimisation du temps de campagne et d'exploitation des résultats, nous avons décidé de diviser cette première campagne en deux sous-campagnes avec des gammes de tensions différentes. La première sous-campagne (**S1-1**) consiste à étudier les effets de surtensions ayant une amplitude comprise entre 5 volts et 45 volts, la seconde (**S1-2**) utilise des amplitudes entre 45 volts et 80 volts. Des surtensions seront appliquées sur les fronts descendant de l'horloge avec des amplitudes allant de 45 volts à 80 volts (**S2**) lors la seconde campagne. Pour les deux campagnes (**S1** et **S2**), les surtensions seront appliquées avec plusieurs durées allant de 10 ns à 100 ns par pas de 10 ns et ceci durant les cycles 49 à 60 (2 rondes complètes : 2 rondes normales et 2 rondes de vérification).

6.3.2.3 Campagnes par tirs laser

Pour les campagnes d'attaque par tirs laser, nous avons réalisé les injections de fautes uniquement lors de l'avant dernière ronde, c'est-à-dire pendant les cycles d'horloge 49 à 54 pour reproduire partiellement l'attaque de Piret-Quisquater. En effet dans [Piret 2003], l'attaque d'un crypto-processeur AES est présentée avec des injections de fautes lors de l'avant dernière et dernière opération "*MixColumns*". La campagne que nous avons conduite, consiste à balayer une zone d'étude de 500 µm de hauteur et de 600 µm de largeur regroupant une partie des S-Box et de la partie linéaire du calcul de la ronde. Pour cette étude, nous avons choisi d'utiliser le spot laser de diamètre 20 µm, une durée d'éclairement de 1000 ns et un pas de déplacement spatial de 15 µm selon les 2 directions X et Y. Puisque l'horloge est stoppée permettant d'injecter la faute uniquement lors d'un seul cycle d'horloge, la durée de l'impulsion laser a peu d'importance sur le déroulement séquentiel. Dans [Pouget 2008], la durée de l'impulsion laser a une

importance car elle est limitée par la valeur de la période. La valeur de déplacement de 15 µm a été utilisée afin d'obtenir un recouvrement des zones éclairées. Sur le composant testé, plus de 1400 tirs laser par cycle d'horloge ont été appliqués soit plus de 8600 tirs pour la campagne complète qui a duré près d'une semaine.

6.3.3. Définitions

Pour caractériser les effets des injections de fautes sur le crypto-processeur sécurisé AES-DDR, nous définissons 5 catégories de perturbations générées lors du chiffrement. Pour les catégories 2 à 5, la configuration du composant (bascules utilisateurs et/ou configuration de l'architecture) a été modifiée par l'injection de fautes.

Catégorie 1 : On appelle **"Tir sans effet"** ou **"Surtension sans effet"**, toute perturbation ne modifiant pas la configuration (bascules utilisateurs et/ou configuration de l'architecture) du composant.

Catégorie 2 : Toute faute dans la configuration permettant d'obtenir le chiffré espéré sans que l'erreur ne soit détectée est appelée **"Faute silencieuse"**.

Catégorie 3 : Lorsqu'une faute est injectée lors du chiffrement et que cette faute permet d'obtenir le bon chiffré alors que la détection d'erreur est activée, on parle de **"faux positif"**.

Catégorie 4 : On appelle **"Erreur détectée"**, toute faute conduisant à un chiffré différent de celui espéré et à une détection de l'erreur de chiffrement.

Catégorie 5 : Lorsque le chiffré obtenu n'est pas celui espéré et que l'erreur n'est pas signalée, alors il existe une **"Erreur non détectée"**. Dans le cas où cette erreur non détectée est exploitable pour récupérer la clef de chiffrement (*Differential Fault Analysis : DFA*) alors on parle de **"Faille de sécurité"**.

6.3.4. Attaque en boîte noire par application de surtensions

Actuellement, il existe des travaux présentant l'effet d'une sous-alimentation de composant électronique implantant des crypto-processeurs sur ASIC ([Selmane 2008]) ou sur FPGA ([Khelil 2008], [Selmane 2009]). Grâce à cette technique de sous-alimentation et à l'attaque de Piret-Quisquater [Piret 2003], les clefs utilisées pour chiffrer des données ont pu être retrouvées. Cependant, les attaques que nous avons menées ne consistent pas à sous-alimenter le composant mais à appliquer une surtension à la tension nominale.

6.3.4.1 Sur front montant de l'horloge

Lors des campagnes S1-1 et S1-2, respectivement 1080 et 960 surtensions ont été appliquées sur le FPGA lors du chiffrement. La répartition des fautes selon les catégories définies dans la section 6.3.3 de ce chapitre est donnée dans les Tableau 6.2 et Tableau 6.3 respectivement pour les campagnes S1-1 et S1-2.

Tableau 6.2 : Types de perturbations obtenus par application de surtensions comprises entre 5 volts et 45 volts sur front montant de l'horloge lors des 2 dernières rondes.

Catégories de perturbations	Nombre	Pourcentage
Surtension sans effet	1027	95,1%
Faute silencieuse	0	0%
Faux positif	16	1,5%
Erreur détectée	37	3,4%
Erreur non détectée	0	0%

Dans la majorité des cas (Tableau 6.2), les injections de fautes par application de surtensions dans la gamme 5-45 volts ne permettent pas de corrompre la configuration du composant (ligne "*Surtension sans effet*") quelque soit le couple de surtension (durée et amplitude). Par ailleurs, aucune faute "silencieuse" n'a été obtenue montrant que si une surtension génère une erreur de configuration alors celle-ci a un effet sur le fonctionnement du crypto-processeur. Pour des surtensions comprises entre 5 volts et 45 volts, la détection d'erreur est activée dans près de 5% des surtensions et dans 1,5% des cas le chiffré est correct. Pour cette gamme de tensions, aucune "erreur non détectée" n'a été trouvée, contrairement à la gamme de tension supérieure (campagne S1-2) où 0,7% des injections de fautes ont permis de générer des erreurs non détectées (Tableau 6.3). Cependant, la contremesure implantée dans le circuit semble assez efficace puisque lorsqu'une erreur de chiffrement est générée alors cette dernière est détectée systématiquement pour la campagne S1-1 et dans 92,7% des cas pour la campagne S1-2. Pour des surtensions plus importantes en amplitude à celles utilisées lors de la campagne S1-1, 15% des surtensions modifient la configuration du composant (Tableau 6.3) au lieu de 5% et plus de 9% des surtensions déclenchent des fausses alarmes (catégorie "*Faux positif* ") ce qui signifie que la faute a été injectée non pas dans le premier calcul d'une ronde, mais dans la copie servant de contremesure.

Tableau 6.3 : Types de perturbations obtenus par application de surtensions comprises entre 45 volts et 80 volts sur front montant de l'horloge lors des 2 dernières rondes.

Catégories de perturbations	Nombre	Pourcentage
Surtensions sans effet	809	84,3%
Faute silencieuse	0	0%
Faux positif	55	9,3%
Erreur détectée	89	5,7%
Erreur non détectée	7	0,7%

Maintenant que nous avons montré la possibilité de générer tel ou tel type d'erreur, il faut regarder plus précisément les instants permettant de conduire à ces erreurs. Les Figure 6-9 et Figure 6-10 représentent la répartition des catégories de perturbations obtenues selon le cycle d'injection, pour des surtensions appliquées sur front montant respectivement comprises entre 5 volts et 45 volts et entre 45 volts et 80 volts. On remarque sur ces 2 figures qu'aucune faute de type "Faute Silencieuse" n'a été obtenue.

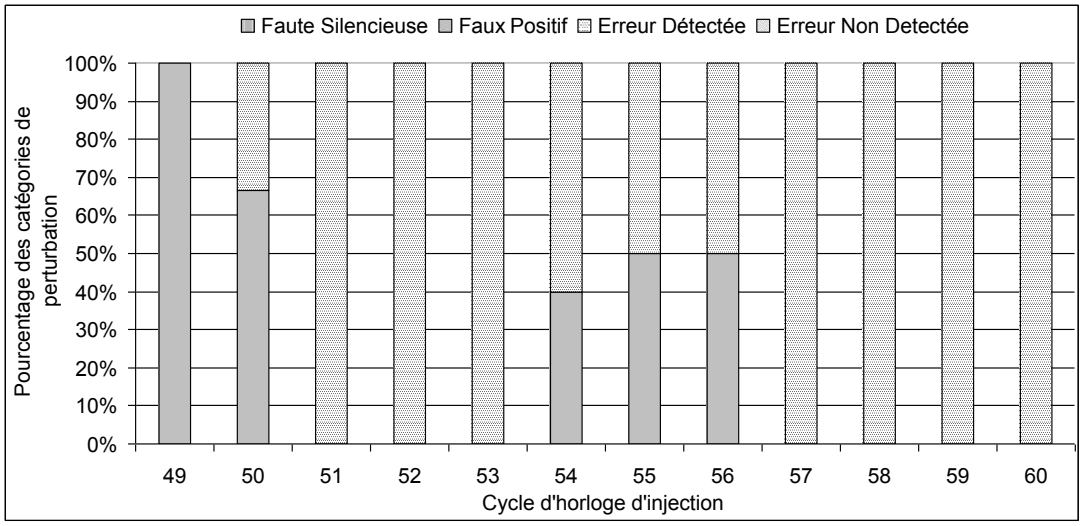


Figure 6-9 : Pourcentage des catégories de perturbations selon le cycle d'injection pour des surtensions comprises entre 5 volts et 45 volts

Sur la Figure 6-9, on note qu'aucune erreur de chiffrement non détectée n'a été obtenue pour des surtensions inférieures à 45 volts. Cependant, les surtensions ont bien un effet sur le chiffrement puisqu'un grand nombre de couples conduisent à des erreurs détectées. Si l'amplitude de la surtension est supérieure à 45 volts (Figure 6-10) et lorsque des erreurs de chiffrement sont présentes, ces erreurs sont dans la majorité des cas détectées. Un résultat intéressant du point de vue de l'attaque est l'apparition d'erreurs non détectées uniquement lors du dernier cycle d'horloge. Par contre, ces erreurs ne sont pas exploitables pour récupérer la clef de chiffrement car elles ont lieu lors de l'ajout de la sous-clé de la ronde et/ou aux écritures dans les différents registres.

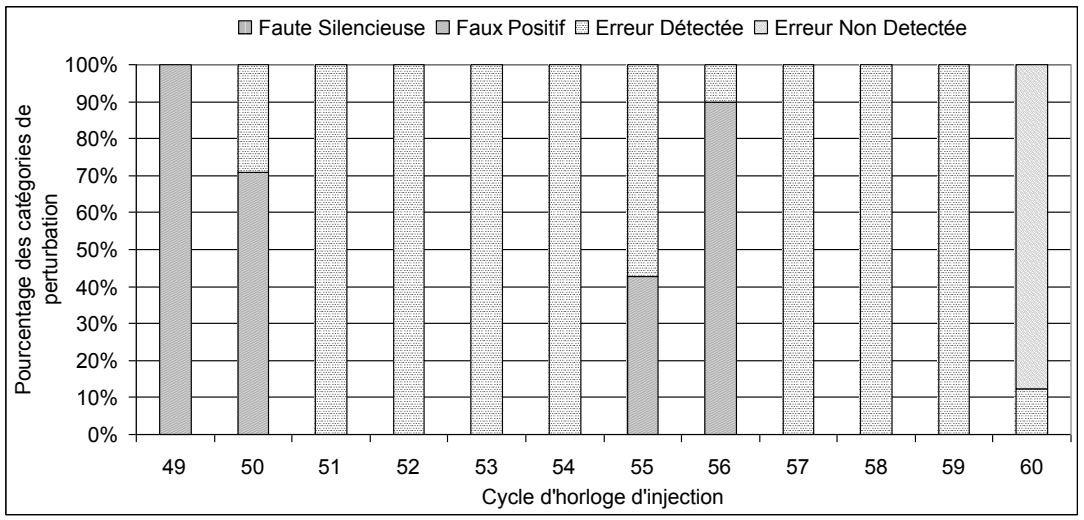


Figure 6-10: Pourcentage des catégories de perturbations selon le cycle d'injection pour des surtensions comprises entre 45 volts et 80 volts

En comparant les résultats de la Figure 6-9 et de la Figure 6-10, on observe que des cycles sont plus propices à certains effets indépendamment de l'amplitude de la surtension. Par exemple, lorsque l'injection de fautes se produit durant les 2 premiers cycles des rondes normales (49-50 et 55-56), on remarque que nous obtenons des faux positifs. Par contre, lorsque l'injection de fautes se déroule durant les autres cycles, nous obtenons principalement des erreurs détectées, excepté pour des surtensions ayant des amplitudes importantes appliquées durant le dernier cycle de chiffrement où des erreurs non détectées sont présentes. Une explication possible des faux positifs est la modification du signal d'activation du stockage des données dans les registres DDR et/ou du contenu des registres. L'activation du signal "*Enable primary DDR Register*" permet de stocker les valeurs dans le registre primaire alors que le signal "*Enable dual DDR Register*" permet le stockage dans le registre secondaire (Figure 6-11).

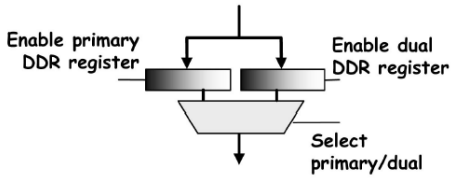


Figure 6-11 : Registre DDR avec la copie de sauvegarde et le signal d'activation [Maistri 2008]

Durant le calcul de la ronde normale où le signal *Enable primary DDR Register* est activé, si l'on modifie uniquement le signal *Enable dual DDR Register* alors de mauvaises valeurs peuvent être stockées dans le registre DDR secondaire et une alarme sera déclenchée sans que le chiffré soit erroné (*Faux positif*). Si le contenu des registres DDR est modifié alors toutes les erreurs de chiffrement induites seront détectées. Si la faute est induite durant la ronde de vérification et que le signal *Enable primary DDR Register* et/ou le contenu des registres sont modifiés alors des erreurs de chiffrement seront obtenues et détectées.

6.3.4.2 Sur front descendant de l'horloge

Puisque le crypto-processeur effectue des calculs sur les deux fronts d'horloge (principe du DDR), des surtensions d'amplitude comprise entre 45 volts et 80 volts ont été appliquées sur les fronts descendants de l'horloge (campagne S2).

Tableau 6.4 : Types de perturbations obtenus par application de surtensions comprises entre 45 volts et 80 volts sur front descendant de l'horloge lors des 2 dernières rondes.

Catégories de perturbations	Nombre	Pourcentage
Surtension sans effet	268	27,9%
Faute silencieuse	12	1,25%
Faux positif	60	6,2%
Erreur détectée	157	16,4%
Erreur non détectée	463	48,2%

Lors de cette campagne, 960 surtensions ont été appliquées sur l'alimentation du cœur du composant et le Tableau 6.4 regroupe les différentes catégories d'effets obtenus. Lors des attaques sur le front montant de

l'horloge, plus de 80% des surtensions n'avaient eu aucun effet sur la configuration du composant, alors que sur le front descendant la proportion de surtensions sans effet est nettement plus faible (27,9%). Ceci montre que le système est nettement plus sensible sur le front descendant de l'horloge que sur le front montant. En effet des efforts de sécurisation du circuit ont été faits sur les fronts montants et les bascules utilisateurs semblent plus sensibles sur le front descendant du fait de leur conception (section 4.4 du Chapitre 4). Contrairement aux résultats obtenus lors de la campagne S1-2 comparables en gamme d'amplitudes de surtension, on remarque que certaines valeurs de surtensions génèrent des fautes silencieuses (1,25%). La proportion de faux positifs est presque constante quelque soit le type de front d'attaque puisque lors de la campagne S1-2, 55 valeurs de surtensions différentes ont permis d'induire des fausses détections d'erreurs contre 60 lors de la campagne S2. Les surtensions sur front descendant permettent de générer nettement plus de détection d'erreurs (157 pour S2 contre 89 pour S1-2). Enfin, plus de 48% des valeurs utilisées ont permis d'obtenir des erreurs non détectées. Cependant, tous les chiffrés obtenus sont à '0' et ne sont pas exploitables par un attaquant pour retrouver la clef de chiffrement. Le cas où tous les registres sont remis à zéro correspond à une remise à zéro du circuit implanté (crypto-processeur et UART) avant la fin du chiffrement puisque tous les registres internes sont à zéro (clefs, messages et chiffrés, erreur). Pour toutes les campagnes par surtensions, y compris dans le cas d'un grand nombre d'erreurs non détectées, aucune faille de sécurité n'a donc été identifiée.

6.3.5. Attaque en boîte noire par tirs laser

6.3.5.1 Les erreurs de chiffrement

Lors des attaques laser du crypto-processeur sécurisé, plus de 1400 tirs ont été effectués par cycle d'horloge lors de son fonctionnement d'où un total de 8610 tirs.

Tableau 6.5 : Types d'effets obtenus lors d'injections de fautes par laser durant les 6 cycles d'horloges de l'avant dernière ronde sur le front montant de l'horloge.

Catégories de perturbations	Nombre	Pourcentage
Tir Sans Effet	1155	13,4%
Faute Silencieuse	3302	38,4%
Faux Positif	44	0,5%
Erreur Détectée	2549	29,6%
Erreur Non Détectée	1557	18,1%

Le Tableau 6.5 représente les résultats obtenus en fonction du type d'effet observé (section 6.3.3). Comme lors les campagnes d'injections statiques présentées dans le Chapitre 4, certaines positions de tir (13,41%) n'ont aucun effet sur la configuration du composant. Cependant lorsqu'un tir laser a un effet sur la configuration du FPGA, plus de 50% des positions de tirs conduisent à des erreurs de chiffrement (34% détectées et 21% non détectées) et plus de 44% des positions à des fautes silencieuses.

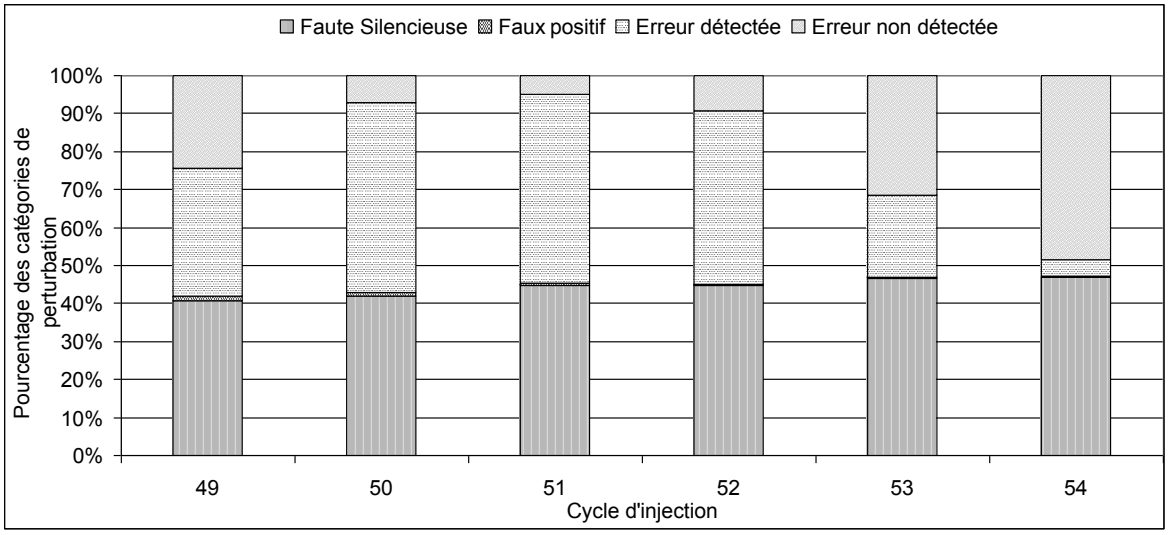


Figure 6-12 : Pourcentage des catégories de perturbations selon le cycle d'injection lors d'attaques sur front montant d'horloge en utilisant un spot laser de 20 µm

Nous venons de montrer la possibilité d'obtenir les différentes catégories d'effet lors de tirs laser et une analyse de ces effets en fonction du cycle d'horloge est donnée dans la Figure 6-12. Quel que soit le cycle d'injection la proportion de tirs laser sans effet reste quasi-constante autour de 12% ainsi que celle des fautes silencieuses (39%). Cependant, lorsque des fautes de chiffrement apparaissent, leur proportion varie en fonction de l'instant d'injection. Ces résultats confirment ceux présentés dans [Pouget 2008] concernant l'attaque d'un crypto-processeur Triple DES non protégé implanté dans un FPGA et montrant que le nombre d'erreurs de chiffrement dépend de l'instant d'injection.

Si l'injection de fautes a lieu durant le calcul de la ronde "normale" de l'AES (cycles d'horloge 49, 50 et 51), la probabilité de détection des erreurs de chiffrement est supérieure à 33%. Alors que si l'injection de fautes se produit lors du dernier cycle d'horloge de la ronde (cycle d'horloge 54 : cycle de vérification et de comparaison), la probabilité de non-détection est très importante (48%) du fait de la rémanence des fautes induites. Le FPGA étant de type SRAM, toute modification de la configuration du circuit est présente jusqu'au chargement d'une nouvelle configuration. Ainsi, si une faute est injectée durant le dernier cycle d'une ronde (fin de la ronde de vérification) alors cette modification de la configuration sera présente durant la ronde suivante (rondes normale et de vérification). La contremesure implantée ne permet pas la détection des fautes rémanentes ce qui explique le résultat obtenu. Une amélioration de la protection sera proposée dans la section 6.3.6.

6.3.5.2 Les fautes induites dans la configuration

Comme expliqué précédemment, les fautes générées dans la configuration du circuit peuvent modifier 2 types d'éléments : les bascules utilisateurs et/ou la configuration. Pour savoir si les bascules utilisateurs ont été modifiées, nous avons utilisé les fichiers de masquage et d'allocation logique fournis par l'outil de compilation. Cependant, tous les bits masqués ne sont pas présents dans le fichier d'allocation logique et la fonctionnalité de ces bits est actuellement inconnue.

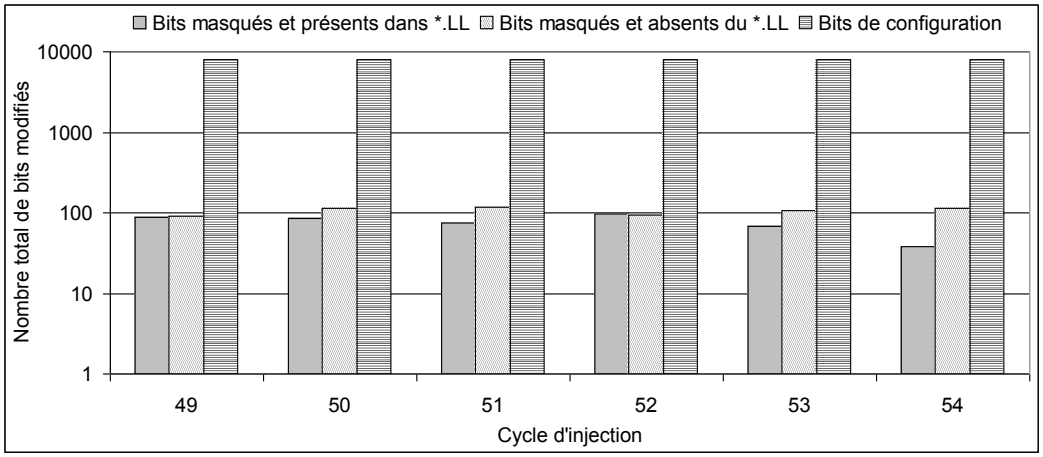
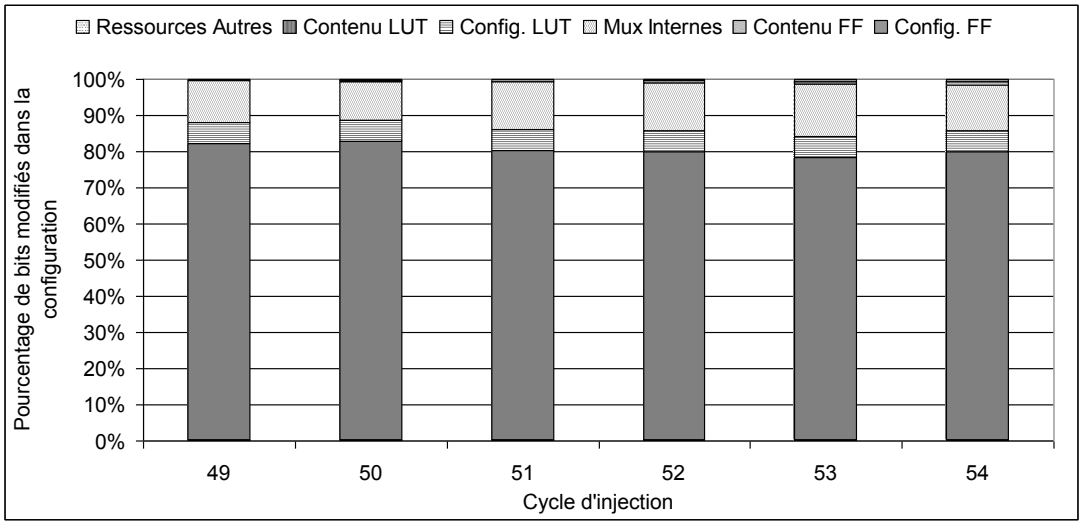


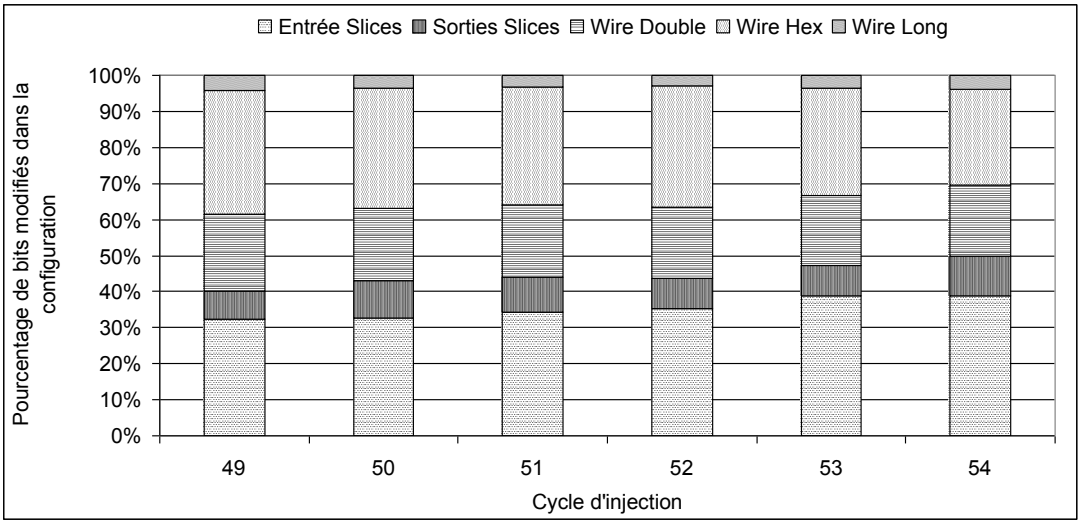
Figure 6-13 : Répartition du nombre total de bits modifiés dans le fichier de relecture en fonction du cycle d'injection lors de la campagne utilisant un spot laser de 20 µm

La Figure 6-13 regroupe la répartition des catégories de bits modifiés (masqués ou non) en fonction du cycle d'injection. Sur cette figure, on note que majoritairement des bits configurant la fonctionnalité des différents éléments autres que les bascules utilisateurs (bits non masqués) ont été modifiés. La proportion de bits masqués modifiés non présents dans le fichier d'allocation logique est quasi-constante quel que soit le cycle d'injection des fautes. Précédemment, nous avons montré que si la faute est injectée durant le dernier cycle d'horloge de l'avant dernière ronde (cycle 54), alors une majorité de positions conduit à des erreurs non détectées. Lors de ce dernier cycle la proportion de bits masqués et présents dans le fichier d'allocation est inférieur aux proportions de la même catégorie de bits des autres cycles d'horloge. Ce résultat tend à montrer que les erreurs de chiffrement non détectées sont peu liées aux modifications induites dans le contenu des bascules utilisateurs. Ceci confirme l'importance de la prise en compte des erreurs rémanentes dans la configuration.

Dans le Chapitre 4, une classification de la sensibilité des éléments constituant les tuiles CLB à été réalisée. La Figure 6-14 représente les résultats obtenus lors de la campagne dynamique (attaque du crypto-processeur en fonctionnement) qui corroborent ceux des campagnes statiques. En effet, près de 80% des erreurs induites dans la configuration des éléments de logique interne concernent le contenu des LUT et dans une moindre mesure les multiplexeurs internes (Figure 6-14-a). Les entrées des slices et les connexions de type Hex sont également les éléments d'interconnexions ayant la sensibilité la plus importante lors des campagnes statiques et dynamiques (Figure 6-14-b).



a)



b)

Figure 6-14 : Pourcentage des erreurs de configuration des éléments constituant la logique (a) et les interconnexions (b) des tuiles CLB

Les bits ayant les sensibilités les plus importantes (i.e. le contenu des LUT, les multiplexeurs internes, les entrées des slices et les connexions de type Hex) sont également ceux qui ont été le plus modifiés lors des erreurs de chiffrement. Il n'est donc pas possible d'identifier une catégorie de bits conduisant plus particulièrement à des erreurs fonctionnelles lors du test dynamique.

6.3.5.3 Le cas des fautes uniques

Nous avons introduit dans les chapitres précédents le cas particulier des fautes uniques dans la mémoire de configuration. On parle de faute unique lorsque le tir laser n'a modifié qu'un seul bit du fichier de relecture (configuration et/ou bascule utilisateur). Aucun des tirs laser des campagnes dynamiques n'a permis de modifier un seul bit configurant des bascules utilisateurs. Ainsi, les seuls cas de fautes uniques sont ceux où uniquement un bit de la configuration a été modifié.

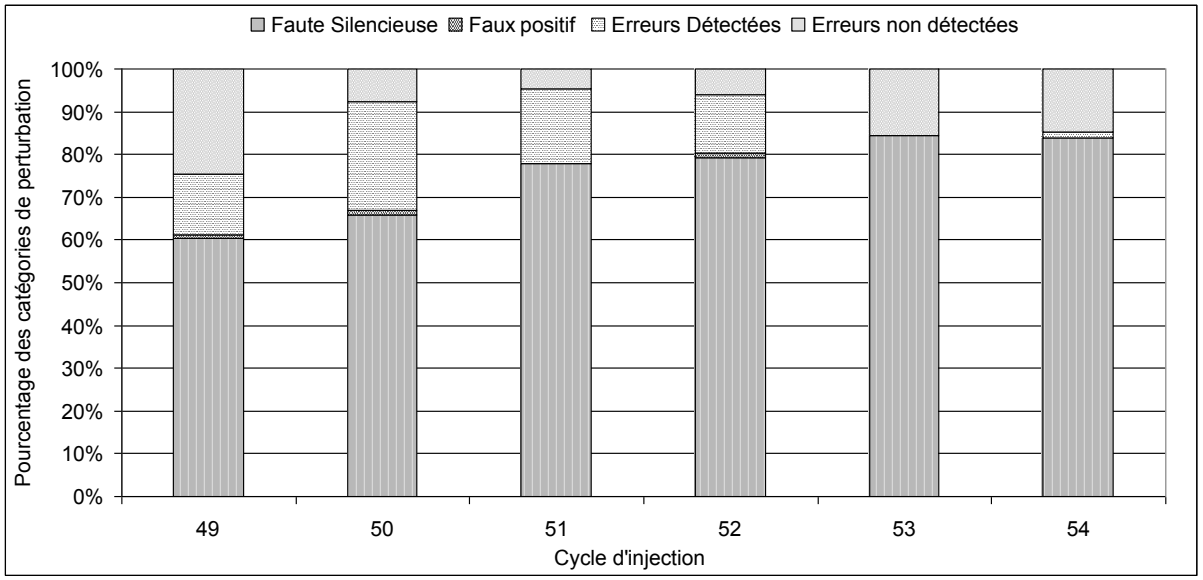


Figure 6-15 : Pourcentage des catégories de perturbations selon le cycle d'injection dans le cas de fautes uniques induites dans la configuration lors de la campagne utilisant un spot laser de 20 µm.

La Figure 6-15 permet de connaître la répartition des effets des fautes uniques de configuration sur le chiffrement selon le cycle d'injection. Une majorité des fautes uniques dans la configuration conduit à des fautes silencieuses (>60%). De plus, on note que des fautes uniques induisent des erreurs de chiffrement non détectées avec une proportion plus importante lorsque l'injection de fautes se produit au début de la ronde normale ou à la fin de la ronde de vérification. Comme précédemment, ce résultat peut être

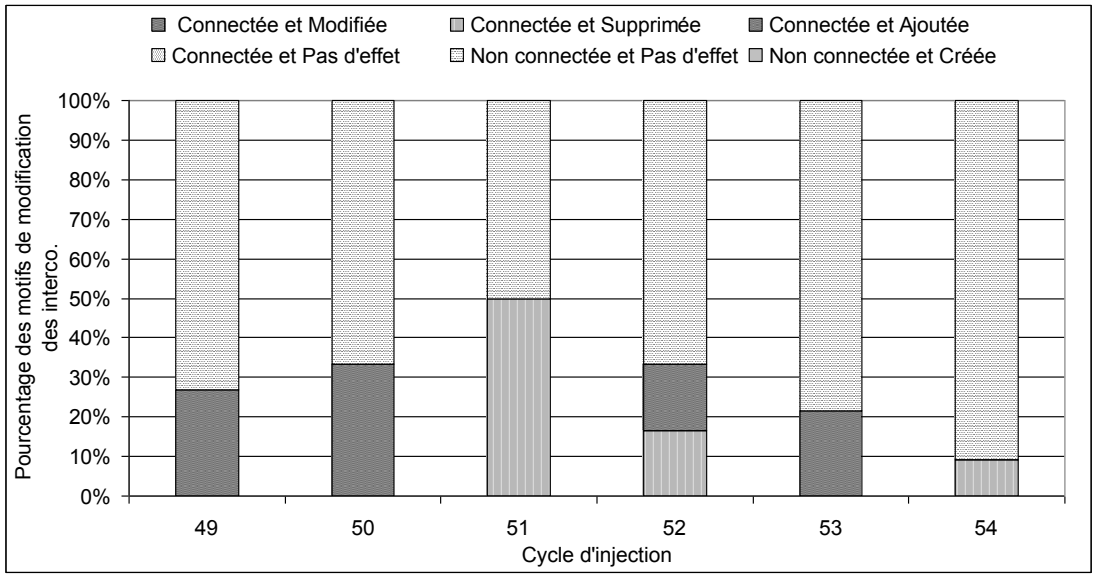


Figure 6-16 : Répartition des modifications de connexions définies par 2 bits obtenues selon le cycle d'injection dans le cas d'erreurs de chiffrement non détectées et de fautes uniques induites dans la configuration

expliqué par la rémanence de la faute induite présente durant tous les cycles suivant l'injection. Par comparaison avec la Figure 6-12, il apparaît par ailleurs que la répartition obtenue pour les fautes uniques est très similaire à celle obtenue pour l'ensemble des tirs lorsque ces derniers ont lieu durant les cycles de la ronde normale. Les fautes uniques ne semblent donc pas avoir d'effet très spécifique dans le cas d'une attaque en boîte noire.

Dans le Chapitre 4, nous avons présenté les différents motifs de modification des connexions définies par plusieurs bits, lesquels dépendent de l'état initial de ces connexions. La Figure 6-16 montre la répartition de ces motifs obtenus dans le cas d'erreurs de chiffrement non détectées en fonction du cycle d'injection. La modification de la configuration des interconnexions concerne principalement une connexion inexistante. Un résultat surprenant montre que cette modification n'a eu aucun effet sur cette connexion alors qu'une erreur de chiffrement est présente. En analysant les fichiers de relecture avec notre outil d'analyse (section 3.5 du Chapitre 3), il ressort que les connexions apparaissent non utilisées et qu'aucune nouvelle connexion n'a été créée. Ainsi, il est possible de supposer que des phénomènes autres que la modification de la configuration et/ou des bascules utilisateurs sont induits par le tir laser mais également que certains éléments ne peuvent pas être analysés par notre outil.

6.3.6. Une nouvelle contremesure

6.3.6.1 Présentation

Nous avons vu que la contremesure implantée n'est pas suffisante pour se prémunir des attaques par injection de fautes laser lorsque le crypto-processeur est utilisé sur une plateforme FPGA de type SRAM. En effet, la protection à redondance DDR permet de se protéger efficacement contre les fautes transitoires telles que les fautes induites par les surtensions mais pas contre les fautes rémanentes.

Ainsi, quelques protections supplémentaires ont été implantées dans l'architecture du crypto-processeur. Chaque ronde (normale et vérification) partage la logique combinatoire non-linéaire c'est-à-dire les S-box. Le calcul étant fortement symétrique, le parallélisme de l'architecture implantée peut être exploité. A chaque ronde de vérification, une rotation de chaque colonne de la matrice d'état a été ajoutée. Ainsi, la ronde normale est effectuée comme précédemment alors que durant la ronde de vérification les colonnes de la matrice sont décalées avant d'entrer dans les boîtes de substitution. Par la suite, les colonnes sont remises dans le bon ordre grâce à l'opération *ShiftRows*. Cette technique permet la détection des fautes rémanentes pouvant être induites dans les boîtes de substitution.

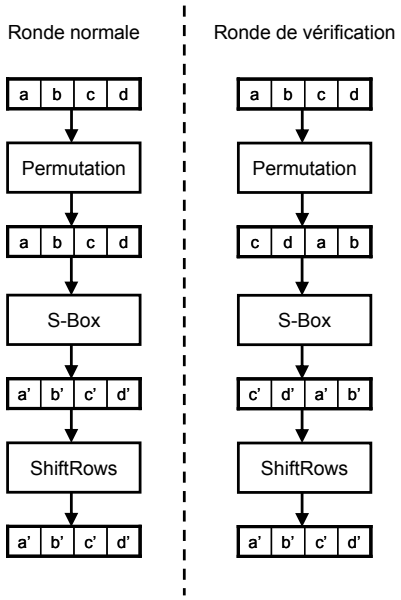


Figure 6-17 : Principe de fonctionnement de la nouvelle contremesure

Dans un même temps, d'autres contremesures mineures ont été implantées. La détection des erreurs induites dans le contrôleur a été modifiée afin de déclencher des alarmes lorsque les fautes sont générées sur les fronts montant et/ou descendant de l'horloge. Le générateur de clefs a également été modifié et plus particulièrement la partie logique qui était partagée avec les opérations de chiffrement. Précédemment, les sous-clés de la ronde étaient calculées uniquement lors du dernier cycle d'horloge de la ronde précédente et plus spécifiquement lorsque les S-box n'étaient pas utilisées par la ronde de vérification. Désormais, le calcul des sous-clés est également réalisé durant la ronde normale, et les clefs de ronde sont temporairement sauvegardées pour être comparées par la suite lors du calcul des sous-clés de la ronde suivante.

6.3.6.2 Les résultats des attaques laser

Précédemment, nous avons montré que les attaques par surtensions ne permettent pas de générer des erreurs de chiffrement non détectées exploitables grâce à la protection implantée. Par contre lors d'attaques laser, cette contremesure est nettement moins efficace pour des fautes rémanentes. De ce fait, uniquement des injections de fautes par tir laser ont été réalisées sur le crypto-processeur implantant la nouvelle protection.

En implantant ce nouveau circuit dans le FPGA sous test, 1699 slices sur les 5120 disponibles (33%) et 19% des bascules utilisateurs (2023 sur 10240) ont été utilisées. Sans contraintes de placement-routage, ce circuit utilise 41 slices de moins que la précédente version (1699 actuellement contre 1740 précédemment) et 54 bascules utilisateurs ont été ajoutées (2023 contre 1969).
Pour l'étude de cette nouvelle contremesure, nous avons utilisé des contraintes de placement-routage similaires à celle de la précédente version et des zones équivalentes aux précédentes campagnes étudiées.

Cependant, la netlist est légèrement différente et devra être prise en compte lors des comparaisons des résultats.

Tableau 6.6 : Types d'effets obtenus lors d'injections de fautes par laser durant les 6 cycles d'horloges de l'avant dernière ronde

Catégories de perturbations	Nombre	Pourcentage
Tir Sans Effet	4698	54,6%
Faute Silencieuse	1548	18,0%
Faux Positif	89	1,0%
Erreur Détectée	2263	26,3%
Erreur Non Détectée	11	0,1%

Du fait de la nouvelle implantation du circuit dans le FPGA, un nombre de tirs laser plus important n'a eu aucun effet sur la configuration (Tableau 6.6). Les tirs laser sans effet représentent 50% des positions alors qu'ils ne représentaient que 13% lors de l'utilisation du précédent circuit. Ce résultat est dû à l'occupation des slices du FPGA qui diffère entre les deux circuits à cause de la nouvelle netlist. Lorsque des modifications apparaissent dans la configuration, nous notons qu'elles ont principalement pour effets de générer des fautes silencieuses (39%) ou des erreurs détectées (58%). Les erreurs non détectées représentent moins de 0,3% des cas de perturbations contre 21% précédemment. Ainsi, la majorité des erreurs non détectées de la précédente version du circuit sont maintenant correctement détectées. La

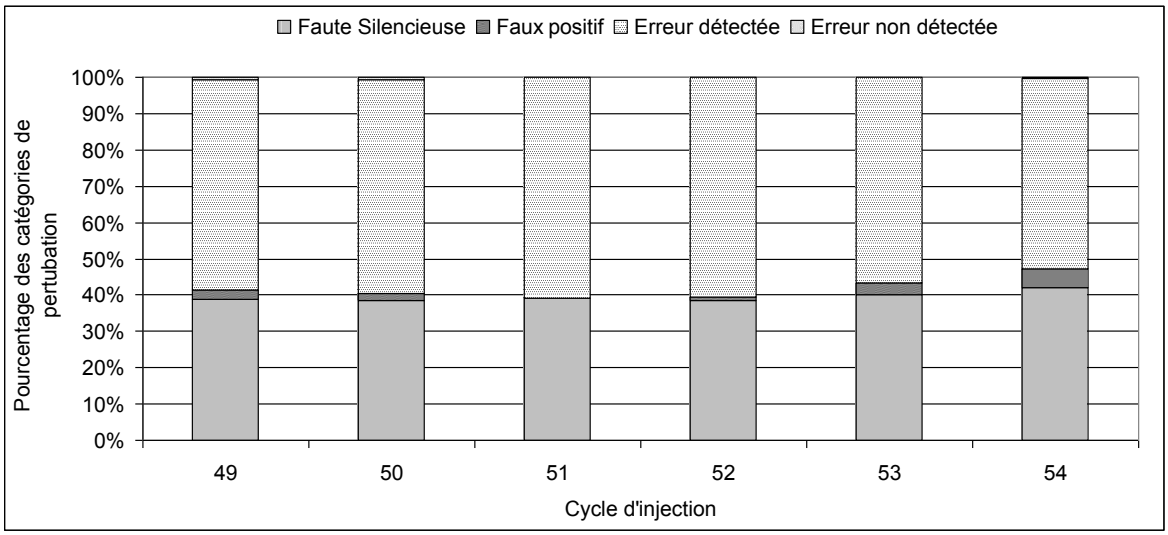


Figure 6-18 : Pourcentage des catégories de perturbations selon le cycle d'injection lors d'attaques sur front montant d'horloge en utilisant un spot laser de 20 µm (nouvelle contremesure)

nouvelle contremesure est plus sécurisée puisque environ 0,5% des erreurs de chiffrement sont non détectées contre 38% avec la version précédente.

Sur la Figure 6-18, nous notons que lorsque la configuration est modifiée, indépendamment du cycle d'injection, la modification conduit majoritairement à des fautes silencieuses (40%) ou à des détections d'erreurs (57%). Lors de l'utilisation de la précédente version d'architecture, on observait des erreurs de chiffrement non détectées dans tous les cycles étudiés avec une majorité pour le cycle d'horloge 54. Avec cette nouvelle contremesure, les quelques erreurs non détectées sont présentes uniquement durant les cycles 49, 50 et 54 mais dans une proportion nettement plus faible. Malgré le fait que la proportion d'erreur non détectée est très faible, il faut cependant vérifier l'exploitabilité des erreurs pour retrouver les clefs de chiffrement et cette analyse est faite dans la section suivante.

6.4. Exploitation des erreurs de chiffrement

Lors de la campagne d'injection de fautes sur la nouvelle version de contre-mesure, nous avons obtenu 11 positions de tirs conduisant à des erreurs non détectées. Ces erreurs sont dues à des modifications de bits configurant des interconnexions de type Double ou le contenu de LUT. Le Tableau 6-7 regroupe les coordonnées de tirs (X; Y), le cycle d'injection et le résultat du chiffrement.

Tableau 6-7 : Résultats du chiffrement dans le cas d'erreur non détectée

Numéro de faute	X	Y	Cycle d'injection	Résultat du chiffrement
0	0	0	0	69 C4 E0 D8 6A 7B 04 30 D8 CD B7 80 70 B4 C5 5A
1	8030	7045	49	69 C4 E0 **44** 6A 7B 04 30 D8 CD B7 80 70 B4 C5 5A
2	8015	7210	49	69 C4 E0 **B9** 6A 7B 04 30 D8 CD B7 80 70 B4 C5 **9F**
3	8015	7225	49	69 C4 E0 **B9** 6A 7B 04 30 D8 CD B7 80 70 B4 C5 **9F**
4	8180	7465	49	69 C4 E 0**C**8 6A 7B 04 **2**0 D8 CD B7 **9**0 70 B4 C5 **4**A
5	8300	7045	50	69 C4 E0 **90** 6A 7B 04 30 D8 CD B7 80 70 B4 C5 5A
6	8030	7060	50	69 C4 E0 **44** 6A 7B 04 30 D8 CD B7 80 70 B4 C5 5A
7	8015	7210	50	69 C4 E0 **B9** 6A 7B 04 30 D8 CD B7 80 70 B4 C5 **9F**
8	8015	7225	50	69 C4 E0 **B9** 6A 7B 04 30 D8 CD B7 80 70 B4 C5 **9F**
9	8180	7465	50	69 C4 E0 **C**8 6A 7B 04 **2**0 D8 CD B7 **9**0 70 B4 C5 **4**A
10	8015	7210	54	69 C4 E0 D8 6A 7B 04 30 D8 CD B7 80 70 B4 C5 **9F**
11	8015	7225	54	69 C4 E0 D8 6A 7B 04 30 D8 CD B7 80 70 B4 C5 **9F**

Nous notons dans le Tableau 6-7 qu'il existe uniquement 5 erreurs de chiffrement différentes : fautes 1 et 6 ("*0x44*" au lieu de "*0xD8*"), fautes 2, 3, 7 et 8 ("*0xB9*" et "*0x9F*" au lieu de "*0xD8*" et "*0x5A*"), fautes 4 et 9 ("*0xC*", "*0x2*", "*0x9*", "*0x4*" au lieu de "*0xD*","*0x3*", "*0x8*","*0x5*"), fautes 10 et 11 ("*0x9F*" au lieu de "*0x5A*") et faute 5 ("*0x90*" au lieu de "*0xD8*").

Tableau 6-8 : Valeurs obtenues lors des opérations inverses de chiffrement [FIPS PUB 197]

Round[10].output	69	C4	E0	D8	6A	7B	04	30	D8	CD	B7	80	70	B4	C5	5A
Round[10].k_sch	13	11	1D	7F	E3	94	4A	17	F3	07	A7	8B	4D	2B	30	C5
Round[10].s_row	7A	D5	FD	C6	89	EF	4E	27	2B	CA	10	0B	3D	9F	F5	9F
Round[10].s_box	7A	9F	10	27	89	D5	F5	0B	2B	EF	FD	9F	3D	CA	4E	A7
Round[10].start	BD	6E	7C	3D	F2	B5	77	9E	0B	61	21	6E	8B	10	B6	89
Round[09].k_sch	54	99	32	D1	F0	85	57	68	10	93	ED	9C	BE	2C	97	4E
Round[09].m_col	E9	f7	4E	EC	02	30	20	F6	1B	F2	CC	F2	35	3C	21	C7
Round[09].s_row	54	D9	90	A1	6B	A0	9A	B5	96	BB	F4	0E	A1	11	70	2F
Round[09].s_box	54	11	F4	B5	6B	D9	70	0E	96	A0	90	2F	A1	BB	9A	A1
Round[09].start	FD	E3	BA	D2	05	E5	D0	D7	35	47	96	4E	F1	FE	37	F1

Pour vérifier l'exploitabilité de la faute, nous devons à partir du chiffré fauté faire les opérations en sens inverse. Nous savons que la ou les fautes ont été injectées durant l'avant dernière ronde de chiffrement (ronde 9). Ainsi, nous étudierons uniquement les deux dernières rondes pour vérifier l'exploitabilité des erreurs. Les mêmes notations que celles définies dans [FIPS PUB 197] seront utilisées et les résultats des différentes opérations sont celles données dans le Tableau 6-8.

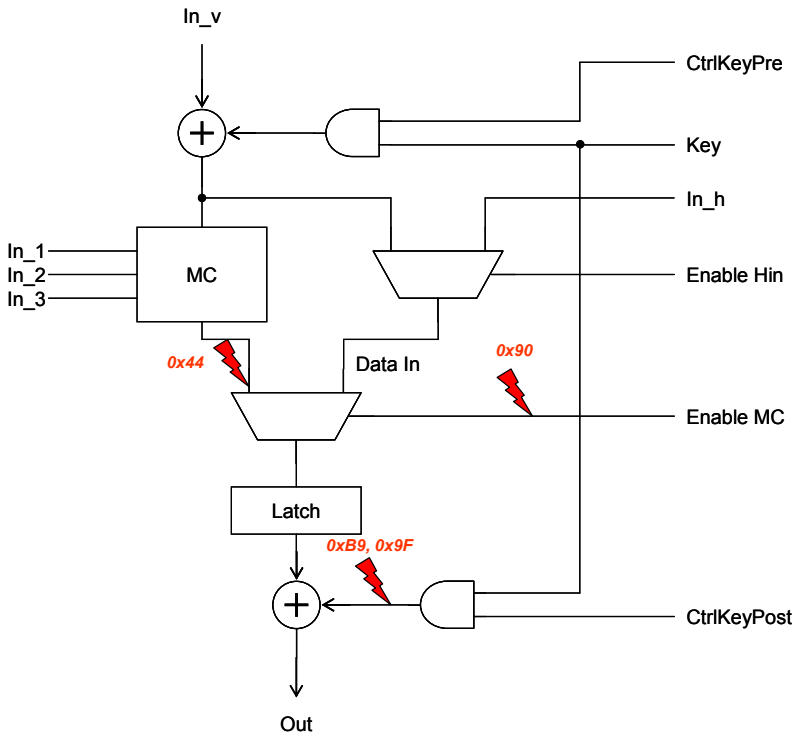


Figure 6-19 : Architecture des cellules de données et localisation des fautes

La Figure 6-19 présente l'architecture des cellules de données avec les différents signaux utiles. Les signaux *In_v* et *In_h* sont les entrées utilisées pour les données. Le signal *Key* est la clef de ronde, *CtrlKeyPre* et *CtrlKeyPost* sont les signaux de contrôle utilisés pour les sous-clés respectivement lors des opérations de déchiffrement et de chiffrement. Le signal *EnableMC* permet de réaliser ou non l'opération "*Mixcolumns*". Durant les 10 rondes, les mêmes opérations sont effectuées excepté dans la ronde 10 où l'opération "*MixColumns*" n'est pas présente. Enfin, le signal *Out* correspond à la sortie de la cellule de données.

Sur la Figure 6-19, nous avons également positionné les fautes induites dans l'architecture du crypto-processeur. Pour localiser ces fautes, nous avons étudié au cas par cas chaque valeur de chiffré erroné.

6.4.1. Cas de l'erreur "0x9F"

Dans le cas de l'erreur "0x9F", un seul octet a été modifié par notre injection de fautes lors du cycle 54. Ce cycle correspond aux dernières opérations de la ronde 9. Ainsi, uniquement les opérations inverses de la ronde 10 seront réalisées pour trouver l'effet de la faute. L'octet fauté se situe dans la dernière colonne de la matrice d'état du fait des contraintes de placement-routage.

Plusieurs hypothèses sont possibles concernant la localisation de la faute dans l'architecture : modification de la S-Box, modification de la clef ou de son chemin, suppression d'opérations, etc….

Dans le cas de l'hypothèse de la S-box, toute modification de cette dernière aura pour effet de modifier une colonne entière de l'état puisque la même S-box est utilisée pour les différentes lignes de l'état. Puisque nous avons uniquement modifié un octet, nous pouvons dire que cette hypothèse est peu probable.

Tableau 6-9 : Valeurs des opérations obtenues en sens inverse (faute "0x9F")

Round[10].output	69	C4	E0	D8	6A	7B	04	30	D8	CD	B7	80	70	B4	C5	**9F**
Round[10].k_sch	13	11	1D	7F	E3	94	4A	17	F3	07	A7	8B	4D	2B	30	C5
Round[10].s_row	7A	D5	FD	C6	89	EF	4E	27	2B	CA	10	0B	3D	9F	F5	**5A**

L'autre hypothèse concerne la modification de la clef ou de son chemin conduisant à une modification d'une ligne de la matrice d'état. Puisque nous avons contraint le placement-routage à une seule colonne, nous pouvons supposer que cette hypothèse est très probable pour modifier un seul octet (une seule ligne et une seule colonne). Nous supposons que la clef n'est pas modifiée pour localiser la faute. Nous

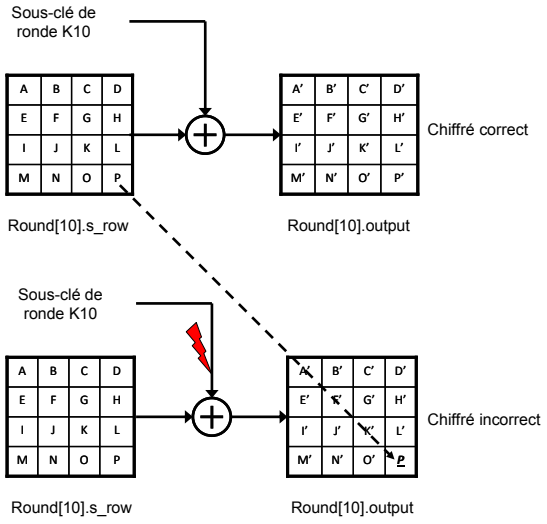


Figure 6-20 : Localisation de l'erreur "0x90" induite

remarquons que la valeur de l'octet modifié vaut "0x9F" (ligne "*Round[10].output*" du Tableau 6-9) et correspond à la même valeur que l'octet de la ligne "*Round[10].s_row*" du Tableau 6-8.

Pour obtenir le résultat du chiffrement "*Round[10].output*", un ou-exclusif est réalisé entre les données " *Round[10].s_row*" et la clef de ronde. Lors de notre attaque, nous remarquons qu'un octet est identique dans l'opération correcte "*Round[10].s_row* " et "*Round[10].output*" lors du chiffrement fauté (Figure 6-20). Ainsi, la seule explication de ce résultat est le forçage à zéro d'un octet de la clef de ronde (forçage ou suppression d'une connexion). En réalisant un ou-exclusif entre les chiffrés correct et fautés, nous obtenons la valeur de la sous-clé de ronde 10 à savoir "*0xC5*". La faute induite est donc exploitable pour retrouver un octet de la clef de chiffrement.

6.4.2. Cas des erreurs "0xB9" et "0x9F"

Dans le cas des erreurs "*0xB9*" et "*0x9F*", nous notons que nous avons modifié le même octet que celui présenté précédemment (l'octet de valeur "*0x9F*"). Pour obtenir ces erreurs de chiffrement, nous avons injecté la faute durant les cycles 49 ou 50 correspondant aux premières opérations de la ronde 9. Puisque nous avons uniquement modifié deux octets se situant sur la même ligne, nous pouvons supposer que comme précédemment, nous avons modifié le chemin de clef. En effet, pour obtenir la valeur "*0x9F*", nous avons forcé à "*0x00*" la clef lors de calcul "*AddRoundKey*" de la ronde 10 (cf. cas de l'erreur "*0x9F*"). Grâce à la connaissance de cette erreur, nous pouvons en déduire la faute conduisant à l'erreur "*0xB9*" qui est également due à un forçage de la clef. En effet, si nous forçons à la valeur "*0x00*" la clef de la ronde 9, et que nous réalisons l'opération "*AddRoundKey*" avec la valeur en sortie de l'opération "*MixColumns*" du Tableau 6-8 alors la valeur obtenue vaut "*0xC7*" (Tableau 6-10).

Tableau 6-10 : Valeurs des opérations obtenues en sens inverse (fautes "0xB9 et 0x9F")

Round[10].output	69	C4	E0	**B9**	6A	7B	04	30	D8	CD	B7	80	70	B4	C5	**9F**
Round[10].k_sch	13	11	1D	7F	E3	94	4A	17	F3	07	A7	8B	4D	2B	30	C5
Round[10].s_row	7A	D5	FD	**C6**	89	EF	4E	27	2B	CA	10	0B	3D	9F	F5	**9F**
Round[10].s_box	7A	9F	10	27	89	D5	F5	0B	2B	EF	FD	9F	3D	CA	4E	**C6**
Round[10].start	BD	6E	7C	3D	F2	B5	77	9E	0B	61	21	6E	8B	10	B6	**C7**
Round[09].k_sch	54	99	32	D1	F0	85	57	68	10	93	ED	9C	BE	2C	97	4E

Lors de cette attaque, l'injection de faute a donc permis de forcer à "*0x00*" les octets de poids faibles des sous-clés des rondes 9 et 10. Cependant, pour retrouver un octet de la sous-clé de la ronde 9, nous avons besoin de connaître la sous-clé de la ronde 10, ce qui n'est normalement pas le cas lors d'une attaque. La faute induite ne permet donc pas de retrouver un octet de la clef de ronde 9. Par contre, puisque la faute est similaire à celle présentée dans la section précédente, nous avons retrouvé la valeur d'un octet de la sous-clé de la ronde 10. Ainsi, l'erreur induite est exploitable pour retrouver un octet de la clef de chiffrement.

6.4.3. Cas de l'erreur "0x44"

Dans le cas de l'erreur "0x44", nous avons modifié l'octet de poids fort de la dernière ligne de la matrice d'état. Pour obtenir cette erreur, nous avons injecté la faute durant les cycles 49 ou 50. Nous supposons toujours que nous n'avons pas modifié la clef de ronde. Comme précédemment, en réalisant les opérations en sens inverse, nous remarquons qu'un quartet en sortie de l'opération "*MixColumns*" a été forcé à zéro puisque nous obtenons "*0x07*" (Tableau 6-11) au lieu de "*0xC7*". Cette analyse a été possible du fait que nous connaissons la valeur de la clef de chiffrement. Cependant, avec les valeurs de chiffrés fautés obtenus, nous pouvons dire que ce résultat est potentiellement exploitable pour retrouver une partie des clefs de chiffrement.

Tableau 6-11 : Valeurs des opérations obtenues en sens inverse (faute "0x44")

Round[10].output	69	C4	E0	**44**	6A	7B	04	30	D8	CD	B7	80	70	B4	C5	5A
Round[10].k_sch	13	11	1D	7F	E3	94	4A	17	F3	07	A7	8B	4D	2B	30	C5
Round[10].s_row	7A	D5	FD	**3B**	89	EF	4E	27	2B	CA	10	0B	3D	9F	F5	9F
Round[10].s_box	7A	9F	10	27	89	D5	F5	0B	2B	EF	FD	9F	3D	CA	4E	**3B**
Round[10].start	BD	6E	7C	3D	F2	B5	77	9E	0B	61	21	6E	8B	10	B6	**49**
Round[09].k_sch	54	99	32	D1	F0	85	57	68	10	93	ED	9C	BE	2C	97	4E
Round[09].m_col	E9	f7	4E	EC	02	30	20	F6	1B	F2	CC	F2	35	3C	21	**07**

6.4.4. Cas de l'erreur "0x90"

Dans le cas de l'erreur "*0x90*", nous avons modifié le même octet que précédemment pour l'erreur "0x44" lors d'une injection lors des cycles 49 ou 50. En réalisant les mêmes opérations en sens inverse et en connaissant la valeur de la clef de chiffrement, nous notons que nous avons supprimé l'opération "*Mixcolumns*". En effet, nous obtenons la valeur "*0x2F*" en sortie de l'opération "*MixColumns*" (Tableau 6-12) et cette valeur correspond à la valeur en sortie de l'opération "ShiftRows" (ligne *Round[09].s_row* du Tableau 6-8).

Tableau 6-12 : Valeurs des opérations obtenues en sens inverse (faute "0x90")

Round[10].output	69	C4	E0	**90**	6A	7B	04	30	D8	CD	B7	80	70	B4	C5	5A
Round[10].k_sch	13	11	1D	7F	E3	94	4A	17	F3	07	A7	8B	4D	2B	30	C5
Round[10].s_row	7A	D5	FD	**EF**	89	EF	4E	27	2B	CA	10	0B	3D	9F	F5	9F
Round[10].s_box	7A	9F	10	27	89	D5	F5	0B	2B	EF	FD	9F	3D	CA	4E	**EF**
Round[10].start	BD	6E	7C	3D	F2	B5	77	9E	0B	61	21	6E	8B	10	B6	**61**
Round[09].k_sch	54	99	32	D1	F0	85	57	68	10	93	ED	9C	BE	2C	97	4E
Round[09].m_col	E9	f7	4E	EC	02	30	20	F6	1B	F2	CC	F2	35	3C	21	**2F**

La suppression de l'opération est possible du fait de l'utilisation du signal "*EnableMC*" (bit de sélection de l'entrée du multiplexeur) défini dans la Figure 6-19. Ainsi la faute induite a modifié le signal "*EnableMC*" et cette modification n'est pas exploitable pour retrouver une partie de la sous-clé de chiffrement. En effet, cette modification conduit à réaliser 8 rondes normales puis 2 rondes 10 (sans l'opération "*MixColumns*"). Si cette attaque était exploitable alors on serait en mesure d'attaquer directement l'AES complet. Par contre, si la faute est injectée au début du chiffrement, alors l'attaque est potentiellement exploitable avec plusieurs valeurs de chiffrés fautés puisque nous supprimerons toutes les opérations "*MixColumns*" et

l'AES ne serait plus que des opérations de substitution, de décalage et d'ajout de clefs. Avec une analyse différentielle, l'attaque permettra de retrouver potentiellement des valeurs de clefs.

6.5. Conclusion

Dans ce chapitre, nous avons présenté un crypto-processeur AES qui avait été sécurisé contre les attaques par fautes pour une intégration ASIC. Les contremesures utilisent les approches de redondance temporelle et de DDR réalisant les calculs sur les deux fronts de l'horloge. Pour vérifier la robustesse de ce circuit, des injections de fautes par émulations avaient permis de montrer que la contremesure est très efficace contre les fautes transitoires [Maistri 2007].

L'efficacité de la contremesure a été vérifiée lors de l'utilisation de surtensions pour induire des fautes de chiffrement. En effet, nous avons montré que les surtensions permettent de modifier uniquement le contenu des bascules utilisateurs et toute nouvelle écriture dans les bascules permet d'effacer la faute induite. Ce type de faute est par définition une faute transitoire qui est détectable par la contremesure implantée. Cependant, certaines erreurs de chiffrement n'ont pas été détectées lors de surtensions sur les fronts montant ou descendant de l'horloge. En effet, cette catégorie a été observée lorsqu'on applique une surtension sur le front montant du dernier cycle d'horloge de la dernière ronde. Ces erreurs sont cependant non exploitables pour récupérer la clef de chiffrement puisqu'elles sont présentes lors de la dernière opération du chiffrement. Si les fautes sont induites sur les fronts descendants de l'horloge, des erreurs non détectées ont également été obtenues mais pour quasiment tous les cycles d'horloge. Comme précédemment ces erreurs ne sont pas exploitables puisqu'elles correspondent à des remises à zéro du circuit.

Des attaques laser ont également été conduites sur le crypto-processeur après avoir réalisé un placement-routage particulier afin de réduire la durée des campagnes. Lors des précédents chapitres, nous avons montré que l'utilisation du laser permet d'obtenir des fautes rémanentes dans la configuration et du fait de l'architecture du système sécurisé nous avons supposé que la contremesure est moins efficace face à ce type de fautes. En effet, lorsque nous implantons le crypto-processeur sécurisé sur une plate-forme reconfigurable FPGA, la détection des erreurs de chiffrement par la contremesure à base de redondance DDR n'est pas assez efficace puisque près de 18% des positions de tirs ont conduit à des erreurs non détectées.
Pour sécuriser notre circuit, une recherche des éléments de tuiles CLB (logique ou interconnexions) conduisant aux erreurs non détectées a été réalisée. Cependant, aucune catégorie spécifique n'a été identifiée comme plus critique vis-à-vis des erreurs de chiffrement. Le cas des fautes uniques et plus particulièrement les modifications induites sur les interconnexions ont également été discutés.

La contremesure implantée dans le crypto-processeur étant nettement moins efficace contre les fautes rémanentes, nous avons proposé une modification de la contremesure précédente. Cette nouvelle

contremesure a été implantée dans le circuit FPGA et évaluée uniquement par des injections de fautes laser car les surtensions n'avaient pas eu d'effets exploitables. Cette amélioration a permis de rendre la détection des erreurs de chiffrement nettement plus efficace contre les fautes rémanentes puisque seulement 0,1% des positions de tirs conduisent à des erreurs non détectées contre 18% dans la version précédente. Cependant même si la contremesure permet de détecter nettement plus de fautes induites, nous avons pu montrer que certaines erreurs non détectées sont exploitables pour retrouver des octets des sous-clés de ronde. Avec différentes valeurs de chiffrés erronés, il est potentiellement possible de retrouver partiellement ou complètement les clefs de chiffrement. Cette amélioration de la contremesure a permis d'augmenter la proportion de détection de fautes induites dans les S-box mais pas dans les chemins de clefs. Nous avons montré que nous avons forcé à zéro un octet de la clef ou supprimé une connexion. Cette suppression de la connexion a par exemple permis de "sauter" une opération. Ainsi, en définissant cette connexion sur plusieurs bits tels qu'en Dual-Rail, il devrait être nettement plus difficile de fauter cet élément.

Conclusions et Perspectives

Dans notre société actuelle, la sécurité et/ou la sûreté de fonctionnement de systèmes à base d'électronique deviennent de plus en plus importantes. A cause des avancées technologiques miniaturisant de plus en plus les semi-conducteurs, les systèmes à base d'électronique sont nettement plus sensibles aux perturbations intentionnelles ou non. Ces perturbations peuvent conduire à des conséquences plus ou moins importantes sur le fonctionnement des systèmes mais également permettre d'obtenir des informations confidentielles. Dans un souci de réduction des coûts de développement et de fabrication, un grand nombre de systèmes sécuritaires ou non utilisent des plateformes configurables et/ou reconfigurables. Nous avons donc décidé d'étudier la sensibilité d'une famille de plate-forme reconfigurable face aux injections de fautes par tirs laser et par surtensions.

L'étude a permis de montrer que les circuits FPGA de type SRAM sont sensibles aux perturbations extérieures et plus particulièrement aux tirs laser après une préparation du circuit ainsi qu'à l'application de surtensions.

Les premiers résultats de tirs laser sur la configuration du circuit montrent que le nombre de fautes induit par ces tirs n'est pas déterministe. En effet lors d'un tir laser unique, indépendamment de la localisation spatiale et de la taille du spot laser, nous avons observé que certaines positions de tirs peuvent induire soit aucune faute soit des erreurs simples ou multiples de la configuration. Un résultat intéressant de cette étude est qu'il est possible d'obtenir des positions de tirs conduisant à des fautes uniques même avec un spot laser assez large et ce cas particulier à été étudié.

Les FPGA étant principalement constitués de blocs de logique configurable (CLB), une classification de la sensibilité des différents éléments les constituant (logique et interconnexions) a été dressée. Il ressort de cette étude que les interconnexions sont les éléments les plus sensibles des tuiles CLB. L'utilisation d'un outil développé au laboratoire TIMA a permis de réaliser une analyse plus fine des différents types de bits configurant ces éléments en fonction de la taille du spot laser. Lorsque des tirs laser changent l'état de bits configurant la logique interne, la majorité des modifications concerne le contenu des LUT et les multiplexeurs internes. Les entrées des slices et les connexions de type Hex sont les éléments dont la probabilité de modification est la plus importante pour les éléments constituant les interconnexions. Une étude a également été réalisée sur les tuiles de mémoire embarquée (BRAM).

Puisque les interconnexions possèdent une sensibilité importante, une étude des effets des modifications des connexions a été conduite. Les effets des modifications dépendent de l'état initial de la connexion : initialement présente ou non. Dans la majorité des cas, les tirs laser touchent des connexions initialement absentes et ne permettent pas de créer une connexion. Si la connexion existe, des modifications de la configuration auront pour principaux effets d'ajouter d'autres segments de connexion ou de supprimer la connexion.

Nous avons donc montré la sensibilité des différents éléments des tuiles CLB face à des attaques laser ainsi que les principaux effets des modifications des connexions. Contrairement aux tirs laser, l'application de surtensions sur l'alimentation du cœur du composant permet uniquement de modifier le contenu des bascules utilisateurs. Les surtensions sont nettement plus dangereuses que les sous-tensions dans le cas d'attaques car seules les surtensions ont permis de modifier le contenu des bascules utilisateurs. Il faut par ailleurs atteindre des amplitudes d'impulsion élevées pour observer des effets significatifs.

Dans de précédents travaux de recherche, il a été montré une différence de sensibilité entre les bits de configuration initialement à '1' et à '0'. Pour comprendre ces différences, une étude des zones de sensibilité et de leurs formes a été conduite. Pour des bits initialement à '1', la forme de la zone de sensibilité est proche d'un cercle alors que celle de bits à '0' s'apparente à un croissant. Ces différences de formes peuvent être expliquées par une dissymétrie de la structure du point mémoire à 5 transistors.

Pour conduire efficacement une attaque laser, un attaquant peut influer sur plusieurs paramètres du banc. Par exemple, il peut réaliser plusieurs tirs sur une même position ou changer la durée de l'impulsion laser. Nous avons montré que les tirs laser ne sont pas reproductibles à 100%. Certaines positions ne permettent pas d'obtenir à chaque fois des erreurs de configuration ou des listes de bits modifiés identiques lors de plusieurs tirs. Cependant, le taux de reproductibilité est assez élevé permettant de ne pas réaliser un nombre de tirs trop important pour obtenir un effet sur la configuration. Si l'on désire par contre obtenir un motif donné, le nombre de tirs nécessaire peut être plus important car la reproductibilité en nombre de bits est nettement plus faible.
L'autre paramètre que l'attaquant peut modifier est la durée de l'impulsion qui influe directement sur la valeur de l'énergie émise par le laser. Ce paramètre a une influence sur le nombre d'erreurs induites dans la configuration. Cependant du fait du non-déterminisme du nombre de fautes, il est assez difficile de trouver la "bonne" durée d'éclairement permettant d'obtenir un nombre de fautes désiré. Par ailleurs, la sensibilité des bits est également dépendante de l'énergie émise. Lors de notre étude des bits ont vu leur état modifié uniquement à partir de certaines durées d'éclairement.

Suite à toutes ces caractérisations statiques, des attaques par applications de surtensions et par tirs laser ont été menées sur un crypto-processeur AES sécurisé contre les injections de fautes. Les attaques par surtensions ont permis de montrer l'efficacité de la contremesure face à des fautes transitoires. Par contre, lors des attaques laser, nous avons pu montrer que la contremesure utilisant de la redondance temporelle

est nettement moins efficace lorsqu'elle est implantée sur un FPGA plutôt que sur un ASIC. Cette inefficacité est principalement due aux fautes rémanentes induites par les tirs laser dans la configuration. Une amélioration rendant plus efficace la contremesure a été proposée, implantée. Cette amélioration permet d'augmenter très nettement la détection d'erreur de chiffrement lorsque les fautes sont induites dans les S-box. Cependant, certaines erreurs de chiffrement non détectées nous ont permis de retrouver des octets des sous-clés de ronde et ceci du fait que les fautes sont induites dans les chemins de données. Pour supprimer une opération, nous avons juste du modifier un bit d'un multiplexeur de l'architecture AES. Ainsi pour contrecarrer ce type d'attaque, il faudrait sécuriser ces chemins en ajoutant par exemple soit coder les signaux de commande sur plusieurs bits soit utiliser le Dual-Rail.

Ce travail de thèse offre plusieurs perspectives de recherche.

Nous avons étudié les effets des attaques par injections de fautes sur une famille ancienne de FPGA de type SRAM, qui est a priori plus résistante aux perturbations que les familles dans les technologies les plus récentes. Il serait intéressant de valider les résultats obtenus sur une technologie de fabrication récente de FPGA de type SRAM mais également sur une autre technologie de points de mémorisation telle que la Flash.

Nous avons vu que la configuration du FPGA étudié est sensible aux perturbations laser. Pour se prémunir des injections de fautes, il faudrait utiliser la reconfiguration partielle présente sur différentes familles de plateformes reconfigurables. En effet en détectant la position de la faute, il est possible de reconfigurer partiellement la zone dans laquelle la ou les fautes sont présentes pour les supprimer.

Outre l'utilisation des techniques d'injection de fautes par application de surtensions ou par tir laser, d'autres techniques peuvent être utilisées. L'électronique étant sensible aux variations de champ électromagnétique, il faudrait vérifier si des attaques par fautes électromagnétiques conduisent ou non aux mêmes effets que les attaques laser.

Bibliographie

[Actel 2002] Actel, *"Design Security with Actel FPGA"*, Presentation, August 2002

[Actel 2003] Actel, *"Implementation of Security in Actel's ProAsic and ProAsic Plus"*, Application Notes, September 2003

[Actel 2009] Actel, *"ProAsic3 Flash Familly FPGAs"*, Datasheet, February 2009

[Altera 2002] Court issues preliminary injunction against Clear Logic in Altera litigation, Altera Corp., July 2002.

[Altera 2005] Altera Corporation vs. Clear Logic Incorporated (D.C. No. CV-99-21134), United States Court of Appeals for the Ninth Circuit, April 2005.

[Altera 2006] Altera, *"Protecting Intellectual Property Through FPGA Design Security"*, Design Perspective, October 2006

[Altera 2008-1] Altera, *"Error Detection and Recovery Using CRC in Altera Devices"*, Application Note 357, July 2008

[Altera 2008-2] Altera, *"Configuration Handbook (Complete Two-Volume Set)"*, Device Documentation, November 2008

[Anderson 1996] R. Anderson, M. Khun, *"Tamper Resistance a cautionary Note"*, USENIX Workshop on Electronic Commerce Proceedings, pages 1-11, November 1996

[Anderson 1997] R. Anderson, M. Khun, *"Low cost Attacks on Tamper-Resistant Devices"*, International Workshop of Security Protocols, volume 1361 of LNCS, pages 125-136, April 1997

[Anis 2005] M. Anis, M. H. Aburahma, *"Leakage Current variability in Nanometer Technologies"*, International Workshop on System-on-Chip for Real-Time Applications, pages 60-63, July 2005

[Bar-El 2006] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, C. Whelan. *"The sorcerer's Apprentice Guide to Fault Attacks"*, volume 94 of IEEE, n°2, pages 370-382, February 2006

[Binder 1975] D. Binder, E. C. Smith, A. B. Holman. *"Satellite Anomalies from Galactic Cosmic Ray"*, IEEE transactions on Nuclear Science, volume 22, pages 2675-2680, December 1975

[Birkner 1992] J. Birkner, A. Chan, H. T. Chua, K. Gordon, B. Kleinman, P. Kolze, R. Wong *"A very-high-speed field-programmable gate array using metal-to-metal antifuse programmable elements"*, Microelectronics Journal, volume 23, pages 561-568, 1992

[Brier 2004] E. Brier, C. Clavier, F. Olivier. *"Correlation Power Analysis with a leakage model"*, Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 3156 of LNCS, pages 16-29, August 2004.

[Brown 1996] S. Brown, J. Rose, *"Architecture of FPGA and CPLDs: A tutorial"*, IEEE Design and Test of Computers, volume 13, pages 42-57, 1996

[Chari 2002] S. Chari, R. Rao, P. Rohatgi. *"Template Attacks"*, Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 2523 of LNCS, pages 13-28, August 2002.

[Chow 1962] W. T. Chow, W. H. Henrich, *"Re-programmable PLA"*, United States Patent n°3028659, April 1962

[Dhem 1998] J-F.Dhem, F.Koeune, P-A.Leroux, P.Mestré, J-J.Quisquater, J-L.Willems, *"A pratical Implementation of the Timing Attack"*, CARDIS'98, Volume 1820 of LNCS, pages 167-182, June 1998.

[Dipert 2000] B. Dipert. *"Cunning circuits confounds crooks"*, Electronic Design Strategy News (EDN), October 2000.

[Djellid-Ouar 2006] A. Djellid-Ouar, G. Cathebras, F. Bancel, *"Supply voltage glitches effects on CMOS circuits"*, Design and Test of Integrated Systems in Nanoscale Technology (DTIS), pages 257-261, September 2006

[Douin 2006] A. Douin, V. Pouget, F. Darracq, D. Lewis, P. Fouillat, P. Perdu, *"Influence of Laser Pulse Duration in Single Event Upset Testing"*, IEEE Transactions on Nuclear Science, volume 53, pages 1799-1805, August 2006

[Duzellier 2000] S. Duzellier, D. Falguere, L. Guibert, V. Pouget, P. Fouillat, R. Ecoffet. *"Application of Laser Testing in Study of SEE Mechanisms in 16-Mbit DRAM"*, IEEE Transactions on Nuclear Science, volume 47, pages 2392-2399, December 2000

[Ferrigno 2008] J. Ferrigno, M. Hlavac, *"When AES blinks: introducing optical side channel"*, Information Security IET, volume 2, pages 94-98, September 2008.

[FIPS 140-2] Federal Information Processing Standards Publication, *"Security Requirements for Cryptographic Modules"*, May 2001

[FIPS PUB 197] Federal Information Processing Standards Publication, *"Advanced Encryption Standard (AES)"*, FIPS PUB 197, November 2001

[FIPS PUB 46-3] Federal Information Processing Standards Publication, *"Data Encryption Standard"*, FIPS PUB 46-3, October 1999

[Greene 1993] J. Greene, E. Hamdy, S. Beal, *"Antifuse Field Programmable Gate Arrays"*, IEEE Journal, pages 1042-1056, July 1993

[Guillemenet 2008] Y. Guillemenet, I. Hassoune, L. Torres, G. Sassatelli, *"FPGA non-volatiles à base de cellules mémoires magnétiques à écriture assistée thermiquement"*, Journée Nationale du Réseau Doctoral en Microélectronique (JNRDM), May 2008

[Gunneflo 1989] U. Gunneflo, J. Karlsson, J. Torin. *"Evaluation of error Detection Schemes Using Fault Injection by Heavy-Ion Radiation"*, Symposium on Fault-Tolerant Computing (FTCS), pages 340-347, June 1989

[Habling 1965] D.H Habling. *"Use of Lasers to Simulate Radiation Induced Transients on Semiconductors and Circuits"*, IEEE Transactions on Nuclear Science, volume 12, pages 91-100, December 1965

[Hamdy 1988] E. Hamdy, J. McCollum, S. Chen, S. Chiang, S. Eltoukhy, J. Chang, T. Speers, A. Mohsen, *"Dielectric-based Antifuse for Logic and Memory ICs"*, IEEE International Electron Devices Meetings Technical Digest, pp. 786-789, December 1988

[Hevia 1999] A. Hevia, M. Kiwi. *"Strength of two data encryption standard implementations under timing attacks"*. ACM transactions on Information and Systems Security, volume 2, pages 416-437, November 1999

[Karlsson 1994] J. Karlsson, P. Liden, P. Dahlgren, R. Johansson. *"Using Heavy-Ion Radiation to Validate Fault-Handling Mechanisms"*, IEEE Micro Magazine, volume 14, pages 8-23, February 1994

[Karri 2002] R. Karri, K. Wu, P. M. Yongkook. Kim, *"Concurrent error detection schemes for faulted-based sided-channel cryptanalysis of symmetric block ciphers"*, Computer-Aides Design of Integrated Circuits and Systems, IEEE Transactions on, volume 21, pages 1509-1517, December 2002

[Khelil 2008] F. Khelil, M. Hamdi, S. Guilley, J-L Danger, N. Selmane, *"Fault Analysis Attack on an FPGA AES Implementation"*, New Technologies Mobility and Security (NTMS), pages 1-5, November 2008.

[Kocher 1996] P. C. Kocher, *"Timing Attacks on Implementation of Diffie_Hellman, RSA, DSS and others Systems"*, CRYPTO'96, vol 1109 of Springer-Verlag, pages 104-113, August 1996.

[Kocher 1999] P.C. Kocher, J. Jaffe, B. Jun. *"Differential Power Analysis"*. Cryptology Conference an Advances in Cryptology, volume 1666 of LNCS, pages 388-397, 1999

[Kömmerling 1999] O. Kömmerling, M.G. Khun, *"Design Principles for Tamper-Resistant Smartcard Processors"*, Workshop on Smartcard Technology (SmartCard'99), pages 9-20, May 1999

[Lattice 2006-1] Lattice, *"LatticeXP SysConfig Usage Guide"*, Technical Note 1082, February 2006

[Lattice 2006-2] Lattice, *"LatticeECP2/M SysConfig Usage Guide"*, Technical Note 1108, September 2006

[Lattice 2006-3] Lattice, *"LatticeECP2/M Configuration Encryption Usage Guide"*, Technical Note 1109, September 2006

[Lattice 2009] Lattice, *"LatticeECP2/M Soft Error Detection (SED) Usage Guide"*, Technical Note 1113, January 2009

[Lesea 2005] A. Lesea. *"Jbits & reverse engineering"* (Usenet comp.arch.fpga), September 2005.

[Lima 2000] F. Lima, E. Costa, L. Carro, M. Lubaszewski, R. Reis, S. Rezgui, R. Velazco, *"Designing and testing a radiation hardened 8051-like micro-controller"*. Military and Aerospace Applications of Programmable Devices and Technologies, 2000

[Liu 1998] S. Liu, D. Lamp, S. Gangopadhyay, G. Sreenivas, S. S. Ang, H. A. Naseem, *"A new Metal-to-Metal Antifuse with Amorphous Carbon"*, IEEE Electron Devices Letters, volume 19, September 1998

[Maingot 2007] V. Maingot, J. B. Ferron, R. Leveugle, V. Pouget, A. Douin, *"Configuration errors analysis in SRAM-based FPGA: software tool and practical results"*, Microelectronics Reliability, Elsevier, volume 47, no. 9-11, pages 1836-1840, September-November 2007

[Maistri 2007] P. Maistri, P. Vanhauwaert, R. Leveugle, *"A Novel Double-Data-Rate AES Architecture Resistant against Fault Injection", September 2007,* International Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC07), pp 54-61, September 2007

[Maistri 2008] P. Maistri, R. Leveugle, *"Double-Data-Rate Computation as a Countermeasure against Fault Analysis",* Computers IEEE Transactions on*,* vol57, pp 1528-1539, November 2008

[Mangard 2003] S. Mangard, M. Aigner, S. Domonukus, "*A hightly regular and scalable AES hardware architecture*", IEEE Transaction on Computers, volume 52, issue 4, pages 483-491, April 2003

[Marple 1994] D. Marple, L. Cooke, *"Programming Antifuses in CrossPoint's FPGA"*, CICC 94, IEEE 1994 Custom Integrated Circuits Conference, pages 185-188, May 1994

[Marx 2008] W. Marx, V. Aggarwal, *"FPGA Are Everywhere – In Design, Test & Control"*, Magazine RTC, April 2008

[Merle 2006] A. Merle. *"Security testing for hardware product: the security evaluations practice"*, Minatec CrossRoad 2006, May 2006

[Messerges 2000] T.S. Messerges,*"Using Second-Order Power Analysis to Attack DPA Resistant Software",* Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 1965 of LNCS, pages 238-251, August 2000.

[More 2000] S. Moore, R. Anderson, M. Kuhn, *"Improving Smartcard Security Using Self-timed Circuit Technology"*, Asynchronous Circuit Design Workshop (ACiD 2000), February 2000

[Moss 2003] S. Moss, S. LaLumondiere. *"Picosecond Lasers for Single-Event Effects Testing"*, Aerospace Crosslink Article: Radiation in the Space Environment, volume 4, pages 20-25 ,Summer 2003

[Mulder 2005] E. De Mulder, P. Buysschaert, S. B. Örs, P. Delmotte, B. Preneel, G. Vandenbosch, I. Verbauwhede, "*Electromagnetic Analysis Attack on an FPGA Implementation of an Elliptic Curve Cryptosystem"*, EUROCON 2005, pages 1879-1882, November 2005

[Normand 1996] E. Normand. *"Single Event Upset at Ground Level"*, Nuclear Science, IEEE Transactions on, volume 43, pages 2742-2750, December 1996

[Note 2007] J-B. Note, E. Rannaud. *"From the bitstream to the netlist"*. Technical report, Département d'informatique de l'École Normale Supérieure, September 2007.

[OpenCores 2007] OpenCores, *"Basic DES Crypto Core"*, http://www.opencores.org, May 2007

[Örs 2003] S. B. Örs, E. Oswald, B. Preneel, *"Power-Analysis Attacks on an FPGA – First Experimental Results"*, Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 2779 of LNCS, pages 35-50, September 2003

[Pacalet 2005] R. Pacalet. *"Security of Security Hardware"*, Cours de l'École Nationale Supérieure des Télécommunications de Paris, December 2005

[Page 1985-1] D. W. Page, L. R. Peterson, *"Re-programmable PLA"*, United States Patent n°4508977, April 1985

[Page 1985-2] D. W. Page, *"Dynamic Data Re-programmable PLA "*, United States Patent n°4524430, June 1985

[Piret 2003] G. Piret, J-J. Quisquater *"A differential fault attack technique against SPN structures, with application to the AES and KHAZAD"*, Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 2779 of LNCS, pages 77-88, September 2003.

[Postel-Pellerin 2008] J. Postel-Pellerin, *"Fiabilité des Mémoires Non-Volatiles de type Flash en architectures NOR et NAND"*, Thèse de doctorat, December 2008

[Pouget 2007] V. Pouget, A. Douin, D. Lewis, P. Fouillat, G. Foucard, P. Perronnard, V. Maingot, J.B. Ferron, L. Anghel, R. Leveugle, R. Velazco, *"Tools and Methodology Development for Pulsed laser Fault Injection in SRAM-Based FPGAs"*, Latin American Test Workshop (LATW), March 2007

[Pouget 2008] V. Pouget, A. Douin, G. Foucard, P. Perronard, D. Lewis, P. Fouillat, R. Velazco, "*Dynamic Testing of an SRAM-based FPGA by Time-Resolved Laser Fault Injection*", IEEE International On-Line Testing Symposium (IOLTS), pages 295- 301, July 2008.

[Samson 1997] J. R. Samson, W. Moreno, F. Falquez. *"Validating Fault Tolerance Designs Using Laser Fault Injection"*, IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), pages 175-183, October 1997.

[Samson 1998] J. R. Samson, W. Moreno, F. Falquez. *"A Technique For Automated Validation Of Fault Tolerant Designs Using Laser Injection (LFI)"*, IEEE Symposium on Fault Tolerant Computing (FTCS), pages 162-167, June 1998.

[Samyde 2002] D. Samyde, S. Skorobogatov, R. Anderson, J-J. Quisquater, *"On a New Way to Read Data from Memory"*, First International IEEE Security in Storage Workshop, pages 65-69, December 2002

[Sauveron 2005] D. Sauveron, P. Dusart. *"Les cartes à puces. Sécurités et Attaques"*, Programme transversal SEFSI – Projets Supports Amovibles Légers et Sécurisés, May 2005

[Schindler 2000] W. Schindler. *"A Timing Attack against RSA with the Chinese remainder Theorem"*, Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 1965 of LNCS, pages 109-124, August 2000.

[Schmidt 2007] J. M. Schmidt, M. Hutter, *"Optical and EM Fault-Attacks on CRT-based RSA : Concrete Results"*, Proceeding of the Austrochip 2007, Verlag der Technischen Universität Graz, pages 61-67, 2007

[Selmane 2008] N. Selmane, S. Guilley, *"Practical Setup Time Violation Attacks on AES"*, European Dependable Computing Conference (EDCC), pages 91-96, May 2008

[Selmane 2009] N. Selmane, S. Bhasin, S. Guilley, T. Graba, J-L. Danger, *"WDDL is Portected Against Setup Time Violation Attacks"*, Fault Diagnosis and Tolerance in Cryptography, A paraître, September 2009

[Shamir 2004] A. Shamir, E. Tomer, *"Acoustic cryptanalysis – On nosy people and noisy machimes"*, Rump session of EuroCrypt 2004, May 2004

[Shang 2002] L. Shang, A. S. Kaviani, K. Bathala. *"Dynamic Power Consumption in Virtex-II FPGA Family"*. 10th International Symposium on Field-Programmable Gate Arrays (FPGA), pages 157-164, February 2002.

[Sharma 2002] A. K. Sharma, *"Semiconductor Memories: Technology, Testing and Relability"*, Wiley-IEEE Press, ISBN 978-0-7803-1000-1, September 2002

[Skorobogatov 2002] S. P. Skorobogatov, R. J. Anderson *"Optical Fault Induction Attacks"*, Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 2523 of LNCS, pages 2-12, August 2002.

[Skorobogatov 2005] S. P. Skorobogatov. *"Semi-invasive attacks – a new approach to hardware security analysis"*, Technical report 630, University of Cambridge, Computer Laboratory, April 2005.

[Standaert 2004] F-X. Standaert, S. B. Örs, B. Preneel. *"Power Analysis of an FPGA implementation of Rijndael: Is pipelining a DPA countermeasure?"*. Cryptographic Hardware and Embedded Systems Workshop (CHES), volume 3156 of LNCS, pages 30-44, August 2004.

[Telikepali 2003] A. Telikepali, *"Is your FPGA Design Secure"*, Xcell Journal, May 2003

[Trimberger 2007] S. Trimberger. *"Trusted design in FPGA"*. In Design Automation Conference, June 2007.

[Tuan 2007] T. Tuan, T. Strader, S. Trimberger, *"Analysis of Data Remanence in a 90nm FPGA"*, IEEE Custom Integrated Circuits Conference (CICC), pages 93-96, September 2007

[Ulogic 2007] Ulogic FPGA netlist recovery, October 2007.

[Wu 2001] K. Wu, R. Karri, "*Idle Cycles Based Concurrent Error Detection of RC6 Encryption*", IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), pages 200-2005, October 2001

[Xilinx 2005] Xilinx, *"Virtex II Pro and Virtex II Pro X Platform FPGA: Complete Data Sheet"*, Data Sheet 083, October 2005

[Xilinx 2006-1] Xilinx, *"Virtex-4 Configuration Guide"*, User Guide 071, January 2006

[Xilinx 2006-2] Xilinx, *"Viretx-5 FPGA User Guide"*, User Guide 190, October 2006

[Xilinx 2007-1] Xilinx, *"Virtex-II Platform FPGA: Complete Data Sheet"*, Data Sheet 031, November 2007

[Xilinx 2007-2] Xilinx, *"Virtex-II Platform FPGA User Guide"*, User Guide 002, November 2007

[Xilinx 2008] Xilinx, *"Security Solutions Using Spartan-3 Generation FPGA"*, White Paper 266, April 2008

[Xilinx 2009-1] Xilinx, " Xilinx Co-Founder Ross Freeman Honored as 2009 National Inventors Hall of Fame Inductee for Invention of FPGA", Press Release, February 2009

[Xilinx 2009-2] Xilinx, "*Virtex-5 FPGA Configuration User Guide*", User Guide 191, February 2009

[Xilinx 2009-3] Xilinx, "*Virtex-5 FPGA System Monitor*", User Guide 192, March 2009

[Zhao 2007] W. Zhao, E. Belhaire, C. Chappert, "*Circuit intégré non-volatile et reconfigurable basé sur la mémoire magnétique*", Colloque du GDR SOC-SIP (System-On-Chip & System-In-Package), June 2007

[Ziegler 1981] J.F. Ziegler, W.A. Lanford, "*The effect of Sea Level Cosmic Rays on Electronic Devices*", J. App. Phys, volume 52, pages 4305-4312., June 1981

Bibliographie de l'auteur

✓ Conférences internationales avec actes

- 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*** », "*International On-Line Test Symposium*" (IOLTS 2008), pp 289-294, juillet 2008, Rhodes (Grèce)

- G. Canivet, R. Leveugle, J. Clédière, F. Valette, M. Renaudin «***Characterization of Effective Laser Spots during Attacks in the configuration of a Virtex-II FPGA*** », "*VLSI Test Symposium*" (VTS 2009), pp 327-332, mai 2009, Santa Cruz (Californie, USA)

✓ Conférences internationales sans actes

- G. Canivet, R. Leveugle, J. Clédière, F. Valette, M. Renaudin «**Intentional Attacks on SRAM-based FPGAs**», *"Design of Circuits and Integrated Systems"* (DCIS 2008), présentation, novembre 2008, Grenoble (France)

✓ Colloques internationaux sans actes

- V. Maingot, J.B. Ferron, G. Canivet, R. Leveugle «**Fault Attacks on SRAM-based FPGAs: Analysis of Laser-induced Faults in a Virtex-II**», *"Users Suppliers European network for Information Technology security"* (USE-IT 2007), présentation, juillet 2007, Toulouse (France)

- G. Canivet, J. Clédière, R. Leveugle, M. Renaudin, F. Valette «**Intentional Attacks on SRAM-based FPGAs**», *"Paca Security Trends In Embedded Security"* (PASTIS 2009), poster, décembre 2008, Gardanne (France)

✓ Conférences nationales avec actes

- G. Canivet, J. Clédière, R. Leveugle, M. Renaudin, F. Valette «**Injection de fautes sur composant Virtex-II XC2V1000**», *" Journée Nationale du Réseau Doctoral en Micro-électronique*" (JNRDM 2008), mai 2008, Bordeaux (France)

✓ Séminaires nationaux

- G. Canivet «**Attaques par fautes sur plates-formes reconfigurables** », "*Séminaire de cryptologie, codage et infrastructures sécurisées*", janvier 2009, Grenoble (France)

RESUME

La sécurité des traitements numériques est quelque chose d'important dans notre société actuelle. Un grand nombre d'applications nécessite de forts niveaux de sécurité et/ou de sûreté. Pour répondre à ces besoins, les applications utilisent souvent des composants ASICs. Les principaux problèmes de ce type de composant sont qu'ils sont dédiés à une application et nécessitent de forts volumes de production. Une autre approche possible consiste à utiliser des plates-formes reconfigurables telles que des FPGAs de type SRAM. Cependant, la mémoire de configuration de ces FPGAs est sensible aux perturbations, ce qui nécessite une étude spécifique. Cette thèse a pour objectif principal de caractériser les effets des injections de fautes par tirs laser et par application de surtensions dans ce type de composant.

Lors de ce travail, nous avons pu analyser pour un type de FPGA la sensibilité des différents éléments configurant la logique programmable et identifier les principaux types de modification des interconnexions. Les effets obtenus ont été étudiés en fonction de plusieurs paramètres : focalisation du faisceau laser ou amplitude des surtensions, durée des perturbations et énergie. Le déterminisme des effets a également été analysé. Il a été montré pour les attaques par laser que la forme des zones de sensibilité dépend de la valeur initiale du bit et une interprétation a été proposée. Suite à ces différentes caractérisations, un crypto-processeur AES sécurisé contre les injections de fautes a été implanté sur le FPGA et attaqué. Les différences de robustesse avec l'implantation ASIC ont en particulier été analysées et une amélioration des contre-mesures a été proposée, implantée et validée.

MOTS CLEFS

Sécurité, FPGA de type SRAM, Attaques par fautes, Attaques laser, Attaques par surtensions, AES

TITLE

Analysis of fault-based attack effects and secure design on a reconfigurable platform

ABSTRACT

Security of digital processing is important in our society. Many applications require high levels of security and/or safety. To meet these requirements, applications often use ASIC components. The main problems of such devices are that they are dedicated to one application and require high production volumes. Another possible approach is to use reconfigurable platforms such as SRAM-based FPGAs. However, the configuration memory of such FPGAs is sensitive to perturbation, thus requiring a specific study. The main goal of this thesis is to characterize the fault injection effects obtained in such devices using lasers and power glitches.

In this work, we analyze for a given FPGA the sensitivity of the elements configuring the logic and we identify the main types of modification patterns in interconnections. Fault attack effects have been studied with respect to several parameters: laser spot size or power glitch amplitude, perturbation duration and energy. The determinism of the effects was also analyzed. It was shown for the laser attacks that the shape of sensitive areas depends on the initial bit state and an interpretation was proposed. Based on these characterizations, an AES crypto-processor secured against fault-based attacks was implemented on the FPGA and then attacked. Robustness differences with the ASIC implementation were particularly analyzed and countermeasure improvements were proposed, implemented and validated.

KEYWORDS

Security, SRAM-based FPGA, Fault attacks, Laser attacks, Power glitch attacks, AES

ISBN : 978-2-84813-136-8