Système embarque de mesure de la tension pour la détection de contrefaçons et de chevaux de Troie matériels

Maxime Lecomte

N°d'ordre NNT : 2016LYSEM018

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Ecole des Mines de Saint-Etienne

Ecole Doctorale N° 488
Sciences, Ingénierie, Santé

Spécialité de doctorat : Microélectronique

Soutenue publiquement le 05/10/2016, par :
Maxime Lecomte

Système embarqué de mesure de la tension pour la détection de contrefaçons et de chevaux de Troie matériels

Devant le jury composé de :

Belleville, Marc	Directeur de recherche	CEA-LETI	Président
Danger, Jean Luc	Professeur	Télécom ParisTech	Rapporteur
Rouzeyre, Bruno	Professeur	Université de Montpellier	Rapporteur
Renaudin, Marc	CTO	Tiempo	Examinateur
Maurine, Philippe	Maitre de conférence	Université de Montpellier	Directeur de thèse
Fournier, Jacques	Chercheur	CEA-Tech	Encadrant de thèse

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	X. Delorme, Maître assistant
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	Ph. Lalevée, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'Etat ou d'une HDR)

ABSI	Nabil	CR	Génie industriel	CMP
AUGUSTO	Vincent	CR	Image, Vision, Signal	CIS
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BADEL	Pierre	MA(MDC)	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BEIGBEDER	Michel	MA(MDC)	Informatique	FAYOL
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	MA(MDC)	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR1	Génie Industriel	FAYOL
CHRISTIEN	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
DOUCE	Sandrine	PR2	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FOREST	Valérie	MA(MDC)	Génie des Procédés	CIS
FOURNIER	Jacques	Ingénieur chercheur CEA	Microélectronique	CMP
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GAVET	Yann	MA(MDC)	Image Vision Signal	CIS
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche	Mécanique et ingénierie	CMP
NORTIER	Patrice	PR1		SPIN
OWENS	Rosin	MA(MDC)	Microélectronique	CMP
PERES	Véronique	MR	Génie des Procédés	SPIN
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROBISSON	Bruno	Ingénieur de recherche	Microélectronique	CMP
ROUSSY	Agnès	MA(MDC)	Génie industriel	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

Mise à jour : 01/02/2016

«Timeo Danaos et dona ferentes»

Éneide(livre II, v. 49), Virgile

«Look, if we built this large, wooden badger...»

Monty Python and the Holy Grail

Sommaire

Glossaire

AES : Advanced Encryption Standard
RSA : algorithme de cryptographie asymétrique
ASIC : Application-Specific Integreted Circuit
ATPG : Automatic Test Pattern Generation
BEOL : Back End Of the Line
BIST : Built-In Self-Test
CAO : Conception Assistée par Ordinateur
CI : Circuit intégré
CLB : Configurable Logic Blocks
CTM : Cheval de Troie Matériel
CMOS : Complementary metal oxide semi-conductor
DRC : Design Route Checking
EM : Électromagnétique
FEOL : Front End Of the Line
FIB : Focused Ion Beam
FPGA : Field-Programmable Gate Array
GDSII : Graphical Database System
HDL : Hardware Description language
IP : Intellectual Property
layout : représentation d'un circuit intégré en formes géométriques
LUT : Look Up Table
LVS : Layout vs. Schematic
MEB : Microscope Électronique à Balayage
netlist : description des connections des éléments d'un circuit intégré
PCB : Printed Circuit Board
PUF : Physical Unclonable Function
RTL : Register Transfer Level
RO : Ring Oscillator
VHDL : VHSIC Hardware Description Language
VHSIC : Very High Speed Integrated Circuit

1 Introduction

Ce premier chapitre présente le domaine de la sécurité matérielle et les menaces liées à l'intégrité des circuits intégrés. On distingue deux menaces principales, les chevaux de Troie matériels et les contrefaçons. Après avoir introduit les menaces à l'intégrité, nous parcourons l'état de l'art des méthodes de lutte contre ces dernières. Parmi les principales méthodes, une majorité consiste en l'analyse de grandeurs physiques des circuits afin de détecter une différence avec un circuit intègre. Cet état de l'art souligne qu'une information locale des grandeurs physiques d'un circuit permet une meilleure détection qu'une information globale et que l'utilisation de capteurs embarqués est un moyen efficace pour obtenir cette mesure locale. Notre travaille porte donc sur l'utilisation d'un réseau de capteurs embarqués pour détecter les contrefaçons et les chevaux de Troie matériels.

1.1 Sécurité des circuits intégrés

Ces dernières décennies, l'essor des technologies numériques a bouleversé notre société, que ce soit au niveau industriel, économique ou au niveau des modes de vie individuels. Les systèmes et objets connectés se retrouvent aujourd'hui dans une multitude d'applications, dans lesquelles ils traitent de grandes quantités d'information. En effet, tous les secteurs industriels sont concernés, des composants discrets (électroménager, capteurs automobiles, ...) aux systèmes (type système industriel) en passant par les télécommunications, le médical ou l'armement. Par conséquent, ces systèmes intelligents manipulent des données privées ou sensibles à fortes valeurs ajoutées.

Les Circuits Intégrés (CI) constituent la clé de voûte des appareils intelligents ou communicants modernes. Un CI est un ensemble de composants électroniques (transistors qui permettent, en fonction de leur organisation et de leurs interconnexions, de réaliser des opérations logiques élémentaires. L'adjonction de ces blocs élémentaires permet la conception de fonctionnalités complexes. Ces fonctionnalités peuvent être, par exemple, l'exécution d'un calcul numérique, la mémorisation d'une donnée ou l'exécution d'un algorithme numérique.

1.1.1 Transistor

L'informatique utilise la notation binaire pour représenter les données, c'est-à-dire qu'une variable élémentaire n'admet que deux valeurs possibles (par exemple "0" ou "1").

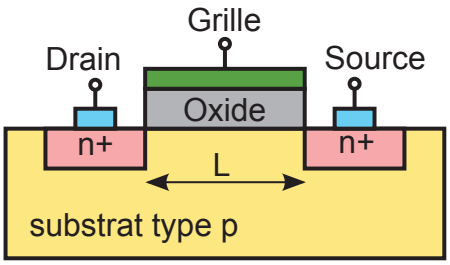

Figure 1.1: Transistor NMOS.

Électroniquement, ces deux états sont associés à deux niveaux de tension, un niveau de tension bas pour "0" et un niveau de tension haut pour "1". Le transistor est donc la brique de base des systèmes numériques qui manipule ces niveaux de tension et informations associées. Physiquement, le transistor est une structure gravée dans du silicium.

Le silicium est utilisé, car ce matériau est un semi-conducteur. On classe les matériaux en fonction de leur conductivité électrique en trois catégories: les isolants, les semi-conducteurs et les conducteurs. La conductivité électrique est la capacité d'un matériau à laisser circuler des charges électriques et donc à laisser un courant le traverser. Un matériau isolant empêche la circulation de courant, alors qu'un matériau conducteur permet la circulation des charges électriques. Un semi-conducteur a une faible conduction, il peut cependant conduire sous l'effet d'un apport d'énergie (excitation thermique ou électrique). De plus, les propriétés électriques d'un semi-conducteur peuvent être modifiées localement en implantant des atomes dans le matériau dans le but d'augmenter la densité d'électrons ou de trous.

La technologie de transistor utilisé aujourd'hui est appelée CMOS, la figure 1.1 illustre un transistor CMOS de type N. Le substrat est dopé P (augmentation de la densité de trous) et deux zones sont dopées N (augmentation de la densité d'électrons). Un oxyde permet d'isoler la grille du substrat. Lorsque la tension entre la grille et la source est suffisamment élevée le passage des électrons entre le drain et la source est possible. Le transistor permet donc de créer un "interrupteur" électronique commandé en tension.

Depuis les années 70, le nombre de transistors présents sur les CIs a grandement augmenté. Il est en effet passé de quelque quelques milliers de transistors dans les années 70 à plus d'un milliard pour certains CIs modernes (2016). En 1965, Gordon E. Moore (cofondateur d'Intel) exprime l'idée que la densité de transistors double chaque année, puis révise cette "loi" par le fait que la densité double tous les 2 ans. Cette règle empirique a orienté la feuille de route de l'industrie microélectronique et s'est donc vérifiée jusqu'à aujourd'hui. Cependant, les limitations physiques font que cette progression atteint une limite, car des dimensions de quelques nanomètres sont mises en jeu ce qui nous rapproche des dimensions limitées par la taille des atomes (de l'ordre de

l'Ångström). Ainsi, ces dernières décennies la densité de transistors présents sur les circuits a augmentée de façon exponentielle permettant aujourd'hui d'atteindre des nœuds technologiques de l'ordre de la dizaine de nanomètres. Une technologie est désignée pas la largeur de grille de ses transistors notée L sur la figure 1.1.

Cette augmentation du nombre de transistors a été rendue possible par leur réduction d'échelle. Cette réduction de la taille des transistors présente trois avantages technologiques. L'augmentation du nombre de transistors sur un circuit permet d'implémenter des fonctionnalités de plus en plus complexes. Ensuite, l'augmentation des fréquences de fonctionnement accélère la vitesse des calculs. Enfin, la réduction de la consommation a permis l'émergence des appareils nomades.

1.1.2 Fabrication des circuits intégrés

Physiquement, les CIs sont composés de différentes couches. Une couche de silicium sur laquelle est gravé un ensemble de transistors et plusieurs couches métalliques qui construisent des connexions entre ces transistors. La couche de métal au-dessus des transistors est appelée "métal 1". Les niveaux supérieurs sont appelés "métal 2", métal 3" .... La n-ème couche est donc nommée "métal n". Les différentes couches de métal sont reliées par des vias. Le nombre de couches d'un circuit complexe peut être supérieur à 20. Le processus de fabrication des CIs est devenu plus complexe et coûteux au fur et à mesure des redimensionnements technologiques. Pour atteindre ces niveaux de complexité, de nouveaux logiciels et de nombreuses ressources sont nécessaires pour concevoir les circuits. Celles-ci ont nécessité de nouveaux procédés de plus en plus fins et de plus en plus difficiles à maîtriser. Cette section vise à expliquer les principales étapes dans la réalisation d'un circuit intégré depuis la conception jusqu'à la distribution.

Spécification Dans un premier temps, à partir des besoins utilisateurs, une spécification est écrite. À partir de l'idée du produit final, on décrit les besoins que doit remplir le CI fini. Cela comprend les protocoles de communication, les fonctionnalités ou bien les performances de fonctionnement. De plus, les contraintes à respecter sont indiquées. Une contrainte peut être la fréquence d'horloge qui influe sur le débit de calcul du CI. La taille du circuit peut être limitée par des problèmes d'intégration et de rendements de fabrication, celle-ci va définir le nombre de transistors maximal en fonction du nœud technologique utilisé. La consommation de courant du circuit s'avère très importante pour les systèmes nomades, qui peuvent être alimentés sur batterie ou par récolte d'énergie (par exemple les badges d'authentification). La chaleur dissipée par le circuit est un autre critère qui doit être considéré, le produit peut être dégradé s'il chauffe de façon trop importante sans un système de dissipation adapté. Les plages de température sur lesquelles le produit doit fonctionner correctement doivent aussi être

prises en compte, cela peut être important pour des usages dans des environnements spécifiques (ex. spatial ou militaire).

Conception Ces spécifications établies, la conception consiste en l'implémentation technique du cahier des charges. Un langage de description haut niveau (HDL) "*Hardware Description Language*" est utilisé, dans un premier temps, pour décrire l'architecture logique ainsi que le comportement des blocs bas niveaux afin d'effectuer les fonctionnalités souhaitées. En parallèle, de la partie logique on retrouve la partie analogique et les mémoires. Les parties analogiques assurent plusieurs tâches comme la gestion d'alimentation, en régulant ou surveillant la tension du circuit. La modulation de signaux de communication par ondes radio (ex: wifi, bluetooth). Un bloc peut prendre en charge la gestion de la fréquence d'horloge, en régulant, multipliant ou divisant la fréquence du signal. Les mémoires sont constituées de grilles régulières de points mémoires contenant "en général" un bit de donnée chacun. Elles représentent une grande partie de la surface utilisée du circuit. Elles sont donc optimisées et générées séparément du reste de la logique. À cela viennent s'ajouter les blocs fournis par des fournisseurs tiers. En effet, la complexité des circuits impose parfois de recourir à des blocs de propriétés intellectuelles (IP) externes.

À partir de la description HDL, un logiciel de synthèse est utilisé pour générer la liste de portes logiques et de leurs interconnexions implémentant l'architecture décrite. Cette liste est appelée "*netlist*". Cette liste est convertie en cellules élémentaires provenant d'une bibliothèque de cellules. Ces cellules sont des structures optimisées pour réaliser des fonctions élémentaires. Pour une même fonction, différentes cellules peuvent être disponibles. En effet, elles peuvent être conçues différemment pour s'adapter aux contraintes de consommations ou de vitesses. Par exemple, elles peuvent être adaptées au nombre de portes qui lui sont connectées en aval, ou en fonction de son temps de traversée. Ces bibliothèques de cellules sont fournies par le fondeur et utilisées par les logiciels de conception.

Une fois les différentes parties obtenues (logique, mémoires, blocs analogiques et IPs des fournisseurs tiers), un logiciel de placement-routage est appliqué pour placer tous les blocs, générer l'arbre d'horloge et la grille d'alimentation et faire les interconnexions entre les blocs. Une phase de vérification est ensuite effectuée. La première d'entre elles est la vérification de la correspondance entre le résultat obtenu et les fonctionnalités souhaitées. Cette étape s'appelle LVS ("*Layout vs. Schematic*"). La seconde phase appelée DRC ("*Design Route Checking*") est ensuite appliquée. Elle permet de vérifier que les contraintes physiques liées aux procédés de fabrication (ex. écartement entre deux pistes) sont bien respectées.

Une chaîne de logiciels est donc utilisée pour les différentes étapes énoncées précédemment (synthèse, placement routage, vérification, etc.) Celle-ci est coûteuse et le marché est dominé par quatre compagnies (Cadence, Mentor Graphics, Synopsys et Magma Design Automation). Le résultat de conception est appelé "layout". Le layout représente la géométrie des différentes couches physiques du circuit permettant ensuite de fabriquer le circuit, le fichier associé est appelé *GDSII* ("*Graphical Database System*").

Fabrication Le GDSII, obtenu lors de l'étape de conception, est ensuite envoyé dans une fonderie pour fabrication. Le fichier GDSII est utilisé pour créer un jeu de masques qui est utilisé pendant les étapes de la fabrication. Certaines entreprises prennent en charge la conception et la fabrication, d'autres sous-traitent la fabrication à des entreprises extérieures parfois dans des pays tiers. La fabrication est un processus long (~ 3 mois) nécessitant une salle blanche (un environnement sans poussière). Ces salles blanches sont équipées d'équipements coûteux.

Le silicium utilisé pour fabriquer les transistors est utilisé sous forme de "wafer". Les wafers sont des disques de silicium sur lesquels sont fabriqués les circuits. Le nombre de circuits fabriqués par wafer dépend de leur taille. Un wafer peut contenir plusieurs milliers de circuits pour les plus petits.

Pour travailler à une échelle nanométrique, un procédé de lithographie est utilisé. Il consiste, dans un premier temps, à appliquer une fine couche de résine photosensible sur le wafer. Ensuite, cette résine est exposée à un rayonnement ultraviolet au travers d'un masque et d'une lentille. Sur le masque est inscrit le motif à graver et une lentille permet de réduire l'échelle du motif imprimé sur la résine. On applique ensuite un solvant sur la surface. Les parties de la résine qui ont été exposées (qui n'ont pas été protégées par le masque) vont être dissoutes. Ainsi on peut graver la résine avec une grande précision, cette résine gravée est utilisée pour protéger une partie du silicium et travailler localement. Le processus de fabrication consiste en différentes étapes de dépôt et de gravure sur des wafers. La photolithographie permet de graver ou de doper des zones précises. On procède ainsi par dépôt et gravure pour construire les couches successives jusqu'au niveau de métal supérieur.

Vérification Suite à la fabrication, une phase de vérification et de test est appliquée sur les wafers en utilisant des cartes à pointes. Une série de tests fonctionnels ou paramétriques est effectuée pour sélectionner les circuits ne présentant pas de défaillances. Pour cela, des vecteurs de test logique sont appliqués sur les entrées des circuits et les résultats obtenus sur les sorties sont comparés avec ceux attendus. En plus des tests logiques, des tests paramétriques sont effectués afin de vérifier que les spécifications (vitesse, consommation) sont bien respectées. Cela consiste en la mesure

des grandeurs physiques telles que la fréquence maximale de fonctionnement ou la consommation de courant.

Découpage et intégration Suite à la séquence de test sous pointe, les wafers sont découpés afin de séparer les circuits. Puis, chaque circuit est encapsulé (mis en boitier et connecté électriquement aux broches du boitier). Les circuits encapsulés sont alors testés une nouvelle fois pour vérifier que cette étape n'a pas altéré leur fonctionnement. Les circuits sont ensuite intégrés sur le PCB du produit final, ils sont soudés au circuit imprimé et donc potentiellement connecté à d'autres circuits intégrés. Le produit final peut ensuite être distribué sur le terrain. Dans le cas d'une carte à puce, le circuit est directement encarté pour être ensuite vendu.

De sa conception à sa distribution, un circuit peut voyager à travers plusieurs continents et passer dans les mains de nombreux acteurs auxquels ont peu accorder un niveau de confiance variable en fonction du contexte de fabrication pour chaque acteur de la chaine de production. Pour une entreprise prenant en charge la conception et la fabrication, toutes les étapes peuvent être considérées comme sûres. Dans le cas d'une entreprise sous-traitant sa production, le niveau de confiance attribué aux étapes de fabrication est plus faible, voire nul.

1.1.3 Sécurité des circuits intégrés

Les CIs étant utilisés dans de nombreuses applications, dont certaines sont critiques (militaire, santé, finance). Ces dernières demandent un haut niveau de fiabilité et de sécurité. Ainsi l'étude des moyens d'attaques et de protections des systèmes d'information est cruciale dans un contexte où ces systèmes sont de plus en plus répandus. Les deux principaux motifs d'attaques d'un composant intégré sont:

- la récupération de données sensibles ou privées (ex. clé de chiffrement),
- la volonté de nuire au fonctionnement d'un système applicatif plus complexe.

Les principaux moyens d'attaques d'un système manipulant de l'information sont:

- le hacking social: cela consiste à influencer ou espionner les personnes concevant ou utilisant le système,
- le hacking informatique qui consiste à utiliser des failles dans les algorithmes mis en œuvre ou dans les protocoles associés,

- le hacking matériel qui consiste à utiliser une faille dans l'implémentation matérielle des algorithmes et protocoles.

La fonction d'un circuit intégré est d'être le support physique de l'information et de sa manipulation. La sécurité matérielle consiste à étudier les attaques et contremesures pouvant être mises en œuvre pour garantir la confidentialité, l'intégrité et l'authenticité des données manipulées.

Confidentialité : les données ne sont pas compréhensible à un individu ou une entité non autorisée.

Intégrité : les données ne peuvent pas être modifiées de façon non autorisée.

Authenticité : les circuits sont ce qu'ils indiquent, ils ne peuvent pas être remplacés.

Dans le but de garantir ces propriétés, des protocoles sécurisés et des algorithmes cryptographiques sont intégrés dans le circuit. Ces algorithmes sont considérés mathématiquement sûrs. On distingue trois types d'algorithmes principaux, les algorithmes de chiffrement symétriques, les algorithmes de chiffrements asymétriques et les fonctions de hachage. Les algorithmes de chiffrement symétrique utilisent une clé unique pour chiffrer et déchiffrer des données. Cette clé est un secret partagé par l'émetteur et le récepteur des données. Il est donc nécessaire que les deux parties possèdent la même clé au préalable. Ce type d'algorithme permet d'assurer la confidentialité des données. Par exemple, l'"*Advanced Encryption Standard*" (AES) est un algorithme de chiffrement symétrique par blocs largement utilisé. Les algorithmes de chiffrement asymétrique utilisent une clé publique pour chiffrer et une clé privée pour déchiffrer les données, ce qui permet, par exemple un échange de clé privée. Cet algorithme peut aussi être utilisé pour générer et vérifier des signatures ce qui permet d'assurer l'authenticité de l'émetteur. Un exemple d'algorithme largement utilisé est le RSA. Il est basé sur la complexité de la factorisation de produit de deux grands nombres premiers. Les fonctions de hachage permettent de créer une empreinte de la donnée. Ainsi par comparaison d'empreintes ce type de fonction permet de s'assurer de l'intégrité des données.

Bien que ces algorithmes soient mathématiquement robustes, leur implémentation matérielle peut présenter des failles. Dans le cadre de la sécurité matérielle, on distingue deux types d'attaques: les attaques par injection de fautes et les attaques par observations, dites par canaux auxiliaires.

1.1.4 Attaques par injections de fautes

Les attaques par injection de fautes consistent à perturber physiquement le circuit cible de façon à modifier une donnée manipulée ou stockée dans ce dernier. Les principaux modèles de faute sont le "bit set" mettre la valeur d'un bit à '1', "le bit reset" mettre la valeur d'un bit à '0', le "bit flip" inverser l'état d'un bit ('0' vers '1' ou '1' vers '0') et le "stuck at" verrouiller l'état d'un bit. Les principaux moyens d'injections sont décrits dans les paragraphes suivants.

Perturbation du signal d'horloge Le principe est de perturber le signal d'horloge. On peut en effet introduire des violations de contraintes temporelles. Par exemple, en ne donnant pas le temps nécessaire à un calcul pour se terminer, le résultat échantillonné sera erroné [Ago+10]. Un défaut de cette technique est que l'ensemble du circuit est altéré, affecté.

Perturbation sur l'alimentation L'idée est de modifier temporairement la tension de l'alimentation [Bar+10]. Ce qui a pour effet de modifier les délais de propagation des portes et d'induire des violations de contraintes temporelles. Là encore, l'intégralité du circuit est affectée par l'attaque.

Injection d'impulsion électromagnétique L'idée est de générer localement de forts courants transitoires au sain du circuit en générant une impulsion EM à son voisinage [Deh+12]. Cette technique permet d'injecter des fautes de manière relativement locale.

Injection laser L'idée est d'illuminer une partie du circuit avec un faisceau laser afin d'induire des photocourants qui vont modifier transitoirement la valeur de sortie d'une ou plusieurs portes [SA03]. L'injection laser permet d'injecter des fautes avec une grande résolution spatiale et temporelle, mais nécessite des équipements coûteux.

Perturbation de la tension de polarisation du substrat L'idée est d'injecter directement dans le circuit des courants (et donc des fautes) au moyen d'une pointe au contact du substrat. Cette méthode permet d'induire des fautes de manière localisée [Mau+12]. Cette méthode est semi-invasive, car elle requiert un accès direct au substrat.

Des méthodes de cryptanalyse ont été développées pour exploiter de telles fautes. La finalité des attaques sur les algorithmes peut être de récupérer la clé secrète (cela a pour effet de rompre la confidentialité) ou encore de modifier le déroulement du code afin d'en exécuter certaines parties sans en avoir les droits. Par exemple, l'attaque sur AES de [PQ03] utilise plusieurs couples de chiffrés fautés et non fautés pour récupérer la clé sécrète utilisée par cet algorithme.

1.1.5 Attaques par canaux auxiliaires

La manipulation de l'information par un circuit intégré provoque différents effets physiques mesurables qui peuvent être dépendants des données manipulées ou des instructions exécutées. Les attaques par canaux auxiliaires ont pour but d'exploiter ces grandeurs mesurables pour extraire les secrets manipulés, par des moyens statistiques. Les principaux phénomènes observés sont présentés dans les paragraphes suivants:

Le temps de calcul de certaines parties des algorithmes peut dépendre des données manipulées. En mesurant ce temps il est donc possible d'obtenir des informations sur les données manipulées [Koc96].

La consommation de courant des circuits varie en fonction des données manipulées. Elle est également exploitée pour extraire des informations [Koc+11].

Les émanations électromagnétiques dépendent elles aussi des données manipulées, c'est en effet une image locale du courant. Cette localité permet d'extraire plus facilement les secrets [QS01].

Les émissions acoustiques Des fuites peuvent aussi être détectées dans le domaine acoustique [Gen+13]. Toutefois ce canal est peu exploité.

Les émissions de photons Les composants CMOS émettent des photons lors de la commutation des portes et donc en fonction de leur activité. Analyser les photons émis permet donc de récupérer des données [Sch+12].

De façon analogue aux attaques par injection de fautes, les analyses des canaux auxiliaires peuvent être utilisées pour récupérer les clés secrètes par des méthodes statistiques. L'attaque par analyse différentielle (DPA) proposée dans [Koc+99] est emblématique à ce sujet. Elle permet à partir d'un jeu de traces, obtenu par analyse d'un canal auxiliaire, de récupérer la clé secrète manipulée par un algorithme.

1.1.6 Attaques invasives

Les attaques invasives consistent à accéder directement aux composants du circuit pour modifier son comportement ou récupérer des informations. Ce sont des attaques qui détériorent le circuit. Par exemple, des procédés de rétro-conception [TJ11] permettent de récupérer des informations sur le circuit. La finalité de ces méthodes peut être de retrouver l'implémentation des blocs de propriétés intellectuelles. Ces méthodes peuvent

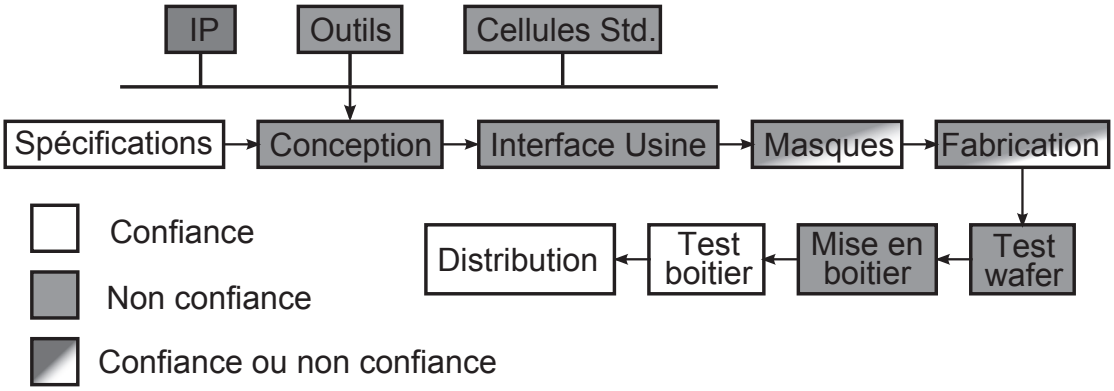

Figure 1.2: Niveaux de confiance des différentes étapes de production [DAR07]

utiliser des technologies de microscopie optique ou électronique. Dans le cas de méthodes optiques, il est nécessaire de retirer le boitier grâce à des méthodes mécaniques ou chimiques. Un procédé peut également consister en la suppression successive de chaque couche du circuit afin de pouvoir obtenir une image de chaque couche et ainsi retrouver un maximum d'information sur les interconnexions.

Les attaques par observation ou par injection de fautes ont été étudiées dans le domaine de la sécurité matérielle depuis bientôt 20 ans. Aujourd'hui, des contremesures prévenant des injections de fautes et des analyses par canaux auxiliaires sont largement implémentées dans les circuits sécurisés. Les recherches continuent aujourd'hui. Cependant, ce mémoire n'adresse pas ces problématiques. En effet, dans un contexte des modifications des modes production et de distribution des circuits qui sont advenues ces dernières décennies, d'autres menaces doivent être considérées. Ce mémoire adresse ces menaces qui sont présentées dans la section suivante.

1.2 Menaces sur l'intégrité physique des composants

Avec la délocalisation d'une partie des moyens de production des circuits et d'une partie des équipes de conception, la vérification de l'intégrité des circuits dans le but de détecter des modifications malveillantes est apparue récemment comme un enjeu important. Cela vient s'ajouter aux contrefaçons potentielles. En effet, de la phase de spécification jusqu'à l'obtention des premières puces, et plus particulièrement lors des étapes de conception matérielle, un design peut être corrompu par des personnes malintentionnées. La figure 1.2 montre les différentes étapes de la chaîne de production des CIs et donne une estimation des niveaux de confiance. Toute étape dont la confiance n'est pas assurée peut permettre à un employé ou une entité mal intentionnée d'altérer le produit final et donc le système dans lequel il est exploité. Une telle modification est appelée insertion de Cheval de Troie Matériel (CTM).

Additionnellement à cette 1-ère menace sur l'intégrité des composants, des circuits non originaux ou réutilisés peuvent être insérés dans la chaine de distribution. De tels circuits sont alors qualifiés de contrefaçons. Elle peut être une copie non autorisée. Elle peut être fabriquée par une entité non accréditée. Elle peut avoir des caractéristiques hors spécifications, être un produit en fin de vie vendu comme neuf ou encore avoir un marquage incorrect [Gui+14].

1.2.1 Moyens d'infections

Afin d'évaluer les risques liés à l'intégrité des CIs, il est nécessaire d'identifier à quelles étapes de la conception et de la fabrication un CTM peut être inséré.

Spécification

La phase de spécification est considérée comme sûre selon [DAR07]. Cependant si cette étape est sous-traitée, il est possible de modifier les spécifications afin de réduire l'efficacité du circuit en dimensionnant un paramètre ou de faciliter une future infection (par exemple s'assurant que le circuit offrira une surface disponible suffisante pour une infection), ou ajouter une des fonctions malveillantes.

Conception

La phase de conception est l'étape durant laquelle il est le plus aisé de modifier les fonctionnalités ou les paramètres du CI. En effet, un employé mal intentionné est alors en mesure de modifier la description HDL afin d'y inclure des fonctions malveillantes. De plus, l'utilisation d'outils extérieurs (cellules standards ou logiciels propriétaires IP) constitue une source de vulnérabilité. Des cellules logiques mal dimensionnées peuvent réduire l'efficacité du circuit ou augmenter son coût.

Composants extérieurs

Pour respecter les contraintes «Time To Market», des blocs préconçus (IP: Intellectual Property) sont utilisés. Pour des raisons de propriété intellectuelle, les IPs fournis par une tierce partie et intégrés dans un circuit le sont de manière à ne pas pouvoir remonter au schéma original de ces blocs. Cela empêche de vérifier le contenu du bloc ajouté sur le circuit et que ce dernier n'inclut pas de fonction non désirée. Par exemple, dans le cas

où cette brique fonctionnelle serait reliée au bus principal d'un processeur, elle pourrait accéder ou modifier une grande quantité d'information circulant dans le circuit.

Logiciels de conception

La conception des circuits intégrés repose sur une suite de logiciels prenant en charge différentes étapes (synthèse, placement-routage . . .). De façon similaire au développement logiciel, la confiance accordée aux résultats de compilation dépend de la confiance accordée aux éditeurs des logiciels utilisés. Les rares outils de Conception Assistée par Ordinateur (CAO) utilisés pour la conception microélectronique sont des outils propriétaires (Synopsys, Cadence Design Systems, Mentor Graphics ou Magma Design Automation). Les codes de ces outils n'étant pas accessibles il est difficile de vérifier leur contenu. Une faille dans l'un d'entre eux permettrait d'ajouter des fonctionnalités pendant la synthèse, ou de modifier des paramètres modifiant par conséquent les performances du CI. Cependant, ces logiciels sont en général considérés comme sûrs.

Bibliothèque de cellules standards

Ces bibliothèques contiennent plusieurs centaines d'éléments. La modification de l'une de ces cellules standards peut avoir un impact sur les performances globales du circuit ou son fonctionnement. Une cellule modifiée peut par exemple avoir une consommation accrue ou une efficacité temporelle dégradée. Cela peut influencer le comportement global du circuit ou sa longévité [Lin+09].

Fabrication

Durant cette étape, un attaquant peut vouloir modifier les fonctionnalités du circuit. Un obstacle à cette infection est que le fondeur ne possède pas les informations fonctionnelles du circuit ce qui rend une infection difficile. Cependant, le risque d'obtenir les informations correspondant au design par des moyens détournés doit être considéré. De plus par rétro-ingénierie un attaquant peut potentiellement identifier des points d'intérêts nécessaires pour mener une infection discrète impactant le fonctionnement du CI. Le cas où la netlist est connue pendant la fabrication doit donc être considéré. Considérant que l'attaquant possède les moyens d'interpréter et de modifier le GDSII, le risque que ce dernier puisse concevoir un ou plusieurs masques infectés est réel. De plus en utilisant une sonde ionique focalisée (FIB: Focused Ion

Beam) il est possible de modifier la structure d'un circuit en coupant ou ajoutant des connexions ou bien retirer une épaisseur donnée de silicium [Hel+13].

Test

La phase de test ne permet pas en elle même une infection. Mais celle-ci doit être considérée dans notre analyse. En effet, lors de cette étape, des vecteurs de test, déterminés lors de la conception, doivent être appliqués sur les entrées du circuit afin de vérifier que ses réponses correspondent à celles attendues. Une infection modifiant trop visiblement les fonctionnalités du circuit risque donc d'être détectée rapidement [Kar+10]. Pour donner un caractère furtif à son infection, un attaquant a donc trois possibilités. Premièrement, il peut négliger la phase de test et faire en sorte que les vecteurs incriminants soient évités. Il peut également prendre connaissance des vecteurs de test avant infection et infecter le circuit en conséquence. Ainsi le test ne couvrira pas les nœuds liés au CTM. Enfin, le CTM peut être conçu pour être déclenché sur de très rares évènements (c.-à-d. de valeurs précises sur plusieurs connexions ou séquences particulières d'état difficilement accessible lors du test). Dans ce cas, un test pensé pour détecter les erreurs de production ne pourra pas détecter l'infection, à moins d'accroître la complexité et le coût du test.

Intégration

La phase d'intégration permet de modifier l'environnement du circuit. De telles modifications peuvent être effectuées dans le but d'intercepter les entrées ou sorties du circuit ciblé. Par exemple, même dans le cas où tous les CIs soudés sur le circuit imprimé sont de confiance, un attaquant peut faire en sorte qu'une piste laisse fuir des données [Moe+15b].

Ainsi par leur caractère furtif et leur dangerosité, les CTMs représentent un risque pour la sécurité numérique et ont été abordés dans plusieurs rapports militaires, notamment par le département de la défense américain [def05] et par le département de la défense australien [Bea+11]. Par ailleurs, le problème de la délocalisation d'une partie de la chaîne de fabrication des circuits à usage militaire a été abordé dans [Ade08]. De plus, en 2012, la commission européenne a financé le projet HINT [Hin]. Ce dernier est un projet d'une durée de trois ans dont l'objectif principal a été de développer un système de vérification d'intégrité et démontrer sa pertinence pour un usage dans une application sur le terrain. Les travaux présentés dans ce mémoire sont en grande partie réalisés dans le cadre de ce projet HINT.

1.2.2 Cheval de Troie matériel

Un cheval de Troie matériel est une modification malveillance d'un circuit intégré. Cela se traduit par un layout non intègre et distinct de celui attendu par l'entité cliente ou l'usager (ex. entreprise de conception sous-traitant la fabrication). Un cheval de Troie est composé de deux parties: le déclencheur et l'actionneur.

Déclencheur

Le déclencheur, appelé «Trigger» en anglais, est une micro fonction intégrée, scrutant l'apparition d'une condition rare de signaux analogiques ou logiques, connu seulement des personnes malveillantes. Une fois la condition rare admise, le déclencheur active l'actionneur. L'évènement déclencheur peut être généré à l'extérieur (signal externe ou conditions physiques spéciales) ou généré à l'intérieur du circuit (état interne, configuration spéciale des données ...). De plus, le déclencheur peut être combinatoire lorsque la condition recherchée est le résultat d'une opération logique sur plusieurs signaux, ou séquentiel lorsque la condition de déclenchement est une succession d'états particuliers du circuit ou d'une sous partie du circuit (ex.: la machine à état finit).

Actionneur

L'actionneur réalise quant à lui la fonction nocive. Cette fonction peut provoquer un déni de service, c'est-à-dire mettre hors service une partie du CI ou tout le circuit. Dans ce cas l'effet de l'actionneur peut se traduire par une diminution des performances du circuit (ex. diminution de la fréquence de fonctionnement maximale du circuit ou vieillissement accéléré) ou par un rendement de production inférieur. Dans une application de sécurité, l'actionneur peut provoquer la fuite des données sensibles telles que les clés secrètes utilisées par un algorithme cryptographique. En fonction de son effet, un actionneur peut donc être explicite lorsque des signaux ou des blocs logiques sont directement ajoutés, supprimés ou désactivés ou implicite lorsque l'effet ne pas être directement et aisément observé, comme dans le cas de l'adjonction d'informations cachées sur des canaux auxiliaires du circuit.

La figure 1.3 illustre les deux parties qui constituent un CTM en présentant un exemple de porte NOR infectée. Dans cet exemple le déclencheur est une porte ET qui surveille les signaux d'entrés A et B. Dans le cas d'un évènement spécifique (ici A=1 et B=1) le déclencheur envoie un signal à l'actionneur. Si l'actionneur reçoit un signal provenant du déclencheur il active l'effet néfaste du CTM (inversion de la sortie de la porte NOR).

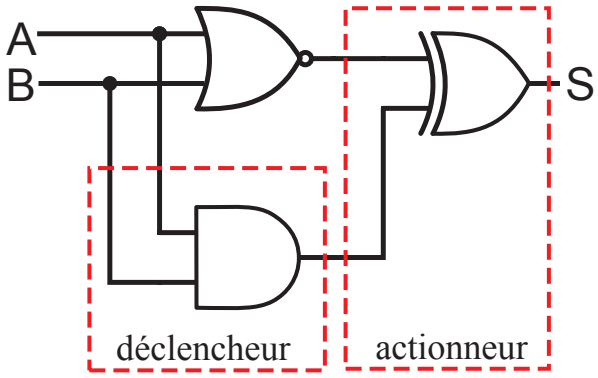

Figure 1.3: Porte NOR infectée

Plusieurs taxonomies de CTM ont été proposées. Tout d'abord, en 2009, une première taxonomie est basée sur les caractéristiques du déclencheur et de l'actionneur [Cha+09a]. La même année, une seconde taxonomie est proposée [Wan+08a], celle-ci s'appuie sur trois caractéristiques: les caractéristiques physiques du CTM (taille, impact sur le layout...), les caractéristiques du déclencheur et enfin les caractéristiques de l'actionneur. En 2010, cette taxonomie a été étendue dans [Kar+10], afin, de prendre en compte le niveau d'abstraction auquel a eu lieu l'infection et l'emplacement du CTM parmi les blocs constitutifs du circuit. Enfin en 2015, une taxonomie étant elle même une adaptation et une extension de la précédente [Moe+15b] a été proposée. Cette dernière, représentée par la figure 1.4. Elle s'articule autour de 8 critères qui sont:

- l'étape d'**insertion** durant le procédé;
- la **fonctionnalité**, le CTM a-t-il un impact fonctionnel?;
- le niveau d'**abstraction** dans le flot conception auquel a eu lieu l'infection ;
- le mécanisme d'**activation** (le déclencheur);
- l'**effet** (l'actionneur);
- l'impact sur la **structure** physique du CI;
- le **type logique** (combinatoire ou séquentielle);
- la **position** du CTM dans le circuit.

Ces critères sont développés dans les paragraphes suivants.

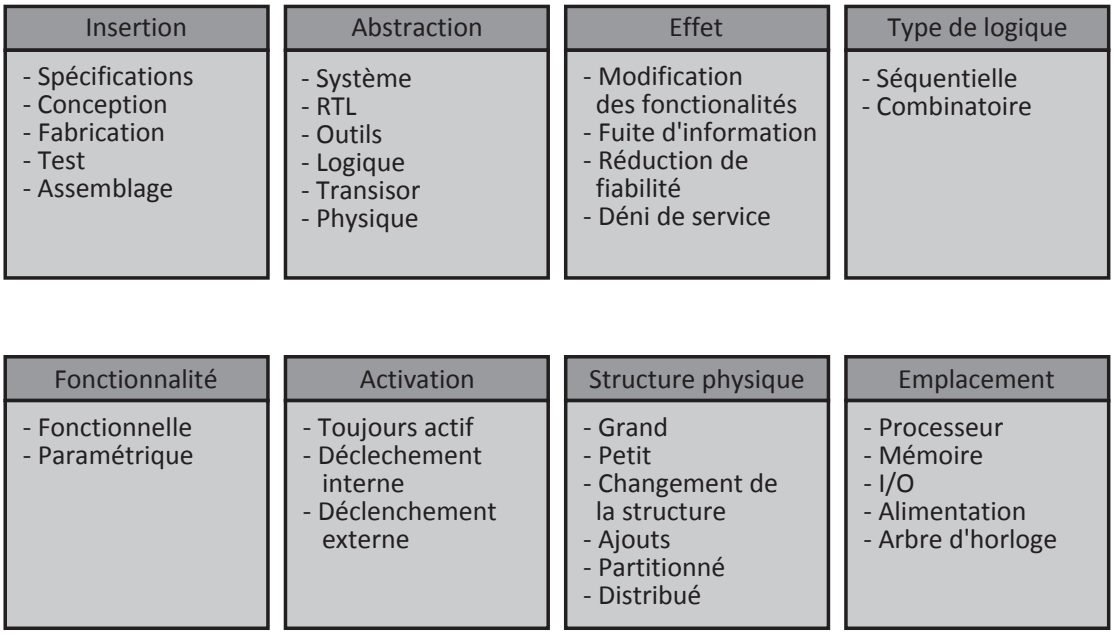

Figure 1.4: Taxonomie des CTMs [Moe+15b]

Insertion Cette caractéristique considère l'étape de la chaîne de production durant laquelle l'infection a eu lieu. Celle-ci peut avoir lieu à partir de la phase de spécification jusqu'à la phase d'intégration. (comme évoque précédemment dans la section 1.2.1)

Fonctionnalité Ce critère indique si le CTM affecte les fonctionnalités du circuit infecté ou s'il modifie les paramètres intrinsèques du circuit tels que sa consommation de courant ou ses délais de propagation. Dans le cas où le CTM modifie des fonctionnalités du circuit (CTM "fonctionnel"), cela signifie que des portes logiques ou des connexions ont été retirées, modifiées ou ajoutées. Si le CTM est "paramétrique", cela signifie que des modifications des caractéristiques géométriques ou physiques des éléments du circuit ont été effectuées.

Abstraction Ce critère indique à quel niveau d'abstraction l'insertion du CTM a lieu. Un haut niveau d'abstraction peut correspondre aux spécifications du système ou à la description RTL (Register Transfer Level) ou HDL. Un bas niveau d'abstraction peut correspondre à une infection conduite au niveau du layout du circuit. Une insertion à un bas niveau d'abstraction est plus complexe à mettre en œuvre, mais cela permet d'obtenir un CTM plus difficile à détecter. Le niveau d'abstraction de l'infection limite la fonctionnalité du CTM. Une infection au niveau transistor aura généralement un effet paramétrique alors qu'une infection au niveau RTL aura un impact fonctionnel.

Activation Ce paramètre indique les conditions d'activation du CTM. Cela correspond donc aux types de déclencheurs décrits précédemment. Le CTM peut être activé en permanence ou déclenché par un stimulus externe ou interne. La plupart des CTMs

considérés dans la littérature ont un déclencheur interne surveillant des signaux du circuit [Mak08; Rad+08; Cha+09b].

Effet Cette caractéristique permet de classifier les CTMs selon leurs effets sur le CI infecté. Il peut aller de la fuite de données à la dégradation de performances en passant par le déni de service.

Structure physique Cette catégorie correspond aux caractéristiques physiques du CTM. Cela comprend la taille de l'infection, c.-à-d. le nombre de portes logiques utilisées pour réaliser cette dernière et à la dispersion du CTM, c.-à-d., le fait que ses constituants soient éloignés les uns des autres ou pas. Enfin cette catégorie comprend le fait que le CTM utilise des emplacements inutilisés par le circuit ou modifie plus largement le layout original.

Type logique Ce paramètre indique si le déclencheur est séquentiel ou combinatoire. S'il est combinatoire, il est déclenché par une condition spécifique sur certains de ses signaux. S'il est séquentiel, il est déclenché par une suite de conditions spécifiques.

Emplacement Cette caractéristique correspond à la position du CTM parmi les blocs composant le circuit. Il peut être situé dans le processeur, dans la mémoire, dans le bloc de gestion d'alimentation ou encore connecté sur le réseau d'horloge. En fonction de sa position, les effets néfastes potentiels seront différents. Une infection du processeur sera en mesure de modifier une instruction alors qu'une infection du réseau d'alimentation peut créer un déni de service global ou une baisse de performance.

1.2.3 Contrefaçon

On peut identifier sept types de contrefaçon de CI [Gui+14].

Contrefaçon issue des recyclage Un circuit est récupéré sur un circuit imprimé usagé et est marqué à nouveau pour être vendu en tant que circuit neuf.

Remarquage de circuit Les inscriptions sur le boitier ou sur le "die" (circuit sans le boitier) du circuit sont effacées, et de nouvelles inscriptions ne correspondant pas au produit sont inscrites.

Surproduction de circuits Les surproductions sont des lots fabriqués par des acteurs possédant le design original et les moyens de production nécessaires à sa fabrication.

Ceux-ci sont fabriqués hors du cadre du contrat d'origine. Les circuits obtenus sont donc des copies conformes vendues à l'insu des concepteurs.

Circuits hors spécifications Un acteur non autorisé peut vendre des composants hors spécification ou défectueux provenant de la chaine de production originale. Il s'agit là de circuits fonctionnels, mais n'ayant pas les caractéristiques souhaitées.

Clone de circuits Un circuit peut être cloné pour éviter les coûts de développement. La récupération des informations peut se faire par deux moyens: par rétro-ingénierie ou bien en obtenant directement les informations de conception par des moyens illégaux (hacking social).

Documents falsifiés Les documents liés à un produit peuvent être falsifiés, par exemple dans le but d'obtenir une fausse certification.

Altérations de circuit L'altération d'un circuit peut prendre la forme d'un CTM ou bien d'une porte dérobée ou encore de mécanisme permettant de réduire la durée de vie de produit. Il peut aussi s'agir de la modification du logiciel embarqué ou encore de sa structure physique.

Les contrefaçons peuvent donc être considérées comme des copies plus ou moins exactes d'un circuit. En cas de recyclage ou de surproduction par exemple le layout sera identique à celui d'un circuit original. Une comparaison de la structure physique ne permet donc pas de détecter ce genre de contrefaçon. Cependant, pour des contrefaçons qui ne sont pas des copies exactes comme des circuits ayant subits un remarquage ou copiés avec des informations incomplètes, une comparaison de la structure physique doit permettre de les repérer.

1.3 Luttes contre les CTMs et les contrefaçons

La première solution pour lutter contre la menace que présente l'introduction de CTMs est de résoudre le problème à sa source en garantissant la confiance de la chaîne de production. Cette solution est traditionnellement mise en place pour les cartes à puces sécurisées. Cependant pour une partie des CIs produits, cette solution est peu envisageable pour des raisons de coût et n'est pas la tendance adoptée par l'industrie compte tenu de l'évolution du contexte de fabrication des semi-conducteurs. D'autres voies ont donc été explorées.

La seconde solution consiste à vérifier l'intégrité d'un lot de circuits intégrés lors de sa réception après fabrication par une entreprise dont la confiance n'est pas assurée (c.-à-d. pour laquelle on n'a pas les moyens de vérifier toutes les étapes de fabrication). Cette solution implique de réaliser une étape post-fabrication permettant la vérification des performances et des fonctionnalités spécifiées lors de la conception, mais également de s'assurer qu'aucun CTM n'a été inséré. À notre connaissance, les premières recherches de détection de CTM ont été publiées en 2007. Parmi les solutions considérées, on peut proposer une classification. Une taxonomie de ces différentes approches, présentée par Moein et al. [Moe+15a] est donnée dans la figure 1.5. On distingue deux grandes familles, les méthodes dites destructives, car le circuit n'est plus fonctionnel après le processus de vérification, et les méthodes non destructives qui permettent de préserver les fonctionnalités du circuit testé, mais qui sont plus limitées dans leurs possibilités de vérification. Les différents types de techniques de détection sont développés dans des sections ultérieures.

Le troisième solution est la mise en place, durant la conception du circuit, de barrières augmentant la complexité d'une insertion furtive ou d'une modification malveillante. Cette protection peut être mise en place soit en masquant les points d'intérêt du circuit soit en facilitant la détection d'un éventuel CTM.

1.3.1 Obstacles à la vérification d'intégrité de CI

Les processus modernes de production posent des difficultés spécifiques à la vérification de l'intégrité des circuits pour diverses raisons. Premièrement, les réponses paramétriques des circuits sont affectées par les conditions expérimentales. Par exemple, les performances temporelles des portes logiques dépendent de la température ou de la tension d'alimentation. Ces difficultés peuvent être contournées en stabilisant les conditions environnementales et en effectuant des mesures répétées pour faire une moyenne afin de réduire le bruit de mesure.

Le second obstacle est le vieillissement des circuits («aging») [WP11]. C'est une dégradation inexorable des performances des composants au cours du temps due à leur utilisation. La figure 1.6 montre la dégradation de la tension de seuil d'un transistor PMOS pour un vieillissement de 10 ans et cela pour deux températures. L'impact de l'aging sur la détection de CTM n'a pas été approfondi à notre connaissance. Cependant, on peut supposer que dans le cas d'une méthode de vérification d'intégrité n'intervenant qu'en début de vie du produit cela n'impacte pas les résultats. Toutefois, si l'on souhaite vérifier l'intégrité d'un composant au cours de sa vie cet effet ne peut pas être négligé. Une méthode pour prendre en compte ce phénomène peut consister en la prédiction de la dégradation telle que présentée dans [PC07].

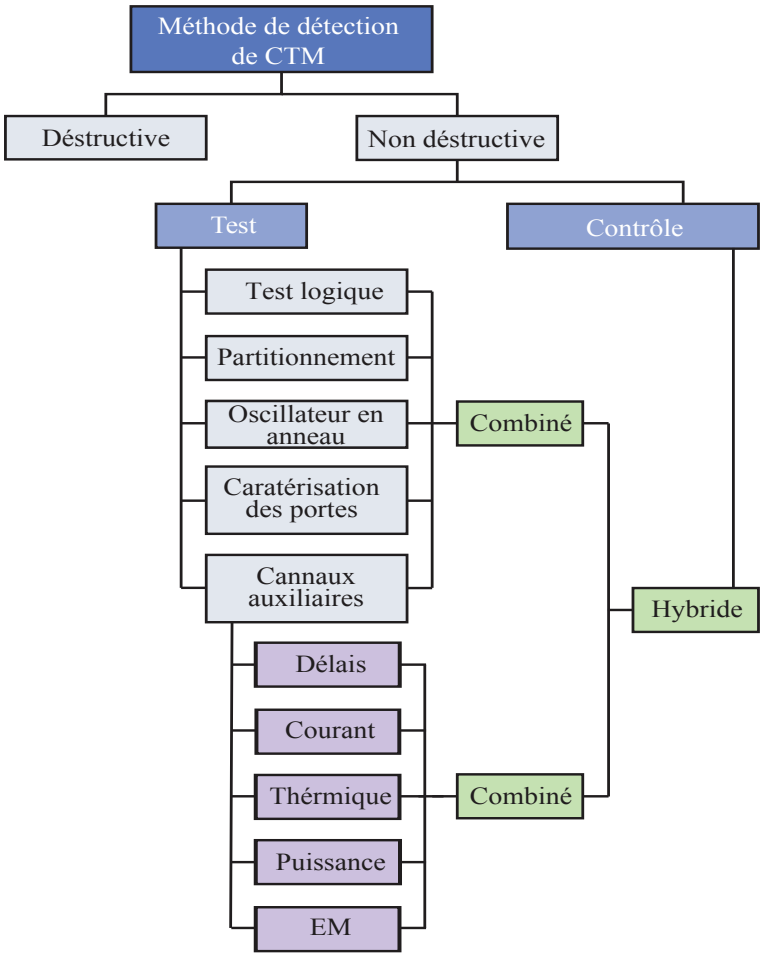

Figure 1.5: Taxonomie des méthodes des détections de CTM [Moe+15a]

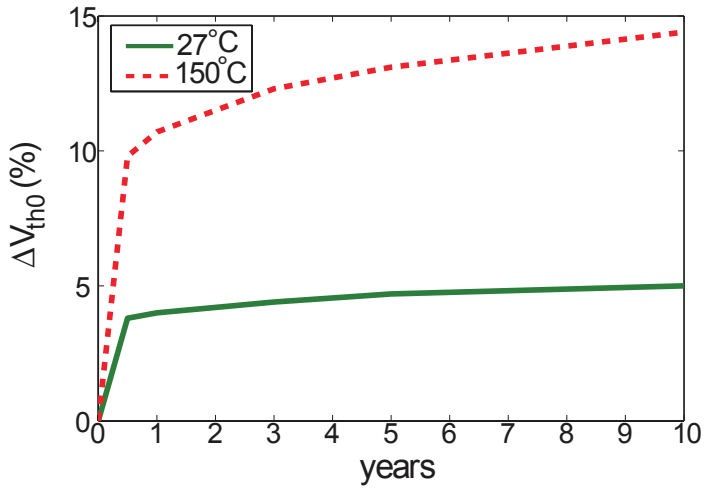

Figure 1.6: Dégradation de la tension de seuil d'un transistor PMOS [An+14].

Enfin, le dernier et principal obstacle à la vérification d'intégrité est l'impact des variations des procédés de fabrication. Avec la miniaturisation des technologies, les processus de fabrication sont plus difficilement contrôlables. Par exemple, il est plus difficile d'obtenir une bonne précision pour les grandeurs topologiques critiques ou pour les niveaux de dopant. Ce phénomène de variations incontrôlées s'accentue à chaque saut technologique [Pel+89]. Leurs causes sont trop diverses pour être modélisées et anticipées, elles sont donc considérées comme aléatoires. Ces variations ont pour effet de rendre chaque circuit unique et donc de rendre les méthodes de vérifications paramétriques difficiles à mettre en œuvre.

Par conséquent, une méthode de vérification doit être capable de distinguer l'impact d'une modification du design ou d'un ajout de logique de celui induit par les variations des procédés de fabrication. Concrètement, cela se traduit par une variation des performances temporelles ou de la consommation des portes logiques. On considère deux types de variations des procédés de fabrication:

- les variations "*inter-die*" qui sont les variations observées entre deux circuits; on les considère constantes pour les éléments d'un même circuit,

- les variations "*intra-die*" qui sont les variations observées entre les différents éléments d'un même circuit.

On considère que ces deux types de variations suivent des distributions normales de moyenne nulle. Elles peuvent induire des variations de plus de $20\,\%$ des performances temporelles en fonction des technologies considérées. L'étude de ces variations et des moyens de les contourner sera abordée dans les chapitres suivants.

1.3.2 Méthodes destructives

Vérifier l'intégrité d'un circuit est équivalent à vérifier que l'ensemble des composants présents sur le circuit est bien celui prévu lors de la spécification et la conception. Ainsi on cherche à vérifier qu'il n'y a pas d'éléments en plus ou en moins par rapport à ceux attendus et que leurs caractéristiques sont les bonnes. On nomme rétro-conception le processus qui sert à extraire les informations d'un circuit.

Le principe des méthodes destructives de vérification d'intégrité des composants est donc analogue à celui des méthodes de rétro-conceptions. Il consiste à accéder physiquement aux différentes couches constituant le circuit afin de pouvoir ainsi accéder aux signaux électriques ou de pouvoir inspecter par procédé optique les éléments du circuit. Dans le cas d'une analyse directe des portes logiques, il est nécessaire de retirer toutes les couches métalliques afin d'accéder à la couche de silicium. Une telle inspection permet

d'identifier les portes logiques. Un procédé de rétro-conception visant à remonter aux fonctions logiques nécessite les informations provenant des couches métalliques et de la couche de silicium. Ce type d'inspection ne permet pas de connaître l'état des points mémoire.

La complexité de tels procédés de rétro-conception augmente avec l'évolution des semi-conducteurs, car celle-ci s'accompagne d'une augmentation du nombre de portes logiques et d'une réduction de la taille physique des composants élémentaires constituant le circuit. Ceci implique un équipement coûteux permettant d'atteindre une résolution optique de l'ordre du nanomètre. Parce qu'une méthode destructive conduit à la décapsulation complète du circuit testé, elle ne permet pas de valider l'intégrité des composants sur le marché. Cependant, cette méthode peut être appliquée en complément d'autres méthodes. Ces autres méthodes (voir les sections 1.3.4 et 1.3.5) nécessitent un jeu de mesures provenant de circuits considérés comme intègres et authentiques afin de construire un modèle. Ce modèle de référence est appelé "*Golden Model*". L'idée est alors, une fois le modèle construit, de confronter les circuits au modèle afin de décider s'ils sont intègres ou non.

Une méthode destructive de détection de CTM appelée SEMBA est proposée dans [Cou+15] ("*Scanning Electron Microscope Based Acquisition technique*". Les auteurs proposent une méthode de rétro-conception moins coûteuse à base de MEB (Microscope électronique à balayage). Elle a pour but d'identifier et de localiser les caissons des transistors des circuits. Elle consiste en une décapsulation du circuit, d'un retrait des couches métalliques, d'une analyse par microscopie électronique à balayage puis de l'application d'outils de reconnaissances. La figure 1.7 donne deux images obtenues avec la technique SEMBA. Celle de gauche provient d'un circuit infecté, et celle de droite provient du même circuit non infecté. Les zones entourées permettent de déterminer visuellement qu'il y a une différence entre les deux layouts et donc de déduire la présence d'une infection. Dans cet exemple, la méthode nécessite un modèle de référence provenant d'un circuit de référence, mais la méthode peut également être appliquée à partir du layout seulement. Cette méthode est plus rapide qu'une méthode de rétro-conception classique, soit 40 min pour préparer le circuit, acquérir les images et les traiter. Cependant, les informations récupérées ne permettent pas de remonter aux fonctionnalités du circuit, car l'information liée aux niveaux de métal est perdue. Dans le cadre de la détection de CTM, cette méthode permet de détecter tout ajout, suppression ou modification logique donc elle n'est pas dépendante de la taille ou de l'activité électrique du CTM.

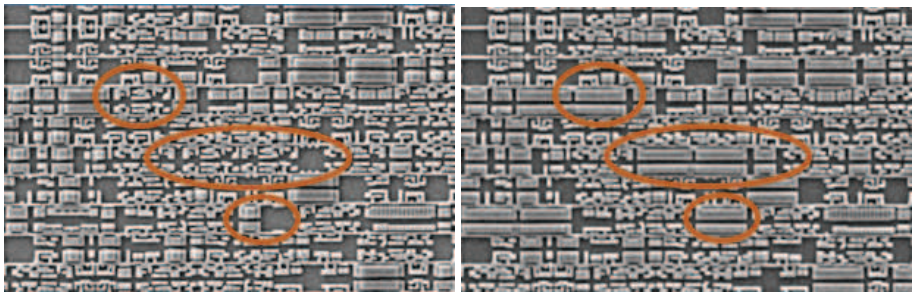

Figure 1.7: Détection de CTM par comparaison d'image provenant d'analyse au MEB [Cou15]. Circuit infecté à gauche, circuit non infecté à droite.

1.3.3 Test logique

Le test logique de circuit intégré est une étape incontournable avant la mise des pièces sur le marché. Elle permet de détecter les défauts de production sur un circuit intégré. Lors de cette étape, une succession de vecteurs de test appliquée en entrée du circuit et une liste de résultats attendus correspondant aux vecteurs choisis sont utilisées pour vérifier que le circuit est conforme. Cela nécessite donc de connaître de façon exacte l'intégralité de la «netlist» (la description de l'ensemble des portes logiques et de leurs interconnexions).

Deux principaux modèles de fautes sont considérés, le collage à 0 lorsque la valeur d'une connexion reste bloquée à l'état bas quelle que soient les entrées, et le collage à 1 qui correspond au blocage d'une connexion à l'état 1. L'objectif du test est donc de maximiser le nombre de fautes couvertes, le cas idéal étant de couvrir les collages à 0 et à 1 pour chaque connexion du circuit. La contrainte appliquée au choix des vecteurs est le nombre de vecteurs utilisés pendant le test. En effet, chaque vecteur supplémentaire augmente la durée du test et donc son coût. Deux notions sont utilisées, l'observabilité et la contrôlabilité. L'observabilité d'un nœud correspond à la capacité que l'on a de lire son état. Et la contrôlabilité correspond à la capacité que l'on a de forcer son état. Ainsi, pour chaque nœud logique, on cherchera à écrire son état, puis à le lire, afin de vérifier son bon comportement.

Le test est utilisé pour détecter les défauts. Donc son utilisation pour détecter des modifications dans la logique due à la présence d'un CTM parait appropriée. Cependant, les vecteurs de test devront être adaptés. En effet, pour tester un collage logique on peut considérer chaque connexion de manière indépendante. Mais dans le cas d'une infection par un CTM fonctionnel, se pose un problème d'observabilité. En effet, afin d'observer l'effet du CTM sur l'état de signaux, il faut que l'actionneur soit actif et donc que le déclencheur soit correctement sollicité. Or, il est raisonnable de considérer qu'afin d'obtenir un CTM se déclenchant lors d'évènements rares, un attaquant choisira comme évènement déclencheur une combinaison des signaux à faible contrôlabilité ou

probabilité d'occurrence. Par conséquent, un test visant à détecter et donc déclencher un tel CTM devra considérer les combinaisons de signaux et non les signaux indépendamment.

Pour pouvoir tester efficacement les circuits finis, la phase test est prise en compte tout au long de la conception. Par exemple en ajoutant une chaîne de scan qui permet de propager des valeurs dans les nœuds internes des circuits et ainsi augmenter la contrôlabilité.

Les techniques usuelles de générations de tests logiques conventionnels ne pouvant pas être facilement étendues pour la détection de CTM, plusieurs travaux ont abordé la problématique de la génération de vecteurs de tests dédiés. Leur but est d'augmenter le taux de détection obtenu.

Les auteurs de [JJ08] présentent une signature probabiliste permettant de détecter les CTMs. La signature est mesurée par une analyse des réponses du circuit à des stimulis aléatoires qui sont affectées par la présence de la logique malicieuse. Le procédé est testé sur des circuits de test combinatoire et permet de détecter l'infection dans 10 cas sur les 12 testés.

Parallèlement, dans [Wol+08], les auteurs utilisent une méthodologie en deux étapes pour faciliter la détection des CTMs. La première étape consiste à rechercher de vecteurs permettant de déclencher les signaux rares. La seconde étape consiste à développer un ATPG (Automatic Test Patern Generation) permettant de générer les vecteurs de tests pertinents qui permettent de propager et donc de détecter les fautes. Les résultats obtenus avec cette approche ont un taux de couverture de $60\,\%$. Cependant, aucune information sur les taux de détection n'est donnée. Suite à ces travaux, les auteurs de [Cha+09b] proposent une méthodologie appelée *MERO* (Multiple Excitation of Rare Occurence). L'idée est de détecter les signaux logiques à faible contrôlabilité, car ces derniers sont les plus susceptibles d'être utilisés par le déclencheur. Une fois ces signaux candidats déterminés, les vecteurs de test sont choisis de manière à les faire commuter de multiple fois (N fois). Ainsi, si N est suffisamment grand, la condition d'un déclencheur composé de plusieurs de ces signaux a une forte probabilité d'être réalisé. Les auteurs montrent dans que dans le cas étudié (dans le cas de CTMs combinatoires), une sélection de 100K vecteurs aléatoires permet d'obtenir une augmentation de $120\,\%$ du taux de couverture par rapport à un ensemble de vecteurs de tests généré avec une méthode de test classique [MS99]. *MERO* permet donc une réduction moyenne de $85\,\%$ du nombre de vecteurs utilisé par rapport à une sélection aléatoire, tout en offrant une couverture similaire ou supérieure. Dans le cas d'une infection séquentielle, on observe une augmentation maximale de $12\,\%$ pour un CTM à 32 états. La figure 1.8 montre l'influence du nombre de noeuds pris en compte par le déclencheur par rapport au taux de couverture d'un test suivant la méthodologie *MERO*. Il en résulte que l'efficacité de la

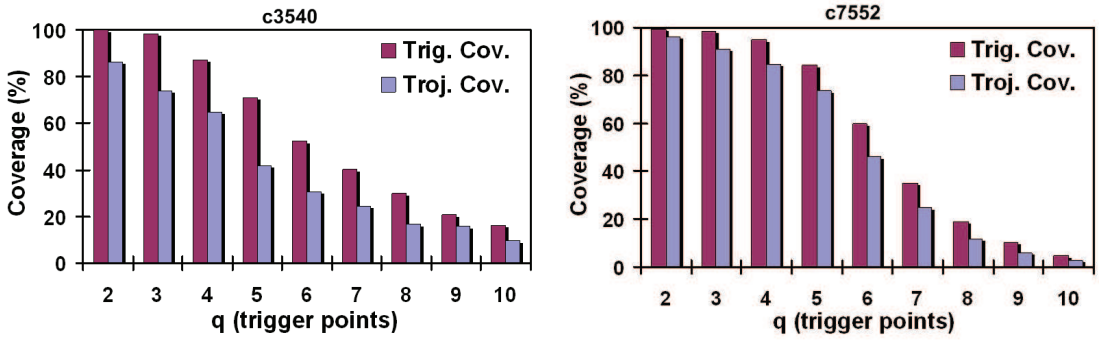

Figure 1.8: Taux de couverture en fonction de la taille du déclencheur pour 2 designs [Cha+09b].

méthode décroit rapidement avec la taille du déclencheur. En effet, dans les deux cas étudies une taille supérieure à 6 bits implique un taux de couverture inférieur à $50\,\%$.

Algorithme génétique

En 2015, une utilisation d'algorithmes génétiques pour améliorer la sélection des vecteurs de test est proposée dans [Sah+15]. Un algorithme génétique est utilisé pour générer des vecteurs de test qui maximisent les combinaisons difficiles à activer, là où *MERO* considère les bits indépendamment. En outre, pour détecter une faute induite par le CTM, il faut qu'elle soit propagée jusqu'à un nœud observable (ex. une sortie primaire). Or, certains vecteurs peuvent déclencher un CTM sans propager la faute provoquée par l'actionneur. Donc, après la génération des vecteurs, une étape de sélection est effectuée afin de conserver les vecteurs permettant l'observation de l'effet du CTM une fois activé. Les auteurs montrent une augmentation moyenne du taux de couverture de $88\,\%$ par rapport à *MERO* pour des infections ayant des déclencheurs combinatoires de 4 bits. Dans les cas de CTMs séquentiels à 4 états les auteurs montrent une amélioration moyenne de $319\,\%$ du taux de couverture. Cependant, seulement 3 circuits de test ont été utilisés pour cette analyse. La figure 1.9 permet de comparer les deux méthodes pour un circuit de test donné. Il apparait une nette amélioration du taux de couverture de la technique basée sur un algorithme génétique par rapport à la méthode MERO présentée par [Cha+09b].

Sélection des points d'intérêt

La première étape pour la sélection de vecteurs est l'identification des points d'intérêt pour une infection. C'est-à-dire l'identification des signaux à faible contrôlabilité. Les

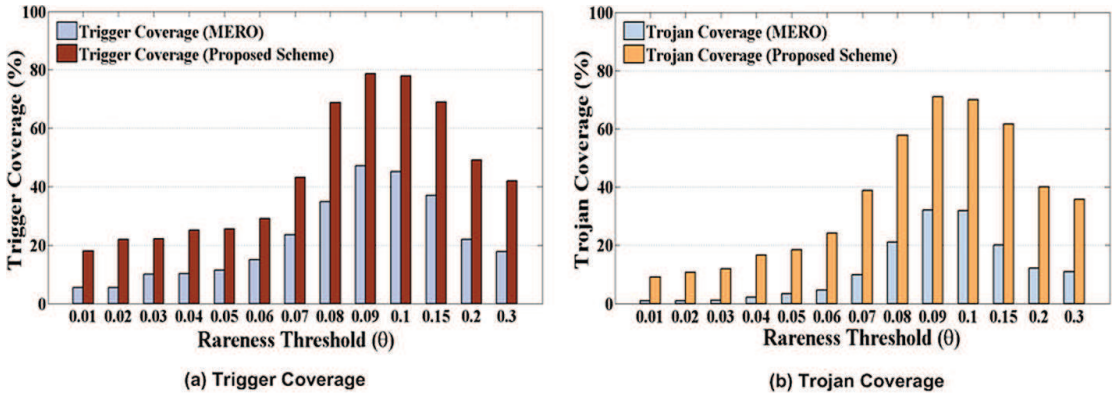

Figure 1.9: Taux de couverture en fonction de la rareté des signaux du déclencheur pour *MERO* et l'approche par algorithme génétique [Sah+15].

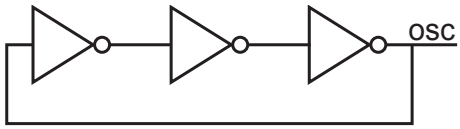

Figure 1.10: Oscillateur en anneaux à 3 étages.

auteurs de [Dup+13] proposent une méthode de détection des points d'intérêt (d'infection) basée sur trois critères:

- la faible contrôlabilité des signaux,
- la non-altération des performances temporelles du circuit,
- la possibilité physique d'insérer un CTM au voisinage des nœuds identifiés ("trous" dans le layout)

Le test logique permet donc de détecter les CTMs dont le déclencheur possède peu d'entrées. Cette catégorie de test est plus efficace pour détecter les CTMs combinatoires que ceux qui sont séquentiels. En effet, pour activer un déclencheur séquentiel le nombre de vecteurs nécessaires augmente fortement avec le nombre d'étages mis en jeu par le déclencheur du CTM considéré.

1.3.4 Oscillateurs en anneaux

Les oscillateurs en anneaux (RO: "Ring oscillator") sont des structures oscillantes constituées d'une chaîne bouclée de portes inverseuses. La figure 1.10 représente un RO à trois étages, composé de trois inverseurs. La fréquence d'oscillation d'un RO dépend du temps de propagation de chacun de ses constituants. La période d'oscillation d'un RO constitué d'inverseur est égale au double du temps de propagation de ses constituants

[MS]. Un oscillateur à n étages ayant chacun un temps de propagation de t oscille donc à une fréquence de:

$$f = \frac{1}{2 \times t \times n}$$

Le délai d'une porte logique, et donc t, dépend de nombreux paramètres technologiques et de conception, mais aussi de:

- la tension d'alimentation (V) ,
- la température (T),
- la qualité des procédés de fabrication (P).

Ainsi, la fréquence d'oscillation dépend de la tension locale dans le circuit. Cela permet donc de faire des mesures de la tension d'alimentation régnant localement en un point du circuit. Ainsi les travaux présentés dans les paragraphes suivants proposent des méthodes de détection de CTM basées sur l'utilisation de RO. Ces méthodes cherchent à détecter les chutes de tension provoquées par les commutations des CTMs.

Une première approche est présentée dans [Raj+11a]. Les auteurs suggèrent de convertir les différents chemins logiques du circuit en ROs. En utilisant les portes logiques présentes sur le circuit, ils limitent ainsi l'ajout de matériel dans le circuit. L'idée sur laquelle repose cette approche est simple et s'appuie sur le constat que l'ajout d'une porte logique supplémentaire connectée à la sortie d'une autre porte logique augmente le fan-out (la charge électrique sur la sortie) de cette dernière et donc son délai de propagation. Chaque chemin étant converti en RO, un ajout de délais sur l'un de ces chemins modifie la fréquence de fonctionnement du RO utilisant le chemin impacté. Si cette approche est séduisante, elle nécessite toutefois que l'impact du CTM modifie la fréquence de fonctionnement. Et cette modification doit être assez importante pour que la fréquence dépasse les valeurs limites prévues en simulation et que l'impact puisse être distingué des variations de procédés de fabrication. Cette méthode a été validée expérimentalement sur un seul FPGA donc sans prise en compte des variations de procédés de fabrication.

Apprentissage

Dans le même genre d'idée, les auteurs de [XT11] intègrent un réseau de ROs pour détecter l'activité électrique des CTMs. L'utilisation d'un réseau de ROs à la place d'un seul RO est intéressante à deux points de vue. En premier lieu, cela permet de couvrir l'ensemble de la surface du circuit. En second lieu, cela permet d'analyser le

Tableau 1.1: Taux de détection obtenue avec un réseau d'oscillateurs [Kar+15]

Taille de l'apprentissage	PCA	SVM	GA + SVM
8	66,5 %	92,7 %	96,9 %
16	70,1 %	94,7 %	97,8 %
24	6,8 %	95,5 %	99,6 %

comportement de chaque RO par rapport au comportement de tous les autres. Une Analyse en Composantes Principales (ACP) est utilisée pour analyser les résultats en prenant en compte les relations entre ROs. L'ACP nécessite une population de vecteurs pour être appliquée.

Dans ce cas, chaque individu de cette population est un circuit et chaque vecteur est constitué de l'ensemble des fréquences des ROs du circuit considéré. Cette méthode s'appuie sur un circuit de référence ou une population de circuit de référence. Cette méthode a été expérimentalement validée sur 24 FPGA infectés et 24 FPGA sains. Les fréquences d'oscillation sont mesurées grâce à un oscilloscope externe. Un LFSR est utilisé pour générer des vecteurs de test induisant des commutations électriques dans les CTMs implémentés. Trois CTMs ont été implémentés (nommés T7, T8 et T9) de tailles respectives égales à 0,17 % 0,25 % et 0,33 % de la taille du circuit. Les résultats expérimentaux montrent un taux de détection de 100 % pour les CTMs T8 et T9 et 80 % pour T7. Ces travaux ont été portés sur ASIC ("Application-Specific Integrated Circuit") 90 nm dans [Fer+12]. La figure 1.11 représente l'implémentation du système. Les ROs sont connectés à un unique compteur au travers d'un multiplexeur afin de permettre une évaluation embarquée de leurs fréquences d'oscillation. Les étages des ROs sont organisés verticalement afin que chaque composant d'un RO soit adjacent à une ligne de cellules standards différente. 40 ASICs ont été conçus, sur un même wafer, chacun possédant 7 CTMs combinatoires pouvant être contrôlés individuellement. Les auteurs considèrent que l'impact passif des CTMs est négligeable. Les tailles des CTMs vont de 0,12 % à 0,81 % de la surface totale du circuit. L'impact des CTMs sur les fréquences des ROs est compris entre 0,5 % et 2,5 %. Le taux de détection est compris entre 50 % et 50 % pour les CTMs les plus petits et de 80 % et 90 % pour les 2 CTMs les plus grands. Le taux de faux positif est de 20 %.

33 ASICs de ce même lot ont été analysés en utilisant une machine à vecteurs de support et un algorithme génétique dans [Kar+15]. Les taux de succès (nombre de cas classés correctement sur nombre total de cas) des trois méthodes sont donnés dans le tableau 1.1 en fonction de la taille de la population utilisée pour l'apprentissage. L'utilisation d'algorithmes génétiques avec une population d'apprentissage suffisamment étendue permet d'atteindre un taux de détection de 99,6 %.

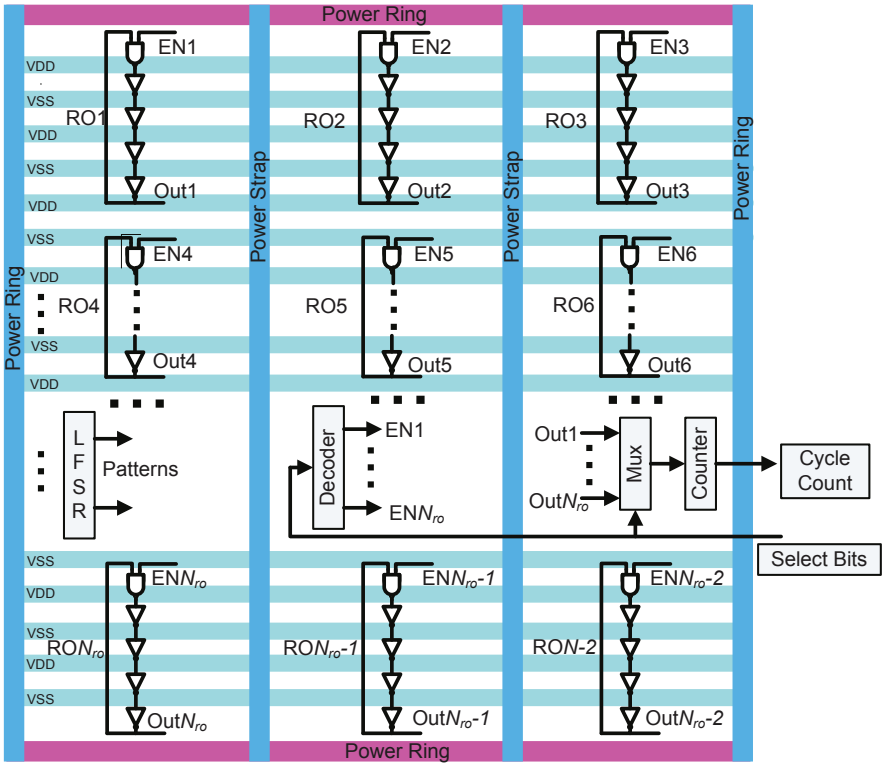

Figure 1.11: Implémentation des oscillateurs en anneaux [Fer+12]

Détection de signaux invasifs

Les travaux, basés sur des ROs, précédents cherchaient à détecter la charge induite par le déclencheur sur les lignes d'alimentation. Les auteurs de [Lam+11] évaluent l'impact des signaux invasifs (connections parasites dues à un CTM) sur les délais des ROs. Ce type d'infection est illustrée dans la figure 1.12. Les modèles de CTMs utilisés sont l'ajout de connexions ou de portes logiques directement sur un RO. Des expérimentations ont été menées sur 20 FPGA XUP-V2Pro (ces circuits sont dans une technologie $130\,\mathrm{nm}$). Les fréquences des oscillateurs à 9 étages considérés dans ces travaux varient entre $140\,\mathrm{MHz}$ et $148\,\mathrm{MHz}$ à cause des variations de procédés intra-die et subissent une variation inter-die allant jusqu'à $3{,}5\,\%$. Une compensation des variations inter-die est faite en utilisant un RO comme référence et en considérant comme circuits infectés ceux pour lesquels les valeurs s'éloignent de plus de 3σ (3 fois l'écart type) de la moyenne. Cette méthode permet d'obtenir un taux de détection de $78\,\%$. Cela démontre que l'impact d'une infection invasive des ROs est supérieur à l'impact des variations de procédés dans la majorité de cas.

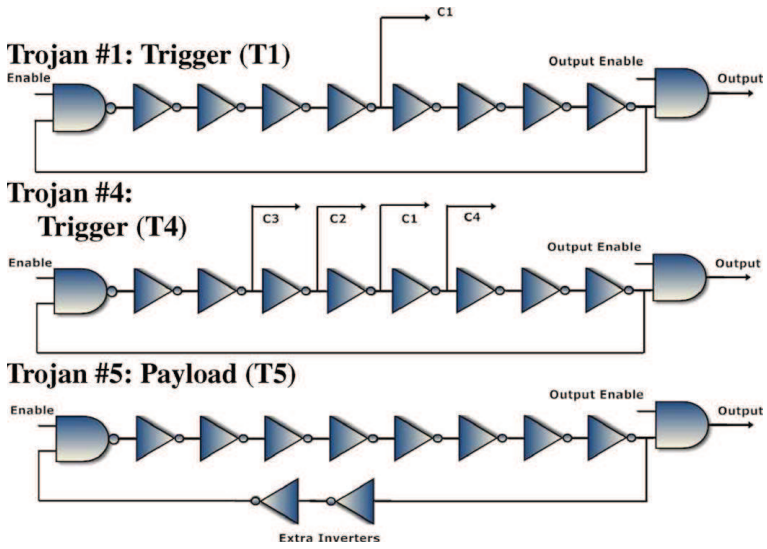

Figure 1.12: Infections invasives de ROs [Lam+11]

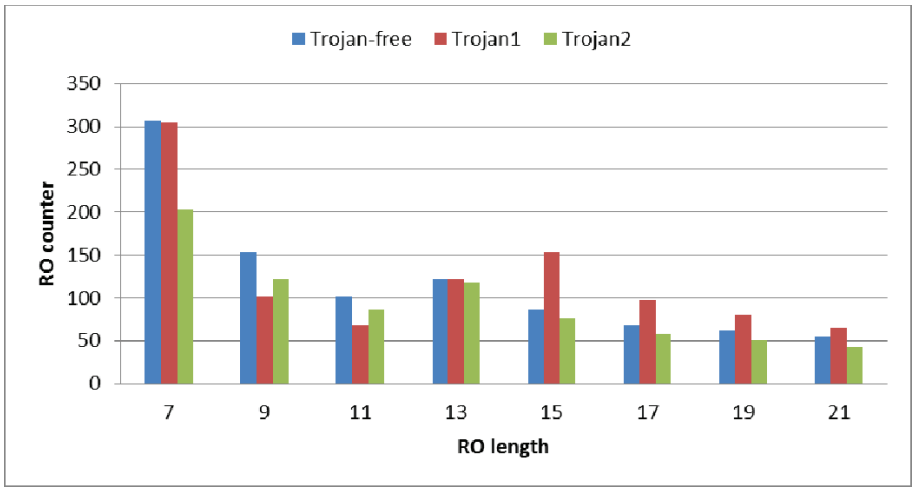

Figure 1.13: Infections invasives de ROs [KV14]

Longueur des ROs

Ensuite, afin de quantifier l'impact de l'implémentation des ROs sur leur sensibilité aux CTMs, une étude de l'impact du nombre d'étages d'un RO a été menée dans [KV14]. Un CTM combinatoire et un CTM séquentiel ont été simulés à partir d'un placement-routage de FPGA Spartan6. Un compteur est utilisé pour mesurer la fréquence des ROs. Trois circuits ont été développés (référence, CTM1 et CTM2). Les contraintes de placement étant faibles, les placement-routages sont légèrement différents entes deux designs. La figure 1.13 représente les résultats obtenus pour différentes longueurs de ROs. Ils montrent une sensibilité différente en fonction du type de CTM et de la longueur du RO. Cependant, considérant les conditions expérimentales, cette différence peut être expliquée par les petites différences de placement-routage. Dans ce cas, cela signifie que l'impact d'un CTM sur des ROs est sensible aux détails de l'implémentation des ROs.

Au regard des travaux mentionnés ci-dessus, il apparait que les oscillateurs en anneaux ont fréquemment été utilisés comme capteur de l'impact de CTMs avec plus ou moins de succès. Des implémentations sur ASIC et sur FPGA ont été réalisées. Ces expériences mettent en valeur que l'activité de commutation électrique des CTMs (combinatoire ou séquentiel) a un impact significatif sur les fréquences d'oscillations. Cependant, l'impact des variations des procédés de fabrication, que ce soit les variations inter-dies ou intra-die sur les fréquences d'oscillations est nettement supérieur à l'impact des CTMs. Cela représente le principal obstacle à ce type de détection. La prochaine section présente une autre source d'information sensible à la présence des CTMs.

1.3.5 Analyse des canaux axillaires

L'étude des canaux auxiliaires permet d'extraire des informations sur les données manipulées ou les instructions par le circuit. Les CTMs pouvant manipuler de l'information ou dans une moindre mesure avoir un impact sur les grandeurs physiques émises sur le circuit, différentes méthodes de détection de CTM utilisant les analyses des canaux auxiliaires ont été proposées.

Consommation de courant

La première solution de détection de CTM par analyse des canaux auxiliaires a été proposée dans [Agr+07] et est basée sur la mesure de courant global consommé. Cette publication montre par simulation, qu'un CTM induit des variations sur le canal auxiliaire consommation. Les auteurs concluent qu'un CTM représentant $0{,}01\,\%$ de la surface du circuit (en présence de variations des procédées de $7{,}5\,\%$) peut être détecté par une analyse de la consommation de courant. Cependant, l'étude est réalisée par simulation (postsynthèse) et le modèle des variations des procédés est une simple variation des paramètres des cellules standard. Aucun bruit de mesure n'est considéré.

Ensuite pour améliorer la détection par analyses de consommation des travaux proposent de prendre plusieurs points de mesure de courant sur un même circuit. Afin d'avoir une information plus locale.

Premièrement, dans [Rad+08], les auteurs explorent les avantages et inconvénients de différentes techniques de calibration des analyses des courants statiques et dynamiques. Les principes de la calibration sont de construire une matrice de transformation visant à corriger les mesures. Cette matrice de transformation est construite à partir de réponses du circuit à des stimuli de courant continus ou variables. Neuf points de mesure différents sur la grille d'alimentation sont utilisés pour mesurer le courant. En sélectionnant des pairs parmi ces 9 points de mesure, 12 traces de courant sont

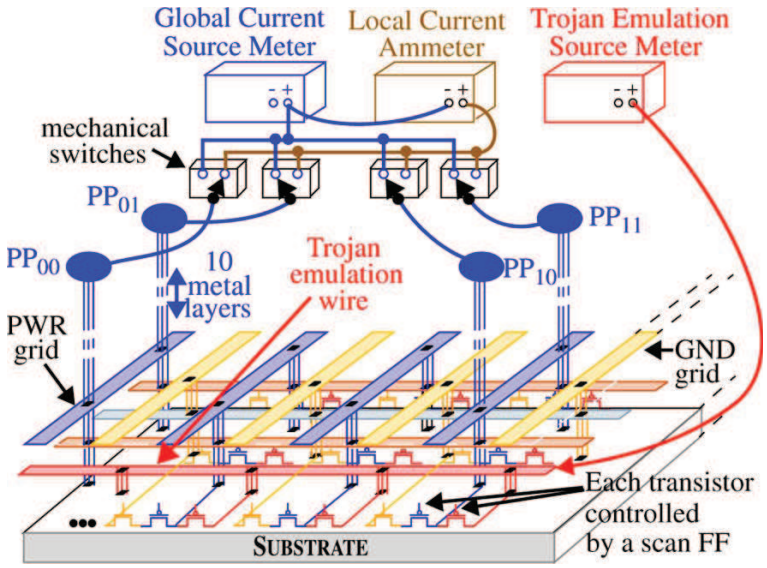

Figure 1.14: Circuit de test [Aar+10]

mesurées. Les méthodes de calibration permettent de réduire les différences entre circuits. Les résultats obtenus montrent que ces techniques de calibration permettent d'augmenter la capacité de détection et de discriminer l'impact d'un CTM composé de 32 portes NAND des variations des procédés. Encore une fois, seuls des résultats de simulation sont fournis. Ensuite dans [Aar+10], la méthode a été mise en œuvre sur un lot de 45 circuits fabriqués en technologie CMOS $65\,\mathrm{nm}$. Le circuit de test est illustré sur la figure. 1.14. Les ports PP correspondent aux points de mesure. Des transistors contrôlés par des bascules D internes (contrôlées de façon externe) sont utilisés pour émuler des CTMs. Leurs sources sont connectées à une alimentation externe pour pouvoir contrôler leur impact en courant local. Les résultats montrent que considérer les acquisitions multiples de courant par rapport à une analyse globale augmente de plus de 49 fois la sensibilité du système de détection considéré. La méthode employée a permis de détecter des courants parasites de $8\,\mu\mathrm{A}$. Parmi les 4050 combinaisons émulant des CTMs, 14 restent toutefois indétectées.

Toujours dans l'optique d'utiliser plusieurs points de mesure de courant. Les auteurs de [Wan+08b] étudient par simulation l'impact des CTM sur les courants de différents points de mesure. Les différentes mesures sont ensuite intégrées pour obtenir une trace globale. Les variations de procédés sont modélisées de façon à obtenir des cas favorables, neutres ou défavorables. Un circuit est déterminé comme sain si sa consommation sort des limites déterminer par la mesure des circuits de référence subissant différentes variations de procédés de fabrication. La conclusion obtenue est que dans les cas où les variations des procédés sont défavorables, les plus petits CTMs (ex. $0{,}1\,\%$ de la surface) sont indétectables.

Une approche différente de l'analyse de la consommation de courant globale est proposée dans [He+15]. Les auteurs montrent que lors de leurs études (sur FPGA) une analyse fréquentielle des traces de courants permet de distinguer l'impact d'un CTM là où une analyse temporelle échoue. Cependant, aucun détail d'implémentation du CTM ou du circuit n'est fourni.

Température

Un autre canal auxiliaire étudié est basé sur la mesure de température d'un circuit. En effet, la température est fonction de la consommation de courant, la modification de consommation causée par l'activité du CTM doit aussi se refléter sur le profil thermique du CI.

Une première approche utilise des capteurs de températures. Ces capteurs sont classiquement utilisés pour prévenir des problèmes de fiabilité ou réguler la consommation des circuits. Dans [Bao+15], les auteurs proposent l'ajout de capteurs de température pour détecter l'activité de CTM. Des cartographies de consommation avec et sans CTMs ont été simulées en utilisant un profil de consommation post synthèse, et les informations de position post placements. Différents capteurs sont simulés. Les résultats donnent un taux de succès de $100\,\%$ pour tous les circuits tests considérés. Cependant, la prise en compte des variations des procédés de fabrication de $10\,\%$ augmente le taux de faux positifs jusqu'à $8\,\%$ et une variation de $20\,\%$ jusqu'à $17\,\%$. Cette méthode permet de détecter une déviation de consommation de $1{,}54\,\%$, mais est très sensible aux variations des procédés de fabrication.

Ensuite, dans [Now+14], les auteurs proposent une analyse de température prise par un capteur de température externe. Ils proposent une estimation du courant à partir d'une cartographie thermique. Deux cas sont envisagés: en premier l'utilisation de l'ACP (Analyse de la composant principale) pour la mise en valeur de l'activité du CTM et en second lieu l'utilisation d'apprentissage non supervisé pour la prise de décision. Les résultats de simulation, intégrant des variations des procédés de $30\,\%$, montrent un taux de succès de $25\,\%$ pour un CTM ayant une consommation de $0.158\mu W/m^2$ à $100\,\%$ pour un CTM ayant une consommation de $0.421\mu W/m^2$. Il est difficile de comparer ces résultats avec les CTMs étudiés dans d'autres travaux, une analyse expérimentale avec une infection logique serait nécessaire pour déterminer l'efficacité de cette méthode.

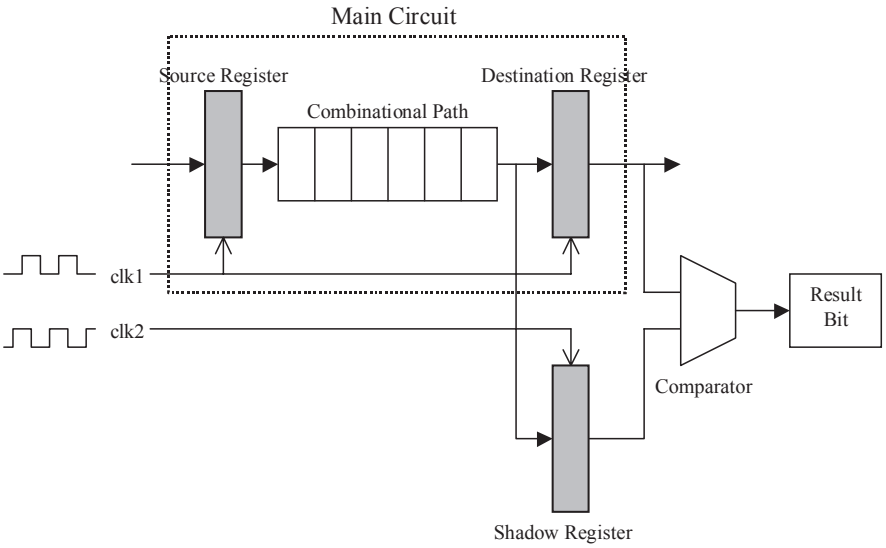

Figure 1.15: Capteur de délais réalisé dans [LL07]

Délais

Après le déploiement de techniques lors des phases de test, et les propositions de méthodes de détection exploitant le canal auxiliaire de consommation, les auteurs de [LL08] proposent d'analyser les délais de propagation des chemins logiques en utilisant des capteurs embarqués. La figure 1.15 représente le capteur utilisé. Le principe est de déphaser la deuxième horloge "CLK2" en retard par rapport à la première horloge "CLK1" jusqu'à ce qu'une erreur d'échantillonnage survienne et soit détectée par le comparateur. Le dernier déphasage pour lequel aucune erreur n'apparait correspond ainsi à la différence entre la période d'horloge et le temps de calcul du chemin mesuré. L'insertion d'un CTM pouvant altérer les délais des chemins voisins au point d'infection, ainsi on peut mesurer l'impact d'un CTM par mesure de délais. Une évaluation expérimentale a été faite sur des cartes FPGA Virtex-II Pro. Les résultats obtenus montrent que le délai mesuré est sensible aux variations de procédés de fabrication et de température sans donner de résultats probants quant au taux de détection des CTMs.

Dans [Mak08] les auteurs proposent une méthode de détection prenant en compte les variations des procédés de fabrication est proposée. La première étape de celle-ci consiste à sélectionner des vecteurs de test permettant de mesurer les délais d'un nombre maximum de chemins. Dans une seconde étape et afin de réduire la dimension des mesures (complexité du problème), une analyse de la composante principale est appliquée et une enveloppe convexe est générée pour discriminer la population saine de celles des circuits infectés. Cette méthode a été validée par simulation sur quatre types de CTMs, trois combinatoires et un séquentiel. Les variations de procédés ont été simulées en générant des librairies de synthèse incluant des variations aléatoires de $7{,}5\,\%$. La méthode d'acquisition des délais envisagée dans la pratique n'est toutefois pas

précisée. Les CTMs combinatoires sont des compteurs 2-bit représentant $0{,}13\,\%$ de la surface du circuit testé et sont détectés dans $100\,\%$ des cas. Le CTM séquentiel est un compteur 4-bit occupant environ $0{,}76\,\%$ de l'aire du circuit. Le taux de détection associé est de $36\,\%$. Les résultats montrent que des CTMs combinatoires ont un impact plus important sur les mesures de délais que les CTMs combinatoires.

Dans la lignée de [Mak08] les auteurs de [RL09] caractérisent l'efficacité de la méthode précédente. Pour cela ils utilisent le même capteur que [LL08] représenté figure 1.15 en prenant en compte les variations de procédés. Ils concluent que la détection de CTMs par analyse de délais est pertinente en considérant que les variations des procédés sont aléatoires, mais que l'impact d'un CTM est le même sur tous les circuits infectés. Aucun taux de détection n'est donné dans cette application.

Dans [LP12], un dispositif embarqué de mesure des délais nommés REBEL ("*REgional dELay Behavior*") est présenté. La logique de REBEL est ajoutée à celle de la chaîne de SCAN pour limiter son impact sur la surface utilisée. Le principe a été validé sur 62 ASICs en technologie CMOS 90 nm. Des CTMs sont émulés par des structures ajoutant des capacités de charge ajoutant ainsi des délais sur les lignes mesurées. La méthode de détection repose sur une régression linéaire des relations entre chemins et une analyse des valeurs aberrantes à 3 σ (on considère suspecte, une valeur s'éloignant de plus de trois fois l'écart type de la moyenne). Les résultats montrent un taux de détection de $100\,\%$ pour des capacités ajoutées équivalentes à $25\,\mathrm{fF}$ et diminuent fortement avec la capacité, en dessous de $10\,\mathrm{fF}$, pour atteindre $0\,\%$ pour des capacités de $1\,\mathrm{fF}$. Ces résultats montrent la pertinence de l'analyse de délais en pratique, mais le modèle de CTM utilisé ne permet pas de conclure sur la nature des CTMs détectables.

L'analyse des délais requiert de tester le plus grand nombre de chemins possibles pour être efficace. Cela peut être coûteux en temps. Dans le but d'optimiser la sélection des chemins mesurés et donc des vecteurs de test, une méthode est proposée dans [Dav+13]. Cette méthode nécessite plusieurs circuits pour être appliquée. L'optimisation de la sélection des chemins mesurés a pour objectif de réduire le nombre de circuits devant être testés pour détecter une infection. Pour chaque site d'insertion potentiel, on cherche le chemin le plus court pouvant être testé. En effet, sur ces chemins courts l'impact relatif du CTM est plus important. Ce phénomène est illustré dans la figure 1.16. Une phase de calibration utilisée pour réduire les variations inter-die est aussi appliquée. Le t-test est utilisé pour déterminer la présence de l'infection. Les variations des procédés sont simulées avec une approche de Monte Carlo. Les résultats de simulations montrent que l'utilisation de la calibration permet une réduction de coût en nombre de circuits nécessaires par 2,1 et que l'algorithme de recherche des chemins courts et la calibration offrent une réduction du coût par 4,51.

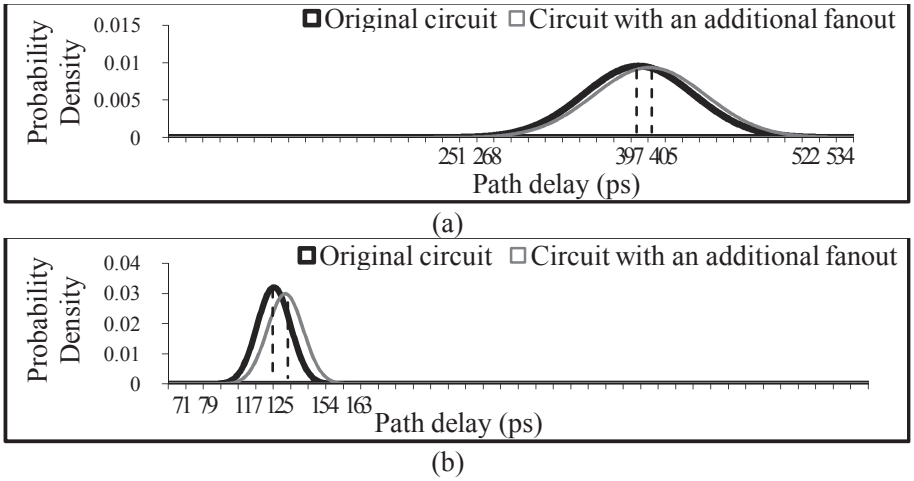

Figure 1.16: Distribution des délais sous l'influence de variation des procédés et de l'impact d'un CTM [CG13]. (a) Chemin long, (b) Chemin court

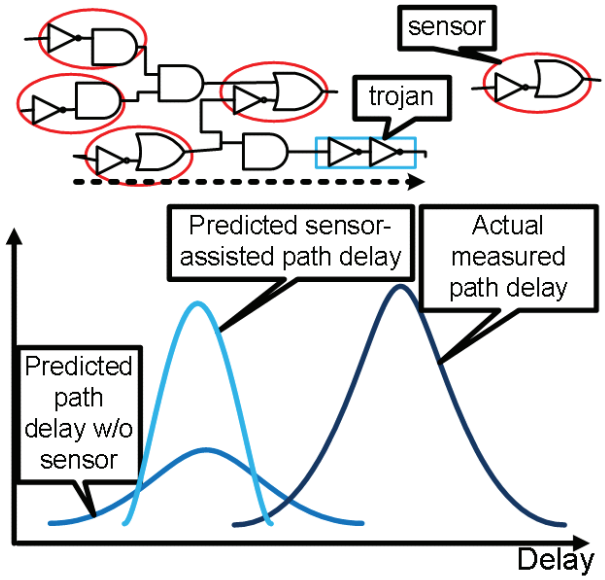

Figure 1.17: Exemple de référence de délais [Dav+13].

Dans [Dav+13] les auteurs proposent une méthodologie visant à se passer de circuit de référence. Le principe est de développer des capteurs embarqués dépendant du layout. La méthodologie consiste à analyser la netlist et à identifier l'ensemble de portes logiques fréquemment instanciées. Les capteurs implémentés seront similaires à une partie du design et serviront de références pour des mesures de délais, un capteur est implémenté par groupe similaire. La figure 1.17 illustre le principe. L'utilisation de capteurs permet d'estimer avec plus de précision les délais des structures similaires. Si une mesure pratique est en dehors des mesures prédites grâce aux capteurs, le circuit est considéré comme infecté. Les résultats donnent un taux de détection variant entre 50 % et 85 % sur les circuits simulés.

Les auteurs de [Exu+15] proposent une méthode de détection basée sur la mesure de délais de le 10ème tour d'un AES 128-bit. Le principe est de proposer une méthode de mesure des temps critiques sur les chemins de donnée pour des algorithmes cryptographiques de type AES. La mesure consiste à réduire progressivement une période du signal d'horloge synchronisée avec le 10ème tour de l'AES. Ainsi, pour chaque circuit considéré, on peut mesurer une valeur de délais par bit de l'AES et par message (à chiffrer) utilisé. Dans un premier temps, le principe est de limiter l'impact des variations inter-die, en considérant que tous les chemins sont impactés de la même manière. Un modèle de référence est construit à partir de plusieurs circuits sains. Cela fait, les auteurs proposent de construire une matrice de non-pertinence. Celle-ci consiste à mesurer l'écart type de chaque point de mesure au travers des circuits, afin de déterminer les points de mesure les moins impactés par les variations des procédés. Les points ainsi déterminés seront considérés prioritaires pour la détection pour éviter les faux positifs. Aucune explication sur la pertinence physique de cette méthode n'est donnée. Cette méthode a été validée expérimentalement sur FPGA Spartan 3AN avec 3 circuits de référence et un circuit test avec deux infections (combinatoires et séquentiels) qui occupent respectivement $0{,}19\,\%$ et $0{,}36\,\%$ des ressources du FPGA.

Émanation électromagnétique

De façon analogue à l'analyse de consommation, l'analyse EM permet aussi de détecter l'activité électrique d'un CTM. Dans [Rad+08; Aar+10; Wan+08b], les auteurs cherchent à obtenir différents points de mesure de courant. Dans le cas d'une analyse EM, la prise d'information locale est facilitée permettant ainsi de cartographier l'activité électrique du circuit. Dans [Sol+] les auteurs décrivent une méthode de détection de CTM basée sur l'utilisation de cartographies des émanations électromagnétiques d'un circuit encapsulé. Les traces EM sont traitées de la façon suivante. Premièrement, un alignement temporel est effectué afin de corriger le bruit et les variations d'horloge. Puis dans un second temps, les points d'intérêt des traces EM sont identifiés en calculant la variance du vecteur de différence absolue (référence - testé). Le point avec la plus grande variance sera considéré comme le point le plus impacté par le CTM. La procédure a été testée expérimentalement sur une carte FPGA Virtex-II Pro. La cible est une AES 128-bit et l'infection est constituée de 30 bascules D (15 slices). Les cartographies des différences absolues, pour le point d'intérêt sélectionné sont données de la figure 1.18. Chaque cartographie correspond à une position d'infection différente. Les résultats montrent un impact très important du routage sur les émissions EM. De plus, la position du CTM semble impacter la détectabilité du CTM, celui-ci n'ayant pas un impact significatif à toutes les positions. Les auteurs concluent donc qu'il semble difficile de détecter tous les CTMs par cartographie électromagnétique.

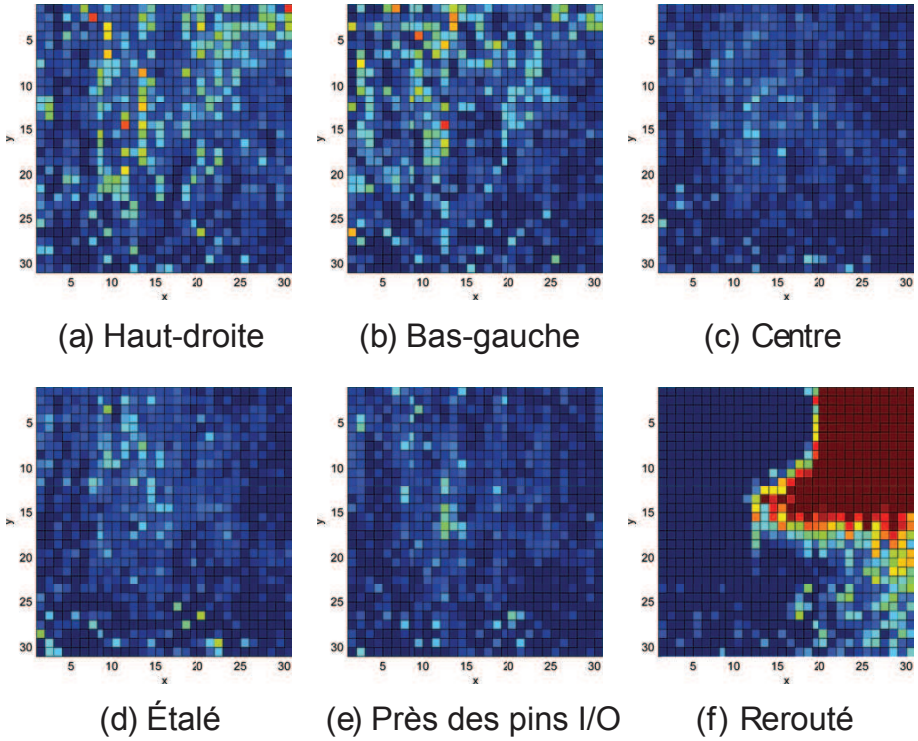

Figure 1.18: Cartographies des différences entre les designs infectés et le design de référence [Sol+].

Néanmoins, dans [Bal+15], une technique plus efficace est utilisée pour interpréter les traces EM. Les auteurs de [Sol+] utilisent la différence de moyenne alors que le t-test est préféré dans [Bal+15]. La valeur utilisée pour générer les cartes est la valeur maximale de la statistique obtenue à chaque point de la cartographie. Dans ce papier, les auteurs insistent sur la nécessité de contrôler les conditions expérimentales. La figure 1.19 présente des résultats obtenus à partir d'un unique circuit et d'une même implémentation. La carte de gauche est obtenue à partir de deux jeux de mesures obtenus consécutivement, la carte correspond à des jeux de mesures acquis à des jours et heures différents. Ceci est le premier résultat montrant l'importance des conditions expérimentales sur la comparaison des mesures des canaux auxiliaires EM pour la détection de CTM. Cela montre que les conditions environnementales ont un impact significatif sur les mesures, de mauvaises conditions pouvant donc provoquer une augmentation du taux de faux positifs. Cela étant, les auteurs concluent qu'il est possible de détecter de très petits CTMs en considérant avec précaution la température de l'environnement pendant la mesure. Les auteurs ont ensuite implémenté un CTM représentant $1{,}3\,\%$ de la surface du FPGA à différentes positions (le CTM n'a pas d'actionneur de façon à limiter sa surface). La figure 1.20 représente les résultats obtenus. L'impact du CTM est plus important que dans [Sol+] et toutes les positions permettent de détecter l'infection. Cependant, la figure 1.21 montre que pour de plus petites infections la position de l'infection impacte fortement la détectabilité.

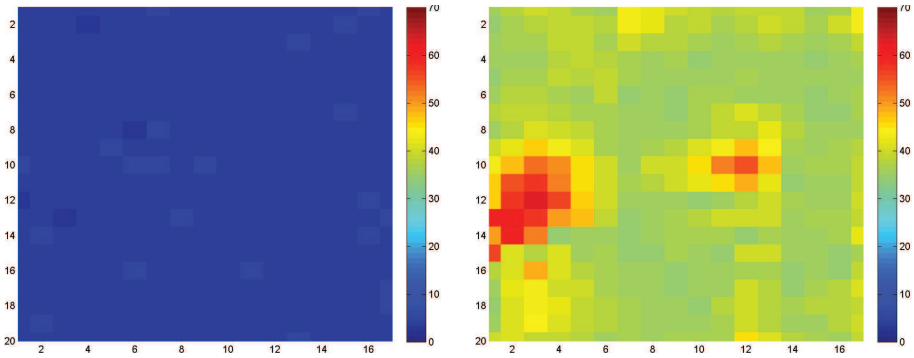

Figure 1.19: Cartographie des valeurs de t-test provenant des jeux de mesures d'un même design [Bal+15].

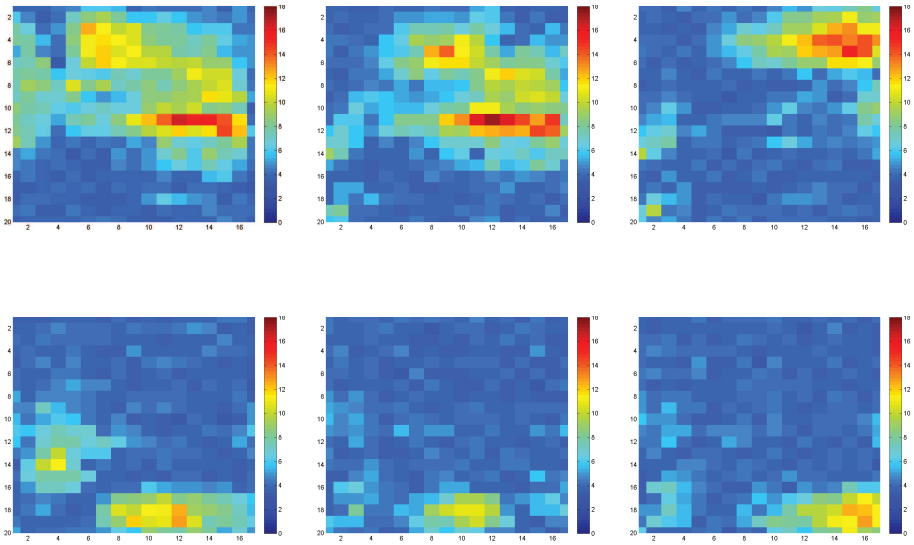

Figure 1.20: Cartographie des différences entre les designs infectés et le design de référence [Bal+15].

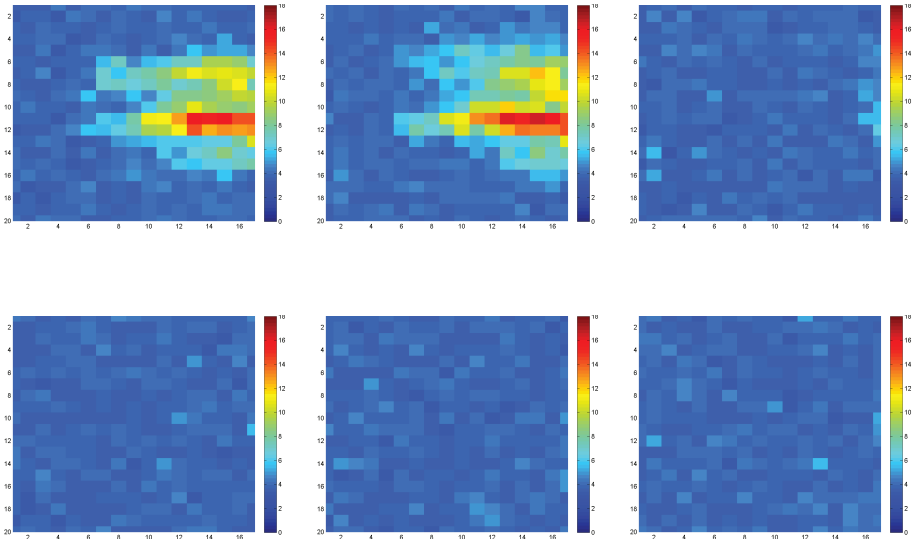

Figure 1.21: Cartographie des différences entre les designs infectés par de petites infections et le design de référence [Bal+15].

Les résultats de [Sol+] et [Bal+15] montrent que la position d'une infection modifie l'impact d'un CTM sur une analyse EM. Cela peut empêcher la détection pour des infections trop petites. Les auteurs de [Bal+15] émettent l'hypothèse que les niveaux de métallisation mis en jeux dans le routage de l'infection sont la cause de cette différence d'impact. Les lacunes de cette approche sont l'impact des conditions environnementales qui sont à prendre en considération (cela doit être vrai pour la majorité des approches), les temps de cartographie peuvent être importants (1h30 par circuit) de plus les variations de procédés n'ont pas été étudiées.

Finalement, dans [XT+15] les auteurs proposent une méthode d'analyse des émanations EM de circuits FPGAs en prenant en considération les variations de procédés. Dans un premier temps, un modèle de référence est construit à partir d'un lot de circuits sains. Ce modèle correspond à la moyenne des traces obtenues à partir du lot sain, cela permet de limiter l'impact des variations de procédés sur le lot de référence. De plus, une sélection des points d'intérêt est effectuée à partir du lot de traces de référence de façon à sélectionner les points permettant de minimiser l'impact des variations de procédés. Ensuite, les points de la trace considérée sont localisés au front d'horloge là où le CTM a son activité électrique. Pour déterminer si un circuit est infecté, les auteurs utilisent la somme des différences entre la référence construite et une trace du circuit à tester. Cette méthode a été expérimentalement validée sur un lot de 10 FPGAs Virtex 5. Trois tailles de CTMs combinatoires ont été implémentées, avec des déclencheurs de 32, 64, 128 bits représentant respectivement $0{,}5\,\%$, $1\,\%$ et $1{,}7\,\%$ de la surface de la cible (AES 128-bit). Les taux de faux négatifs calculés pour des infections de $1\,\%$ et $1{,}7\,\%$ sont de $0{,}017\,\%$ et $0{,}011\,\%$. Cependant, le taux de faux négatif de la plus petite infection est de $24\,\%$. De plus, trois placements de CTM ont été testés. Le premier placement est dans l'AES, le second hors de l'AES et hors de la zone directement ouverte par la sonde EM et le troisième est étalé hors de l'AES. Il en résulte un impact similaire des deux premiers placements et un impact renforcé pour le troisième placement. Cela doit être dû à l'impact des lignes de routage plus longues augmentant les effets RC et le nombre de buffers utilisés.

Les résultats de cette méthode tendent à confirmer les résultats précédents montrant que l'analyse EM semble efficace, mais limitée pour les petits CTMs. De plus, la zone d'impact présentée figure 1.21 et les résultats obtenus avec le placement dispersé semblent montrer qu'une haute résolution de cartographie n'est pas nécessaire. En effet cette figure montre que lorsque l'impact du CTM est observable une grande partie des points des cartographies permet cette observation.

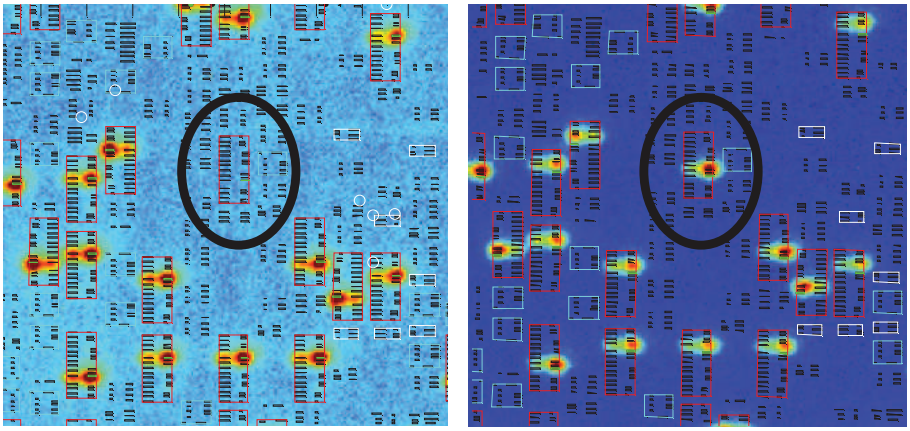

Figure 1.22: Cartographie provenant de deux circuits avec (à droite) et sans activité (à gauche) de la bascule entourée [Ste+14].

Émission de lumière

Dans [Ste+14] les auteurs proposent une méthode de détection des émissions de lumière. En effet, les portes logiques émettent des photons lorsqu'elles commutent. Cette émission est intrinsèquement liée à l'opération effectuée par la porte. La lumière est émise dans l'infrarouge proche et peut être détectée grâce à des techniques développées pour l'analyse de défauts. En combinant l'acquisition des cartographies d'émission, le traitement d'image et des algorithmes d'assemblages, on obtient une cartographie de l'état ou de l'activité du circuit (en fonction de l'activité de l'horloge). La méthodologie a été testée sur des circuits fabriqués dans une technologie $90\,\mathrm{nm}$. La figure 1.22 reporte des cartographies obtenues à partir deux circuits différents. Elle montre que l'activité d'une bascule peut être observée et comparée sur différents circuits. Une image de référence est générée à partir du layout de référence et est comparée avec les émissions mesurées du circuit testé. Ainsi, si la réponse lumineuse n'est pas celle attendue, on en déduit que le circuit a été altéré.

Cette étude est, à notre connaissance, la seule sur l'utilisation de la photoémission pour la détection de CTMs. Toutefois, les équipements mis en œuvre sont coûteux et cette solution ne semble donc pas viable en l'état pour la production en volume.

Les différentes études de détection de CTM par analyse des canaux auxiliaires ont montré leur pertinence que ce soit pour des infections combinatoires ou des infections séquentielles. L'impact d'un CTM sur les grandeurs physiques mises en œuvre dans le circuit est donc mesurable. Les méthodes les plus efficaces ont en commun de pouvoir obtenir des informations locales sur le circuit. Là où les méthodes de détection par test logique sont limitées par la taille du déclencheur (un déclencheur important est moins susceptibles d'être déclenché) les méthodes d'analyse par canaux auxiliaires sont limitées

dans la détectabilité des petites infections. Une combinaison des deux semble donc une solution pour couvrir les différentes tailles de CTMs.

Partitionnement

Le partitionnement d'un circuit consiste à découper de manière réelle ou virtuelle le floorplan de ce dernier. Ceci peut être fait pour différentes raisons. Dans [Min+14] les auteurs proposent une méthode de partitionnement et de génération de vecteurs de test permettant de provoquer une activité de commutation électrique dans une région ciblée et de limiter l'activité de fond des autres régions. Le principe de partitionnement est de connecter la chaîne de scan, servant au test, par région. Pour tester l'effet du partitionnement, une analyse par simulation utilisant la méthode de [Aar+10] (présentée dans la section 1.3.5) est effectuée. Trois CTMs ont étés implémentés occupant jusqu'à $0{,}005\,\%$ de la taille du circuit. À noter que le paramètre des variations des procédés utilisés dans cette simulation a été réglé à $1\,\%$, ce qui parait très optimiste pour une technologie moderne. La méthode de partitionnement montre une augmentation de l'impact des CTMs, par rapport à un CI ou "une approche" non partitionnée, sur l'amplitude des signaux obtenue par analyse de consommation allant de 6.6X à 26.2X en fonction de la taille du CTM et du circuit considéré.

Une autre méthode de partitionnement est proposée dans [Cao+14] et illustrée dans la figure 1.23. Le principe est de séparer le circuit en plusieurs parties électriquement indépendantes, chaque partie étant alimentée au travers d'un dispositif mesurant la durée et l'amplitude de l'activité électrique de la partition. Ainsi on obtient une signature contenant ces deux informations pour chaque partition. Le dispositif utilisé est un capteur dédié. Les résultats de simulations montrent un taux de succès de $100\,\%$ pour les infections représentant plus de $1{,}1\,\%$ de la surface du circuit. Et un taux d'erreur allant de $1\,\%$ à $4\,\%$ pour des infections allant de $0{,}54\,\%$ à $0{,}78\,\%$ de la surface du circuit.

Dans [Abd+16], les auteurs proposent quant à eux un partitionnement logique du circuit. La figure 1.24 représente le découpage d'un AES. Des multiplexeurs sont ajoutés afin de commander l'activité électrique des principaux blocs séparément. À partir de cette division, on peut obtenir 4 signatures distinctes provenant d'un même circuit. D'autres divisions permettraient d'augmenter le nombre de signatures possibles. L'effet de ce partitionnement a été expérimentalement étudié sur un composant FPGA embarquée sur carte Sakura-G. Un CTM avec un déclencheur composé de 12 portes 2-NAND (porte ET à deux entrées) et 1 6-AND (porte ET à 6 entrées) est inséré et connecté au bloc "*MixColumns*" de l'AES. Il en résulte que l'impact du CTM est visible uniquement lorsque le "*MixColumns*" est actif. Cette technique permet donc de localiser l'infection et de réduire le bruit de fond. Le coût de cette division est un nombre de slices

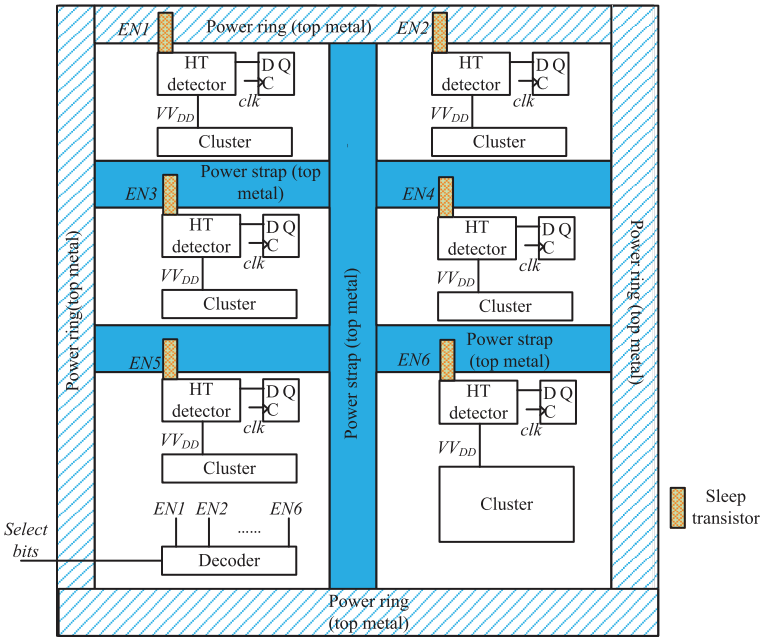

Figure 1.23: Partitionnement d'un layout en régions d'alimentations distinctes [Cao+14].

augmenté de 25 % et une réduction de la fréquence maximale de fonctionnement de 57 %. On notera que les variations des procédés de fabrication n'ont pas été prises en compte dans cette étude, car le même circuit est utilisé en tant que circuit de référence et circuit de test.

Méthodes mixtes

Dans [Nar+10], les auteurs proposent d'utiliser la corrélation existante entre la fréquence maximale de fonctionnement d'un circuit, F_{max}, et sa consommation dynamique. Cette approche a cependant comme défaut le difficile problème de mesurer F_{max} sur des systèmes complexes [Bow+10]. Les auteurs contournent ce problème par une évaluation de la fréquence maximale en utilisant un RO de 15 inverseurs étalé sur la surface du FPGA. Cette approche a été testée sur 10 FPGAs Virtex 2. Les auteurs montrent que la prise en compte de la fréquence du RO permet d'améliorer le rapport entre l'impact du CTM et les variations de procédé de fabrication. Cependant, une telle utilisation d'un RO ne permet d'obtenir des informations que sur les variations de procédés inter-die. Or dans [Rad+08], une méthode de calibration visant à réduire les différences entre les circuits est proposée. Ici, les auteurs ne comparent pas l'efficacité des méthodes utilisant la combinaison de deux canaux (fréquence et consommation) avec l'efficacité de méthodes utilisant un seul canal. Sans comparaison, on ne peut conclure à l'apport de la mesure de la fréquence en complément de la consommation sur la détection de CTM, par rapport à la mesure du courant seule ou des résultats cumulés d'une mesure de courant et d'une mesure de fréquence.

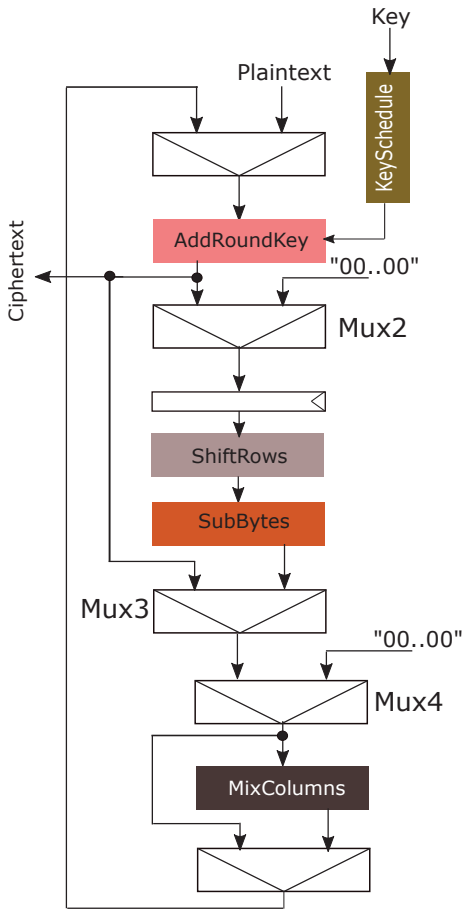

Figure 1.24: Partitionnement le l'architecture d'un AES [Abd+16]

Dans [Zha+13] les auteurs proposent de combiner l'analyse de consommation avec celle d'un réseau de ROs. Pour cela, ils adaptent la méthode proposée dans [XT11](voir section 1.3.4) en ajoutant les informations de consommation dans l'ACP. Cette méthode a été testée avec 60 FPGA sains et 28 FPGA infectés. 24 ROs ont été implémentés sur chaque FPGA. Sur 8 CTMs considérés 6 sont détectés avec un taux de $100\,\%$ les deux restant avec un taux de $73\,\%$ et $86\,\%$. La méthode semble donc efficace. Cependant, l'apport de la combinaison $\{\mathrm{RO} + \mathrm{consommation}\}$ considéré n'a pas été expérimentalement quantifié.

L'utilisation de méthodes combinées a donc été peu explorée. À notre connaissance, aucun résultat à ce jour ne met en valeur un apport supérieur à l'union des ensembles de CTMs couverts par les méthodes respectives.

1.3.6 Méthode de prévention contre les chevaux de Troie matériel

Les mesures préventives ont pour but d'empêcher l'infection d'un circuit en rendant cette dernière impossible. Comme toute contremesure en sécurité, elles cherchent à augmenter le coût, l'effort et les connaissances nécessaires à la réussite d'une attaque. Dans le cas présent, on cherchera donc à rendre l'infection physiquement plus difficile à mettre en place ou encore à rendre les connaissances nécessaires pour mener l'attaque à terme inaccessible.

La conception en vue du test "*Design for Testability*" (DfT) a pour but de réduire le coût du test et d'augmenter son efficacité. Cela a pour but de diminuer les temps de test et d'augmenter leur couverture. Appliquée à la problématique d'intégrité des circuits, on utilise le terme "*Design for Hardware Trust*" [Raj+11b; Raj+14]. Cela comprend l'ensemble des techniques permettant de prévenir ou de faciliter les modifications malveillantes du circuit. Cette section traite des différentes méthodes de prévention existantes.

"Built-In Self-Test"

Le "Built-In Self-Test" (BIST) est un mécanisme que l'on intègre lors de la conception du circuit afin de faciliter le test fonctionnel effectué par les équipements de test automatique (ATE: Automatic Test Equipment). En plus du fonctionnement normal, un mode de test au circuit est donc prévu. Ce mode désactive les fonctionnalités normales du circuit. Il permet un accès en lecture et en écriture à certains nœuds internes du circuit, augmentant ainsi l'observabilité et la contrôlabilité globale du circuit.

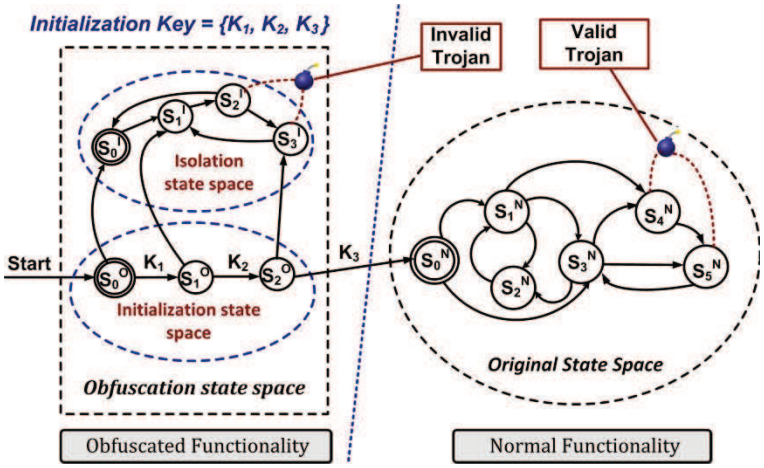

Figure 1.25: Schéma d'obfuscation pour cacher les évènements rares[Cha09].

Dans le cadre de la détection de CTMs, ce mécanisme permet d'améliorer les méthodes présentées dans la section 1.3.3. Un ajout matériel, dans le but d'améliorer les tests logiques pour la vérification d'intégrité, est présenté dans [Cha+08]. Il consiste en l'ajout d'une FSM (Finite-State Machine), activée par une clé secrète, qui a pour but de mieux contrôler les signaux à faible contrôlabilité et de générer, à partie des signaux à faible observabilité, une signature à ressortir au travers des sorties primaires. Une différence entre la signature obtenue et la signature attendue indique une infection.

Obfuscation

L'obfuscation consiste à modifier un circuit de manière à ce qu'il conserve ses fonctionnalités tout en rendant les processus de rétro-ingénierie plus complexes à mettre en œuvre. Les auteurs de [Cha09] proposent une méthode d'obfuscation des circuits destinée à complexifié les infections de CTM. Le but de ce schéma est d'avoir trois espaces de graphe pour masquer une fonctionnalité. Premièrement, un espace d'initialisation sert à atteindre les deux autres espaces. Si la séquence d'initialisation est correcte, le prochain état sera dans l'espace de la fonctionnalité originale sinon dans un espace d'isolation bouclant sur lui même. La figure 1.25 illustre ce principe. Ainsi toute infection dans l'espace d'isolation sera sans effet, cela complique la tâche consistant à identifier les points auxquels il est possible d'insérer un CTM.

Camouflage

Le camouflage est une technique s'appliquant sur le placement-routage d'un circuit avec pour objectif d'empêcher un attaquant d'extraire la netlist d'un circuit grâce aux images des différentes couches. Dans [Cho+07], une méthode est proposée. Elle permet de

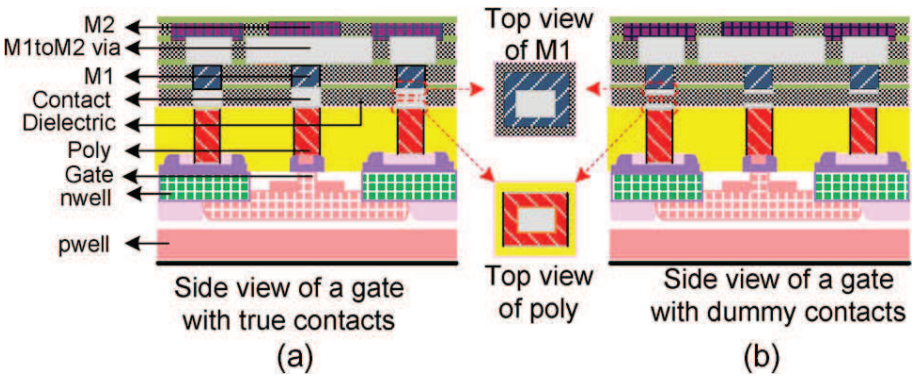

Figure 1.26: Cellule camouflée par contact factice. (a) Une cellule avec un vrai contact. (b) Une cellule avec un contact factice [Cho+07].

créer des contacts factices difficilement distinguables d'un vrai contact. Cela a pour but d'empêcher un attaquant de deviner la fonction d'une cellule standard camouflée. La figure 1.26 illustre ce principe appliqué à une cellule. Dans le cas de l'insertion de CTM, empêcher l'accès à la netlist permet de limiter les possibilités qu'aura un attaquant d'identifier des points d'infections viables et pertinents.

Chiffrement logique

Le chiffrement logique est utilisé pour cacher les fonctionnalités ou autres détails de l'implémentation d'un circuit [Roy+08; Raj+12]. Cette technique consiste à rajouter des portes logiques qui prendront en entrée un bit de clé servant à déverrouiller les fonctionnalités du circuit. Pour que le circuit ait la bonne fonctionnalité, donc que les valeurs de sortie soient correctes, la bonne clé doit être fournie au circuit et transmise aux portes "clés". Ce procédé empêche de retrouver la fonctionnalité du circuit s'il ne possède pas la bonne clé. La qualité du chiffrement peut être évaluée par le nombre de sorties fausses après application de clés aléatoires fausses. Le chiffrement logique empêche donc de cibler correctement un point d'infection. La figure 1.27 représente un circuit chiffré avec trois portes XOR, les sorties O1 et O2 seront correctes seulement si les bits de la clé (K1, K2 et K3) sont corrects.

Dans [Dup+14], la technique de chiffrement vise à changer virtuellement la contrôlabilité des signaux. Ainsi un attaquant ne pourra pas exploiter les signaux à faible contrôlabilité permettant de créer des déclencheurs à évènements rares. De plus, l'addition de portes permet l'ajout des points d'entrées dans le circuit, augmentant la contrôlabilité des signaux internes et facilitant le test logique.

Les auteurs de [Nej+15] choisissent les portes insérées de façon à augmenter le nombre de chemins courts présents dans sur le circuit. Considérant que la variation de délais d'un chemin est le résultat de l'accumulation des variations de délais de ses constituants,

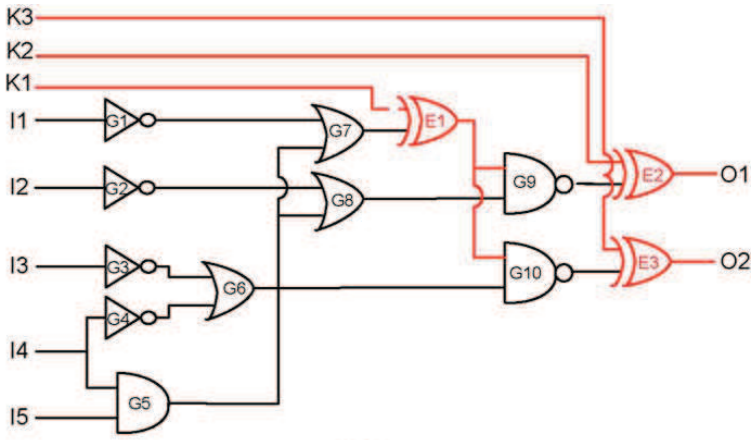

Figure 1.27: Circuit chiffré avec trois portes XOR [Raj+12].

un chemin court est moins soumis aux variations des procédés de fabrication. Un chemin court permet donc d'améliorer les méthodes de détection basées sur les délais. Cependant, un chemin trop petit peut ne pas être mesurable à cause de la fréquence maximale de l'arbre l'horloge.

Ainsi, le chiffrement matériel permet de rendre plus complexe la rétro-ingénierie d'un circuit et donc l'insertion d'un CTM. De plus, plusieurs critères de sélection pour l'insertion des portes de chiffrement peuvent être utilisés pour rendre plus difficile la recherche de points d'intérêts pour l'insertion de CTMs:

- la contrôlabilité du signal,
- la longueur du délai du chemin,
- les emplacements vacants proches permettant une infection.

"Split manufacturing"

La division de la production ("split magnufacturing") a pour but de réduire les menaces pesant sur l'intégrité des circuits induite par la délocalisation tout en conservant un coût de production amoindri. Le principe est de séparer les étapes de production en deux, la première partie, la plus coûteuse sera effectuée dans une usine classique (donc probablement à l'étranger pour des raisons de coût). La seconde étape sera effectuée dans une usine de confiance. De cette façon, le layout du circuit est séparé entre le "*Front End Of the Line*" (FEOL) et le "*Back End Of the Line*" (BEOL). Les couches FEOL sont constituées des transistors et des métallisations inférieures. Les couches BEOL correspondent aux couches supérieures de métal. Cette division est représentée dans la figure 1.28. Plus la division est effectuée à un bas niveau de métal, plus les

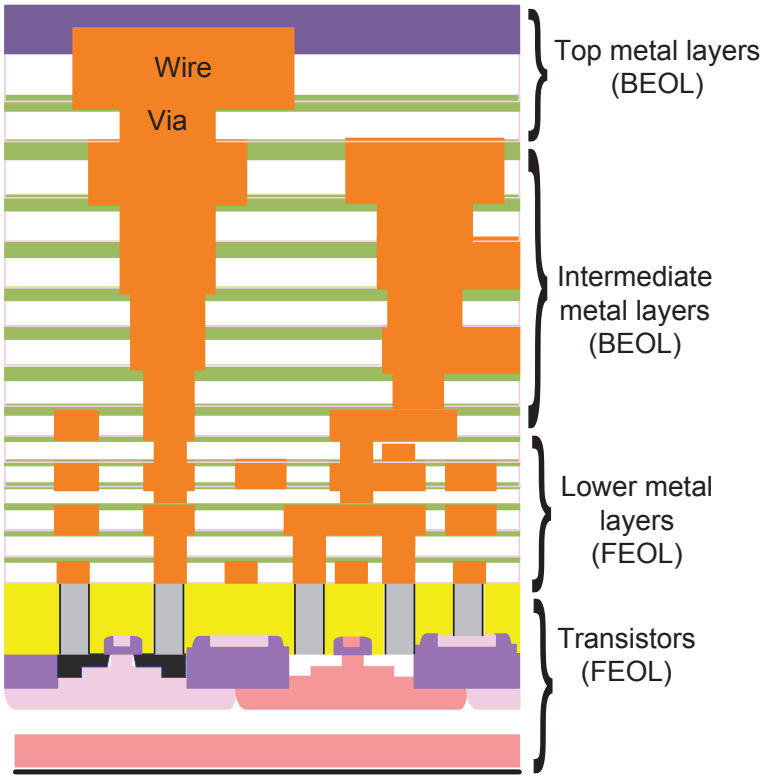

Figure 1.28: Vue en coupe d'un layout de circuit [Raj+13]

fonctionnalités du circuit sont difficiles à reconstituer, mais plus le coût de production devient élevé. La principale difficulté de ce processus réside dans l'alignement en raison de l'échelle nanométrique des interconnexions. Cet alignement est effectué en utilisant des techniques d'alignement électrique, mécanique ou optique [JM07]. Cet alignement est facilité par les différences de taille entre les éléments des différentes couches.

Ainsi les informations accessibles à un attaquant lors de la première partie de la production ne lui permettent pas d'avoir une idée des fonctionnalités ni de savoir où insérer un CTM. Cependant, une attaque ("*Proximity attack*") utilisant les propriétés des algorithmes de placement-routage est présentée dans [Raj+13], elle permet de reconstituer les informations manquantes. Les auteurs proposent une méthode d'échange de broches pour réduire l'efficacité de l'attaque. Cela montre que le "*split manufacturing*" doit considérer le placement-routage pour être efficace et ainsi permettre de cacher les informations fonctionnelles. C'est pourquoi une métrique d'évaluation est proposée dans [Jag+14]

Les auteurs de [Val+13] proposent d'utiliser l'intégration 3D pour assurer la sécurité des composants. L'intégration 3D consiste à combiner plusieurs composants dans un même CI. Ainsi il est possible de séparer les éléments du circuit à protéger du reste. Un composant contenant les parties sensibles du circuit sera produit dans une fabrique de confiance et une puce intégrant le reste des fonctionnalités sera produite dans une usine

délocalisée. De plus, toujours selon [Val+13], la puce sécurisée peut intégrer des fonctions de surveillance de la puce non sécurisée.

Remplissage du circuit

Le placement et le routage d'un CI sont effectués par des algorithmes dédiés. La complexité du procédé implique un taux de remplissage inférieur à $100\,\%$. Le taux de remplissage est le ratio de la surface occupée par les cellules standards sur la surface totale de la puce. Ces emplacements "vides" sont souvent remplis par des cellules de remplissage sans autre utilité que de découpler les alimentations. Ces emplacements offrent donc des possibilités d'ajouts malicieux de logique avec un impact limité sur le layout original.

Dans [Bha+13], les auteurs, étudient l'influence du remplissage sur les possibilités d'infection et leur impact sur le layout. Ils ont observé l'influence de deux paramètres sur le layout. Premièrement la taille du déclencheur du CTM constitué d'une série de portes NAND (1 à 128 portes) et le taux de remplissage du circuit ($50\,\%$ à $99\,\%$). Il en résulte que pour un taux de remplissage supérieur à $80\,\%$ les infections importantes ($>$ 64 portes AND) impactent significativement le layout. Un haut taux de remplissage est nécessaire pour que des infections plus petites aient un impact significatif. Par exemple, un remplissage supérieur à $95\,\%$ est nécessaire pour observer l'impact d'un CTM avec un déclencheur de 16 portes NAND.

Ainsi, un taux de remplissage élevé réduit les possibilités d'infections. C'est en considérant cette observation que les auteurs de [Ba+15] ont proposé une méthode pour augmenter le taux de remplissage du circuit. Le principe est de remplir les espaces vacants du circuit par des blocs fonctionnels dont on peut vérifier le fonctionnement. Ces blocs doivent être suffisamment simples pour permettre de les placer et les connecter sur le reste du circuit. De plus, dans cette proposition, les espaces ciblés sont prioritairement les espaces qui sont proches de signaux à faible contrôlabilité, plus susceptibles d'être choisis pour insérer un CTM. Cette méthode est illustrée dans la figure 1.29. Les registres à décalage sont utilisés pour contrôler et observer les blocs fonctionnels et ainsi s'assurer qu'ils n'ont pas étés altérés ou supprimés pendant un processus infection. Les résultats expérimentaux montrent une augmentation comprise entre $60\,\%$ et $80\,\%$ du taux de remplissage des circuits tester. Les taux de remplissage finaux vont de $90\,\%$, $100\,\%$, ce qui permet de limiter les infections plus importantes selon [Bha+13].

Les auteurs de [Xia+15] proposent une méthode de remplissage nommée OBISA. Cette méthode est couplée avec l'approche du "*split manufacturing*" et de l'obfuscation. Elle a pour but de prémunir les fabricants de CIs de la rétro-ingénierie et l'infection matérielle.

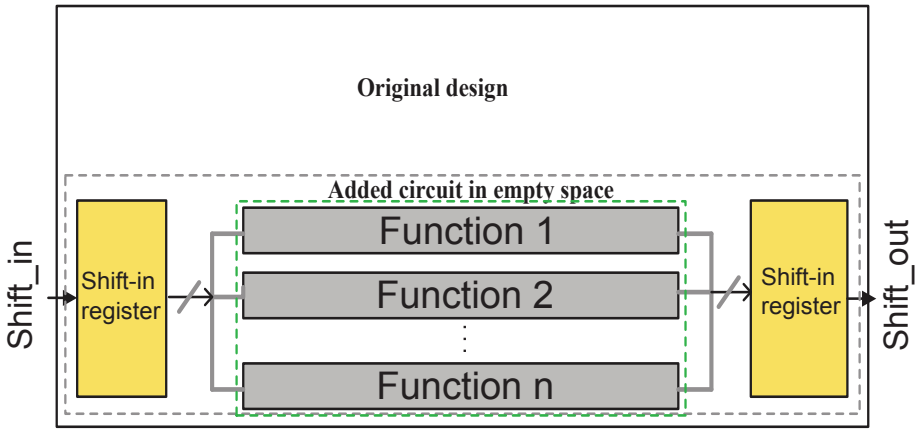

Figure 1.29: Méthode de remplissage [Ba+15]

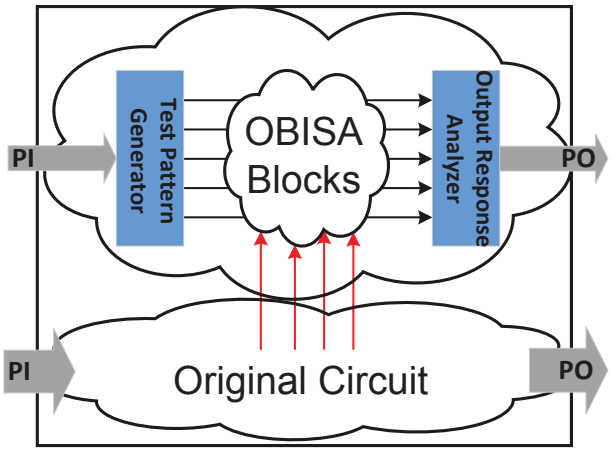

Figure 1.30: Méthode OBISA [Xia+15]

Le principe est d'utiliser le "*split manufacturing*" pour cacher les fonctionnalités et d'augmenter cette obfuscation en connectant des portes logiques supplémentaires au circuit principal. De plus, ces portes sont placées dans les espaces disponibles du layout, sont connectées entre elles et ont une sortie permettant de vérifier leur intégrité. Contrairement à la méthode présentée dans [Ba+15] il n'y pas de registres de décalage pour contrôler et vérifier le fonctionnement des blocs ajoutés, mais un LFSR et un "Multiple Input Shift Register" (MISR) sont utilisés à cette fin. De plus, l'ensemble des portes ajoutées forme un circuit auxiliaire qui peut être testé à une fréquence de fonctionnement réduite au travers du LFSR et du MISR. La figure 1.30 illustre cette proposition.

Encodage des états logiques

Les auteurs de [Ngo14] proposent une méthode appelée "circuit encodé". Considérant que les registres sont les composants du circuit les plus facilement identifiables sur un layout à cause de leur taille [Noh+10], le but de cette approche est de protéger lesdits

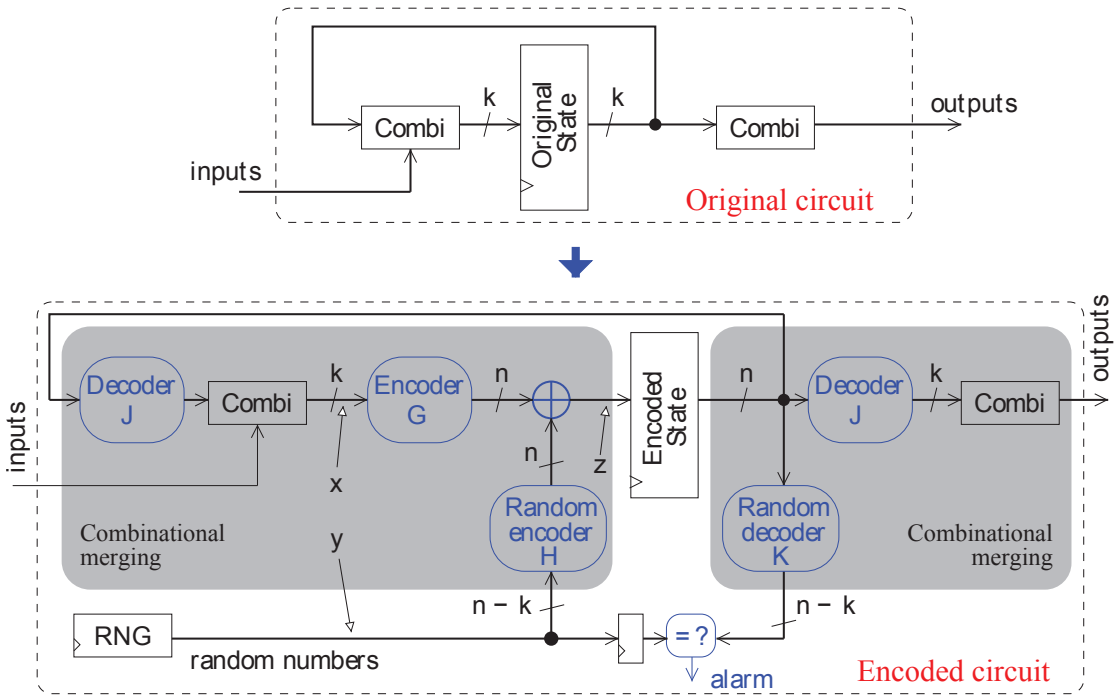

Figure 1.31: Circuit encodé [Ngo14]

registres. Pour cela, ils cherchent à encoder tous les registres sensibles du circuit et ainsi à masquer l'état des registres. La figure 1.31 illustre le schéma de protection. Tout d'abord la sortie de la logique combinatoire initialement sur k bits est encodée sur n bits. Parallèlement, un nombre aléatoire généré sur $n-k$ bits est encodé sur n bits. Une opération XOR est appliquée sur les deux valeurs encodées et le résultat est échantillonné dans un registre de n bits. Les états sont ainsi masqués avec une variable aléatoire empêchant ainsi d'anticiper les valeurs des signaux et de deviner l'utilité des signaux. Cela permet de prévenir l'insertion de CTM en empêchant la sélection de point d'intérêt puisque l'attaquant ne connait pas le rôle des signaux. La différence $d = n - k$ est le paramètre de sécurité. Son élargissement augmente le nombre de registres nécessaires pour échantillonner l'état encodé et donc le coût de la protection.

En plus du système d'encodage, un système d'alarme est déployé. Celui-ci vérifie que le nombre aléatoire utilisé pour masquer les entrées des registres est le même que celui décodé à partir de la sortie des registres. Si ces deux valeurs sont différentes, on en déduit que l'actionneur d'un CTM altère un signal et une alarme est alors déclenchée.

1.3.7 Méthodes spécifiques à la lutte contre la contrefaçon

Nous avons vu dans la section 1.2 que les contrefaçons peuvent être des copies fonctionnelles plus ou moins conformes à un circuit original. Les sections précédentes présentent différentes techniques permettant de détecter les CTMs. Ces méthodes de détection consistent, pour la plupart, à détecter des différences de structures physiques

entre des circuits de référence et des circuits testés. On peut considérer la détection de CTMs et la détection de contrefaçons comme les deux extrêmes d'un spectre représentant le même problème. Dans le cas des contrefaçons, tout le circuit est modifié contrairement aux cas des CTMs où seule une partie du circuit est modifiée. Cette modification peut ne représenter qu'une partie très réduite de la surface totale du circuit. Cette considération n'est pas valable pour les contrefaçons étant des copies conformes du design original (structure physique identique).

Ainsi, les méthodes permettant de détecter les CTMs permettent généralement de vérifier l'intégrité d'un circuit et donc de détecter certaines contrefaçons. Cette section présente des méthodes spécifiques à la lutte contre les contrefaçons. Plusieurs méthodes peuvent être utilisées pour détecter les contrefaçons. La première d'entre elles est l'inspection physique. Elle peut aller de l'inspection visuelle aux techniques à base de rayons X, en passant par différentes techniques de microscopie [Gui+14]. De façon générale, les procédés de rétro-ingénierie permettent de détecter une contrefaçon. Ensuite, une vérification des réponses électriques du circuit peut être appliquée. Cela permet de procéder à des tests fonctionnels ou paramétriques avec une plus grande finesse.

Des méthodes d'authentification comme les PUF (Physical Unclonable Function) permettent d'authentifier un circuit [SD07; Lim+05; Dev+08] et ainsi de s'assurer de la provenance d'un circuit. Les PUF servent à générer une clé unique et inclonable par circuit. Pour cela on utilise les propriétés uniques de chaque composant. Concrètement, on cherche à mesurer des grandeurs physiques impactées par les variations des procédés de fabrication (qui sont aléatoires et non reproductibles), et à en extraire une clé numérique permettant d'authentifier individuellement chaque circuit considéré.

Les CIs sont sujets au phénomène de vieillissement. Il est donc possible d'intégrer des capteurs exploitant ce phénomène afin de mesurer l'aging d'un CI. La figure 1.32 présente un capteur basé sur des ROs [Zha+12], permettant d'évaluer le nombre d'utilisations du circuit. Ce type de capteur est utile dans la lutte contre la contrefaçon. En effet, ils permettent de détecter des circuits remis sur le marché après recyclage.

Les techniques permettant de détecter les CTMs permettent de détecter les circuits avec des layouts non conformes, ce qui représente une partie des possibles contrefaçons. En effet, lorsqu'une contrefaçon est une copie conforme à un circuit original, le layout est identique et donc les méthodes détectant des modifications de structure sont inefficaces. Dans ce cas, un chiffrement du circuit peut permettre de contrôler la distribution des circuits au design original, ou une solution d'authentification telle que les PUF peut être mise en place. Enfin, des capteurs basés sur les phénomènes de vieillissement permettent de détecter les circuits de récupérations.

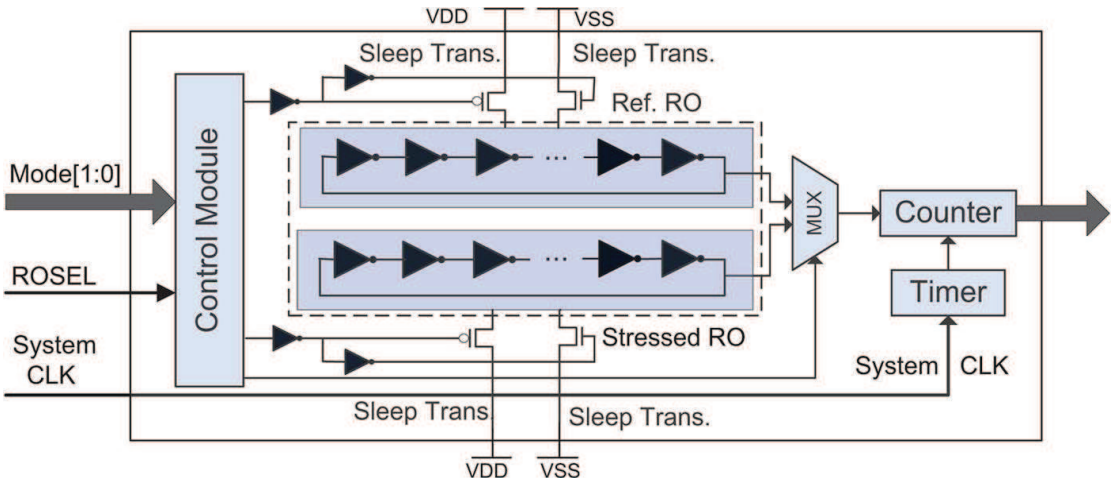

Figure 1.32: Capteur de contrefaçons [Zha+12]

1.4 Positionnement de la thèse

Comme indiqué dans la section 1.2, les menaces pesant sur l'intégrité des composants d'un CI sont nombreuses. La figure 1.33 synthétise les différentes étapes, depuis la conception jusqu'à l'exploitation de CI et les menaces sur l'intégrité. La première vulnérabilité se trouve à l'étape de conception. Un bloc matériel provenant d'une tierce partie possédant une fonctionnalité malveillante peut être implémenté dans le produit (Menace (1) dans la figure. 1.33) ou un employé compromis de l'entreprise ou encore une faille de sécurité peut permettre l'introduction d'un CTM dans la description HDL (Menace (2) dans la figure 1.33).

La seconde étape vulnérable est la fabrication (Menace (3) dans figure. 1.33). Par exemple, les cellules de remplissage peuvent être substituées par de la logique induisant un déni de service ou réalisant une fonctionnalité malveillante plus complexe, un fusible peut être désactivé, etc. C'est l'étape la plus vulnérable à l'insertion de CTM.

La dernière menace est celle des contrefaçons. Elle peut consister en la vente de produits de qualité inférieure, des produits hors spécifications, des produits de seconde main ou des copies fonctionnelles. Ces ventes créent des pertes financières ou des risques pour la fiabilité ou la sécurité (Menace (4) dans la figure 1.33).

Le travail présenté dans cette thèse porte sur les menaces liées à la production par des tiers (Menaces (3) et (4) dans la figure 1.33) c.-à-d. les altérations de la structure physique du circuit par rapport à un circuit de référence. Considérant les difficultés logistiques et de production liées au cycle de vie des CIs actuels, l'infection d'une seule pièce ne sera pas considérée dans notre modèle de menace. La méthodologie proposée n'a pas pour but de déterminer si un CI est infecté, mais vise à vérifier l'intégrité d'un lot de CIs. Nous cherchons à vérifié si l'ensemble des CIs d'un lot sont infectés par un CTM.

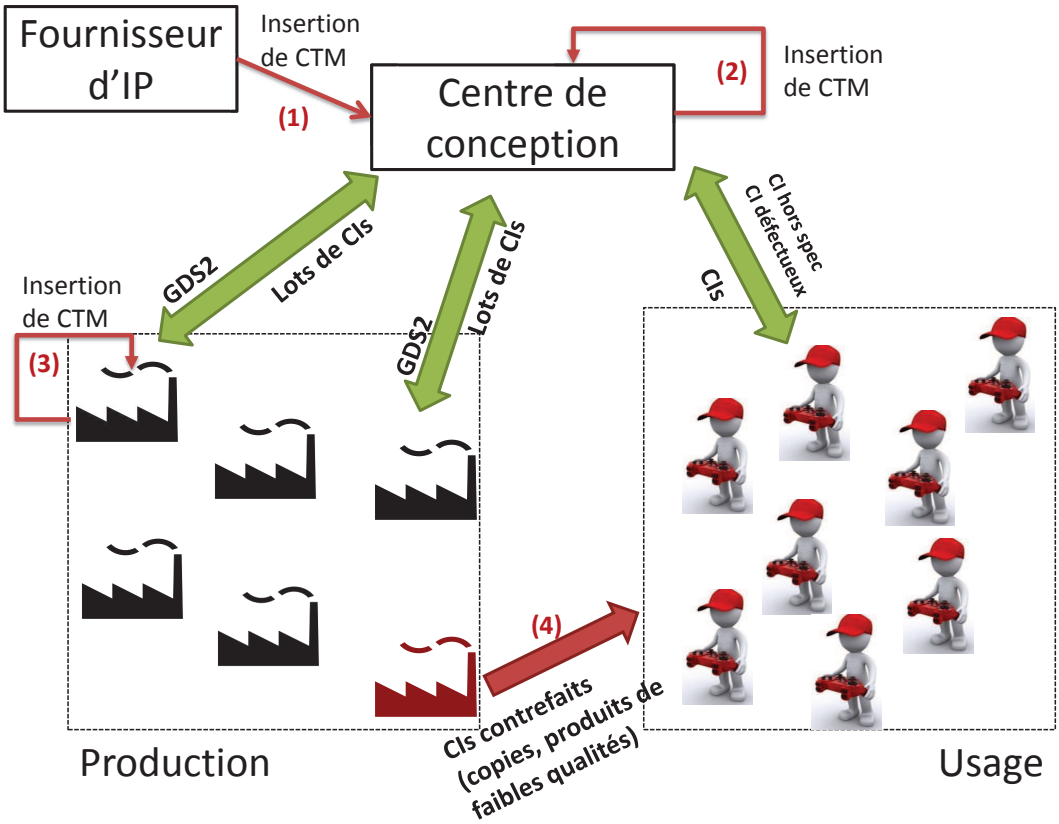

Figure 1.33: Menaces principales pesant sur l'intégrité des CI

Cependant, le scénario consistant à déterminer si un CI est une contrefaçon reste pertinent. Cette approche vise donc à pouvoir déterminer si une pièce est contrefaite ou non. Les contrefaçons n'étant pas caractérisées par une modification de la structure physique du design (c.-à.-d. une copie conforme) ne seront pas considérées.

Les études reportées précédemment dans ce chapitre montrent que les mesures locales semblent les plus pertinentes en termes d'efficacité pour détecter la présence de CTMs. Parmi les solutions permettant une prise locale d'information, l'utilisation de capteurs intégrés semble permettre d'atteindre de hauts taux de détection. De plus, cette solution semble industriellement viable. En effet, le coût de l'équipement et les temps d'acquisition des données sont réduits car les tests peuvent être faits en parallèle. Cette thèse a donc pour but de proposer une méthode de détection embarquée de détection de CTM et de contrefaçons, cela implique des mesures internes et une analyse simple pouvant être embarquées sur une surface limitée.

Les techniques de détections présentées dans ce chapitre ne permettent pas une détection de 100 % des CTMs. Dans ce travail, nous cherchons à proposer une méthode de détection de CTM permettant d'atteindre un taux de détection de 100 % pour un ensemble de CTMs clairement défini (en taille, position et activité électrique) en fonction de l'implémentation du système de mesure.

Le point de départ de ce travail est l'utilisation de réseaux d'oscillateurs en anneau analogues à ceux déployés dans [XT11; Fer+12; Kar+15]. Les analyses utilisées sont basées sur des méthodes statistiques d'apprentissage. Ces analyses représentent un fort coût d'intégration. Dans le but de proposer une méthode embarquée, on cherche à une procédure d'analyse ayant un coût d'intégration inférieur à celles déjà proposées. De plus, les travaux de [Sol+; Bal+15; XT+15; Abd+16] ont montré que l'activité d'un CTM induit des fuites par émanations électromagnétiques. À partir de ce résultat, nous testerons si la méthode de détection de CTMs développée pour exploiter un réseau de capteurs embarqués est adaptable à une analyse EM.

Ensuite, les différentes méthodes par canaux auxiliaires présentées dans ce chapitre nécessitent une activité du déclencheur du CTM pour être efficaces. Pour cela, des vecteurs de test sont soumis de façon externe. Dans le cas des mesures de délais, les vecteurs externes sont utilisés pour pouvoir mesurer des chemins précis. Toujours dans l'optique d'une détection embarquée, il est nécessaire de pouvoir tester les circuits sans stimuli externes afin de limiter le besoin en équipements de test externes. Dans ce but nous explorerons une possibilité de détection sans stimulation du déclencheur du CTM.

Afin de développer une méthode de détection répondant aux critères cités ci-dessus, les contributions suivantes ont été apportées.

- Une caractérisation expérimentale de l'impact d'une insertion malveillante sur un réseau de capteurs basée sur les ROs. Cette caractérisation est présentée dans le chapitre 2.

- Un nouveau modèle de variation des performances des structures CMOS dans un circuit réel (différent des circuits dédiés aux mesures fines des performances des variations intra et inter die). Ce modèle est introduit dans le chapitre 3 puis expérimentalement validé dans le chapitre 4.

- Une méthodologie de détection utilisant un réseau de capteurs embarqués. Celle-ci permet de détecter une altération de la structure physique d'un circuit induite par la présence d'un CTM, ou par des modifications de son placement routage ou de son floorplan. Cette méthode est introduite dans le chapitre 3 et validée dans le chapitre 4.

- Un nouveau distingueur permettant de décider si un lot est infecté. Il permet de réduire le taux de faux positifs et une grande capacité de détection. Ce distingueur est dédié à la détection de CTMs furtifs, c.-à-d. de CTMs ayant un impact spatial restreint (petite consommation ou petite taille) et/ou un impact réduit dans le temps (le déclencheur ne consomme que pendant un intervalle de temps réduit). Il

est indépendant du moyen de mesure et peut s'appliquer sur des mesures externes ou obtenues par un réseau embarqué de capteurs. Ce distingueur est introduit dans le chapitre 3 et expérimentalement évalué dans le chapitre 4.

2

Émulation des chevaux de Troie matériel et mesure de leur impact

Ce chapitre présente une caractérisation de l'impact de l'activité électrique d'un CTM sur un réseau d'oscillateurs en anneau. Ce réseau d'oscillateurs nous permet d'obtenir des cartographies des tensions à l'intérieur des circuits. Cette caractérisation est suivie d'une discussion sur l'utilisation de compteurs embarqués pour mesurer les fréquences d'oscillation des ROs dans le cadre de la détection de CTM. Cette caractérisation et cette discussion ont été publiées dans [Lec+15]. Les résultats montrent que les CTMs ont un impact dynamique très inférieur aux variations des procédés de fabrication et un impact statique très supérieur à l'impact dynamique. Nous en déduisons que l'utilisation de compteur pour évaluer la fréquence des ROs ne permet pas de détecter l'impact dynamique des CTMs, contrairement à leur impact statique qui peut être mesuré ainsi. Enfin on observe qu'une modification du placement-routage du circuit impacte fortement le réseau de capteurs. Ce principe général a fait l'objet d'un brevet [Lec+16c].

Le postulat de départ de l'étude menée dans ce chapitre est que l'insertion de logique additionnelle dans un CI, et plus précisément celle d'un CTM, modifie sa structure interne et donc la distribution statique de la tension d'alimentation dans sa grille d'alimentation et ce même si cette logique reste au repos. La première répercussion de cet impact est l'effet statique de l'insertion de CTM. Un second impact peut être observé, quand celui-ci est activé. En effet, cette logique malicieuse augmente temporairement la consommation en courant avec une amplitude dépendant du nombre de bits commutant. Ceci est l'effet dynamique de l'insertion de CTM. Dans le but de valider et quantifier les impacts statiques et dynamiques de l'insertion d'un CTM, un circuit test, embarquant une structure de mesure et un bloc d'émulation de CTM, a été implémenté. Le circuit a été implémenté sur un FPGA conçu en technologie $90\,\mathrm{nm}$. La structure de mesure est une matrice de ROs déployée pour couvrir une grande partie de la surface du CI. Pour émuler l'impact d'un CTM, une structure séquentielle, dont on peut modifier les modalités de fonctionnement, c.-à-d. modifier son activité de commutation ou son arbre d'horloge, a été implémentée sur le design. Dans un second temps, une implémentation d'un algorithme cryptographique est utilisée pour créer un bruit de fond lié au calcul d'une implémentation. Ceci est fait dans le but de vérifier la validité des mesures dans un

environnement réel. Enfin, à partir des résultats obtenus, l'utilisation d'oscillateurs en anneaux comme capteurs pour une détection embarquée de CTM est discutée.

2.1 Circuit de mesure

Un circuit test a été implémenté afin d'analyser précisément l'impact de l'insertion d'un bloc malicieux. Pour cela un design a été implémenté sur un FPGA Xilinx Spartan-3E 1600E [Fpg] en utilisant les outils de Xilinx (ISE 14.7 et FPGA editor). Cette section détaille les différents blocs utilisés, leurs rôles et les techniques d'implémentation adoptées.

Le layout du circuit est reporté sur la figure 2.1 et l'architecture simplifiée est présentée figure 2.5. On distingue quatre blocs principaux:

- un réseau de 60 oscillateurs en anneau qui peuvent être activés séparément, afin de limiter leurs influences mutuelles (ce nombre de ROs permet une mesure dans un temps limité,
- un registre à décalage à rétroaction linéaire émulant un CTM,
- un bloc de contrôle intégrant une machine à états finis (FSM) qui supervise la mesure,
- un bloc de communication série (RS232) qui prend en charge la communication entre le circuit et l'ordinateur.

2.1.1 FPGA

Le circuit de caractérisation a été réalisé sur FPGA ("*Field-Programable Gate Array*"). L'utilisation de FPGAs permet de modifier les paramètres étudiés (taille, position, activité électrique) pendant la caractérisation. En effet, les FPGAs sont des circuits reprogrammables, les comportements des blocs combinatoires peuvent être modifiés et les connexions entre les blocs peuvent être reroutées. Un circuit ASIC est dédié à une tâche particulière et permet peu de modularité, alors qu'un FPGA permet de changer de placement-routage ou d'implémentation. Malgré les différences entre ASIC et FPGA (tous les éléments sont présents par défaut, le routage est contraint dans des canaux . . .), nous faisons l'hypothèse que les comportements observés sur FPGA seront représentatifs des comportements observés sur ASIC. En effet, cela semble être couramment admis, plusieurs travaux antérieurs ont utilisé des FPGAs à de telles fins [Raj+11a; XT11; Bal+15; XT+15]. L'utilisation de prototypage FPGA permet d'obtenir des résultats

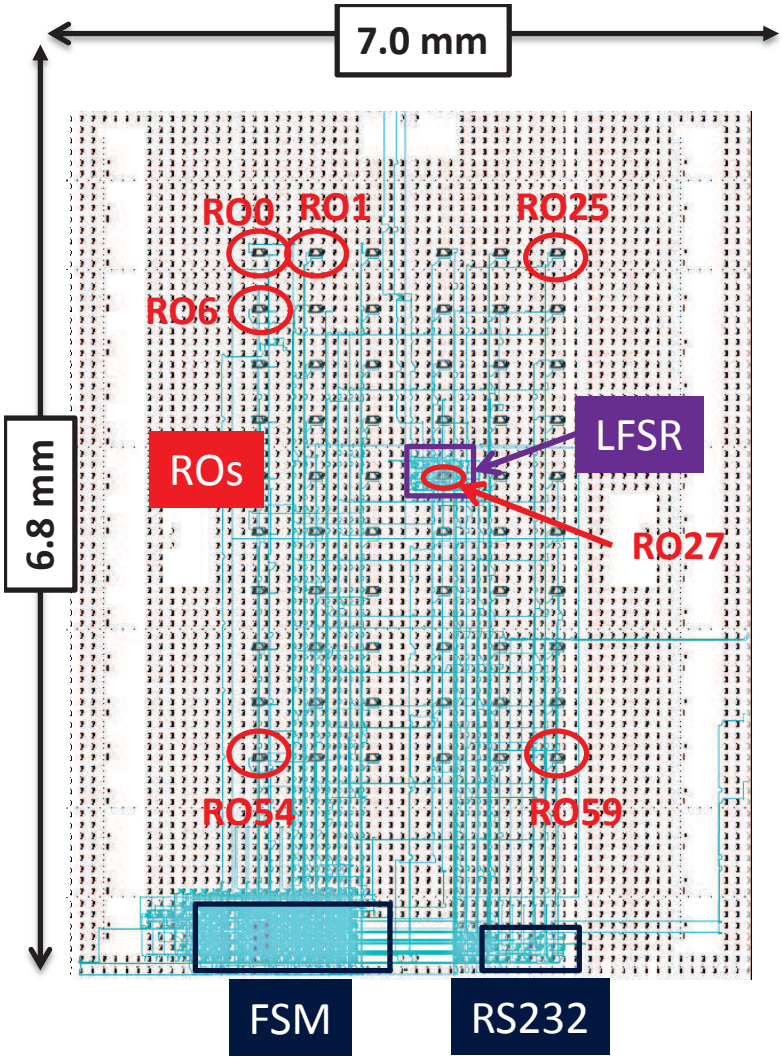

Figure 2.1: Floorplan du circuit test de caractérisation.

expérimentaux à un moindre coût et en un temps court (les coûts et les durées impliqués par les cycles de production de circuits dédiés sont importants, en particulier sur la durée d'une thèse).

Concrètement, les FPGAs sont principalement composés d'une matrice de blocs logiques configurables (CLB) qui réalisent des fonctions combinatoires ou séquentielles. Les CLB sont composés de "*slices*", des matrices de connexion permettent de connecter les slices entre eux en fonction des fonctionnalités décrites.

Les FPGAs utilisés dans ces travaux sont des Spartan 3E-1600 (xc3x16000e) encapsulés dans des boitiers 4fg320. Dans ce modèle, chaque slice comporte 2 bascules D et 2 "*Look Up Tables*" (LUT). Les LUTs sont principalement composées de 4 entrées et d'une sortie, et sont programmables pour effectuer n'importe quelle fonction logique combinatoire). Au total, ces FPGAs sont composés de 14752 slices ce qui donne 29504 bascules D et 29504 LUTs.

2.1.2 Chaîne d'implémentation FPGA

La description du circuit a été faite en VHDL (VHSIC Hardware Description Language) en utilisant la suite de logiciel Xilinx ISE 14.7. C'est un langage de description logique, permettant de décrire le comportement logique du circuit. Un second logiciel, "FPGA editor" également fournit par Xilinx, a été utilisé pour créer des «hard macros». Les «hard macros» sont des blocs programmés, routés et placés à partir d'un point de référence. Ces blocs peuvent ensuite être placés à l'emplacement désiré dans un autre design en conservant leur placement et routage relatif.

En plus de la description HDL et des «hard macros», les contraintes de placement sont décrites dans un fichier ".ucf". Ce fichier contient trois types de contraintes principales qui seront utilisés dans nos implémentations:

- les limites de placements des différents blocs logiques décrits en VHDL,
- l'assignation des différentes entrées/sorties physiques du FPGA,
- l'emplacement précis des «hard macros».

Finalement, trois types de fichiers doivent être écrits pour décrire le design final:

- les fichiers ".vhd" décrivant la logique à implémenter,
- le fichier ".ncm" décrivant la forme exacte des ROs,

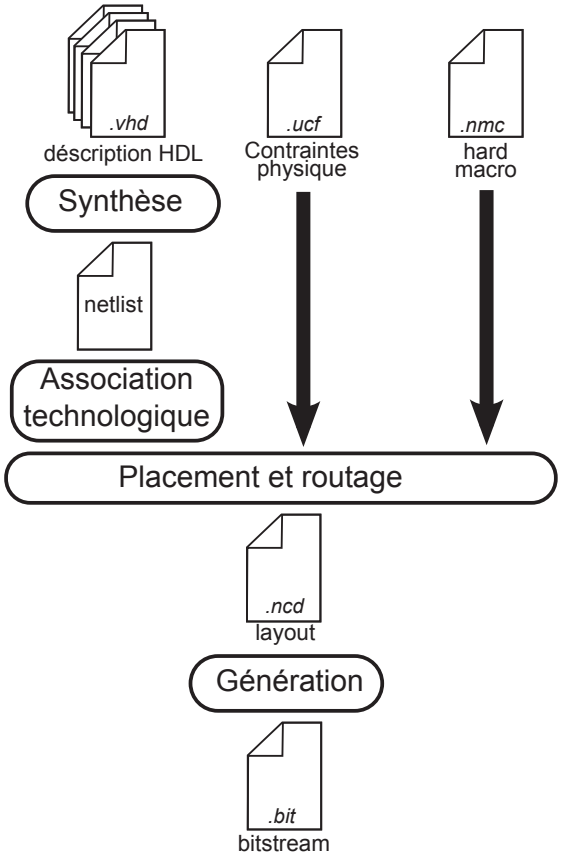

Figure 2.2: Procédure de programmation d'un FPGA.

- le fichier ".ucf" décrivant les contraintes de placement sur le FPGA.

Une fois ces fichiers complétés, on utilise la chaîne de compilation et d'intégration des outils Xilinx. Cette procédure est illustrée sur la figure 2.2 et est composée de trois étapes principales.

Synthèse La synthèse prend en entrée une description logique (dans notre cas en VHDL) et génère la description des portes logiques et de leurs interconnexions (le résultat est appelé "netlist"). Cette étape est indépendante de la cible ASIC ou FPGA.

Implémentation À partir de la netlist, la projection technologique décrit les éléments de la technologie cible qui seront utilisés. Dans notre cas cela correspond à la configuration des slices et de leurs interconnexions ainsi que la programmation des LUTs. Après la projection technologique, l'étape de placement-routage choisit l'emplacement des slices et les chemins de routage. C'est à cette étape que les «hard macro» sont placées et connectées au reste du circuit. Les contraintes de placement décrites dans le ".ucf" sont utilisées par l'algorithme de placement. Le résultat est un fichier ".ncd" (Native Description Circuit) comportant toutes les informations de placement, routage et configuration spécifique au FPGA utilisé.

Génération Enfin, le *bitstream* est généré, c'est un fichier binaire utilisé pour la configuration du FPGA. Il est directement chargé dans le FPGA pour le programmer.

La chaîne d'implémentation est utilisée pour générer les bitsreams qui seront implantés sur FPGA. Cependant, en fonction des besoins nous serons amenés à ajouter des étapes. Les éventuelles modifications seront décrites dans les sections associées ultérieurement.

2.1.3 Oscillateurs en anneaux

Les ROs sont traditionnellement utilisés en tant que capteurs pour mesurer localement la tension d'alimentation interne sur la totalité de la surface du circuit. La FSM et le bloc RS232 sont placés suffisamment loin des ROs de manière à ne pas les influencer comme illustré dans la figure 2.1, figure qui donne le layout du circuit de caractérisation. On peut observer que les 60 ROs forment une matrice 6x10. Ce placement a été adopté pour permettre une analyse spatiale des impacts statiques et dynamiques d'un CTM sur la distribution interne de la tension d'alimentation. Les 60 ROs ont exactement le même design. Ils sont composés de 4 inverseurs et d'une porte NAND2 qui permet de désactiver et d'activer chaque RO séparément. La période d'oscillation (T_i) du RO i est environ de $150\,\mathrm{MHz}$ ($6.667ns$) en fonction de la qualité locale du procédé de fabrication.

Pour améliorer la qualité des mesures des fréquences d'oscillation effectuées au travers des pins de sortie du FPGA, chaque RO est connecté à un diviseur d'horloge qui permet d'obtenir une mesure plus propre de $\mathrm{T_i}/2$ et ainsi de $\mathrm{T_i}$.

La conception des ROs a été effectuée sous la forme de «hard macros». Celles-ci ont été utilisées pour deux raisons. Premièrement, lors d'une synthèse haut niveau les ROs peuvent être optimisés, c.-à-d. réduire leurs nombres d'étages réduits ou carrément être remplacés par une porte simple. Car le synthétiseur prend seulement les comportements logiques en compte. De plus, l'utilisation de «hard macros» permet d'assurer que tous les ROs seront strictement identiques et auront donc la même fréquence intrinsèque.

Le diviseur d'horloge est composé d'une bascule D avec une rétroaction inversée comme illustré dans la figure 2.3. Le signal d'entrée est connecté sur l'entrée d'horloge de la bascule et le signal d'horloge divisé échantillonné est présent sur la sortie Q.

La «hard macro» a été créée sur le logiciel FPGA editor, permettant d'éditer précisément l'implémentation sur FPGA. Une fois la «hard macro» créée, elle peut être intégrée dans le flot de conception haut niveau grâce au ficher généré d'extension «.nmc». La figure 2.4 montre l'implémentation de la «hard macro» sur FPGA editor. Chaque rectangle correspond à un slice. Chaque rectangle rempli en noir (6 en tout) correspond à un slice

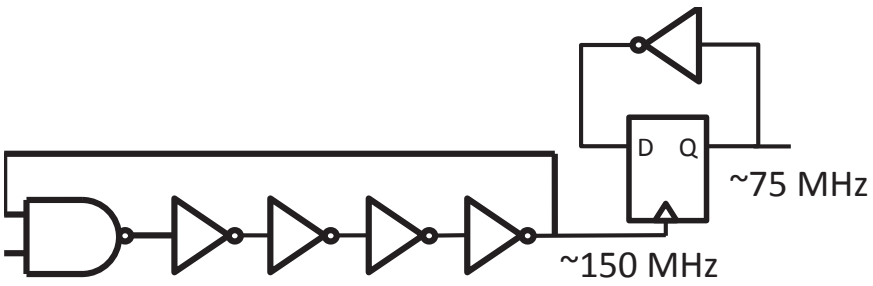

Figure 2.3: Oscillateur en anneau à 5 étages et diviseur de fréquence par 2.

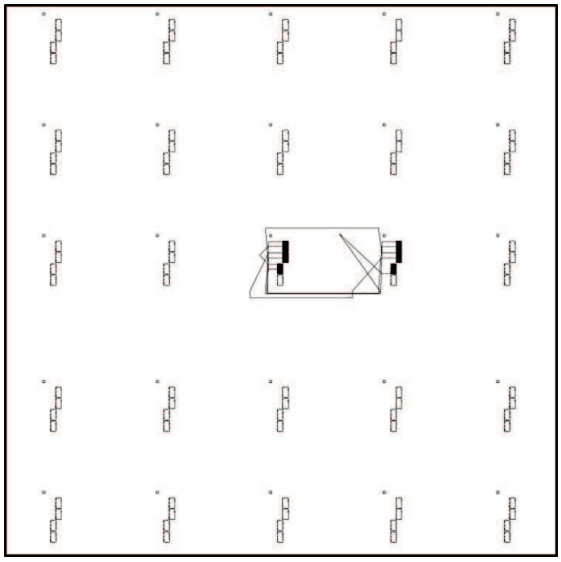

Figure 2.4: Layout du RO à 5 étages

configuré (c.-à-d. utilisé). Un slice pour la porte NAND, 4 slices pour les inverseurs et un slice pour le diviseur de fréquence.

2.1.4 LFSR

Afin d'émuler l'impact d'un CTM, impact lié aux commutations de bascules D, un registre à décalage à rétroaction linéaire (LFSR) a été placé autour du RO27 comme le montre la figure. 2.1. Ce LFSR a été conçu afin de pouvoir contrôler la taille (en bits) des mots qu'il manipule (de 4 et 64 bits). Ceci a été fait pour imiter l'impact dynamique équivalent aux commutations électriques de 2, 8, 16 et 32 bascules D. De plus, l'horloge du LFSR est contrôlable afin de pouvoir observer de façon indépendante l'impact de l'addition d'une branche d'horloge sur le comportement du CI, c.-à-d. quand il n'y a pas de commutation électrique du LFSR lui-même. Le LFSR possède donc 7 modes de fonctionnement. Un mode correspond à l'horloge du LFSR désactivé et 6 modes correspondent à des état lorsque l'horloge est activée. Chacun de ces 6 modes correspond à une taille de mot manipulé par le LFSR (0, 4, 8, 16, 32 ou 64 bits).

Afin qu'une infection soit la plus discrète possible, son impact sur le layout doit être minimum. Or, une modification de la description HDL peut provoquer des changements importants sur le résultat du placement-routage du circuit final. De plus, ces modifications du placement et du routage peuvent avoir un impact important sur les fréquences des ROs. Afin de garder une cohérence dans les mesures et d'obtenir une infection la moins impactante possible, le placement et le routage doivent être modifiés au minimum. Une infection au niveau HDL semble donc peu adaptée. Dans le but de mesurer l'influence de l'implémentation d'un CTM sur le FPGA, il est nécessaire d'avoir un fichier de configuration avec le LFSR et un fichier de configuration sans le LFSR, et les deux fichiers doivent être les plus proches possible en termes de placement-routage.

Pour créer les deux fichiers de configuration, nous avons ajouté une étape dans la chaîne d'implémentation. Cette étape consiste à modifier le fichier ".ncd" obtenu a l'issu du placement-routage avant de générer les bitstreams. Pour cela nous avons utilisé le logiciel FPGA editor. Outre la création de «hard macro», ce logiciel permet d'éditer manuellement le fichier ".ncd" (fichier placement et routage). La procédure de création des fichiers est la suivante. Dans un premier temps, un fichier ".ncd" correspondant au design avec le LFSR est créé. On utilise le fichier de contrainte ".ucf" pour placer le LFSR à l'emplacement souhaité. Dans un second temps, on vérifie que l'implémentation du LFSR est optimisée, c.-à-d. que le minimum de slices a été utilisé. Ensuite, on corrige manuellement, si nécessaire, l'implémentation du LFSR (optimisation et placement). Le layout obtenu sera celui infecté. Pour générer le fichier correspondant à un layout non infecté (sain) on supprime manuellement les slices et fils de routage utilisés par le CTM. En suivant cette procédure, nous limitons au maximum les différences entre les deux fichiers et obtenons un fichier de configuration avec LFSR (infecté) et un fichier sans LFSR (sain).

2.1.5 FSM

La FSM est le bloc de contrôle qui contient en plus d'une machine d'états des registres de configuration, contenant les paramètres de mesure, un démultiplexeur permettant de contrôler le RO actif et un multiplexeur permettant de récupérer le signal oscillant du RO sélectionné. L'architecture simplifiée du circuit test est présentée figure 2.5. En dehors du signal de reset et de l'horloge externe (non représentés), on distingue 4 entrées et sorties du FPGA. Les signaux "rx" et "tx" sont utilisés pour la communication série avec l'ordinateur. Le signal "synchronisation oscilloscope" sert à la synchronisation de l'oscilloscope avec le circuit (il envoie le signal démarrant la mesure) et le signal "ro_out" est le signal oscillant provenant directement des ROs, ce signal est également envoyé à l'oscilloscope.

Le registre de contrôle comporte 24 bits et contient 3 paramètres:

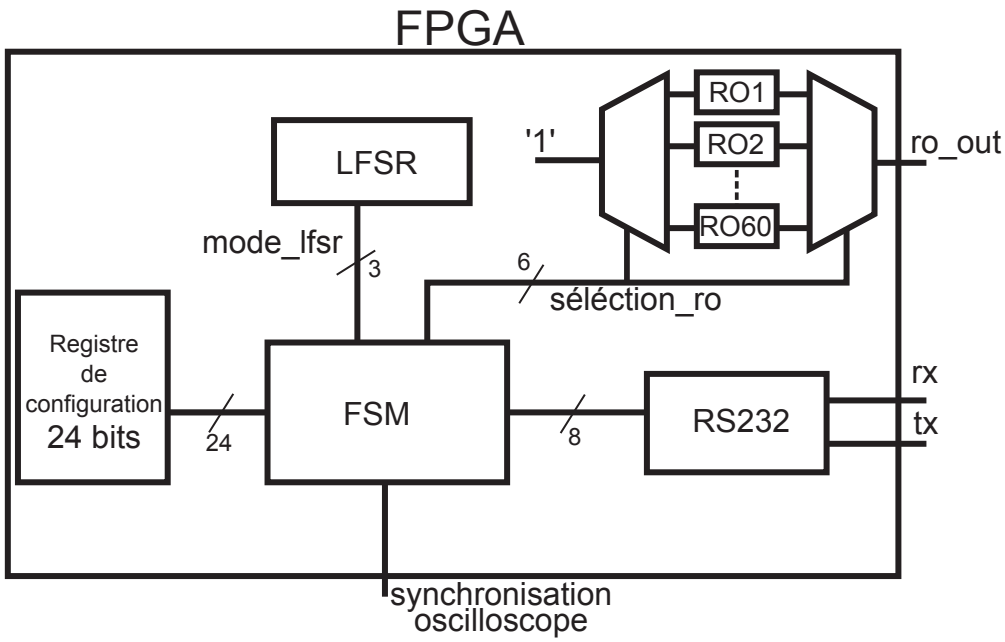

Figure 2.5: Architecture du circuit test.

- le temps de mesure en nombre de cycles d'horloge, cela permet de contrôler le nombre d'oscillations du RO (15 bits),
- Le RO dont on mesure la période d'oscillation (6 bits),
- l'activité du LFSR pendant la mesure, nombre de bits commutant et état de l'horloge (active ou inactive) (3 bits).

Le comportement de la machine d'états pour un point de mesure est le suivant:

1. réception des paramètres de mesures,
2. activation du RO sélectionné, et configuration de l'activité LFSR,
3. mise à l'état actif du signal de synchronisation utilisé par l'oscilloscope,
4. attente du temps de mesure paramétré,
5. mise à l'état inactif du signal de synchronisation,
6. désactivation du RO et du LFSR.

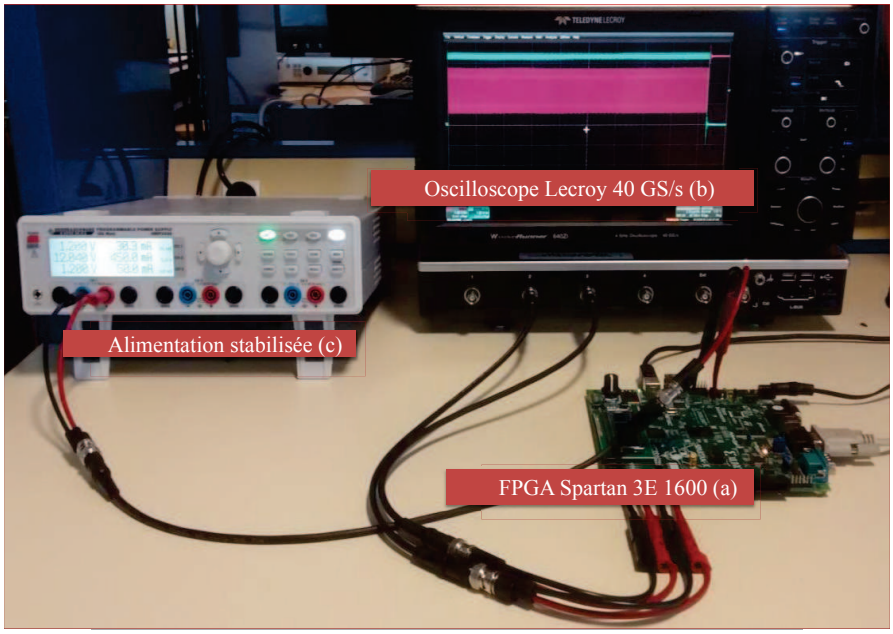

Figure 2.6: Banc de mesure. (a) FPGA, (b) oscilloscope, (c) alimentation stabilisée.

2.2 Protocole de mesure

La mesure des fréquences des ROs se fait de façon externe en utilisant un oscilloscope afin d'obtenir une grande précision. Une alimentation externe est utilisée pour alimenter le cœur du FPGA afin de garantir une bonne stabilité de la tension durant les mesures. Le tout est piloté par un ordinateur. La figure 2.6 est une photo du banc de mesure. Seul l'ordinateur de pilotage n'est pas dans le cadre de la photo. Un script Matlab a été utilisé pour communiquer avec le FPGA par communication série, contrôler l'alimentation et récupérer les courbes fournit par l'oscilloscope. Un second script extrait les fréquences des ROs à partir des courbes d'oscillation acquises. Dans cette section, le matériel et le protocole utilisés pour extraire les oscillations puis les fréquences sont détaillés. Ensuite, la précision de l'ensemble du banc de mesure est expérimentalement quantifiée.

2.2.1 Matériel

Les mesures des fréquences d'oscillation des ROs sont faites avec un oscilloscope à mémoire numérique (DSO) de chez Lecroy possédant un taux d'échantillonnage de $40\,\mathrm{GS/s}$ ($25\,\mathrm{ps}$ entre 2 échantillons). Afin d'obtenir une précision de $\pm 0{,}25\,\mathrm{ps}$ sur la mesure de $\mathrm{F_{RO}}/2$, plus de 100 oscillations de $\mathrm{T_i}/2$ sont acquises et le processus est répété 1000 fois de façon à obtenir, par moyenne, une estimation précise de $\mathrm{T_i}$ mais aussi de l'écart type σ_{Ti}. Les ROs sont des structures dont la fréquence est très sensible à la tension d'alimentation. Il est donc nécessaire d'avoir une tension stable pour limiter l'influence de l'environnement; or le régulateur embarqué sur les cartes FPGAs utilisées

ne permet pas cette stabilité. Pour garantir une tension constante, une alimentation stabilisée avec une précision de $0{,}05\,\%$ est donc utilisée pour alimenter directement le cœur du FPGA sans passer au travers du régulateur de tension du circuit imprimé.

2.2.2 Déroulement de la mesure

Pour chaque RO, 8 types de mesures sont effectués. Une mesure sans LFSR (layout sain), et 7 mesures avec le LFSR (design infecté) une par mode du LFSR. Ce sont ces mesures qui sont répétées 1000 fois. Pour limiter l'impact lié à l'auto échauffement suite à une utilisation continue du RO, on alterne entre les types de mesure. C.-à-d. que l'on répète 1000 fois une rotation entre les types de mesure avant de passé au RO suivant. Cependant, passer d'une mesure sans LFSR à une mesure avec AES requiert de reprogrammer le FPGA, afin d'éviter cela nous avons fait un compromis et utilisé le protocole suivant. Premièrement, on programme le FPGA avec le layout sain. Pour chaque RO, nous faisons 1000 mesures sans LFSR. Puis on reprogramme le FPGA avec le layout infecté et l'on fait 1000 rotations de 7 mesures (une par mode du LFSR). On procède ainsi pour tous les ROs.

2.2.3 Extraction de la période

Pour chaque mesure, on récupère une courbe correspondant à plus de 100 oscillations de RO. Pour chaque configuration de mesure, on récupère 1000 courbes. Pour extraire la période moyenne à partir de ce jeu de courbes, le processus est le suivant. Premièrement, chaque trace est recentrée sur zéro, puis on repère toutes les périodes de chaque trace grâce au front montant du signal oscillant. On considère qu'il y a un front montant lorsqu'une valeur négative est suivie d'une valeur positive. On récupère ainsi pour chaque trace les 100 premières périodes du signal. On obtient donc une matrice de période 1000 par 100 avec une précision de $25\,\mathrm{ps}$. Pour obtenir $\mathrm{T_i}$ on fait la moyenne de toutes les valeurs de la matrice. σ_{Ti} est l'écart type des valeurs de la matrice.

2.2.4 Précision de la mesure

Dans un premier temps, nous avons quantifié la précision de nos mesures (incluant toutes les sources de bruit de mesure: circuit, laboratoire, alimentation, etc) et l'impact des variations intra-die et inter-die. À cette fin, nous avons mesuré la valeur moyenne $\mathrm{T_i}$ et l'écart type σ_{Ti} des différentes périodes de RO et ceci a été répété pour plusieurs cartes. La figure. 2.7 illustre la précision et la qualité de nos mesures. Elle donne les 60 écarts types obtenus sur deux cartes. Nous pouvons observer que les valeurs de σ_{Ti} sont comprises entre $12{,}2\,\mathrm{et}\,13{,}4\,\mathrm{ps}$.

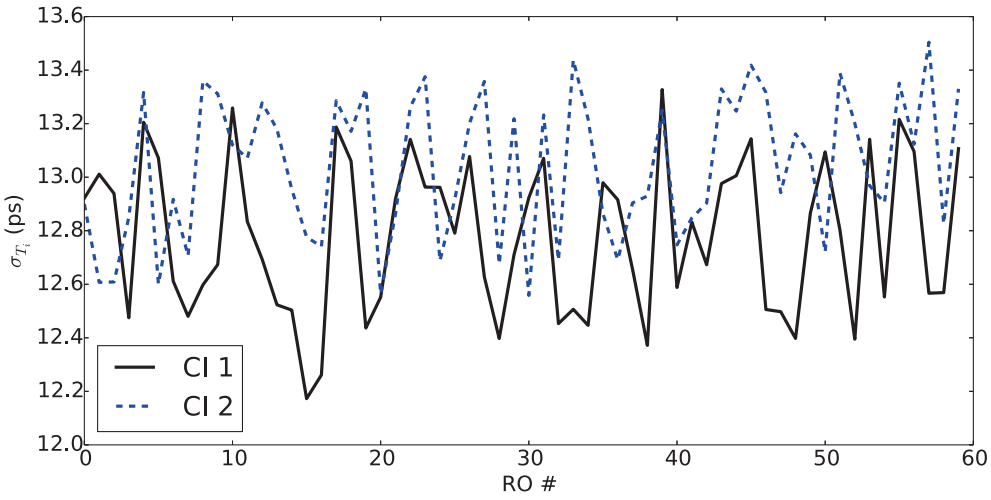

Figure 2.7: Écart type $\sigma_{\mathrm{T_i}}$ de 60 ROs pour les deux cartes.

2.3 Impact dynamique d'un CTM

Dans cette section, nous cherchons à mesurer l'impact dynamique d'un CTM sur le réseau de capteurs. Pour cela, deux paramètres principaux sont analysés. Le premier paramètre est l'amplitude de l'activité du LFSR, c.-à-d. le nombre de bascules ayant une activité de commutation. Le second paramètre est la distribution spatiale de l'impact des commutations électriques sur le réseau de capteur.

2.3.1 Influence de la taille

La figure 2.8 montre la différence, $\Delta\mathrm{T_i}$, entre les périodes des 60 ROs en fonction de l'amplitude de l'activité de commutation parasite du LFSR. La période augmente linéairement avec le nombre de commutations de bascules D (bits) dans le LFSR. $\Delta\mathrm{T_i}$ est donc calculé par $\Delta T_i = T_i^{n\ bits} - T_i^{0\ bit}$, soit la différence de période entre le cas où n bits du LFSR commutent et le cas du LFSR au repos. Cependant, on peut observer que parmi les 60 courbes, celle en rouge montre un comportement spécifique et inexpliqué. Elle correspond à $\Delta\mathrm{T_{27}}$ observé pour le RO27 qui est localisé dans le LFSR. L'augmentation observée est comprise entre $0\,\mathrm{ps}$ et $7\,\mathrm{ps}$ pour la plupart des ROs. Cela correspond à seulement $0{,}048\,\%$ de $\mathrm{T_i}$. Cette valeur est très faible si on la compare à l'effet des variations des procédés de fabrication qui peuvent atteindre $30\,\%$ selon [Now+14].

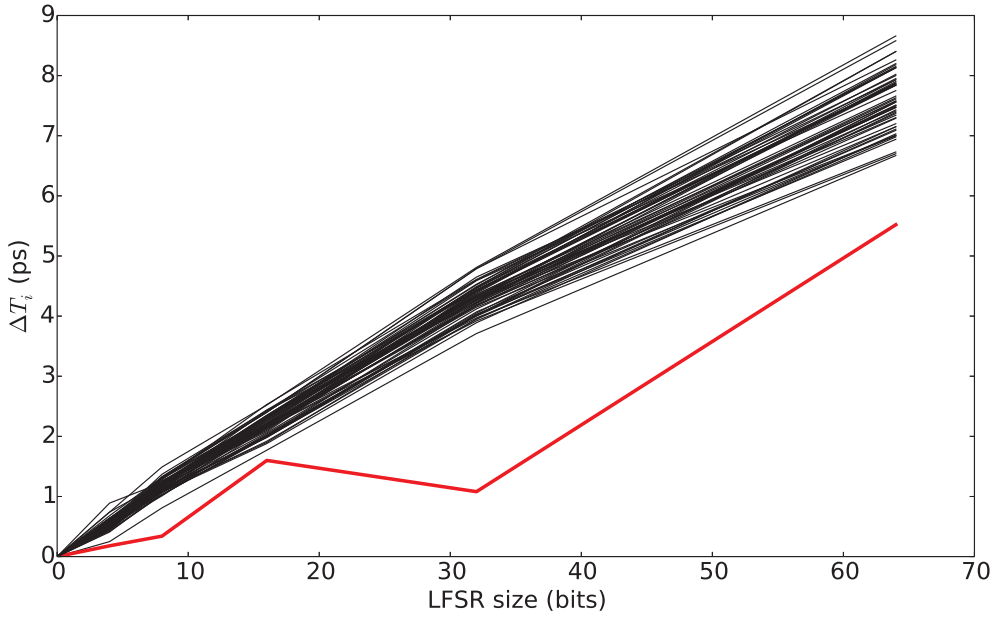

Figure 2.8: Évolution de ΔT_i en fonction de l'amplitude de l'activité parasite de commutation.

2.3.2 Distribution spatiale

La distribution spatiale de l'impact dynamique observé a aussi été caractérisée. À cette fin, le LFSR, configuré pour fonctionner avec des mots de 64 bits, a été placé autour de RO27. Puis tous les ΔT_i ont été mesurés. Cela a été fait pour plusieurs cartes.

La figure. 2.9 montre deux cartes de ΔT_i obtenue à partir de deux circuits Spartan3E-1600 différents. Chaque point de ces cartographies correspond à un RO. La distance entre deux ROs voisins est de 24 slices, slice dont la taille a été estimée à $\sim 120\mu m$ en prenant en compte le nombre de slices (selon X et Y) embarqués sur ces circuits et la dimension des circuits (mesurée aux rayons X sans décapsuler les ICs). Le RO 27 localisé aux coordonnées $(X,\ Y) = (3,\ 4)$.

La figure. 2.9 représente l'impact de l'activité du LFSR (64 bits) sur la période des ROs. Elle montre que l'activité de commutation parasite induit une augmentation de la période de RO27 de $5\,ps$, la valeur minimum obtenue sur toute la surface du CI pour les deux cartographies. Ceci est un résultat surprenant et inexpliqué. En effet, le RO27 est entouré par le LFSR, et l'impact dynamique maximal n'est pas atteint à cet emplacement. L'impact dynamique maximal est atteint pour un ensemble de ROs situés autour du LFSR et est égale à $8\,ps$. L'impact dynamique minimal est égal à $6\,ps$ et est atteint pour les ROs les plus éloignés.

Cette tendance est confirmée par la figure. 2.10 qui donne une projection de deux cartographies selon l'axe Y (voir Fig. 2.18 pour l'orientation). Ces résultats suggèrent

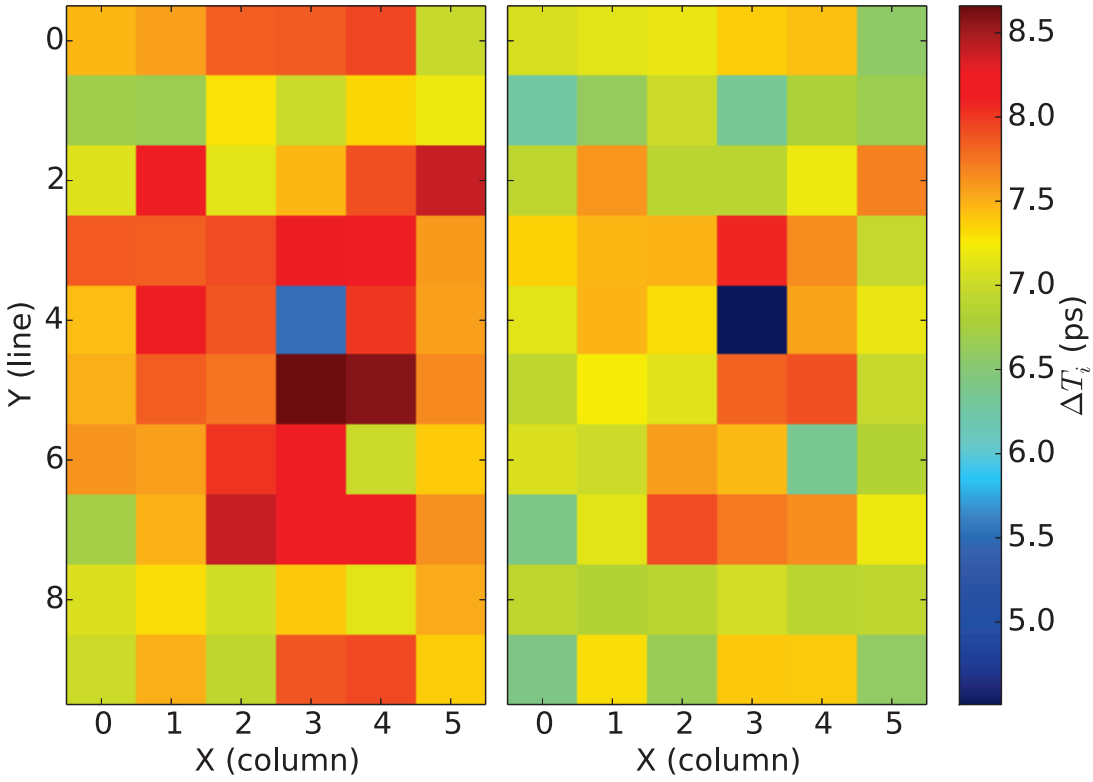

Figure 2.9: Cartographies des ΔT_i obtenues sur 2 cartes. Le LFSR de 64 bits est à la coordonnée (X, Y) = (3, 4).

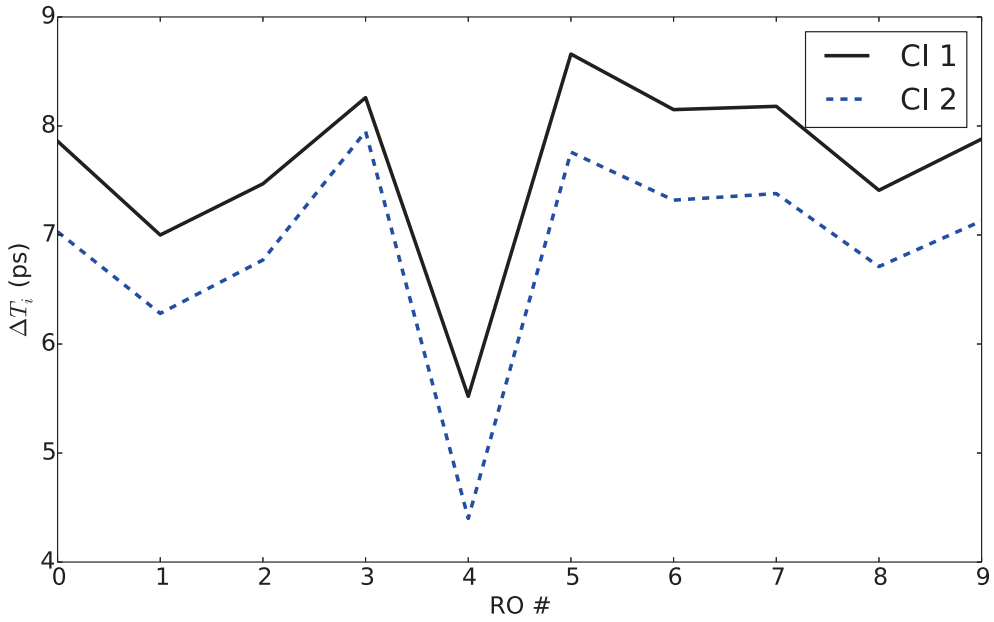

Figure 2.10: Valeurs de ΔT_i (pour deux cartes) des ROs selon la ligne verticale passant par la coordonnée $(3, 4)$ de la figure 2.9

qu'une activité de commutation parasite induit un effet global ($4\,\mathrm{ps}$) sur l'ensemble du réseau d'alimentation du CI, mais aussi un effet local légèrement plus prononcé ($8\,\mathrm{ps}$).

Néanmoins, ces résultats suggèrent également que l'effet induit par la commutation de 32 bascules sur la fréquence est extrêmement faible pour les ROs les plus éloignés de la source d'activité ($6\,\mathrm{ps}$ à $7\,\mathrm{ps}$) comme pour les plus proches ($8\,\mathrm{ps}$). Cela est rassurant puisque ces résultats reflètent le fait que les réseaux d'alimentation et de masse sont conçus pour être le moins résistifs possible dans le but d'éviter des chutes de tension importantes pouvant compromettre les contraintes temporelles. Par conséquent, cela montre que la perturbation de tension est globale avec une très faible amplitude.

Un autre point intéressant mis en valeur par ces résultats est que l'influence des activités de commutation parasite sur les périodes des ROs ($\Delta \mathrm{T_i}$) est relativement constante à une distance donnée de la source d'activité. Cela signifie que cet effet semble être *relativement* indépendant des variations intra-die, et ceci est confirmé sur la figure 2.10.

2.4 Impact statique d'un CTM

Précédemment, nous avons étudié l'impact de l'activité de commutation d'un CTM, c.-à-d. que nous avons observé les effets sur les ROs d'un LFSR, et ce en considérant la taille des mots manipulés. Pour caractériser l'impact statique de l'insertion de CTM, nous avons mesuré l'impact de son implémentation, c.-à-d. le fait d'implémenter ou non le LFSR sur le FPGA, le LFSR restant au repos (horloge coupée une fois insérée). Nous avons ensuite mesuré l'effet de l' activation de l'horloge du LFSR.

2.4.1 Impact de l'horloge

La figure 2.11 montre l'impact de l'activation ou non de l'horloge du LFSR sur les ROs pour les mêmes circuits que précédemment. Sur cette figure, $\Delta T_i = T_i^{avec\ horloge} - T_i^{sans\ horloge}$. Comme le montre la figure 2.11, l'activation de l'horloge du LFSR de 64 bits induit une augmentation globale de $\mathrm{T_i}$ égale à $5\,\mathrm{ps}$ et une augmentation locale (seul l'emplacement infecté est affecté) de $9\,\mathrm{ps}$. Cela est en accord avec les résultats reportés dans la section précédente. En effet, l'impact de l'horloge est du même ordre de grandeur que l'impact dynamique du LFSR entièrement activé (64 bits de commutation). Cependant la localisation de l'impact est différente. Dans le cas de l'activité de commutation électrique, l'impact maximal est observé pour les ROs entourant le LFSR. Contrairement à l'impact de l'horloge qui est maximal seulement pour le RO situé dans le LFSR.

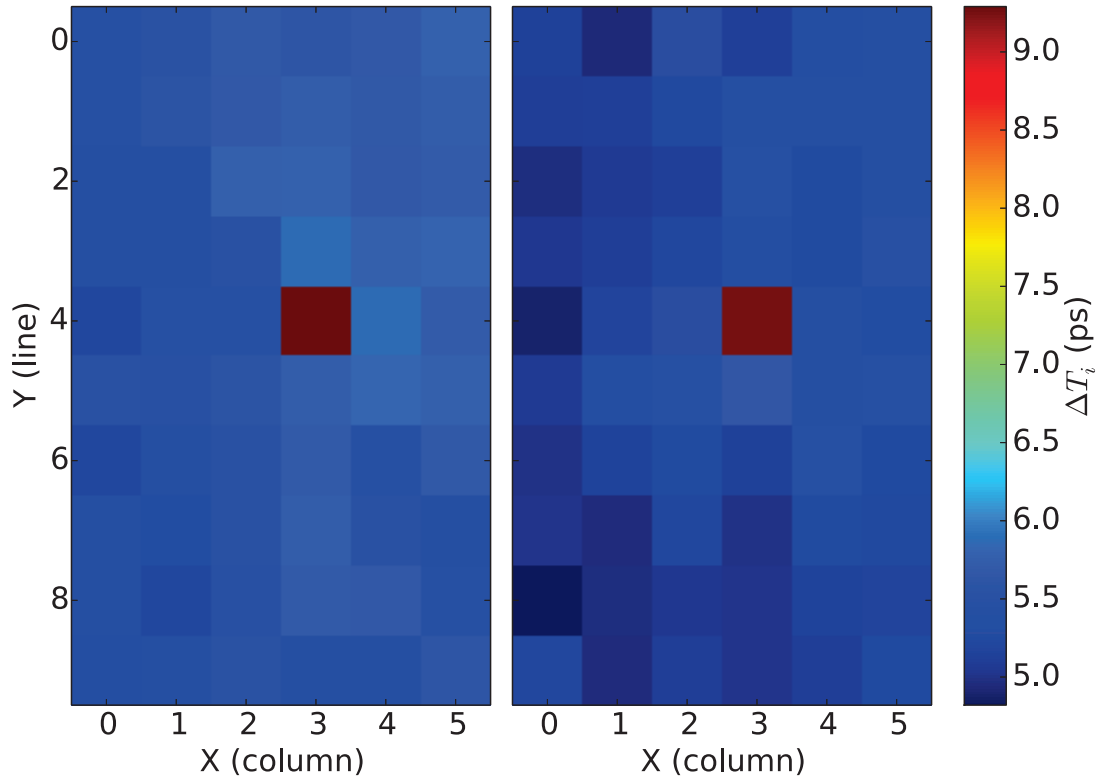

Figure 2.11: Impact dynamique de l'activation de l'horloge du LFSR pour deux cartes.

2.4.2 Impact de l'implémentation

La figure 2.12 montre l'impact de l'implémentation du LFSR sur les ROs pour deux circuits différents,. Pendant les mesures le LFSR est au repos. Comme on peut l'observer, le LFSR a un impact statique global de $20\,\mathrm{ps}$ et un impact statique significatif de $150\,\mathrm{ps}$ à l'emplacement de l'infection. L'implémentation du CTM est donc est donc la principale source de modification de la distribution d'alimentation à l'intérieur du CI. Toutefois, cet impact significatif reste très localisé. En effet, un RO est fortement impacté par l'implémentation.

C'est un résultat important. En effet, cela suggère que l'implémentation d'un CTM induit des modifications significatives des caractéristiques locales du réseau d'alimentation/masse (R, C, et courant statique local). Ces modifications peuvent créer une modification de la tension d'alimentation locale et donc des performances temporelles de la logique environnante. Cette observation est la base de la méthodologie de détection introduite dans le chapitre 3. De plus, elle suggère que les courants transitoires de commutation ont un impact moins important à une fréquence d'horloge de $50\,\mathrm{MHz}$. Cette dernière observation pourrait être différente pour des fréquences d'horloges plus élevées.

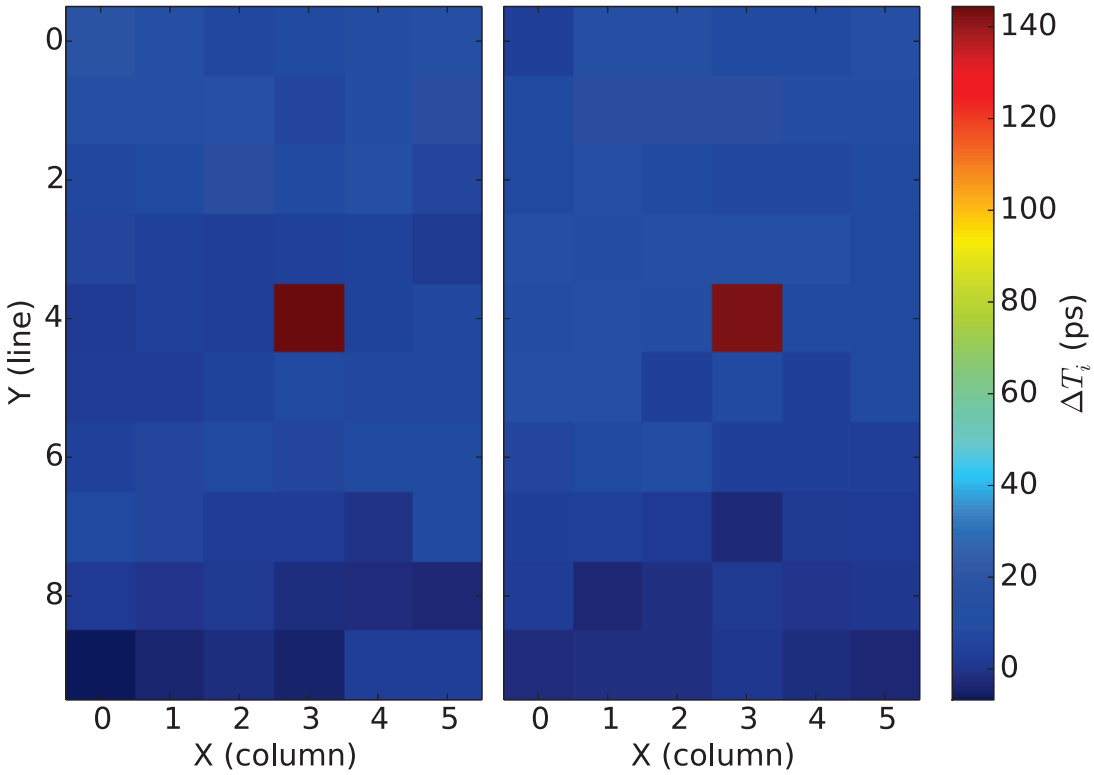

Figure 2.12: Impact statique de l'insertion d'un CTM pour deux cartes.

2.5 Variations des procédés de fabrication

Après avoir étudié l'impact statique de l'implémentation d'un CTM et l'impact dynamique de son activité de commutation parasite, l'influence des variations des procédés de fabrication sur les périodes des ROs a été étudiée.

Nous avons vu dans le chapitre précédent que le modèle classique des variations des procédés les classes en deux catégories: les variations inter-die et les variations intra-die. Les variations inter-die sont les différences en termes de qualité de procédés de fabrication entre deux CIs et donc également les différences de performances. Les variations intra-die dénotent des différences physiques et électriques entre les structures élémentaires (interconnexions, transistors, ...) d'un même circuit.

2.5.1 Variations inter-die et intra-die

Nous avons estimé l'impact de variations inter-die (inter-carte) et intra-die de notre lot de 24 cartes FPGA. En guise d'illustration, la figure 2.13 donne la période de nos 60 ROs pour deux cartes FPGA différentes. Ainsi, cela montre simultanément l'impact des variations inter-die et intra-die. On peut observer qu'il y a des variations inter-die significatives. Elles sont responsables de changements dans les périodes des ROs de plus

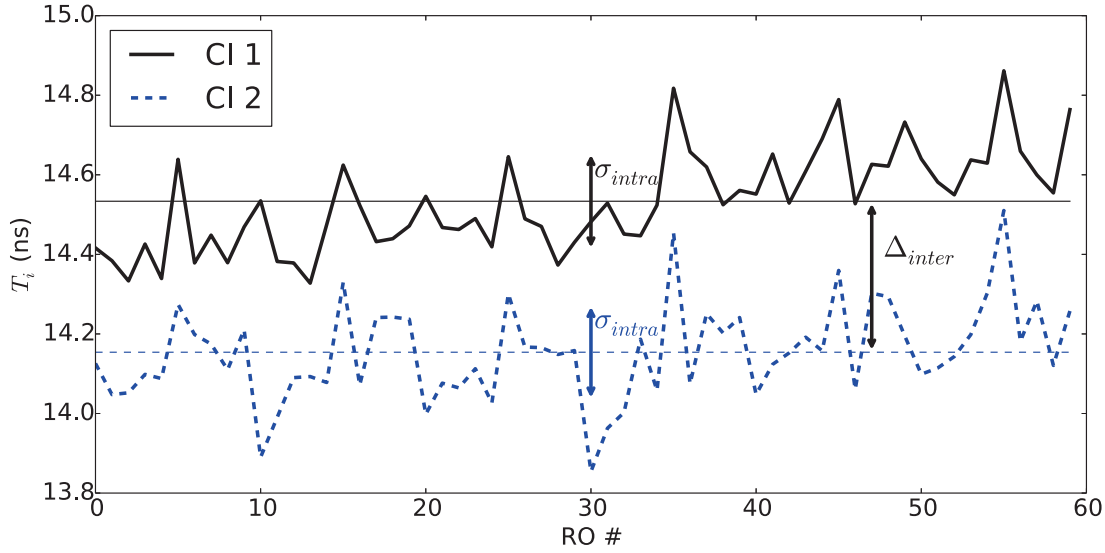

Figure 2.13: Les 60 valeurs de période mesurées sur deux cartes.

de $500\,\mathrm{ps}$ entre les deux CIs. Cette figure montre également que les variations intra-die sont responsables de changement de plus de $200\,\mathrm{ps}$ à l'intérieur de chaque circuit.

2.5.2 Analyse spatiale

La figure 2.14 donne les périodes de chaque RO pour deux cartes. Un lien entre la position d'un RO et sa période d'oscillation apparait clairement. Ce lien peut être expliqué par l'impact de la grille d'alimentation, c.-à-d. sur comment le CI a été conçu. Cependant, les variations des procédés ont un impact plus large. En effet, les courbes des deux distributions période, la figure 2.13, montre qu'elles sont quasiment disjointes.

2.6 Chutes de tension (IR drops)

Dans les paragraphes précédents, les résultats de caractérisation ont indiqué l'importance de la distribution de l'alimentation. Premièrement, ces résultats expérimentaux suggèrent que l'impact principal de l'insertion d'un CTM est l'altération statique et locale de la tension d'alimentation. D'autre part, pendant la caractérisation de l'impact des variations des procédés de fabrication, il a été montré que le placement des ROs influence leur fréquence d'oscillation. Cette variation est due à la distribution d'alimentation. Ces observations nous encouragent à analyser finement l'impact de l'insertion des CTMs sur la distribution de tension interne. Nous avons donc utilisé les résultats précédents pour évaluer les chutes de tension associées aux différents types d'impact du LFSR.

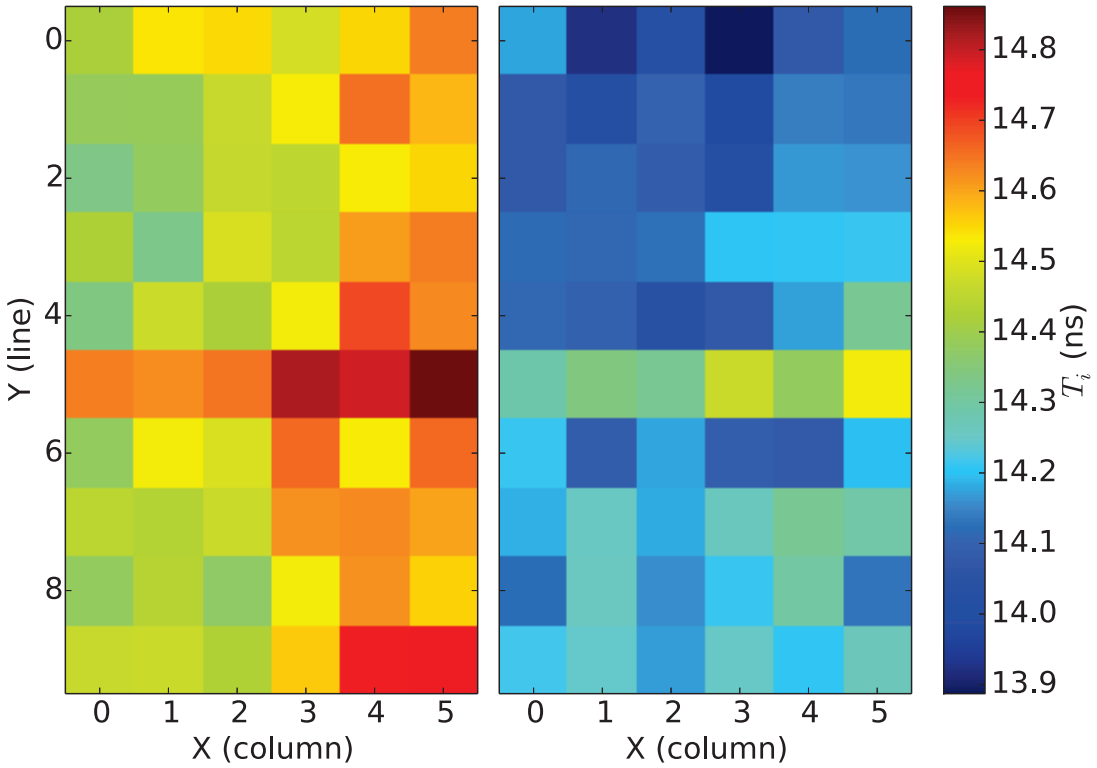

Figure 2.14: Cartographies des périodes des ROs.

2.6.1 Quantification des chutes de tension

Dans le but de cartographier la tension d'alimentation interne, en présence ou non d'un CTM, nous avons caractérisé la sensibilité de la fréquence d'oscillation des 60 ROs à Vdd (la tension d'alimentation). Cela a été fait en utilisant l'alimentation stabilisée sur une gamme de tension de $[1{,}19\,\mathrm{V},\ 1{,}21\,\mathrm{V}]$, en considérant que la tension nominale d'alimentation est $1{,}2\,\mathrm{V}$.

La figure 2.15 montre l'évolution des périodes d'oscillation $\mathrm{T_i}$ en fonction de Vdd. Comme attendu, la période décroit linéairement avec Vdd sur cette courte plage de tension. À partir de ces courbes, nous avons appliqué une régression linéaire sur les résultats de chaque RO afin d'obtenir les fonctions $\mathrm{T_i} = \mathrm{f(V)}$ pour chaque RO. Il en résulte que la pente $\frac{\Delta F}{\Delta V dd}$ est presque identique pour tous les ROs et égale à $(-12\mathrm{ns/V})$. Compte tenu de ce résultat, nous évaluons dans les sections suivantes, en utilisant les observations des sections précédentes sur les changements de fréquence, les chutes de tension locales.

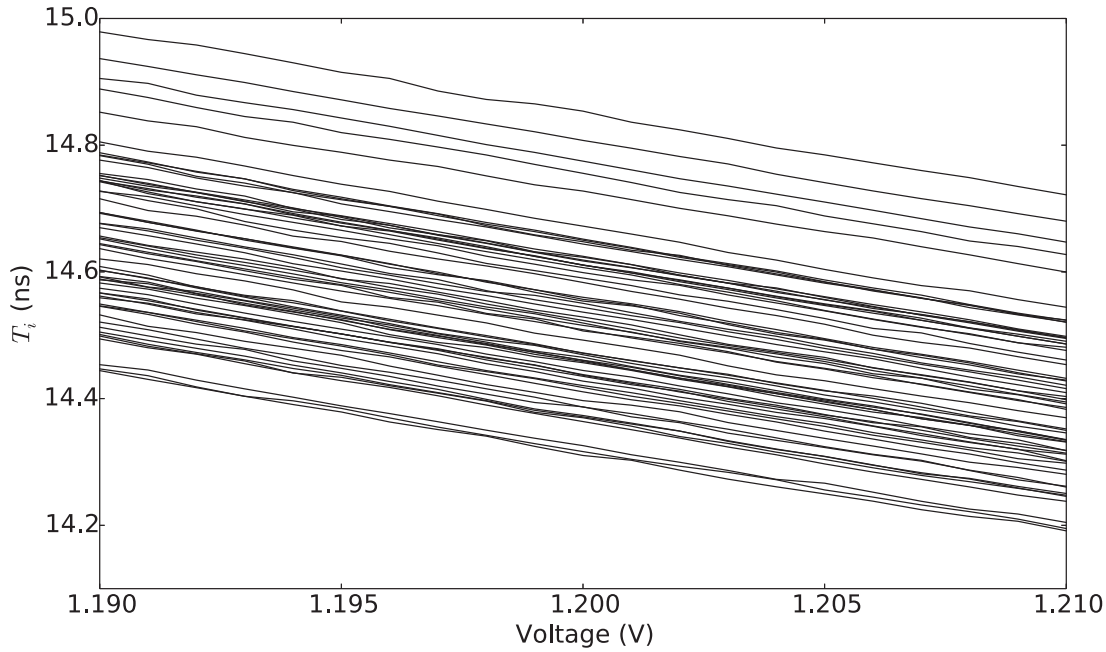

Figure 2.15: Évolution des T_i en fonction de la tension d'alimentation Vdd

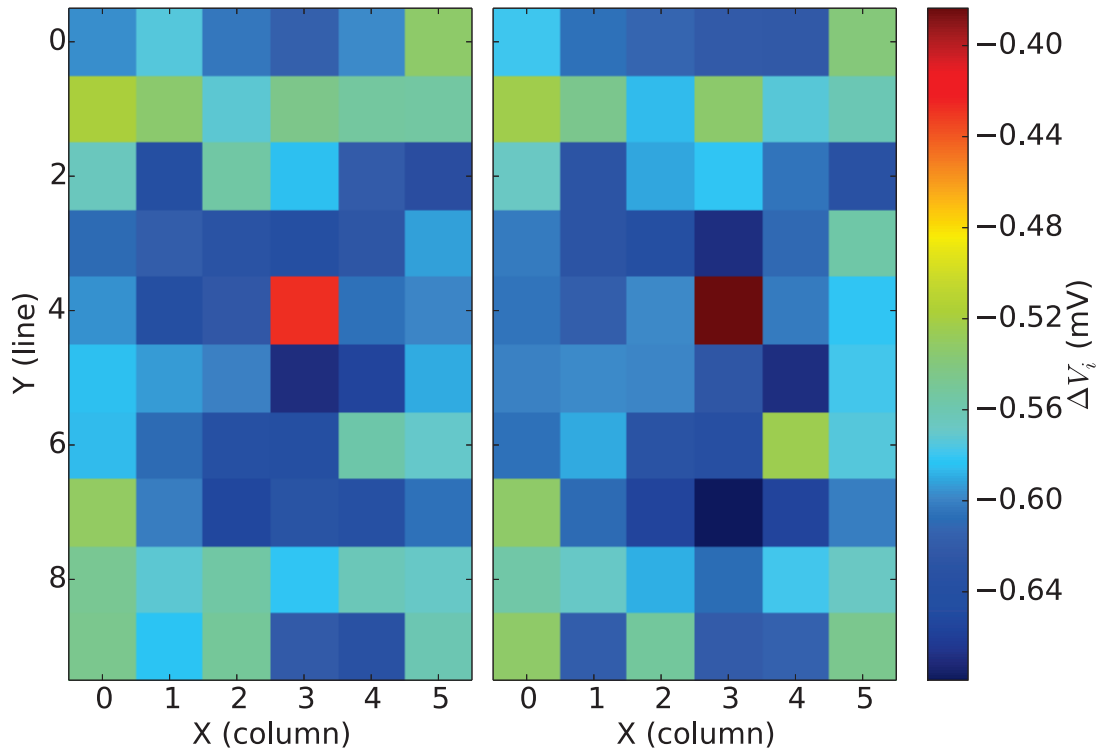

Figure 2.16: Cartographies des chutes de tension induites par une activation des 64 bits du LFSR.

2.6.2 Impact dynamique

La figure 2.16 donne, pour deux cartes FPGA, les cartographies des chutes de tension induites par l'activation du LFSR préalablement implémenté (configuré pour travailler sur des mots de 32 bits). La cartographie montre que la chute de tension maximale (le maximum de la différence de tension absolue) induite par les commutations du LFSR apparait aux emplacements des ROs entourant le LFSR. À ces emplacements, l'amplitude de la chute de tension atteint $0{,}66\,\mathrm{mV}$. Cette chute de tension décroit progressivement lorsque l'on s'éloigne pour atteindre $0{,}54\,\mathrm{mV}$ pour les positions les plus éloignées de l'infection. Comme attendu, ces valeurs de chute de tension sont très inférieures aux valeurs considérées pendant les étapes de conception qui suivent une approche basée sur les "*corners*"; les "*corners*" Vdd sont habituellement fixés à $\pm 10\,\%$ de la tension nominale (ici $1{,}2\,\mathrm{V}$).

2.6.3 Impact statique

La figure 2.17 représente, dans le cas de deux cartes FPGA différentes, les chutes de tension créées par l'implémentation du LFSR elle-même. Une chute de tension globale de $1\,\mathrm{mV}$ apparait ainsi qu'une chute de tension locale de $12\,\mathrm{mV}$. Ceci confirme que l'implémentation a un impact significatif sur la tension d'alimentation interne du CI. L'amplitude de cet impact est 20 fois supérieure à l'impact provoqué par l'activation du LFSR, c.-à-d. 20 fois plus important que l'impact des commutations du LFSR. Notons que nous obtenons des résultats similaires pour les deux cartes. Cela montre que le phénomène est relativement indépendant des variations de procédé. Cela peut paraitre surprenant. Toutefois en songera au fait que les ROs intègrent dans leur fréquence la valeur moyenne de la tension sur plusieurs cycles, et que les appels en courant du LFSR sont intenses, mais brefs par rapport à la durée d'une période. Enfin, on songera que l'insertion de logique induit un courant statique.

2.7 Influence du design

Les résultats précédents ont montré que l'insertion d'un CTM modifie la distribution de la tension interne dans le circuit. Dans le même temps, la figure 2.13 indique que le floorplan du CI, et donc la façon dont est distribué l'alimentation a un impact sur les performances des ROs, et de manière plus générale sur les performances des composants logiques CMOS. Dans le but d'améliorer notre compréhension et de quantifier l'influence du design sur la distribution de tension et donc des performances de composant CMOS, nous avons implémenté de deux manières différentes un même design dans un FPGA.

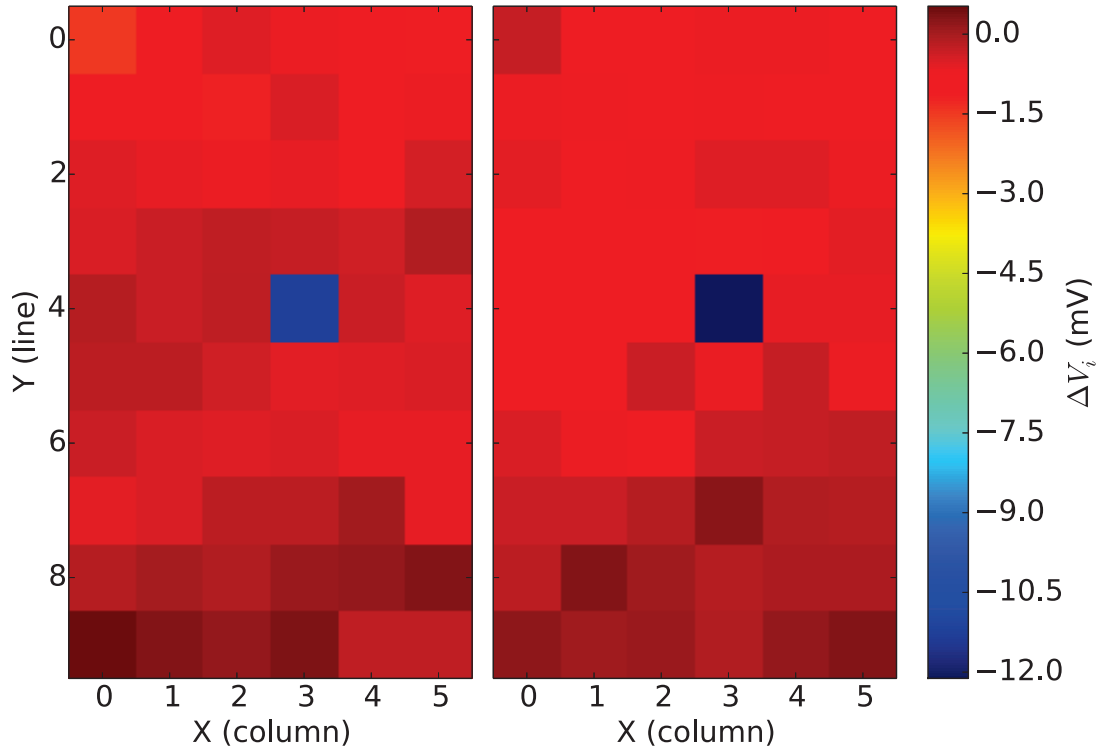

Figure 2.17: Cartographies des chutes de tension induites par le fait d'implémenter le LFSR

2.7.1 Implémentations

Pour cette expérimentation, deux ensembles de contraintes de placement-routage ont été adoptés pour intégrer un "*Advanced Encryption Standard*" (AES) comme illustré sur la figure 2.18. L'AES utilisé est une implémentation du standard de chiffrement du NIST spécifié dans [Aes]. Le design embarque aussi un LFSR afin d'émuler un CTM.

Pour effectuer les mesures nécessaires, trois fichiers de placements et routages sont nécessaires. Deux fichiers implémentant le même AES avec deux placements et routages différents et un fichier contenant la structure de mesure sans AES. Comme pour les expérimentations précédentes, on veut le minimum de différences entre les trois layouts. Pour générer ces fichiers, nous utilisons une procédure similaire à celle décrite dans la section 2.1.2. Dans un premier temps, le design embarquant la structure de mesure, le LFSR et l'AES est généré à partir d'une description HDL et d'un ficher de contrainte ".ucf". Le résultat correspond à un des deux layouts avec AES. Deuxièmement, sous FPGA editor, on supprime le placement et le routage de l'AES, on modifie ses contraintes de placement et routage puis on relance l'algorithme de placement-routage. De cette manière, seule la projection technologique de l'AES est modifiée et la structure de mesure reste inchangée. Nous obtenons ainsi le second layout avec AES. Enfin, nous supprimons tout élément relatif à l'AES afin d'obtenir le layout sans AES.

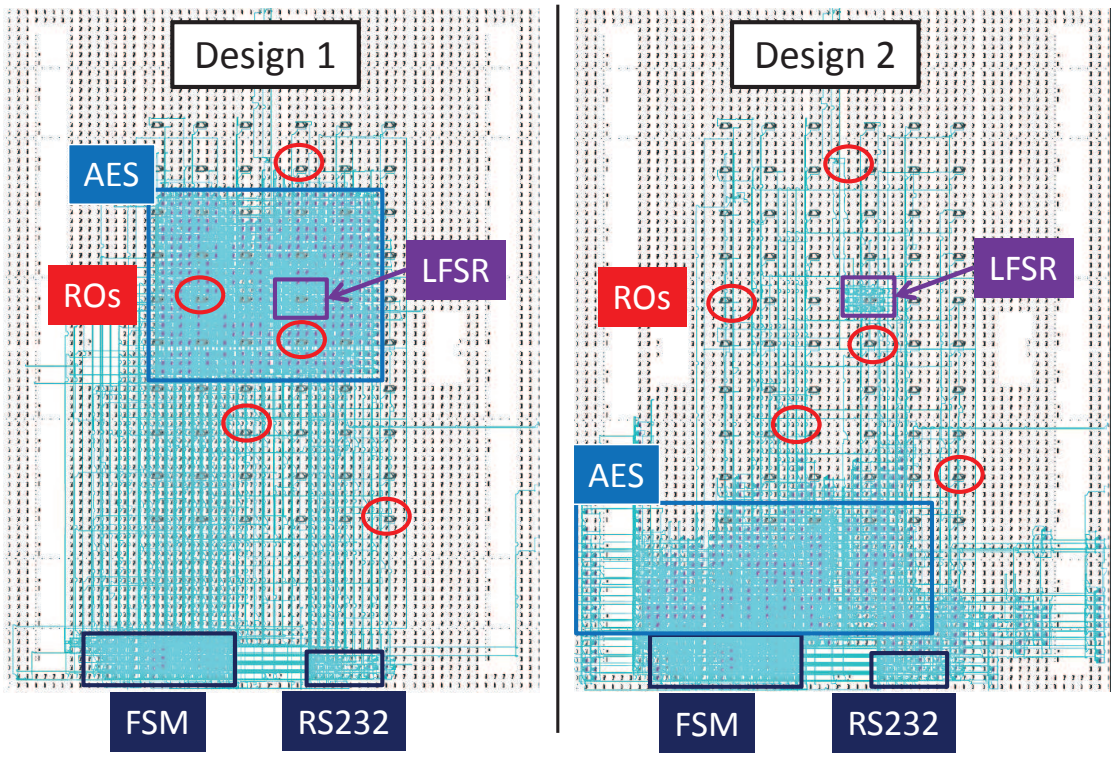

Figure 2.18: Les deux implémentations (floorplan) considérées du même placement-routage.

2.7.2 Résultats

Les figures 2.19 et 2.20 montrent les chutes de tension induites par les deux implémentations de cette même logique. C.-à-d. la différence entre les tensions obtenues dans le cas avec et dans le cas sans AES. Les deux cartographies permettent de retrouver les placements spécifiés, car les ROs situés dans les zones implémentées sont grandement affectés par la présence de l'AES. L'impact atteint localement $80\,\mathrm{mV}$. De plus, on observe un impact global de $30\,\mathrm{mV}$. Malgré ces observations, le résultat principal est que la distribution de tension dépend significativement de la façon dont le design a été physiquement implémenté.

À ce stade, nous pouvons affirmer avec un haut niveau de confiance que l'insertion d'un CTM modifie la distribution de l'alimentation d'un circuit au repos. Bien sûr, l'amplitude de la modification induite dépend de la taille du CTM et peut être très petite. Cette hypothèse constitue la base de la méthode de détection décrite dans le chapitre 3.

2.8 Mesure de l'impact dynamique

Nous avons étudié dans les sections précédentes l'impact de l'activité de commutation parasite sur les périodes d'oscillation des ROs et sur les tensions d'alimentations internes.

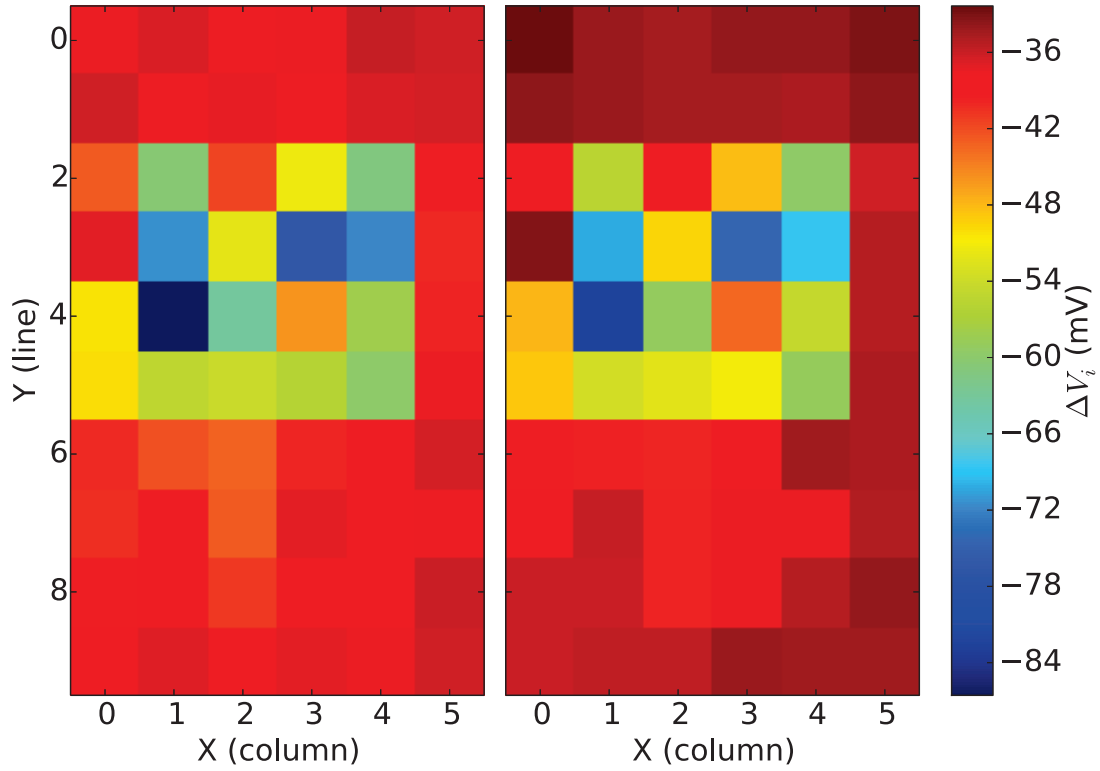

Figure 2.19: Chutes de tension obtenues, sur deux cartes, pour la première implémentation (Design 1) de l'AES

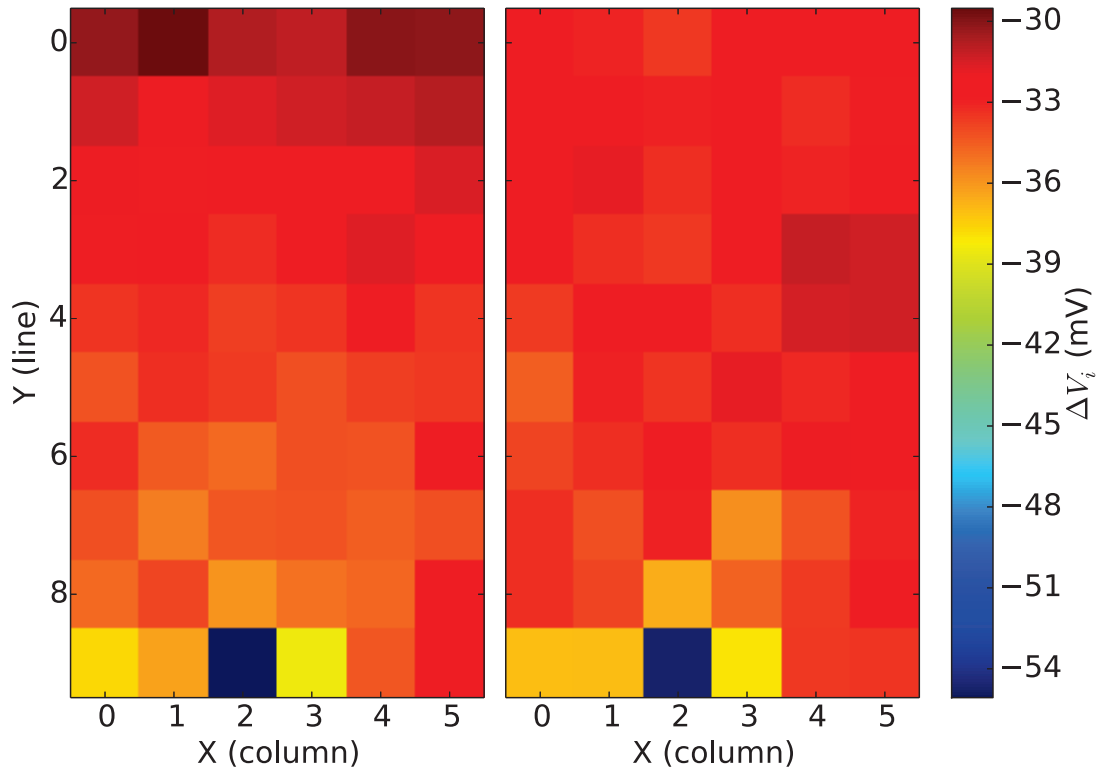

Figure 2.20: Chutes de tension obtenues, sur deux cartes, pour la seconde implémentation (Design 2) de l'AES

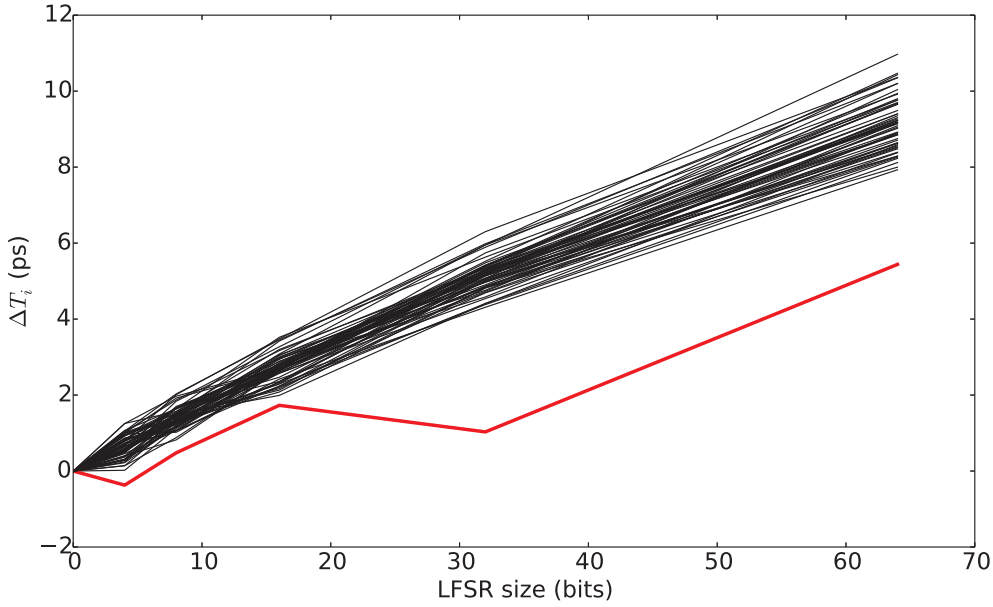

Figure 2.21: Impact dynamique du LFSR en présence de l'AES.

Les mesures précédentes ont montré que l'impact dynamique peut être mesuré en utilisant un oscilloscope avec un taux d'échantillonnage élevé. Cependant, cet impact demeure faible comparé aux variations des procédés de fabrication (intra-die et inter-die) et par rapport aux variations induites par l'implémentation d'un AES. Ces mesures d'impact dynamique ont été menées sans l'activité de calcul qui correspond au fonctionnement normal d'un circuit. Par conséquent, ces mesures ne reflètent pas la réalité du scénario ou le déclencheur est connecté aux nœuds internes du circuit ciblé. Des mesures d'impact dynamique ont donc été effectuées avec une activité de calcul en parallèle (ici des chiffrements AES).

La figure 2.21 montre les différences, $\Delta \mathrm{T_i}$, entre les périodes des 60 ROs en fonction de l'amplitude de l'activité de commutation parasite du LFSR. On observe les mêmes comportements que ceux observés sans AES et reportés figure 2.8. La mesure est toutefois moins précise et plus dispersée. On observe un impact dépassant les $9\,\mathrm{ps}$ pour la plupart des ROs, pour un effet de $7\,\mathrm{ps}$ sans l'AES.

Afin de vérifier la diffusion spatiale de l'impact, la figure 2.22 a été tracée. Elle représente les résultats de cartographies de l'impact dynamique du LFSR avec 64 bits de commutation sur deux cartes différentes. Par comparaison avec la cartographie sans perturbation de l'AES présentée sur la figure 2.16, nous voyons que l'impact global est plus élevé autour de $0{,}65\,\mathrm{mV}$ au lieu de $0{,}54\,\mathrm{mV}$. De plus, l'effet local ($0{,}80\,\mathrm{mV}$) est moins élevé sur la surface du circuit.

Il semble que la présence de l'AES augmente la dispersion des mesures. Malgré des différences d'amplitudes, les comportements restent semblables, cela confirme que les

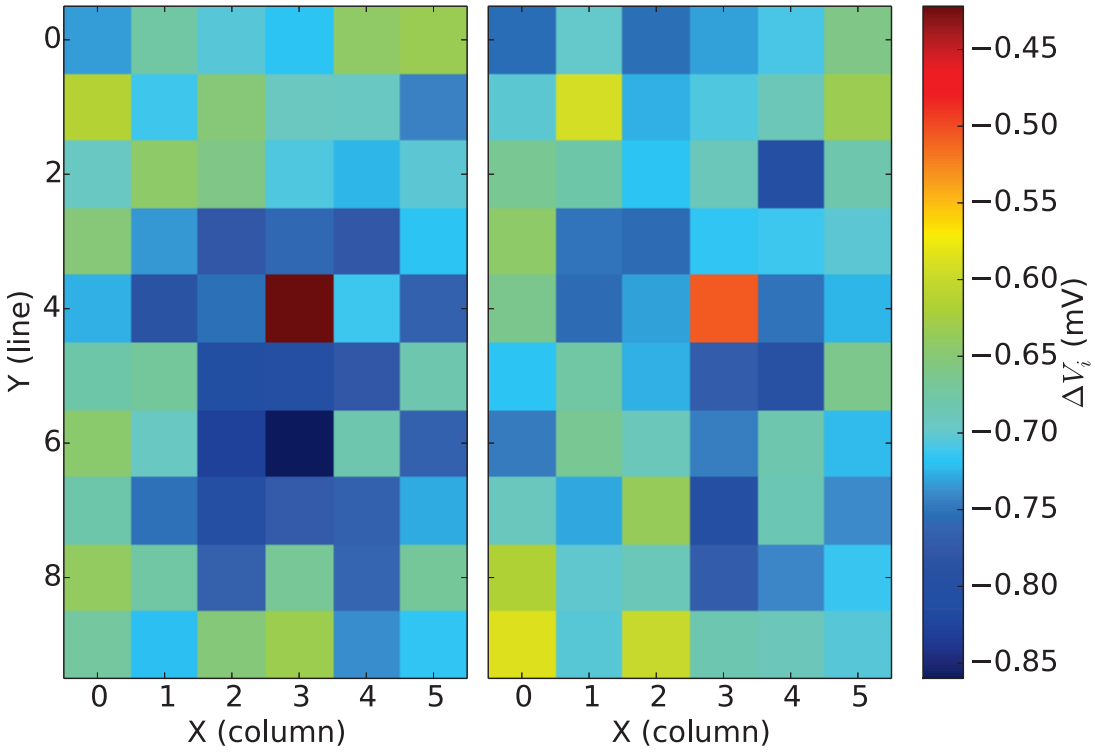

Figure 2.22: Impact dynamique du LFSR avec 64 bits de commutation en présence de l'AES.

résultats précédents restent valides sous l'influence des perturbations d'autres blocs fonctionnels dans le circuit.

2.9 Discussions sur l'utilisation de compteurs pour des mesures embarquées

Les résultats présentés dans les sections précédentes montrent qu'une activité électrique équivalente à la commutation de 8 bits (resp. 32 DFF) induit des accroissements de délai de l'ordre de $3\,\mathrm{ps}$ (resp. $8\,\mathrm{ps}$) et des chutes de tension de l'ordre de $0{,}25\,\mathrm{mV}$ (rep. $0{,}7\,\mathrm{mV}$). La petitesse de ces grandeurs pose la question de la pertinence du choix du couple RO, compteur comme capteur efficace pour la détection embarquée des CTMs comme cela est proposé dans [Fer+12; Zha+13; KV14]. En effet, indépendamment de la période d'oscillation des ROs utilisés, la durée des mesures (de comptage) permettant la capture d'une dérive de leur période de $8\,\mathrm{ps}$ doit être trop élevée pour être applicable dans la pratique, c.-à-d. pour garantir une bonne stabilité des conditions de fonctionnement et être compatibles avec les durées de calculs des blocs fonctionnels constituants les circuits. Cette durée de mesure peut donc devenir grande devant la durée pendant laquelle le détecteur du CTM est actif. Afin d'éclaircir ce point, nous

Tableau 2.1: Nombre de cycles d'horloge nécessaire pour mesurer un écart de 1 avec un compteur.

ΔT_{RO}(ps)	4	8	100	140
n	5000	2500	200	143
temps de mesure (μs)	100	50	4	2,86

avons donc estimé le nombre de cycles d'horloge de période T_{CK} nécessaire à la capture d'une dérive de $8\,\mathrm{ps}$ en utilisant la formule :

$$n = \frac{p \cdot T_{CK}}{\Delta T_{RO}} \tag{2.1}$$

où p est la différence du nombre de fronts captés par le compteur, valeur qui doit être suffisamment grande par rapport aux fluctuations induites par le bruit de mesure.

Le tableau 2.1 donne le nombre de coups d'horloge et le temps nécessaire pour mesurer un écart de 1 sur un compteur relié au RO pour mesurer des différences ΔT_i de $4\,\mathrm{ps}$ à $140\,\mathrm{ps}$. Dans notre cas une période d'horloge de $T_{CK} = 20ns$ est considérée.

Dans le cas où l'on souhaiterait mesurer l'impact de la commutation de 32 bits ($8\,\mathrm{ps}$), il faut une durée de mesure minimale de $50\,\mu\mathrm{s}$ pour observer un écart de 1 sur le résultat du compteur. Si l'on souhaite avoir un résultat significatif, par exemple une différence de 100 oscillations, la durée minimale de mesure est de $5\,\mathrm{ms}$. Pendant ce temps, il est nécessaire de laisser osciller librement les ROs, les conditions de fonctionnement doivent être maintenues parfaitement stables, ainsi que l'activité du CTM. De plus, dans le cas, où l'on souhaite différencier l'impact local ($8\,\mathrm{ps}$) de l'impact global ($4\,\mathrm{ps}$), donc détecter une différence de ($4\,\mathrm{ps}$) le temps nécessaire pour mesurer une différence de résultat de compteur de 1 (resp. 100) le temps de mesure minimal est de $100\,\mu\mathrm{s}$ (resp. $10\,\mathrm{ms}$. Compte tenu des grandeurs obtenues, il semble donc difficile d'utiliser des ROs couplés à des compteurs pour détecter de manière embarquée l'impact dynamique des CTMs.

Cependant dans le cas où l'on souhaiterait détecter l'impact statique du CTM, donc une différence de $140\,\mathrm{ps}$, le temps nécessaire pour mesurer une différence de résultat de compteur de 1 (resp. 100) le temps de mesure minimal est de $2{,}86\,\mu\mathrm{s}$ (resp. $286\,\mu\mathrm{s}$. En considérant le fait que le CTM à un impact statique permanent (il n'est pas nécessaire de garder le CTM activé), il semble donc possible d'utiliser des compteurs pour estimer la fréquence des ROs dans le but de détecter l'impact statique des CTMs.

2.10 Conclusion

Ce chapitre décrit les expérimentations qui ont été conduites de sorte à caractériser les impacts dynamiques et statiques de l'implémentation de CTM sur les tensions internes

d'un circuit et sur un réseau de capteurs de type oscillateurs en anneau. C'est avec cet objectif qu'un circuit de test a été implémenté sur un FPGA. Celui-ci a été pourvu d'une matrice de mesure composée de ROs et d'un LFSR modulaire permettant d'émuler l'impact d'un CTM. Ensuite, un AES a été ajouté pour étudier le comportement de la structure de mesure sous l'influence d'un bruit provoqué par un environnement de calcul.

Les résultats expérimentaux nous permettent de conclure que l'effet dynamique des CTMs sur la tension (moins de $1\,\mathrm{mV}$) et sur les délais est très inférieur aux effets des variations de procédés. On observe cependant un effet statique (de l'ordre de $12\,\mathrm{mV}$) plus de 10 fois supérieur à l'impact dynamique. De plus l'impact statique à un effet plus localisé. En effet, dans le cas de l'impact dynamique, plusieurs ROs sont impactés par un impact local plus prononcé, alors que dans le cas de l'impact statique un seul RO subit un impact plus important que les autres. Bien que cet impact statique soit le plus élevé, il reste inférieur aux variations des procédés de fabrication. Dans le chapitre suivant, nous chercherons donc à utiliser ce fort impact statique afin de proposer une méthode de vérification embarquée de l'intégrité des CIs.

3 Méthodes de détection de cheval de Troie matériel et de contrefaçon

Ce chapitre présente une méthode de détection de CTMs et de contrefaçons. Il présente les types de CTMs et de contrefaçons qui peuvent être détectés par la méthode considérée. La méthode introduite permet de détecter les altérations qui impactent la structure physique des circuits. Cette méthode repose sur un nouveau modèle des variations des performances des structures CMOS. Ce modèle prend en compte les variations des procédés de fabrication et l'impact du placement-routage. Le modèle et la méthode ont été publiés dans [Lec+16b]. En plus de cette méthode, nous introduisons un nouveau distingueur pour déterminer si un circuit est infecté, une première version de celui-ci a été publiée dans [Lec+16a]. Ce dernier s'adapte aux variations des procédés de fabrication et au bruit de mesure, il permet d'atteindre un taux de détection de $100\,\%$ *pour des infections localisés.*

Le Chapitre précédent caractérise les effets d'une infection sur un réseau de capteurs. Cela a montré que l'impact des variations des procédés de fabrication est largement supérieur à celui de l'infection par un CTM. Cependant, nous avons également montré que l'impact statique d'un CTM est significativement plus important que son impact dynamique. À partir de ces résultats, le chapitre présent décrit une méthodologie permettant d'extraire l'impact statique d'une infection en éliminant le bruit induit par les variations des procédés de fabrication. De façon générale, ce chapitre décrit une méthodologie de détection de CTM et de contrefaçons en l'inscrivant dans un scénario donné de production. Le point de vue adopté est celui d'une entité de conception sous-traitant sa production et ayant un besoin de confiance pour les étapes de fabrication et de distribution. Cette méthode est basée sur trois contributions par rapport à l'état de l'art:

- un nouveau paradigme d'infection,
- un nouveau modèle de variation de performances de la logique CMOS au niveau design,
- et un nouveau distingueur pour la prise de décision, il permet de déterminer si un CI est sain, contrefait ou infecté.

3.1 Caractéristiques des circuits infectés ou contrefaits

Nous avons montré dans le chapitre 1 que diverses méthodes de détection de CTM ont été proposées. Parmi elles, une large majorité vise à détecter l'activité parasite de commutation (la consommation de courant additionnelle) générée par le déclencheur du CTM. Cependant, comme il a été montré dans le chapitre précédent, cette activité parasite de commutation n'est pas la seule trace mesurable laissée par les CTMs. Une autre trace est l'altération de la structure interne du CI. En effet, l'insertion d'un CTM modifie les résistances et capacités locales et globales de la grille d'alimentation. Cette modification induit une différence dans la diffusion du courant dans le CI, et donc une distribution statique et dynamique de tension différente (chutes de tension statiques ou dynamiques). Cette section donne les types de CTMs et de contrefaçons qui seront considérés dans ce mémoire.

3.1.1 CTM

La figure 3.1 est une copie de la figure 1.4 sur laquelle nous avons indiqué les types de CTMs considérés par notre méthodologie. Ces CTMs sont des modifications opérées pendant l'étape de fabrication, le niveau d'abstraction est donc celui du transistor ou des éléments physiques du circuit. Nous cherchons à détecter la présence de CTMs indépendamment de leurs effets; tous les effets potentiels sont donc considérés. De la même manière, nous cherchons à détecter une modification indépendamment du type de logique implémentée. Nous considérons les CTMs fonctionnels puisque ceux-ci imposent un ajout ou une modification de la logique. Le type d'activation est également indépendant de l'impact passif du CTM, donc tous les types d'activation potentiels sont considérés. Ensuite, on considère toute modification de la structure. Cependant, il est montré dans [XT+15] que les infections dispersées ont un impact plus important sur les canaux auxiliaires (dans ce cas EM) que les infections localisées. En considérant que le CTM doit être discret, les CTMs distribués ne sont pas pris en compte. De plus, la notion de CTM «petits» ou «grands» étant floue, les limites concernant la taille des CTMs considérés seront abordées dans le chapitre suivant. Enfin, certaines zones du circuit ne sont pas considérées (mémoires, I/Os), car elles ne peuvent pas être couvertes par une structure de mesure.

3.1.2 Contrefaçons

Les contrefaçons considérées par notre méthode de détection sont celles présentant une structure physique différente de celles des circuits originaux. Par conséquent, toutes les catégories de la taxonomie présentée dans la section 1.2.3 ne sont pas prises en compte. Le premier cas considéré est celui du recyclage. Si les circuits recyclés sont revendus

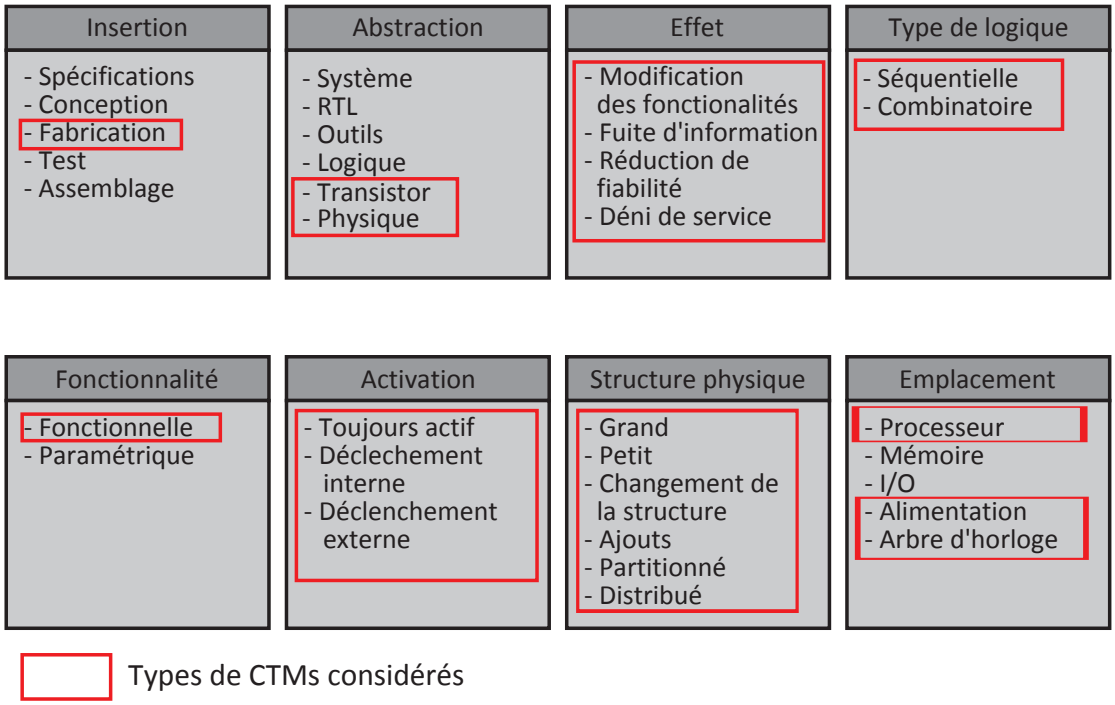

Figure 3.1: Types de CTM considérés par la méthode de détection.

avec les mêmes références que celles utilisées lors de leur première distribution, alors leur structure réelle et leur structure attendue seront les mêmes. Dans le cas contraire, ils pourront être détectés par notre méthode. Dans le cas des circuits ayant subis un remarquage, la structure physique est systématiquement différente de celle attendue, ce cas rentre donc dans notre cadre de détection. De plus, les copies fonctionnelles peuvent ne pas embarquer notre réseau de capteurs.

Ensuite, les surproductions de circuits sont les cas de contrefaçon les plus sensibles. Les circuits sont des copies conformes et possèdent donc la structure physique des composants originaux. Détecter ce type de contrefaçon nécessite une solution d'authentification, ce cas n'est donc pas compris dans notre spectre de détection. De façon analogue, le cas des circuits hors spécifications possède la même structure physique et n'est donc pas pris en compte. On peut cependant noter qu'un test paramétrique doit permettre de détecter de tels circuits.

La détectabilité des clones par notre méthode dépend du niveau d'information et des moyens dont dispose l'attaquant. Dans le cas où l'attaquant dispose de toutes les informations relatives au layout et de moyens de production équivalents à ceux utilisés pour produire les circuits originaux, ces contrefaçons s'apparentent à une surproduction. Ce cas n'entre donc pas dans notre plage de détection. Dans le cas contraire (layout ou moyens de production différents), notre méthode permet de mettre en valeur des différences d'implémentation.

Enfin, la falsification de documentation n'intervient pas au niveau physique des circuits et n'est donc pas considérée. Quant aux altérations physiques, celles-ci rentrent dans le cadre de la détection de CTM.

Ainsi, basées sur la taxonomie donnée dans le chapitre 1, les contrefaçons considérées par notre méthode de détection de contrefaçon sont donc des cas spécifiques de composants recyclés, clonés ou ayant subits un remarquage, qui sont des cas caractérisés par des structures physiques différentes.

3.2 Principe de détection des CTMs et de contrefaçons

Notre méthode de détection est basée sur le *principe suivant: la prise d'empreinte de la distribution statique de la tension d'alimentation sur l'entière surface du CI au repos* (c.-à-d. juste sous tension avec l'horloge active et sans activité logique). Afin de fabriquer cette empreinte, un réseau de capteurs est distribué uniformément sur l'ensemble de la surface du CI pour obtenir une cartographie de la tension d'alimentation interne. Le type de capteur utilisé doit donc être sensible à la tension d'alimentation locale Vdd. Dans les expériences décrites dans le chapitre 4, le même type d'oscillateur en anneau (RO) que ceux utilisés dans le chapitre 2 est utilisé. En considérant que la fréquence f d'un RO est sensible à la tension locale Vdd, la distribution des valeurs de fréquences mesurées f sur la surface du CI, en absence de variation des procédés de fabrication ou de température, est une image directe de la distribution de Vdd.

Dans notre approche, nous devons donc réduire l'effet des variations des procédés inter-die et intra-die. En utilisant cette approche, il devrait être possible de limiter les risques liés à l'introduction de CTMs pendant l'étape de fabrication et de contrefaçons pendant la distribution. Cette section présente le modèle de variation déduit des résultats du chapitre précédent et notre processus d'extraction d'empreinte d'un lot basé sur ce modèle.

3.2.1 Modèle des variations de procédé et modèle de variations de performances de structures CMOS

Soit p, un paramètre inhérent à la technologie de fabrication d'un CI, l'impact des variations des procédés de fabrication est généralement décrit de la façon suivante:

$$p = \bar{p} + \Delta p_{inter} + \Delta p_{intra} \tag{3.1}$$

avec $\bar{p}$ la valeur moyenne (ou typique) du paramètre sur l'ensemble d'un lot de production, $\Delta p_{inter} \sim N(0, \sigma^2_{inter})$ l'effet des variations inter-die considéré comme

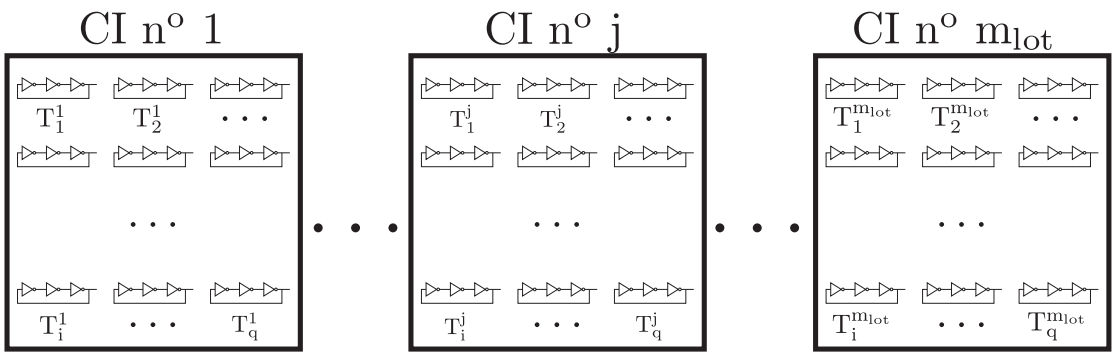

Figure 3.2: Lot de m_{lot} CIs embarquant chacun un réseau de q capteurs.

suivant une distribution normale et, $\Delta p_{intra} \sim N(0, \sigma^2_{intra})$ l'impact des variations à l'intérieur d'un même circuit, c.-à-d. les variations intra-die également considérées comme suivant une distribution normales.

Ce modèle de variation des procédés est connu et largement adopté pour simuler l'effet des variations des procédés sur le paramètre p d'un CI (un paramètre de transistor, une résistance, une jonction pn, ...). Cependant, l'extraction des valeurs d'écart type σ_{intra} et σ_{inter}, quantifiant les variations des procédés, est généralement effectué sur des circuits dédiés à la caractérisation technologique. Un tel circuit de caractérisation peut comprendre une structure régulière de transistors [Kes+05] CMOS ou de cellule SRAM [Bha+05]. Ces structures possèdent des structures physiques régulières et sont conçues pour maintenir la tension d'alimentation. Ce modèle (présenté dans l'équation (3.1)) ne prend donc pas en compte la structure physique du CI du routage de l'alimentation, de routage de l'horloge et de la densité locale de transistors, ...) sur la performance des portes CMOS ou celle d'un capteur embarqué. Dans notre cas, nous proposons d'utiliser le modèle de variation suivant pour les valeurs de sorties $T(x_i, y_i)$ d'un capteur i localisé aux coordonnées (x_i, y_i) sur la surface du CI:

$$T(x_i, y_i) = \bar{T} + \Delta T_{inter} + \Delta T_{intra} + \Delta T(x_i, y_i) \tag{3.2}$$

où $\Delta T(x, y)$ est une valeur déterministe qui dépend de la position du capteur sur le CI, et qui modélise l'impact de la structure du CI sur les performances du capteur et donc sur sa valeur de sortie. Afin de faciliter la lecture, $T(x_i, y_i)$ et $\Delta T(x_i, y_i)$ seront respectivement notés (de façon analogue aux notations précédemment introduites) T_i et ΔT_i. Cette notation temporaire a été adoptée pour mettre en valeur que le modèle de variation considérée un modèle spatial. Cette notation est illustrée dans la figure 3.2. Cette figure représente un lot de m_{lot} CIs embarquant chacun une matrice de q capteur. La période du i-ème capteur du j-ème CI est notée T_i^j.

3.2.2 Prise d'empreinte de la structure de CIs

Considérant le modèle de variation résumé par l'équation 3.2, on peut extraire l'empreinte de la structure d'un design, comprenant un réseau de q capteurs placés selon un motif régulier, de circuits provenant d'un même lot de production de CIs. Pour ce faire, les q valeurs de ΔT_i sont calculées par centrage de l'impact des variations des procédés sur les m_{lot} circuits d'un même lot.

$$\Delta T_i = \frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} T_i^j - \bar{T} = \frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} \Delta T_i^j \tag{3.3}$$

$$\sigma_{\Delta T_i} = \sqrt{\frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} (\Delta T_i^j - \Delta T_i)^2} \tag{3.4}$$

où:

$$\bar{T} = \frac{1}{m_{lot} \cdot q} \cdot \sum_{j=1}^{m_{lot}} \sum_{i=1}^{q} T_i^j \tag{3.5}$$

T_i^j est la mesure de la sortie du capteur i du circuit $j \in \{1, \ldots, m_{lot}\}$ du lot considéré.

Avec ces notations, le vecteur S^{Design} défini comme suit:

$$S^{Design} = [\Delta T_1, \ldots, \Delta T_q, \sigma_{\Delta T_1}, \ldots, \sigma_{\Delta T_q}], \tag{3.6}$$

représente l'empreinte de la structure physique d'un «Design» et est par construction indépendante des variations des procédés de fabrication. Cette empreinte est la base de la méthode de détection de CTM et de contrefaçons proposée de ce chapitre.

Dans le cas principal adressé dans ces travaux, la prise d'empreinte est basée sur un réseau de q capteurs. Cependant, cela peut être généralisé à des mesures externes comme les émanations électromagnétiques d'un CI. Dans ce cas, on note M^j (équation (3.7)) l'ensemble des q valeurs (échantillons) obtenues avec un oscilloscope numérique par analyse EM du j-ème CI. Nous obtenons ainsi un vecteur M^j pour chaque CI et donc un ensemble de vecteurs par lot.

$$M^j = [m_1^j, \ldots, m_k^j, \ldots, m_q^j] \tag{3.7}$$

En remplacement T_i^j par m_i^j dans les équations (3.8) et (3.9) S devient:

$$\Delta M_i = \frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} m_i^j - \bar{M} \tag{3.8}$$

$$\sigma_{\Delta M_i} = \sqrt{\frac{1}{m_{lot}} \cdot \sum_{j=1}^{m_{lot}} (\Delta m_i^j - \Delta M_i)^2} \tag{3.9}$$

où:

$$\bar{M} = \frac{1}{m_{lot} \cdot q} \cdot \sum_{j=1}^{m_{lot}} \sum_{i=1}^{q} m_i^j \tag{3.10}$$

S^{Design} est maintenant défini:

$$S^{Design} = [\Delta M_1, \ldots, \Delta M_q, \sigma_{\Delta M_1}, \ldots, \sigma_{\Delta M_q}] \tag{3.11}$$

Nous obtenons donc une empreinte liée à la structure physique des circuits en prenant en compte les variations de procédés. Cette empreinte ne dépend pas de l'activité de commutation électrique, le circuit ne fonctionnant pas durant la mesure. Seul l'impact statique influence l'empreinte, l'impact dynamique n'étant pas pris en compte. La comparaison de ces empreintes nous permet donc de vérifier l'intégrité d'un lot. La section suivante présente une méthodologie de détection de CTMs et de contrefaçons utilisant ces empreintes.

3.3 Méthodologie de détection

Le point de départ de notre méthodologie est l'addition d'un réseau de capteurs sensibles à la tension d'alimentation. Ces capteurs sont placés de façon à couvrir une grande partie de la surface du CI. La granularité c.-à-d. la distance entre deux capteurs est choisie par le concepteur en fonction du compromis entre capacité de détection et coût de la mesure. La figure 3.3 illustre les différentes étapes de détection de CTMs et de contrefaçons.

Quand le premier lot ou le lot de qualification (qui est moins susceptible d'être infecté, car ce lot est dédié à la caractérisation, ce qui permet la détection de CTM) est reçu, l'intégrité de certains composants est vérifiée pour qualifier le lot entier. Cela peut être fait par rétro-ingénierie ou par l'utilisation de méthode optique comme cela a été présenté dans le chapitre 1. Une fois le premier lot de production qualifié, l'empreinte (voir éq. (3.6)) du design est calculée en utilisant les équations (3.8) et (3.9). Cette signature constitue l'empreinte de référence du design considéré. On notera en outre, des techniques d'estimation de l'«aging» peuvent être appliquées pour dériver l'empreinte du design à différents âges.

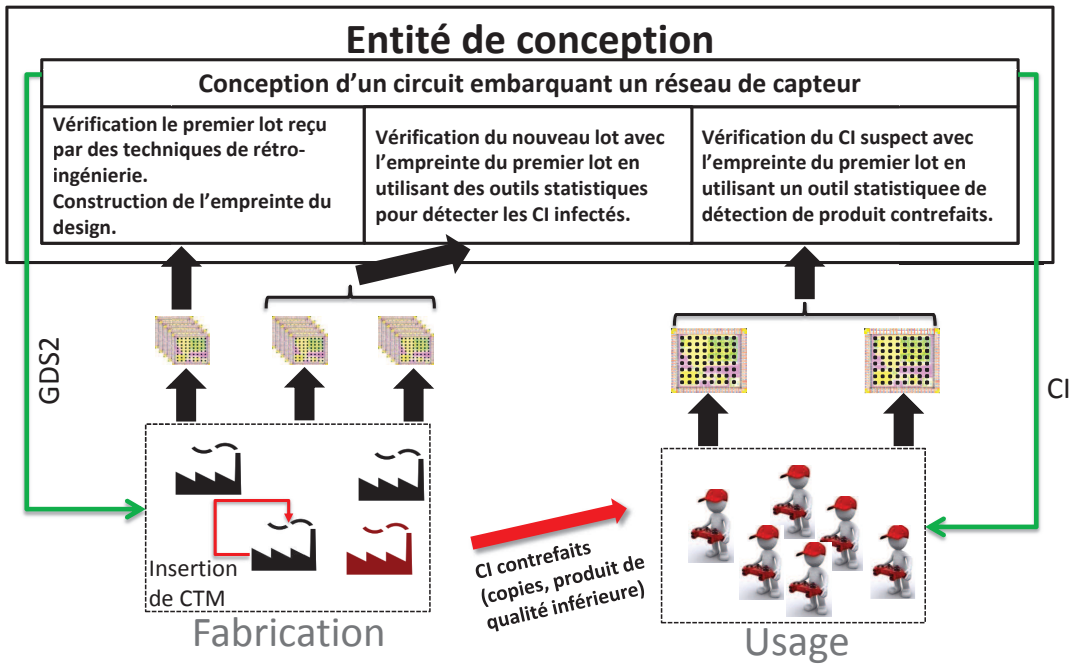

Figure 3.3: Principe de détection de CTMs et de contrefaçons.

L'entité de conception peut alors commander de nouveaux lots de CIs à la même fabrique/usine ou à une autre fabrique/usine proposant le même nœud technologique. Après réception de ces nouveaux lots, leurs empreintes correspondantes sont calculées et sont «comparées» avec celle de référence de façon à vérifier que ces derniers n'ont pas été corrompus.

De manière similaire, à une date ultérieure, le concepteur peut avoir des retours clients qui peuvent être des contrefaçons. En utilisant l'empreinte de référence, l'origine de ces retours peut être vérifiée sans application de méthodes coûteuses, complexes et destructives. Afin d'opérer cette vérification, le concepteur extrait l'empreinte du composant suspect et la compare avec l'empreinte de référence afin d'obtenir la probabilité que le design soit l'original. Si la probabilité est trop basse, une analyse complémentaire (comme un procédé de rétro conception) doit être envisagée.

Ces procédures requièrent la comparaison de la signature de référence, avec celles des nouveaux lots de production à tester afin de détecter un CTM (cas 1). Elles requirent également la comparaison de l'empreinte de référence S^{REF} avec l'empreinte d'un seul composant de façon à détecter les contrefaçons (cas 2).

3.3.1 Cas 1: détection de CTM

Quand l'intégrité d'un nouveau lot de circuits doit être vérifiée, la première étape est de calculer une empreinte S^{NewRun}. Cette empreinte doit, en pratique, être calculée en

utilisant une grande quantité de circuits ($\gtrsim 100$), pour que les grandeurs estimées puissent être considérées comme valables.

Ensuite, il est possible de comparer l'empreinte de référence avec l'empreinte du lot test. Nous avons vu dans le chapitre précédent que l'infection se traduisait pour une augmentation de la période moyenne du capteur. Dans notre modèle cela correspond à une augmentation des ΔT_i (la première partie de l'empreinte). On peut donc mettre en valeur l'impact du CTM par une différence de moyenne:

$$S_i^{Ref} - S_i^{NewRun}, i \in \{1, \ldots, q\}. \tag{3.12}$$

Cependant, la différence de moyenne ne permet pas de poser clairement un seuil de prise de décision. Nous proposons d'appliquer comme distingueur un outil statistique travaillant sur la moyenne comme le T-test, et plus précisément le test de Welch.

$$w_i = \frac{S_i^{Ref} - S_i^{NewRun}}{\sqrt{\frac{(s_{q+i}^{Ref})^2}{m_{lot}} - \frac{(s_{q+i}^{NewRun})^2}{m_{lot}}}}. \tag{3.13}$$

Ceci conduit à définir W, le vecteur composé des q résultats de q t-test (q capteurs):

$$W = [w_1, \ldots, w_k, \ldots, w_q] \tag{3.14}$$

Le t-test est un test statistique qui travaille sur la moyenne. Il permet de décider si deux distributions ont la même moyenne. L'hypothèse nulle, l'égalité des moyennes, est rejetée si la valeur résultant du t-test dépasse une valeur critique t_{critic}.

La valeur critique est choisie à partir d'une table de distribution de la loi de Student en fonction du degré de liberté ddl et d'un indice de confiance fixé par la p-value. Dans le cas de deux distributions de taille m_{lot} le degré de liberté est donné par la formule:

$$ddl = 2 \times m_{lot} - 2$$

Le t-test nous permet donc d'avoir un seuil de prise de décision. De plus, contrairement à la différence de moyenne il utilise toute la signature, ce qui permet une plus grande mise en valeur de l'infection. Donc, dans le cas où l'hypothèse nulle est rejetée pour au moins un capteur on considère que le lot testé est infecté. Une validation expérimentale de cette méthode est présentée dans le chapitre suivant.

3.3.2 Cas 2: détection de contrefaçon

Le cas d'un retour client suspect est plus difficile à traiter, car l'empreinte décrite jusqu'ici pour détecter les CTMs ne peut pas être calculée pour un seul composant: nous possédons seulement les valeurs $T^{Suspected}$ du CI considéré. Dans ce cas, on recentre toutes les valeurs $\bar{T}^{Suspected}$ du CI suspecté (c.-à-d. calculer $T_i^{Suspected} - \bar{T}^{Suspected}$) et ainsi calculer la probabilité que chaque valeur $T_i^{Suspected}$ proviennent de la même distribution normale que le lot de référence:

$$N(0, (s_{q+i}^{Suspected})^2) = N(0, \sigma_{\Delta T_i}^2) \tag{3.15}$$

En effet $\sigma_{\Delta T_i}$ est l'écart type du capteur i estimé avec les CIs du lot de référence. Les probabilités que tous les capteurs aient un comportement normal peuvent être combinées (et plus précisément une distribution normale multivariée est définie avec tous les $\sigma_{\Delta T_i}$) afin d'obtenir la probabilité que le composant considéré soit original.

3.4 t-test adaptatif

L'utilisation du t-test nécessite de placer un seuil de sensibilité au travers du choix de l'indice de confiance. Habituellement, t_{critic} est choisi par l'utilisateur en sélectionnant l'indice de confiance α. Généralement, une valeur α de 0,05 est choisie.

Cela signifie qu'un taux de faux positifs de $5\,\%$ est accepté. En pratique, rejeter l'hypothèse nulle est équivalent à déclarer le CI ou le lot de CIs infecté. Donc, choisir un indice de confiance de $5\,\%$ signifie qu'on accepte de jeter $5\,\%$ de CIs ou de lots non infectés. Dans un contexte de production, cela est trop important. D'un autre côté, choisir un indice de confiance plus bas revient à diminuer les capacités de détection et donc d'accepter d'avoir une probabilité significative de laisser des CIs ou des lots infectés passer les tests.

Pour résoudre ce problème, nous proposons une solution adaptative pour choisir la valeur critique t_{critic} du test. L'idée est de considérer qu'un CTM modifie le résultat de t-test seulement pour une minorité de valeurs, car les CTMs sont discrets et ont un impact temporel et/ou spatial réduit sur le comportement du CI. En effet, en fonction de la source d'information, les valeurs peuvent être considérées comme liées à un emplacement dans l'espace ou un moment dans le temps. Si une matrice de capteurs (ou de traces EM) possède un nombre de capteurs (ou d'échantillons temporels) suffisamment élevé, un petit nombre d'entre eux est affecté par la présence du CTM. Considérant cela, nous supposons que si un grand nombre de t-test est appliqué pour

comparer deux empreintes, la plupart passent le test. Ceci nous conduit à calculer t_{critic}, comme suit. Le calcul de t_{critic} commence par le calcul de $t_{critic_{(-k)}}$, la valeur de t-test critique associée au capteur ou échantillon k:

$$t_{critic(-k)} = \mathcal{W}_{(-k)} + \delta \cdot \sigma_{(-k)} \tag{3.16}$$

où $t_{critic_{(-k)}}$ est la somme de deux termes: la moyenne $\mathcal{W}_{(-k)}$ donnée par l'équation (3.17) et l'écart type $\sigma_{(-k)}$ donné équation (3.18). Le paramètre δ est le seuil qui a été adapté à la source de mesure, ce point étant discuté plus tard.

$$\mathcal{W}_{(-k)} = \frac{1}{q-1} \sum_{i=1, i \neq k}^{q} \mid w_i \mid \tag{3.17}$$

$$\sigma_{(-k)} = \frac{1}{q-1} \sum_{i=1, i \neq k}^{q} (w_i - \mathcal{W}_{(-k)})^2 \tag{3.18}$$

Comme le montre l'équation 3.17, la valeur w_k est rejetée du calcul de $t_{critic_{(-k)}}$ car elle ne doit pas être prise en compte. En effet, si un CTM impacte le k-ème capteur (resp. k-ème échantillon), le k-ème résultat de t-test, w_k, peut devenir significativement plus important que les autres. Donc, le prendre en compte dans le calcul de $\mathcal{W}_{(-k)}$ implique une surestimation de $\mathcal{W}_{(-k)}$ et $\sigma_{(-k)}$ dans le cas où le k-ème capteur ne serait pas impacté par un CTM.

Après le calcul des q valeurs de $t_{critic_{(-k)}}$, on obtient finalement t_{critic} en sélectionnant la valeur minimale $t_{critic_{(-k)}}$:

$$t_{critic} = \underset{k=\{1,\ldots,q\}}{argmin} \{t_{critic(-k)}\} \tag{3.19}$$

Pour illustrer cela, la figure 3.4 montre toutes les valeurs $t_{critick}$ et t_{critic} dans le cas d'une infection localisée autour du RO 48. La courbe verte correspond au vecteur W obtenu par comparaison d'un lot infecté avec un lot sain en utilisant un réseau de 60 capteurs. La valeur de t-test w_{48}, est significativement plus grande que les autres valeurs. La courbe bleue en pointillés représente les 60 valeurs $t_{critic_{(-k)}}$ calculées avec $\delta = 3$. Comme on peut l'observer, la valeur minimale $t_{critic_{(-k)}}$ est obtenue pour $k = 48$, et donc $t_{critic} = t_{critic(-48)}$.

Avec cette définition de t_{critic} nous sommes maintenant capables de sélectionner par un processus adaptatif un seuil de décision pour détecter les CTMs malgré l'influence de

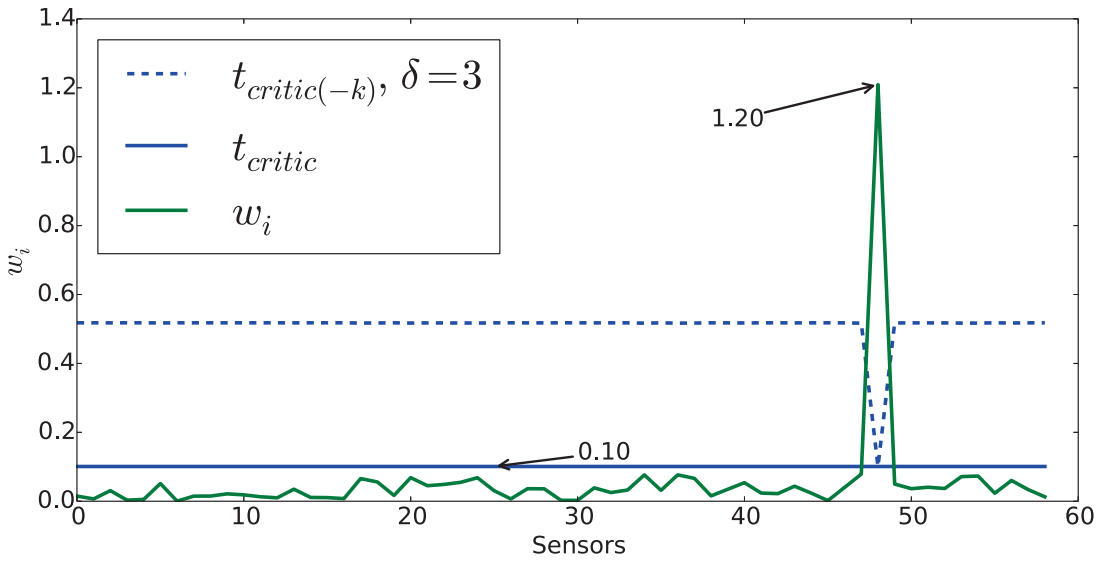

Figure 3.4: t_{critic} adaptatif.

variations de procédés ou de bruit de mesure tout en éliminant les risques liés aux faux positifs.

3.5 Méthodes alternatives de comparaison d'empreintes

D'autres outils statistiques pour comparer les empreintes ont été étudiés. Cette section présente les principales pistes étudiées.

3.5.1 Test de Kolmogrov-Smirnov

Un test de Kolmogrov-Smirnov est un test statistique qui permet de déterminer si une distribution suit une fonction de répartition théorique ou si deux distributions empiriques suivent la même fonction de répartition. On peut l'utiliser dans deux méthodes différentes.

Dans le premier cas, on peut comparer des distributions de période du lot de référence avec le lot testé. Le capteur i suit une loi normale $N((s_i^{Ref})^2, (s_{q+i}^{Ref})^2)$ pour le lot de référence et suit une loi normale $N((s_i^{test})^2, (s_{q+i}^{test})^2)$ pour le lot testé. D'après la section 3.2.1 si le design des deux lots est identique alors leurs distributions seront identiques. Le test de Kolmogrov-Smirnov permet donc de déterminer si les distributions sont identiques, et ce pour chaque capteur. Ainsi si la distribution d'au moins un capteur est différente, le circuit testé est considéré comme infecté.

Dans un second cas, on peut considérer la fusion des lots. Pour un capteur i, si $N((s_i^{Ref})^2, (s_{q+i}^{Ref})^2)$ et $N((s_i^{test})^2, (s_{q+i}^{test})^2)$ sont les mêmes distributions normales alors la combinaison des deux aura la même distribution et sera une normale. Dans le cas où les deux distributions seraient différentes, la composition des deux distributions ne donne pas une loi normale. Ainsi en utilisant un test de Kolmogrov-Smirnov pour vérifier que les distributions des capteurs pour les deux lots suivent une loi normale on peut vérifier que le lot testé n'est pas un lot infecté. Par ailleurs, cette méthode peut être utilisée pour vérifier que tous les circuits d'un lot ont la même structure physique. En effet, si l'on est en présence de deux populations distinctes alors la distribution d'un capteur au travers du lot ne suivra pas la distribution attendue.

3.5.2 Partitionnement

Toujours selon cette idée de comparer les populations centrées, il est également possible d'utiliser des solutions de partitionnement comme les k-moyennes, ou dans le cas de deux lots de tailles identiques la médiane. La méthode des k-moyennes est un algorithme de partitionnement. Il prend pour paramètre k qui est le nombre de partitions attendues. Cet algorithme fonctionne par itération en cherchant à minimiser la distance entre les points de chaque partition. Dans notre cas, $k = 2$ on cherchera à identifier deux ensembles distincts dans la population composée de la fusion du lot testé et du lot de référence.

Pour ce faire, les populations du lot de référence et du lot dont on souhaite vérifier l'intégrité sont mélangées. Ceci fait, la méthode des k-moyennes est appliquée à la population résultante (ou la médiane calculée) et la répartition des pièces appartenant au lot de référence et au nouveau lot sont analysées. Les répartitions doivent être similaires à la répartition de départ dans les deux populations obtenues. En effet, si le circuit est sain, les deux sous-populations obtenues avec ces outils doivent être réparties de façon aléatoire sous l'effet des variations aléatoires des procédés de fabrication. Par exemple, dans le cas où les nombres de pièces du lot de référence et du nouveau lot sont égaux, $50\,\%$ des pièces du lot de référence et $50\,\%$ de pièces du nouveau lot doivent constituer une sous-population. Dans le cas où l'on observerait un biais dans la répartition, on en déduit qu'une différence, provoquer par l'impact d'un CTM, existe dans les populations de départ.

3.6 Conclusion

Ce chapitre a introduit une méthode de détection de CTMs et de contrefaçons. Cette méthode a pour but de détecter des différences de structures physiques dans des lots de circuits. Les cas de CTM ou de contrefaçons couverts par cette méthode ont tout d'abord

été détaillés. Les CTMs et les contrefaçons considérés sont caractérisés par des modifications par rapport à la structure physique des circuits originaux. Dans le cas des CTMs cela correspond aux modifications opérées pendant l'étape de fabrication, qu'importe leur effet. De plus, ces CTMs sont de type fonctionnel et sont localisés dans l'espace. La problématique de leur taille sera abordée dans le chapitre 4. Dans le cas des contrefaçons, nous considérons les cas spécifiques de composants recyclés, clonés ou ayant subits un remarquage.

Ensuite, en s'appuyant sur les résultats du chapitre précédent, un nouveau modèle de variation des performances temporelles des structures CMOS a été proposé:

$$T(x_i, y_i) = \bar{T} + \Delta T_{inter} + \Delta T_{intra} + \Delta T(x_i, y_i)$$

Celui-ci prend en compte les variations de procédés de fabrication en plus de la structure physique des circuits.

À partir de là, deux cas sont considérés. Dans le premier cas, on compare une population de référence avec une population testée afin de détecter l'impact statique d'un éventuel ajout de CTM. Dans un second cas, on compare un lot de référence avec un CI seul afin de vérifier que ce n'est pas une contrefaçon. Enfin, pour pallier certaines limitations de la méthode dans le premier cas, un nouveau distingueur pour la prise de décision a été introduit. Ce distingueur est adaptatif dans le sens ou il s'adapte aux variations de procédé et au bruit de mesure. Il est basé sur le t-test et part du principe qu'une infection est discrète et donc localisé dans le temps ou l'espace. La validation expérimentale des points présentés dans ce chapitre est donnée dans le chapitre suivant.

4

Résultats expérimentaux et analyses

Ce chapitre présente les validations expérimentales des points présentés dans le chapitre 3. La validation du nouveau modèle de variation des performances des structures CMOS et les résultats de détection de CTMs et de contrefaçons ont été publiés dans [Lec+16b]. Les résultats montrent que notre modèle de variation permet de prendre en compte l'influence des variations des procédés de fabrication et de l'impact du placement routage sur un réseau de capteurs. De plus, la méthode basée sur ce modèle permet de détecter un ajout de logique ou des contrefaçons. Les expériences ont montrées des résultats similaires pour une analyse EM. Ensuite, notre proposition de distingueur est évaluée. Il apparait que celui-ci permet de détecter des infections plus petites qu'avec un test classique. Ces résultats ont été publiés dans [Lec+16a]. Suite à cette validation une caractérisation des limites de la méthode a été faite. Les résultats montrent que le nombre de circuits nécessaires pour atteindre un taux de succès de $100\,\%$ *augmente à mesure que la taille des CTMs diminue. Enfin, afin d'améliorer la couverture spatiale de notre méthode nous proposons une implémentation étendue d'oscillateurs en anneaux. Celle-ci permet d'améliorer la portée de détection de petites infections par rapport à une implémentation compacte.*

Dans le but de proposer une solution embarquée de vérification d'intégrité des circuits intégrés, nous avons caractérisé dans le chapitre 2 les impacts statiques et dynamiques des modifications structurelles sur un réseau de capteurs embarqués. Cela nous a permis de proposer, dans le chapitre 3, un nouveau modèle de variation des performances. En nous appuyant sur ce modèle, nous avons alors proposé dans ce même chapitre une méthode embarquée de vérification d'intégrité s'inscrivant dans un contexte de production industrielle globalisée. Le présent chapitre a pour but de valider expérimentalement les principes énoncés dans le chapitre précédent, c.-à-d. le modèle de variation de performance des structures CMOS, la détection de CTM (embarquée et externe), la détection de contrefaçons et le nouveau distingueur. Pour cela, de nouvelles implémentations de test ont été développées et exploitées. Ensuite, nous caractérisons les limites du système proposé pour la détection CTM. Les deux principaux critères sont la granularité du réseau de capteurs, ainsi que la taille des CTMs. Enfin, pour faire suite aux résultats de caractérisation de l'efficacité de la méthode, une amélioration du capteur considéré est proposée.

4.1 Protocole expérimental

Afin de valider et caractériser les réseaux de capteurs, de nouvelles implémentations de circuits de test ont été réalisées. De plus, compte tenu du nombre important de placement-routages à effectuer en raison des différentes caractéristiques de CTM à étudier, un script d'automatisation de l'infection du ".ncd" a été programmé. Cette section décrit le matériel utilisé, les caractéristiques des circuits implémentées, ainsi que le schéma d'automatisation utilisé.

Le protocole de mesure est similaire à celui présenté dans le chapitre 2. Sur chaque FPGA Spartan-3E-1600, un AES utilisant des clés de 128 bits, un bloc de communication série (RS232) et une machine à états finis (FSM) ont été placés et routés. Une matrice de 60 ROs a été ajoutée sur le design. Chaque RO est couplé à un diviseur de fréquence par deux afin d'observer et de mesurer précisément la fréquence au travers d'un multiplexeur connecté sur une broche de sortie. Le supplément de surface induit par notre matériel de détection embarqué est d'environs $3{,}2\,\%$ des ressources du FPGA. La fréquence est mesurée avec un oscilloscope de chez Lecroy possédant une bande passante de $4\,\mathrm{GHz}$ et un taux d'échantillonnage de $40\,\mathrm{GS/s}$.

Dans le but d'obtenir une bonne précision de mesure (précision de $\pm 0{,}025\,\mathrm{ps}$), chaque estimation de la fréquence est faite en mesurant une durée correspondant à plus de 100 périodes de chaque RO: $\mathrm{T}_\mathrm{i}^\mathrm{j}$. Pendant la mesure, le CI est laissé inactif, c.-à-d. juste mis sous tension avec l'horloge active et le minimum de logique active pour superviser la mesure. Le temps de mesure des 60 valeurs T_i d'une carte est d'environs $2\,\mathrm{min}$ ce qui assez court pour affirmer que la température de l'environnement du laboratoire est constante. Afin de garantir une bonne stabilité de mesure, le FPGA est alimenté par une alimentation stabilisée possédant une précision de $0{,}05\,\%$.

Lors de la phase de validation, on utilise, pour émuler l'effet d'un CTM, un LSFR de 64 bits, dont le nombre de bits commutant n'est pas modulable. Il occupe 48 slices ce qui représente $0{,}32\,\%$ de la surface du FPGA. Il est à noter que la surface de l'AES tout seul est de 1778 slices. Le LFSR est cadencé par une horloge de fréquence $50\,\mathrm{MHz}$. Ce CTM peut donc être considéré comme séquentiel.

Pour émuler des contrefaçons, différents placement-routages du même layout ont été effectués. Les 3 différents layouts proviennent de la même description HDL ont été implémenté. L'un d'entre eux (placement-routage 1) est considéré comme original/sain, les deux autres (placement-routage 2 et 3) sont considérés comme des contrefaçons.

Lors de la phase de caractérisation, d'autres tailles et positions seront utilisées. Pour réduire drastiquement les temps mis pour insérer une infection, tout en conservant un

faible impact sur le placement-routage initial, une nouvelle procédure d'infection a été mise en place.

4.1.1 Implémentation

La figure 4.1, décrit la nouvelle procédure d'implémentation et d'infection adoptée dans ce chapitre. Les premières étapes sont identiques à la procédure décrite dans le chapitre 2. Mais, la modification manuelle du fichier de layout (".ncd") est remplacée par une procédure automatique de modification.

La première étape consiste toujours en la conception HDL du circuit infecté. Dans le but de pouvoir choisir la taille de l'infection par la suite, toutes les tailles d'infection souhaitées sont implémentées durant cette phase. Comme dans le processus présenté dans le chapitre 2 l'optimisation des CTMs (nombre de slices) est manuellement vérifiée. En pratique, 4 tailles de CTMs ont été conçues, ces tailles sont résumées dans le tableau 4.1. La première ligne correspond au nombre bits (bascule D) manipulés par le CTM, la seconde ligne est le nombre de slices utilisés pour son implémentation et la dernière ligne correspond au ratio des ressources utilisées par le CTM sur les ressources totales du FPGA. Un fichier de layout (".ncd") contenant le circuit cible et les 4 infections est donc généré après synthèse et placement-routage. Ce fichier sera le fichier de base utilisé pour toutes les infections. À cet effet, un programme a été réalisé pour manipuler automatiquement ce fichier.

Le principe de ce programme est d'utiliser les fonctions de manipulation de ".ncd" offertes par FPGA editor. En effet, ce logiciel peut être commandé, via un fichier de commande. Nous cherchons donc à générer un fichier de procédure d'infection décrivant les étapes pour réaliser le layout attendu. Dans un premier temps, on génère un ficher contenant la taille et la position de l'infection que l'on souhaite obtenir. Dans un second temps, on utilise un outil fourni par Xilinx nommé "ncd2xdl". Celui-ci permet de convertir le fichier ".ncd" en fichier ".xdl" contenant les mêmes informations, mais sous un format textuel et non plus binaire. Ce format nous permet d'exploiter plus facilement les informations.

C'est à partir de ces deux fichiers que notre programme travaille afin générer le fichier de commande utilisé par FPGA editor pour modifier le ".ncd". Celui-ci parse le ".xdl" afin de récupérer la position de tous les éléments du layout. Puis, il écrit dans un fichier de commande les instructions nécessaires pour supprimer les CTMs non désirés de façon à ne garder que celui correspondant à la taille souhaitée. Ensuite, il inscrit les commandes indiquant comment déplacer les composants du CTM restant pour obtenir l'infection à l'emplacement désiré. Enfin, les commandes de routage sont inscrites. Dans le cas où l'on souhaite obtenir un design sain, tous les CTMs sont supprimés.

Tableau 4.1: Tailles des CTMs implémentés

Taille (bits)	8	16	32	64
Taille (slices)	5	11	23	48
Ratio (%)	0,033	0,0074	0,15	0,32

Une fois le fichier de commande réalisé, celui-ci est exécuté par FPGA editor afin de modifier le ".ncd" et d'obtenir le layout infecté à la position voulue et par la taille de CTM souhaitée. Le bitstream correspondant est ensuite généré.

4.1.2 Estimations de σ_{inter} et σ_{intra}

Avant d'évaluer la pertinence de la méthodologie proposée et donc celle du modèle de variation des performances d'un capteur, les impacts des variations inter-die et intra-die a été estimé à partir des cartes FPGA à disposition.

La figure 4.2 reporte les histogrammes des périodes des 60 ROs pour 10 cartes. Comme on peut l'observer, la période moyenne par carte varie de $13{,}5\,\mathrm{ns}$ à $14{,}5\,\mathrm{ns}$ pour ces 10 cartes. L'impact des variations intra-die est donc de l'ordre plusieurs centaines de ps. Ce type de figure nous a permis d'estimer, à partir des 30 cartes, que l'impact de l'inter-die, modélisé par une Normale $N(0, \sigma^2_{inter})$, est tel que $\sigma^2_{inter} = 460\,\mathrm{ps}^2$.

La figure 4.3 illustre sous forme d'histogramme la répartition des 60 valeurs centrées (par carte) des périodes caractéristiques des ROs pour 10 cartes. Comme on peut le constater, les variations intra-die ont un impact de l'ordre de $100\,\mathrm{ps}$. En supposant que la distribution de ces variations est normale, elles sont donc caractérisées par la normale $N(0, \sigma^2_{intra})$ avec σ^2_{intra} estimée à $130\,\mathrm{ps}^2$. Ces deux jeux d'histogrammes (figure 4.2 et figure 4.3) semblent confirmer les distributions normales dues aux variations des procédés de fabrication. La prochaine section valide notre modèle de variation des performances des structures CMOS, notre méthode d'extraction d'empreinte et notre principe de vérification d'intégrité.

4.2 Validation de la détection embarquée

Dans le chapitre 3, nous avons introduit un modèle de variation des performances des composants logiques CMOS et par extension de nos capteurs. Ce nouveau modèle introduit un terme qui représente l'impact de la structure du design sur les performances des capteurs et particulièrement de l'impact de la distribution d'alimentation, tout en étant indépendant des variations de procédés. Ce nouveau modèle est la base de la méthode de détection proposée, nous commençons par évaluer sa pertinence.

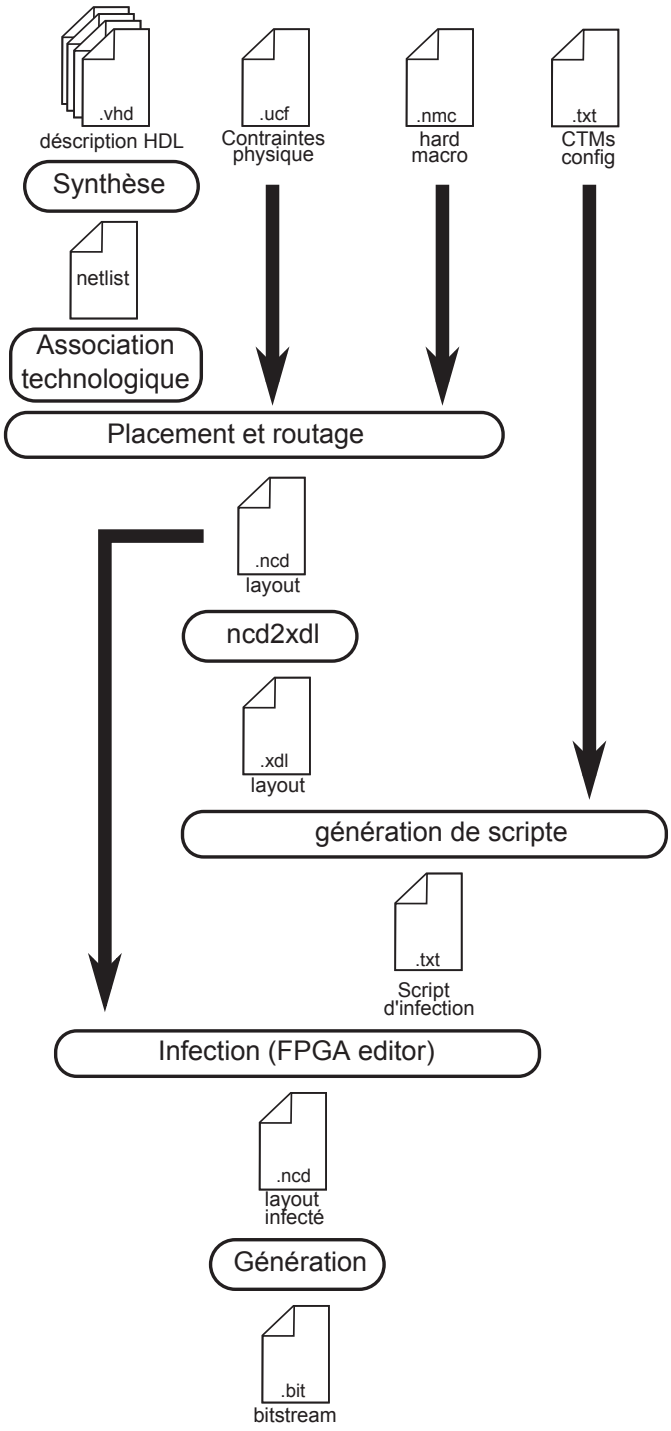

Figure 4.1: Processus d'implémentation et d'infection des FPGAs.

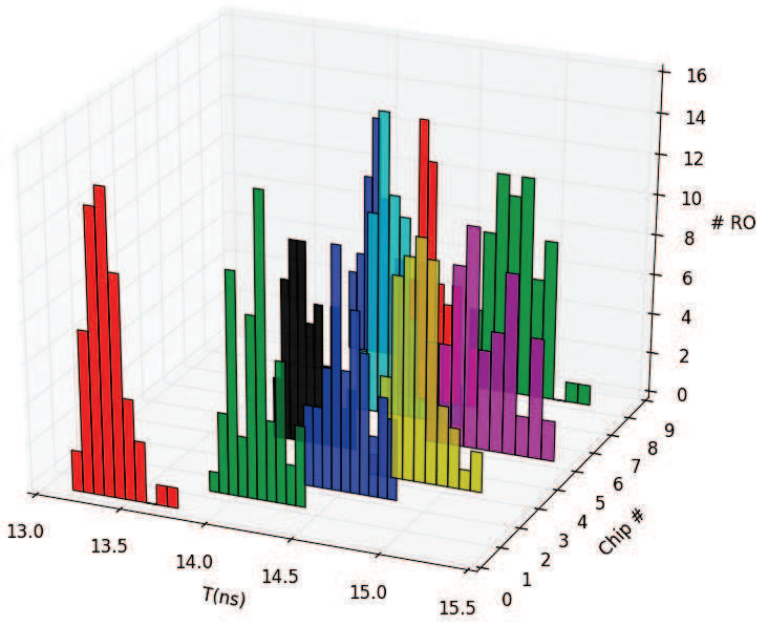

Figure 4.2: Illustration de l'impact des variations de process inter-die

4.2.1 Modèle de variation

Ce modèle est original dans le sens où il introduit un terme déterministe modélisant l'impact de la structure sur les performances des capteurs ou éléments CMOS et plus particulièrement la distribution de l'alimentation sur la puce. Ce modèle étant original et au cœur de notre méthode de détection, nous avons évalué dans un premier temps sa pertinence.

Pour ce faire, nous avons mesuré les fréquences des 60 ROs des trois placement-routages reportés sur la figure 4.4 en utilisant 15 cartes. Ceci fait, les trois empreintes S^{Design} ont été comparées. Les trois placement-routages représentés sont différentes implémentations de la même description HDL. Les différences résident dans les contraintes de placement de l'AES qui ont été modifiées. Ces différentes implémentations ont été effectuées pour illustrer l'impact des contrefaçons sur les empreintes obtenues. Le placement-routage 1 représente celui d'un circuit de référence et les placement-routages 2 et 3 représentent les implémentations de circuits contrefaits. Ce type de contrefaçon est donc une copie fonctionnelle du circuit original et non pas une copie exacte.

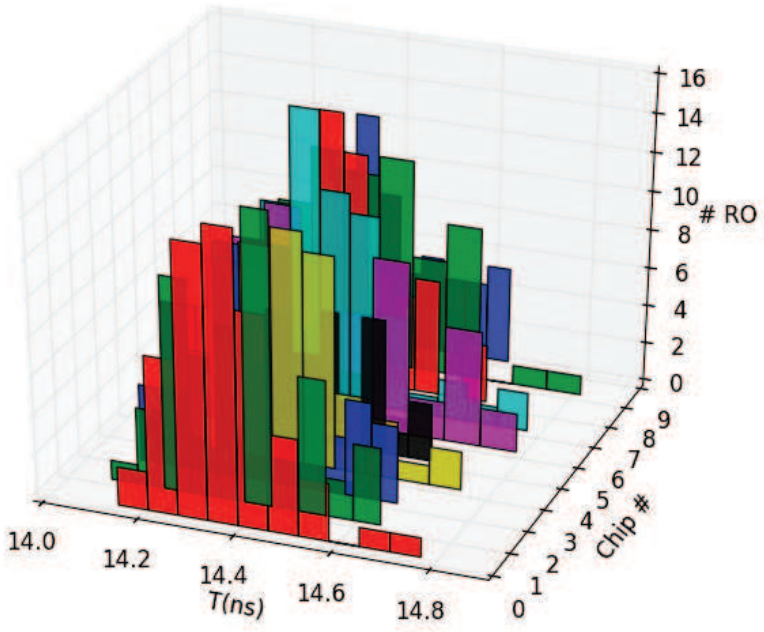

Figure 4.3: Illustration de l'impact des variations de process intra-die

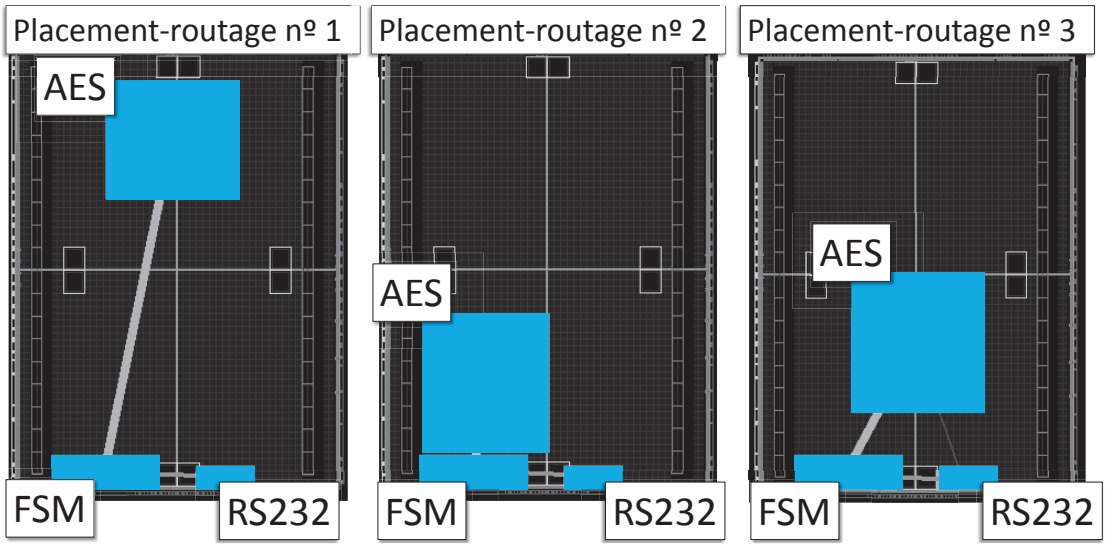

Figure 4.4: 3 placement-routages différents de la même fonction.

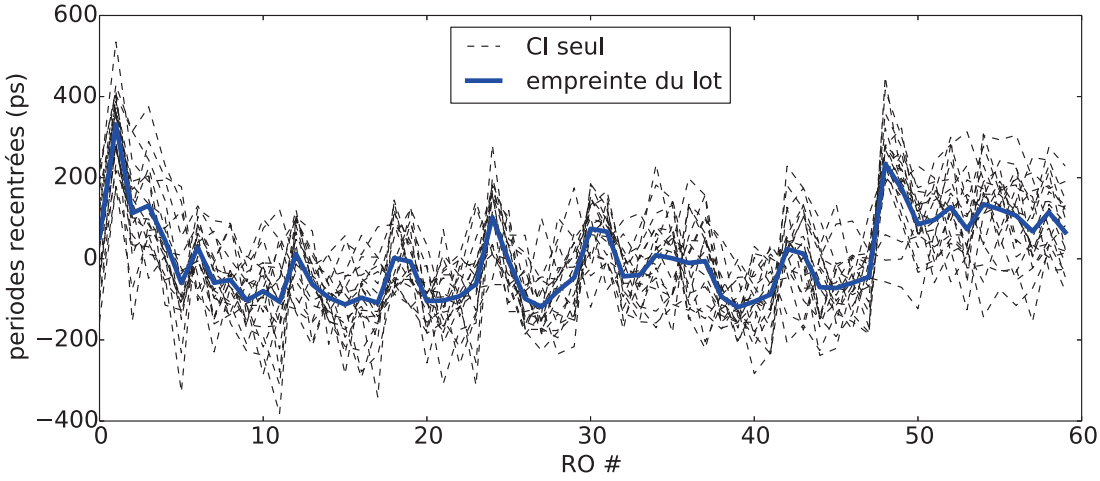

Figure 4.5: Illustration de la structure

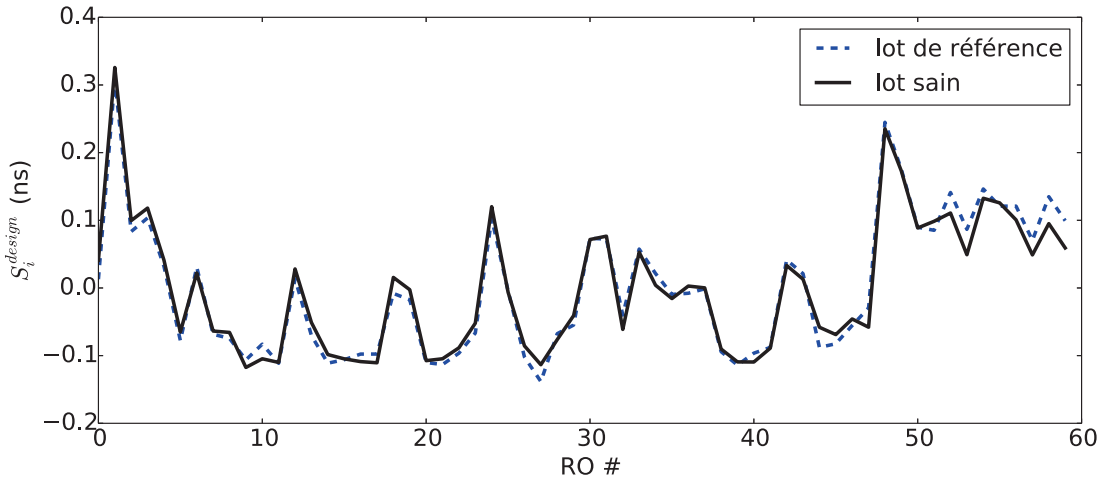

Figure 4.6: Illustration de la structure pour deux lots aux implémentations identiques.

La figure 4.5 reporte l'empreinte du placement-routage 1 (S^{Design}), et les périodes particulières de chacune des 15 pièces qui ont permis son obtention. On observe donc que les périodes des cartes prises individuellement évoluent autour d'une structure commune qui est extraite par l'empreinte. Ce résultat semble confirmer notre méthode d'extraction de la structure commune d'un lot.

La figure 4.6 reporte les signatures obtenues avec deux lots de 15 pièces différents embarquant le même layout (placement-routage 1). On observe que les deux empreintes sont très proches. Cela confirme que notre empreinte permet de réduire l'impact des variations des procédés de fabrication et est liée au placement-routage implémenté.

La figure 4.7 reporte de manière graphique les trois signatures obtenues pour les 3 placement-routages considérés (figure 4.4). Comme on peut le constater, elles sont nettement différentes, et ce bien que les positions des 60 ROs aient été conservées

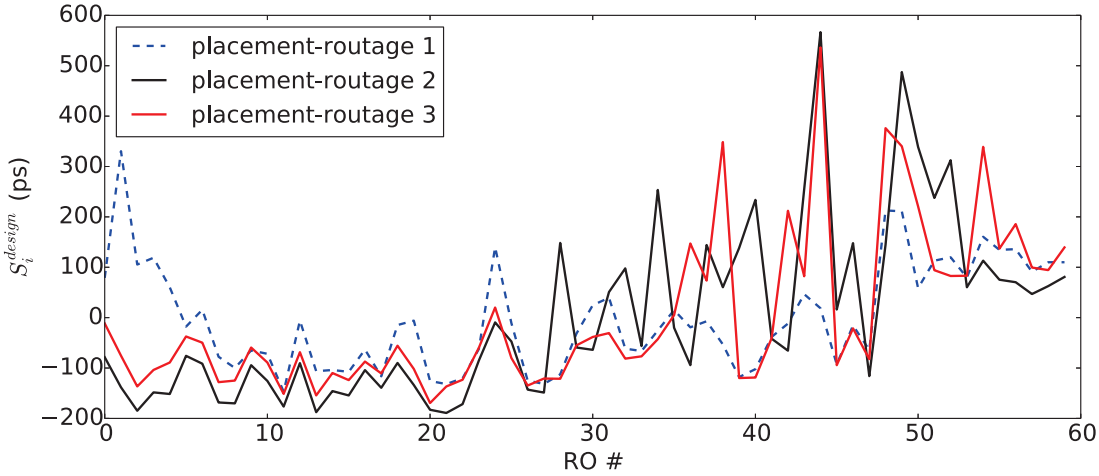

Figure 4.7: Empreintes des 3 placement-routages.

identiques et que les mesures aient été faites sur les mêmes 15 cartes. On notera également que les designs 2 et 3 ont des signatures significativement différentes bien que les deux implémentations soient relativement similaires.

On peut observer que pour les deux placement-routages 2 et 3, les capteurs 30 à 45, positionnés au voisinage ou dans l'AES (voir placement-routages 2 et 3 sur la figure 4.4), sont caractérisés par des valeurs de S_i^{design} élevées. Pour le placement-routage 1, ce sont les capteurs 1 à 10. Ces derniers sont positionnés en haut du FPGA autour et dans l'AES (voir placement-routage 1 sur la figure 4.4). Ceci conforte l'hypothèse émise que le floorplan affecte la performance des capteurs en modifiant localement la valeur de la tension d'alimentation. De plus, cela confirme le potentiel de notre méthode, qui permet de distinguer différentes implémentations de la même fonctionnalité, et donc de détecter des contrefaçons potentielles.

4.2.2 Détection de contrefaçon

Pour démontrer expérimentalement que notre méthode peut déterminer si un CI suspect est une contrefaçon (comparé à un lot de référence) en utilisant notre modèle, nous avons utilisé les placement-routages 1 et 2 présentés sur la figure 4.4. Ces deux implémentations sont fonctionnellement équivalentes. Les fréquences des 60 ROs du premier placement-routage ont été mesurées sur 15 CIs. Puis, les fréquences des 60 ROs du second placement-routage ont été mesurées sur un CI. La figure 4.8 montre l'empreinte complète (calculée à partir de 15 CIs) du placement-routage 1 (courbe bleue), c.-à-d. les valeurs $\Delta T_i \pm \sigma_{\Delta T_i}$, et l'empreinte du composant suspecté (courbe en noire en pointillé). Dans ce cas, il n'y visuellement pas de doute que le composant considéré est contrefait. En effet, les ROs 30, 39 et 40 sont hors des $\pm 3 \cdot \sigma_{\Delta T_i}$ mesurés

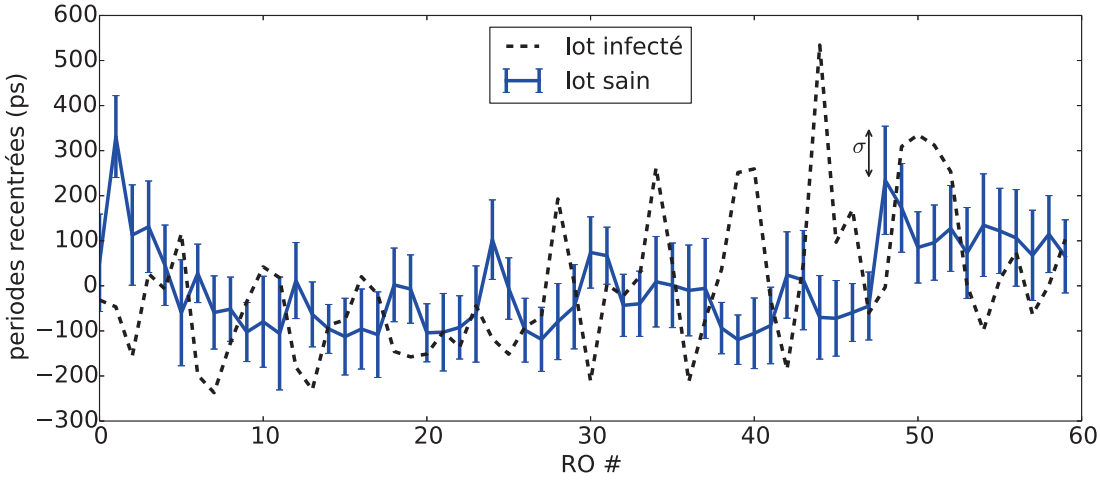

Figure 4.8: $S^{Design1}$ et empreinte d'une contrefaçon

sur le lot de référence. Ce résultat valide notre proposition de méthode de détection de contrefaçons et notre modèle de variation.

4.2.3 Détection de CTM

La méthode de détection d'un lot infecté est quasiment similaire à celle d'un lot contrefait, avec pour différence que dans le cas du CTM l'altération de la structure physique est significativement plus petite et plus localisée. Pour valider la méthode de détection, nous avons utilisé l'implémentation infectée décrite dans la section 4.1. La figure 4.9 représente le placement routage utilisé. L'infection est donc un LFSR de 64 bits. Cette infection est placée autour du RO 33. Il doit être noté que nous cherchons à détecter l'impact statique du CTM. Le LFSR ne fonctionnant pas, les bascules D ne commutent pas.

La figure 4.10 donne deux empreintes provenant d'un lot de 15 CIs. La première empreinte (courbe bleue en pointillés) est obtenue avec le layout de référence. La seconde empreinte (courbe noire) est obtenue avec le layout infecté. On observe que les deux empreintes se superposent pour tout les ROs sauf le RO 33 (point entouré). En effet, pour ce dernier la valeur est significativement plus importante avec l'infection. Cela confirme la sensibilité de notre empreinte à un ajout de logique malicieuse. De plus, cet impact est local ce qui est cohérent avec les résultats du chapitre 2.

Les sections suivantes décrivent comment mettre en valeur cette différence, par comparaison avec un lot de référence, en utilisant les méthodes présentées dans le chapitre 4. Les résultats présentés ont pour but de vérifier l'intégrité de lots de 15 CIs infectés et 15 CIs sains avec le lot de référence de 15 CIs. 30 cartes ont été utilisées et l'infection à base de LFSR de 64 bits autour du RO 33 a été utilisée.

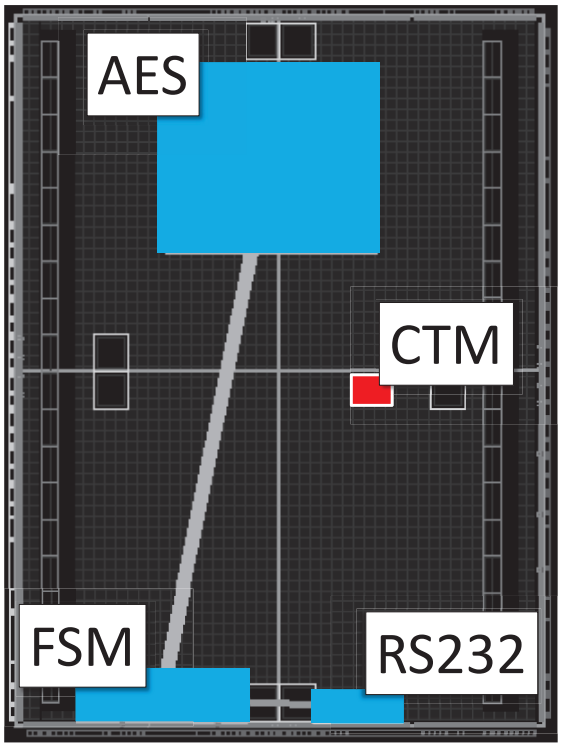

Figure 4.9: Placement-routage avec infection.

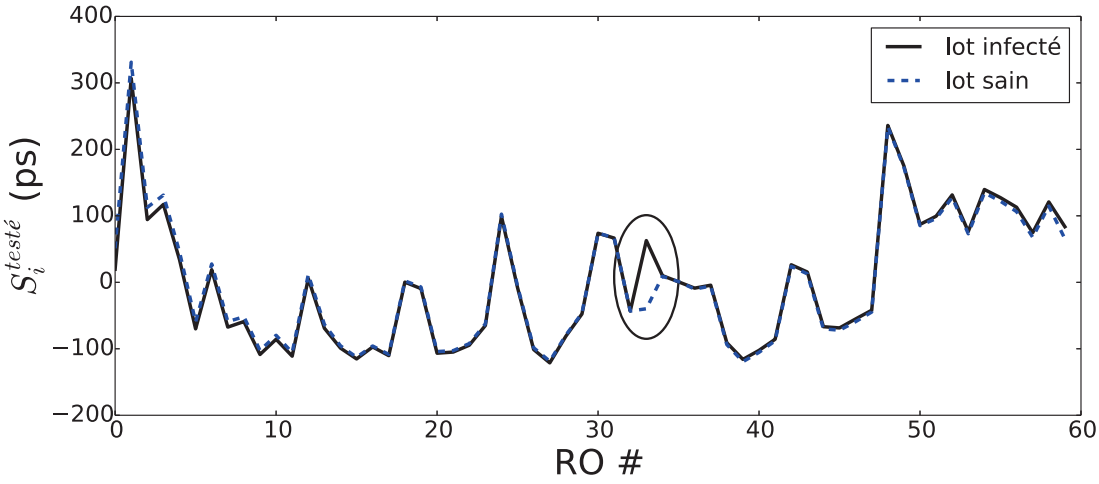

Figure 4.10: Empreintes d'un lot infecté et un lot sain.

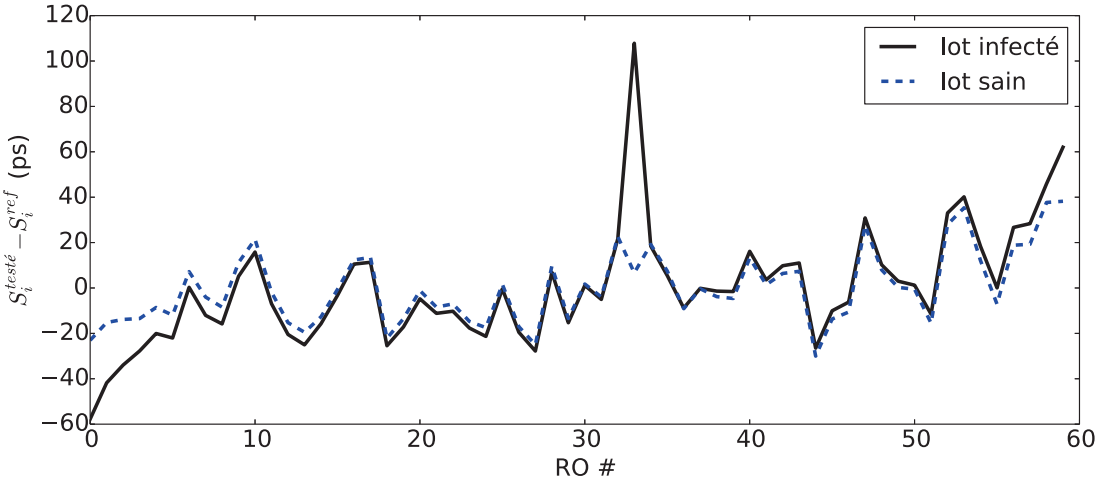

Figure 4.11: Différences des moyennes entre deux lots testés (un lot sain et un lot infecté) et le lot de référence.

Différence de moyenne

La figure 4.11 présente les résultats obtenus en utilisant la différence de moyenne. Cela correspond donc à la différence entre les premières parties d'empreinte. La courbe en pointillés bleue représente la différence d'empreinte entre le lot sain et le lot de référence, c.-à-d. $S^{sain} - S^{ref}$. La courbe en noir représente la différence d'empreinte entre un lot infecté et celle du lot de référence, c.-à-d. $S^{infecté} - S^{ref}$. On observe une valeur anormale pour le RO 33, la différence de moyenne permet donc de mettre en valeur l'infection. Cependant, elle ne permet pas de poser un seuil de décision.

t-test

La figure 4.12 montre les résultats obtenus en utilisant le t-test. La courbe en pointillés bleus représente le t-test appliqué entre le lot sain et le lot de référence. La courbe en noire représente le t-test appliqué entre le lot infecté et le lot de référence. Dans notre cas, avec des lots de 15 pièces, le degré de liberté est de $ddl = 28$. En choisissant un indice de confiance de $5\,\%$ on obtient une valeur critique de $2,04$.

Dans le cas du lot sain, la valeur absolue $\mid w_i \mid$ n'excède pas $2,04$ pour $i \in \{1, \ldots, 60\}$. Le lot sain est donc bien reconnu comme non infecté. Dans le cas du lot infecté, la valeur absolue $\mid w_i \mid$ dépasse la valeur critique de $2,04$ pour le RO 33. Le lot est donc déclaré infecté par une infection proche du RO 33. Ces résultats confirment que l'utilisation du t-test permet de décider quels lots sont sains et quels lots sont infectés.

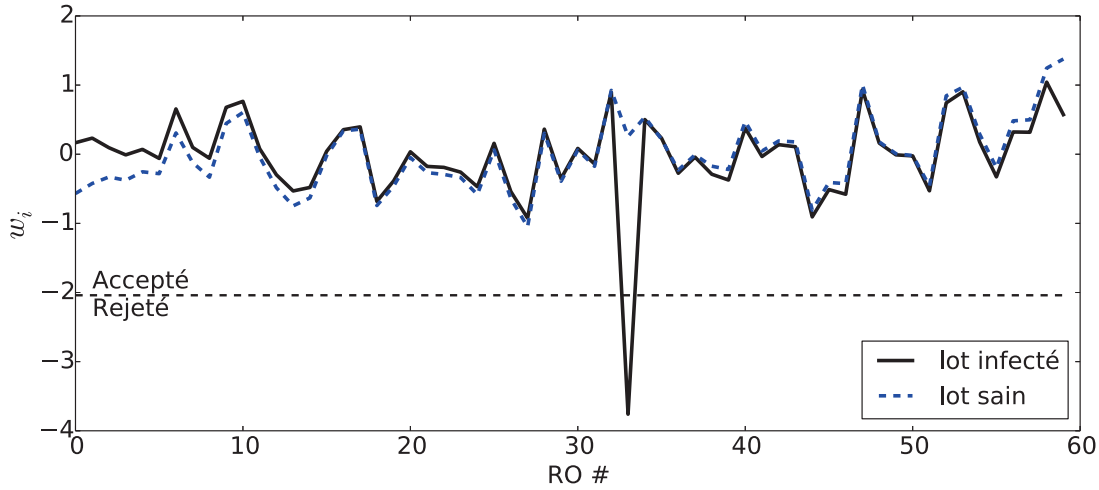

Figure 4.12: t-test entre l'empreinte de 15 CIs infectés et 15 CIs sains avec un lot de référence

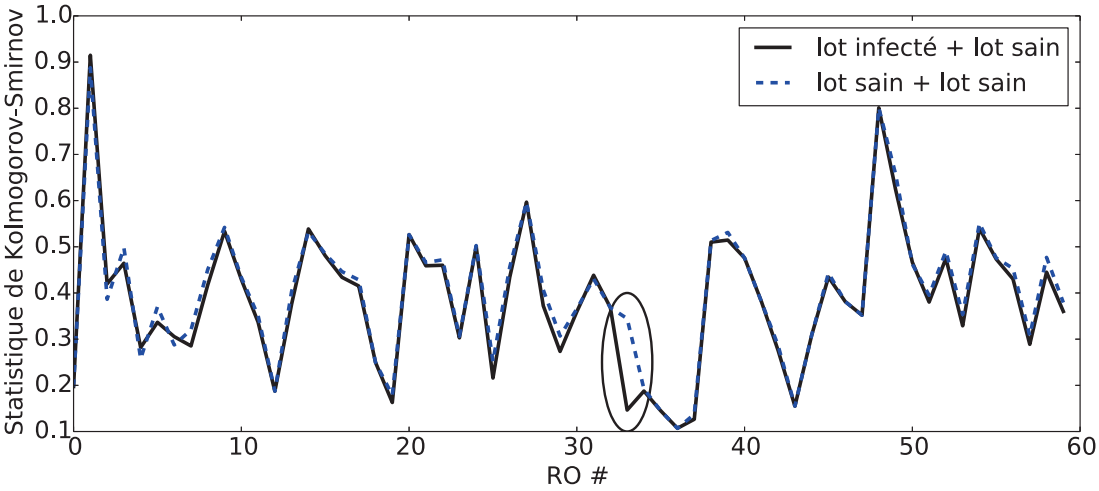

Figure 4.13: Test de Kolmogorov-Smirnov, comparaison de la distribution d'une combinaison de lot à une normale.

Kolmogorov-Smirnov

La figure 4.13 donne les résultats d'un test de Kolmogorov-Smirnov utilisé pour comparer les distributions des compositions de lots à une distribution normale. La courbe bleue en pointillé correspond à la composition du lot sain et du lot de référence. La courbe noire correspond à la composition du lot infecté et du lot de référence. On observe une différence entre les deux courbes, ce qui montre l'infection a un impact sur la distribution. Cependant, le nombre de lots semble trop peu important pour pouvoir comparer les distributions estimées expérimentalement avec une distribution théorique, car on observe des valeurs élevées ($> 0,4$) pour la majorité des points dans les deux cas. Par conséquent, ces résultats ne permettent pas de confirmer cette méthode.

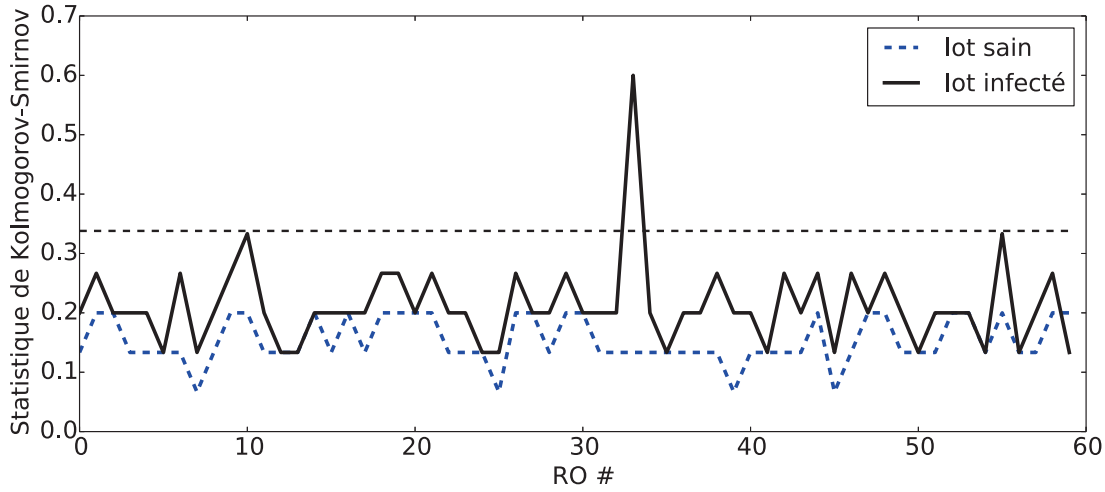

Figure 4.14: Test de Kolmogorov-Smirnov, comparaison des distributions du lot de référence avec les distributions d'un lot testé.

Le test de Kolmogorov-Smirnov peut aussi être utilisé pour comparer deux distributions provenant de mesures expérimentales. La figure 4.14 donne les résultats d'un test de Kolmogorov-Smirnov utilisé pour comparer les distributions mesurées. La courbe bleue en pointillés correspond à la comparaison du lot sain et du lot de référence. La courbe noire correspond à la comparaison du lot infecté et du lot de référence. Dans ce cas d'utilisation, le test permet de décider ou non de l'infection d'un lot.

En effet, en considérant la taille des lots (15 pièces) et en choisissant un indice de confiance de $5\,\%$ on obtient une valeur critique de $0,338$. On observe que dans le cas d'un lot sain cette valeur n'est pas atteinte pour $i \in \{1, \ldots, 60\}$, ce qui permet de déclarer le lot comme sain. Dans le cas d'un lot infecté, la valeur critique est dépassée pour le RO 33. Le lot est donc déclaré comme infecté par une infection proche du RO 33.

Ces résultats n'ont pas permis de valider la première approche utilisant un test de Kolmogorov-Smirnov. Pour cela, plus de circuits seront nécessaires pour estimer correctement les distributions des capteurs. Cependant, la seconde approche a pu être validée. Cela montre que ce test peut être pertinent pour la détection de CTM.

Partitionnement

Les figures 4.15 et 4.16 donnent les résultats obtenus par les méthodes de partitionnements, la première en utilisant la médiane et la seconde en utilisant la méthode des k-moyennes. Sur chacune des deux figures, la courbe bleue en pointillé correspond à partitionnement de la composition du lot sain et du lot de référence et la courbe noire correspond au partitionnement de la composition du lot infecté et du lot de

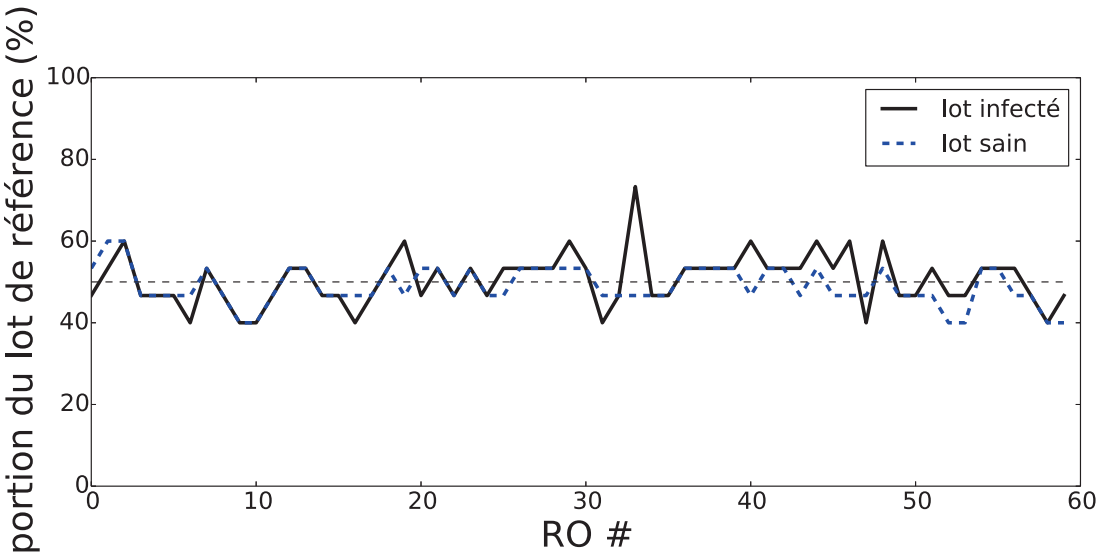

Figure 4.15: Répartition des lots après séparation par la médiane

référence. Dans le cas de lots sains, on attend une valeur de $50\,\%$. Dans les deux cas, on observe des valeurs restant proches de $50\,\%$ pour les lots sains. Dans les cas de lots infectés, on observe également des valeurs autour de $50\,\%$ sauf pour le RO 33 où la valeur atteint $70\,\%$ et $75\,\%$.

Ces résultats montrent donc qu'une méthode de partitionnement permet de mettre en valeur l'impact d'un CTM. Cependant, avec ces méthodes, le rapport entre une valeur impactée par un CTM et une valeur non impactée est plus faible que dans le méthode précédente.

Les résultats des précédentes sections valident la méthode de détection embarquée proposée et donc le modèle des variations des performances des structures CMOS dans un design réel, variations qui sont fortement dépendantes de la distribution d'alimentation dans les technologies CMOS avancées. Par la suite, on utilisera le t-test, car c'est le test faisant ressortir l'impact du CTM le plus clairement et qu'il admet un seuil critique, ce qui permet la prise de décision automatique.

4.3 Validation de la détection externe

Pour démontrer expérimentalement l'efficacité de notre méthode de détection dans le cas où des mesures par canaux auxiliaires sont considérées, un protocole similaire à celui utilisé pour des mesures embarquées a été suivi. Toutefois, contrairement au cas précédent, les mesures exploitées sont collectées lors du fonctionnement du circuit et plus précisément lors du chiffrement par le bloc AES. Pour chaque CI, 10000 courbes EM ont été acquises pour réduire le bruit de mesure par moyenne des traces. Un ensemble

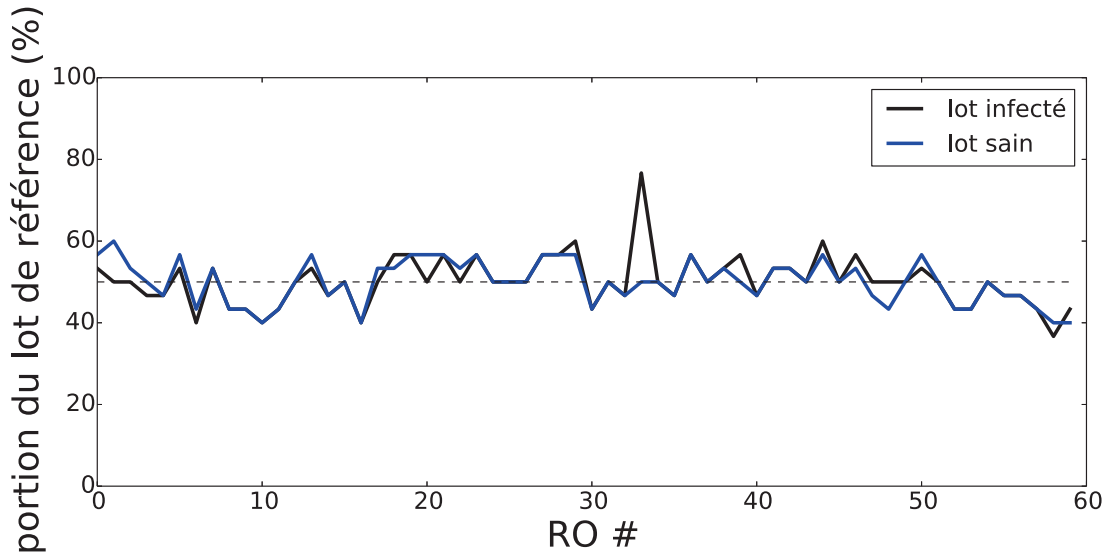

Figure 4.16: Répartition des lots après partitionnement par la méthode des k-moyennes.

de 10000 textes aléatoires a été généré et appliqué de façon identique pour la caractérisation de chaque circuit. Les courbes obtenues ont dû être resynchronisées avec celles d'un circuit choisi comme référence. À cette fin, les pics EM liés à l'exécution des rounds de l'AES sont considérés comme des références temporelles. Chaque acquisition dure $250\,\mathrm{ns}$, et les courbes obtenues sont composées de 10000 échantillons, ce qui nous donne un vecteur empreinte S^{design} composé de $q = 10000$ valeurs. Cette valeur est très supérieure à 60. Afin de valider ce protocole de mesure, nous avons, dans un premier temps, mesuré l'impact d'un CTM sur les émanations EM.

De façon similaire au cas de l'exploitation de capteurs embarqués, un LFSR de 64 bits a été utilisé pour émuler l'effet d'un CTM. Contrairement au cas précédent, nous ne cherchons pas à détecter l'impact passif et statique du CTM sur les mesures, mais son impact dynamique. En effet, cela est impossible avec des mesures EM qui révèlent les variations du courant circulant dans les CIs. Ce protocole a été conçu afin de détecter l'activité de commutation d'un CTM qui a un impact limité dans le temps. Le LFSR est connecté à l'horloge globale et synchronisé avec le 8-ème tour de l'AES. Le LFSR a donc une activité de commutation une fois par chiffrement. Contrairement à la méthode de détection utilisant un réseau de capteurs exploitant l'impact statique de l'infection, nous cherchons ici à détecter son impact dynamique. Ce CTM peut toujours être considéré comme un CTM séquentiel. Il a été placé volontairement dans l'AES, car cela est le cas le plus défavorable pour la détection à cause de l'influence de l'activité de commutation de l'AES. La figure 4.17 représente le placement du circuit.

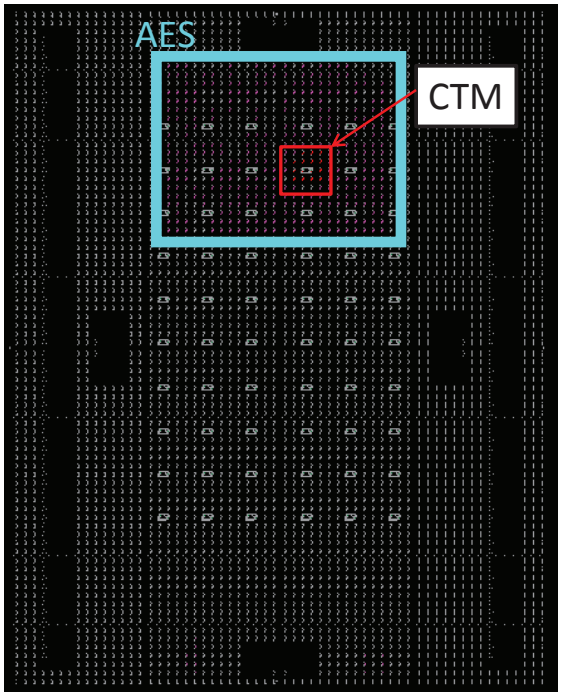

Figure 4.17: Layout pour analyse EM.

4.3.1 Impact du CTM en EM

La figure 4.18 représente les traces moyennes obtenues pour un circuit. La courbe bleue en pointillé correspond aux émanations du circuit sans CTM. La courbe noire correspond aux émanations du circuit avec CTM. On distingue clairement les 11 pics correspondant aux 11 tours de l'AES. Cependant, les deux courbes sont totalement similaires et l'on ne distingue pas visuellement l'impact du CTM.

Afin d'observer l'impact du CTM, on a utilisé le t-test sur une implémentation saine et une implémentation infectée par rapport à une implémentation de référence. La figure 4.19 montre les résultats de t-test. La courbe en pointillés bleue représente le t-test appliqué entre le lot sain et le lot de référence. La courbe en noire représente le t-test appliqué entre le lot infecté et le lot de référence. L'impact du CTM apparait clairement pour les échantillons $i \in \{6000, \ldots, 9000\}$. La figure 4.19 est une représentation spatiale de la valeur absolue maximale du t-test entre l'implémentation de référence et l'implémentation infectée. Par comparaison avec le placement-routage représenté sur la figure 4.17 on observe que le CTM à une influence principalement horizontale et verticale. Ces résultats semblent en accord avec ceux donnés dans [Sol+] et [Bal+15].

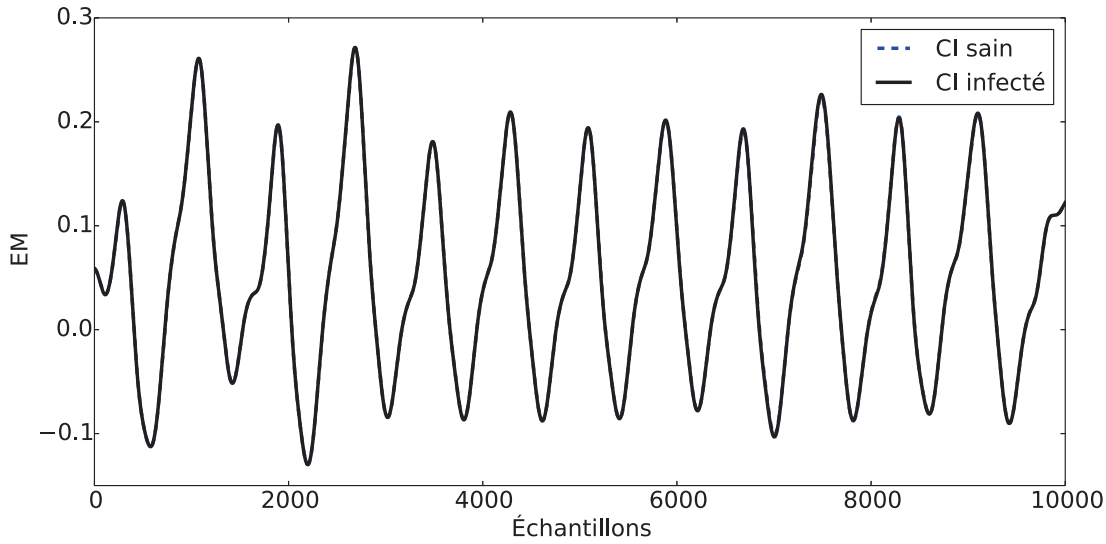

Figure 4.18: Traces EM provenant de deux implémentations.

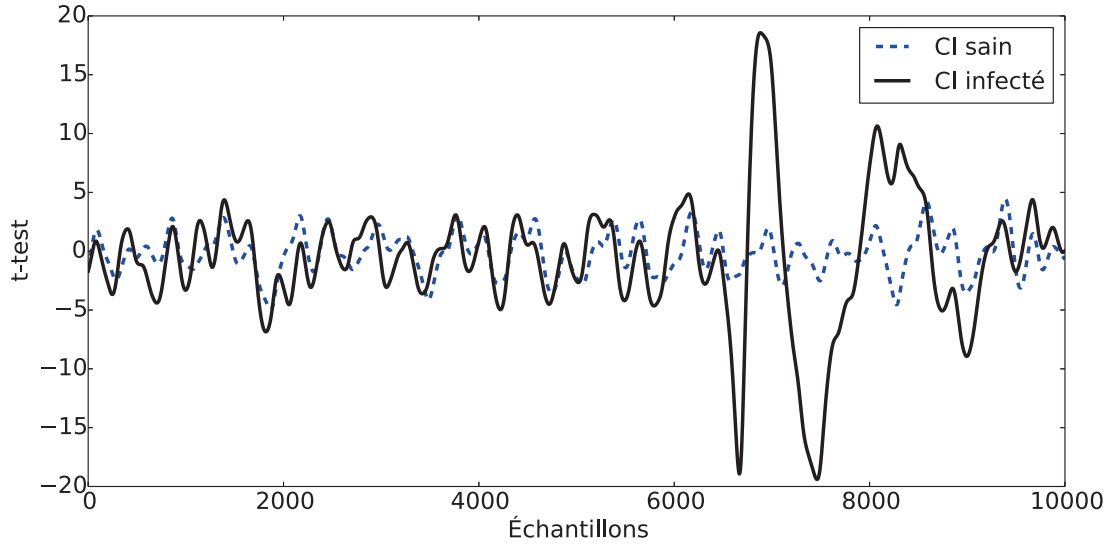

Figure 4.19: t-test appliquée sur les mesures d'un CI suivant deux implémentations.

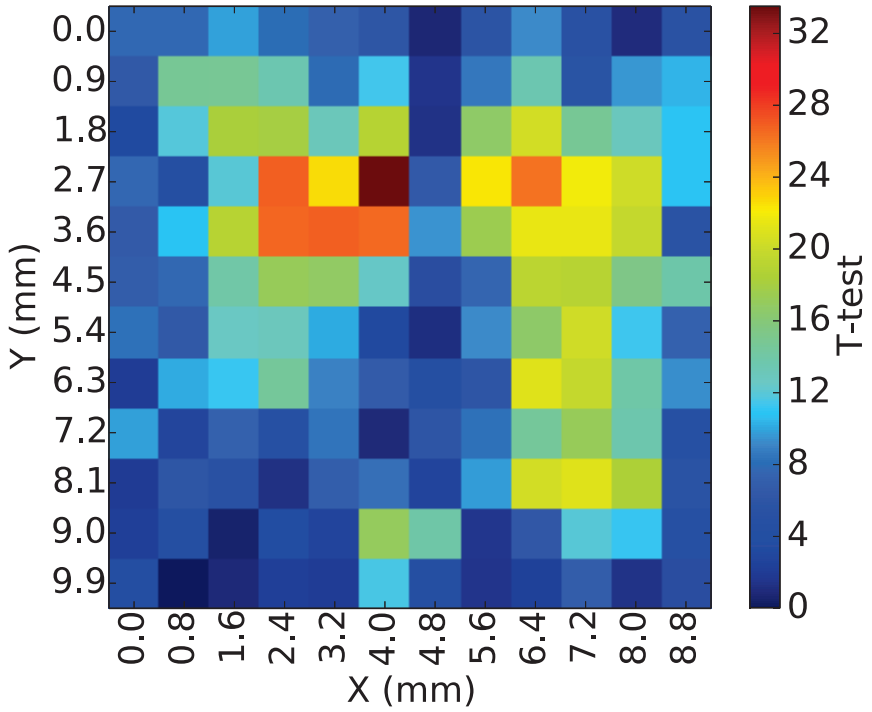

Figure 4.20: Carte des valeurs absolues de t-test obtenues par analyse EM.

4.3.2 Adaptation du distingueur

Dans le cas précédent (réseau de ROs), nous considérions que seul un capteur était impacté par le CTM. Cependant, plus la résolution temporelle ou spatiale de la technique est grande, plus le nombre de capteur ou d'échantillons affectés est élevé. Nous devons donc dans ce cas considérer l'exclusion de plusieurs échantillons à la place d'un seul pour le calcul de t_{critic}. Étant donné r le nombre d'échantillons exclus, les équations (3.17) et (3.18) sont modifiés comme suit:

$$\mathcal{W}_{(-k)} = \frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq \left[k-\frac{r}{2}, k+\frac{r}{2}\right]}}^{n} w_i \tag{4.1}$$

$$\sigma_{(-k)} = \frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq \left[k-\frac{r}{2}, k+\frac{r}{2}\right]}}^{n} (w_i - \mathcal{W}_{(-k)})^2 \tag{4.2}$$

Pour déterminer la taille r de la fenêtre d'exclusion, deux paramètres doivent être pris en compte. Premièrement, l'impact du CTM sur nos mesures en termes de surface (resp. durée) pour une analyse spatiale (resp. temporelle). Deuxièmement, la résolution (spatiale ou temporelle) du protocole de mesure. En prenant en compte ces deux

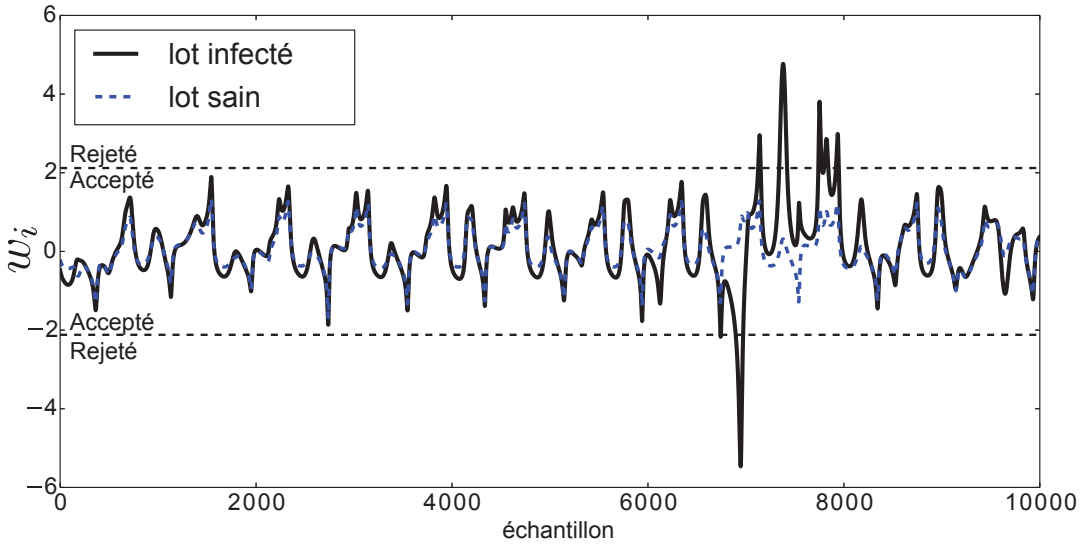

Figure 4.21: Valeurs de t-test obtenues lors de la comparaison de 15 CIs infectés et 15 CIs sains avec un lot de référence.

paramètres, nous pouvons déterminer le nombre maximum de valeurs dans le vecteur M et déterminer largeur de la fenêtre d'exclusion r.

Pour l'analyse EM considérée, la durée de l'activité du CTM est d'un cycle d'horloge soit $20\,\mathrm{ns}$. Le taux d'échantillonnage de l'oscilloscope est $40\,\mathrm{GS/s}$, ce qui correspond à une résolution de $25\,\mathrm{ps}$. En prenant en compte ces paramètres, la fenêtre d'exclusion r peut être configurée à $\frac{20ns}{25ps/S} = 800$ échantillons. Cependant, l'impact de l'activité de commutation d'un LFSR pendant une période horloge, peut durer plus de deux cycles d'horloge avant de disparaitre. Par conséquent, une fenêtre de $50\,\mathrm{ns}$ est choisie afin de s'assurer que la totalité de l'impact du LFSR y est bien exclue du calcul de t_crtic. Donc, $r = \frac{50ns}{25ps/S} = 2000$ échantillons. Il doit être noté que choisir une fenêtre trop fine peut mener à augmenter t_{critic}, et par conséquent à augmenter le taux de faux négatifs. Alors que prendre une fenêtre trop large n'a pas d'influence sur t_{critic} tant que le nombre d'échantillons restant permet une estimation précise de la moyenne et de l'écart type mis en jeu dans l'équation 4.1.

4.3.3 Résultats

La figure 4.21 reporte les valeurs de t-test obtenues à partir de mesures EMs. La courbe bleue en pointillés représente le vecteur W obtenu par comparaison du lot de référence avec un lot sain. La courbe noire correspond au vecteur W obtenu en comparant le lot de référence à un lot infecté. Les deux courbes sont superposées sur une large part du vecteur. Cependant de l'échantillon 6300 à l'échantillon 8200, les deux courbes sont significativement différentes. Ceci révèle l'activité du CTM. Par conséquent, une fenêtre d'exclusion de 2000 échantillons semble pertinente pour détecter une implémentation de

LFSR avec une activité qui dure un coup d'horloge. La courbe en pointillé correspond aux valeurs de t_{critic}. Elles ont été calculées avec cette largeur de fenêtre en suivant l'équation 4.1. Cela montre que la valeur de t_{critic} n'est pas affectée par le CTM, mais seulement par les variations des procédés de fabrication et le bruit de mesure.

Ces résultats montrent que la méthodologie introduite dans le chapitre 3 est applicable pour des mesures externes, en particulier la mesure du champ électromagnétique. Après avoir validé les principes présentés dans le chapitre précédent. Les sections suivantes caractérisent plus en profondeur les limites de cette méthode.

4.4 Taux de succès

Plusieurs CIs sont nécessaires pour appliquer la méthode proposée (1 lot de référence et 1 lot testé). Son efficacité est fortement dépendante du nombre de CIs dans chaque lot. Pour permettre une analyse facilitée de l'efficacité de notre méthode de détection et d'autres méthodes, nous introduisons ici l'idée de taux de succès (SR: «Succes Rate»). Le SR est défini par rapport au nombre de CIs contenus dans chaque lot. Plus précisément, le SR est le pourcentage de CIs infectés dans un lot classé comme infecté (vrai positifs) moins le pourcentage de CIs sains classés comme infecté (faux positifs).

$$SR = \frac{nb\ de\ CIs\ infectés\ classés\ infectés}{nb\ de\ CIs\ infectés} - \frac{nb\ de\ CIs\ sains\ classés\ infectés}{nb\ de\ CIs\ sains} \quad (4.3)$$

Durant notre analyse d'efficacité, nous considérons que le lot de référence et le lot testé ont la même taille. 19 cartes FPGA ont été utilisées pour créer les deux lots. Pour calculer le SR associé à une taille de lot donnée, noté l, l cartes sont aléatoirement tirées comme cartes de références et l cartes sont aléatoirement tirées comme cartes testées. Le tirage est effectué de façon à minimiser l'intersection entre les deux ensembles de cartes. Il n'y a donc pas d'intersection pour $l < 10$, et les ensembles de CIs ne sont pas complètement indépendants pour $l \geq 10$. Par conséquent, l'impact des variations des procédés sur le résultat décroit plus vite avec la taille de lot que si deux lots complètement indépendants avaient été utilisés.

Pour chaque tirage, le t-test est appliqué deux fois entre les deux lots selon deux cas considérés. Dans le premier cas, le lot de référence et le lot testé sont implémentés avec le même design sain (sans LFSR/CTM). Ce cas permet de calculer le taux de faux positifs. Dans le second cas, le lot de référence est implémenté avec le design sain et le lot testé est implémenté avec le design infecté. Cela permet de calculer les taux de vrais positifs. 500 tirages sont effectués pour chaque taille de lot.

4.4.1 Taux de succès avec des capteurs embarqués

La figure 4.22 montre l'évolution du SR en fonction de la taille de lot, pour un CTM de 64 bits. La courbe noire en pointillés représente le SR obtenu avec le t-test avec une p-value choisie à la valeur classique $5\,\%$. Les autres courbes représentent le SR obtenue avec différentes valeurs de δ utilisées pour déterminer la valeur t_{critic} considérée pour le distingueur adaptatif que nous proposons.

Comme on peut l'observer, les résultats changent significativement avec la valeur δ. Si la valeur est trop basse, le taux de faux positifs (le second terme dans l'équation 4.3) augmente et le SR n'atteint pas les $100\,\%$ (avec $\delta = 3$, il n'excède jamais $80\,\%$), car certaines fluctuations induites par les variations des procédés de fabrication ou le bruit de mesure conduisent certaines valeurs de t-test à dépasser t_{critic}. Dans le cas contraire, si la valeur δ est trop élevée, le distingueur accepte la plupart des CIs infectés comme sains. Il en résulte que le taux de vrais positifs (le premier terme de l'équation 4.3) décroit vers $0\,\%$. Dans notre exemple, c.-à-d. la figure 4.22, cette valeur n'est pas atteinte, car le SR atteint les $100\,\%$ avec un le seuil le plus haut $\delta = 6$.

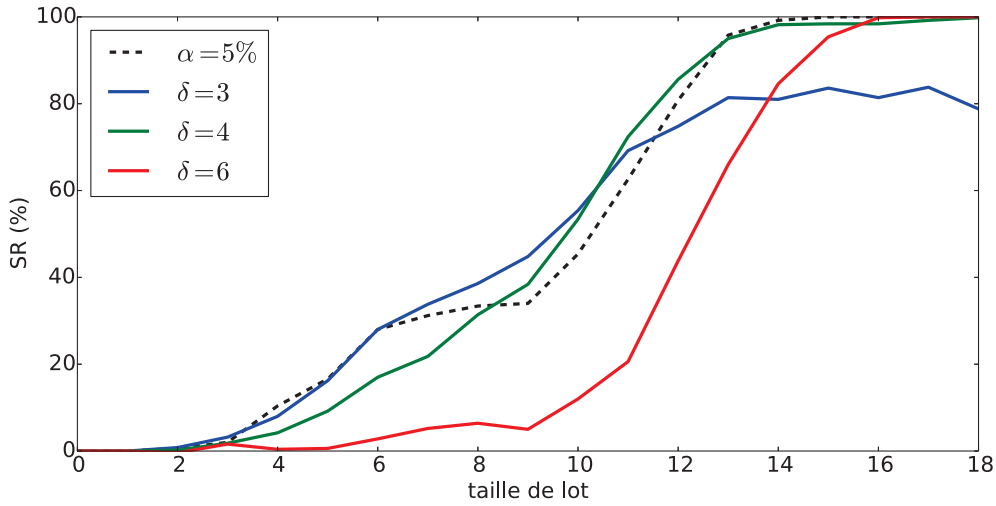

Figure 4.22: SR obtenu avec l'approche capteurs embarqués (approche de détection spatiale): cas de CTM de 64 bits

Nous avons analysé l'impact de la taille des CTMs (en bits) sur l'évolution du SR. Les figures 4.22, 4.23, 4.24 et 4.25 représentent les SRs obtenues pour des infections correspondant à des LFSRs de 64, 32, 16, 8 bits respectivement. Comme attendu, il apparait que le SR décroit avec la taille du CTM. Pour le plus petit CTM, le SR n'atteint pas les $100\,\%$. Plus particulièrement, l'observation des figures révèle un décalage de la courbe de SR vers la droite à mesure que la taille des CTMs considérés décroit. Cela signifie que la taille de lot nécessaire pour un obtenir un haut SR augmente lorsque la taille des CTMs diminue.

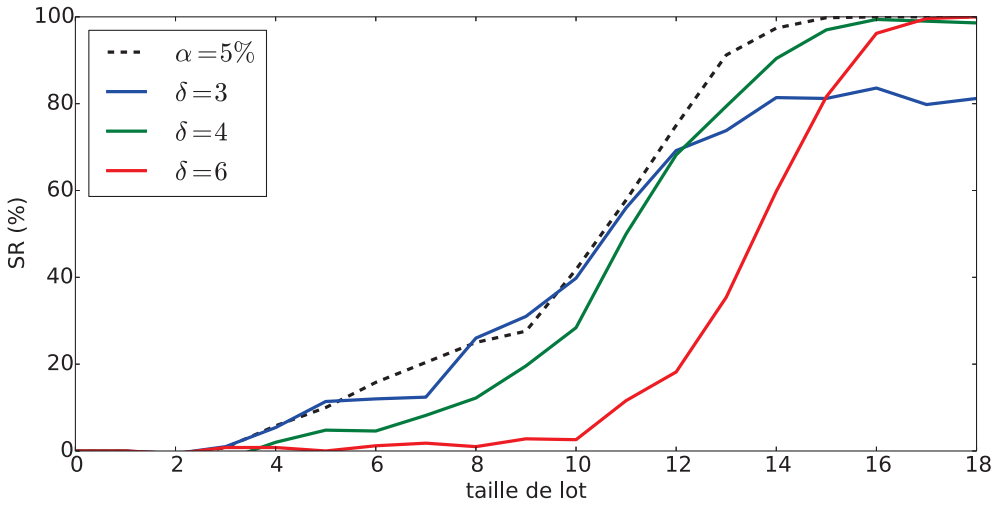

Figure 4.23: SR obtenu avec l'approche capteurs embarqués (approche de détection spatiale): cas de CTM de 32 bits

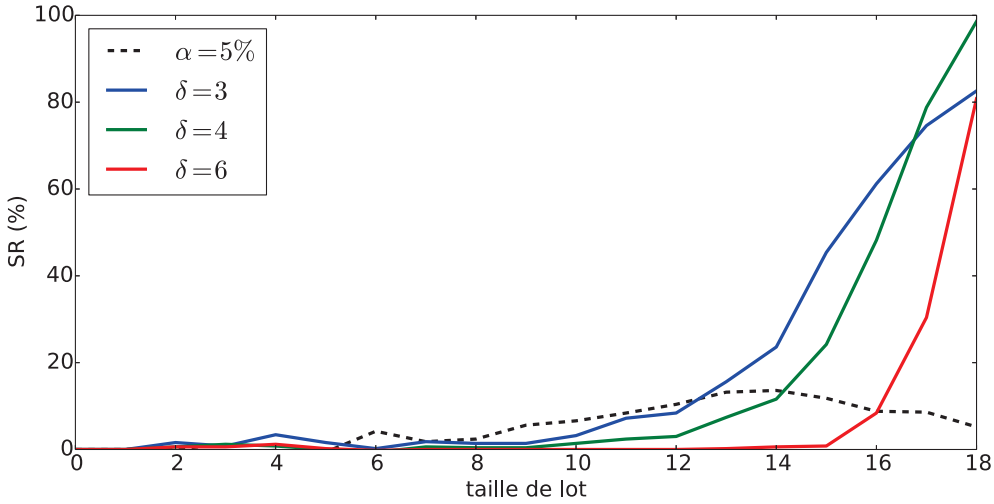

Figure 4.24: SR obtenu avec l'approche capteurs embarqués (approche de détection spatiale): cas de CTM de 16 bits

Pour les tailles de CTM de 8 et 16 bits, nous n'avons pas de lots suffisamment grands pour obtenir un taux de succès de $100\,\%$. Mais comme les courbes correspondant aux infections de 8 et 16 bits et les courbes correspondant aux infections 32 et 64 bits ont la même forme, il semble qu'un ensemble de CIs plus grand permettrait de distinguer les infections au travers des variations de procédés avec un plus grand SR.

De plus, on peut observer que le t-test classique fournit des valeurs de SR similaires à celles obtenues avec à notre méthode adaptative pour des infections de 64 et 32 bits. Cependant, le SR n'atteint pas $20\,\%$ pour des infections de 16 et 8 bits en utilisant cette approche classique alors que c'est le cas avec notre approche adaptative. Cela montre que notre distingueur est pertinent pour de grandes et petites (furtives) infections.

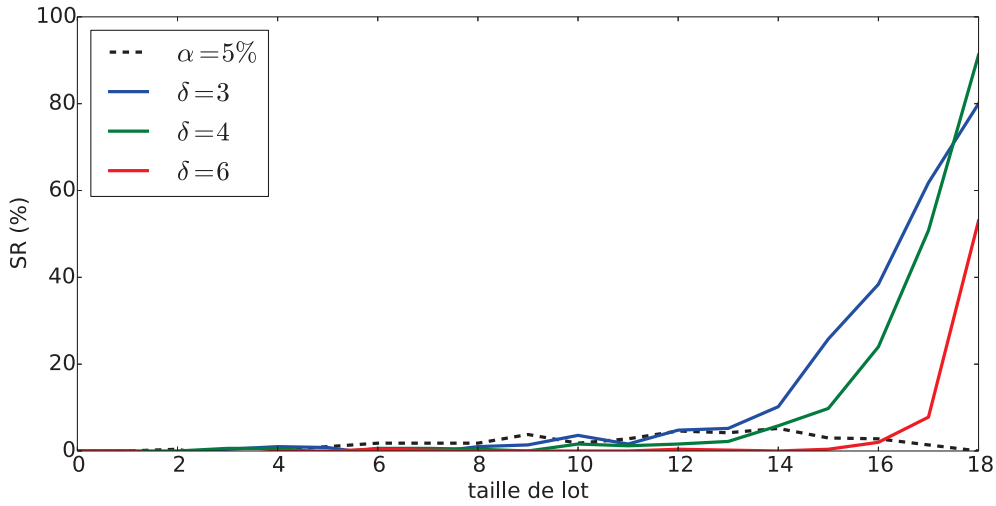

Figure 4.25: SR obtenu avec l'approche capteurs embarqués (approche de détection spatiale): cas de CTM de 8 bits

Finalement, en considérant l'impact de seuil δ, $\delta = 4$ apparait être un bon choix pour optimiser les taux de faux positifs et de faux négatifs avec une taille de lot minimale.

4.4.2 Taux de succès avec une analyse EM

La figure 4.26 montre l'évolution du SR en fonction de la taille du lot considéré quand une analyse EM est préférée à l'approche embarquée. Chaque courbe représente le SR obtenu avec différentes valeurs de seuil δ. La courbe en pointillés représente le SR obtenu avec le t-test classique en choisissant un indice de confiance (p-value) α égal à $5\,\%$. De la même façon qu'auparavant, il a été observé que le choix de seuil fixe n'est pas idéal, car les évolutions des taux de faux positifs et faux négatifs influent fortement sur le SR.

Cependant, ces résultats sont significativement différents de ceux obtenus avec des capteurs embarqués. En effet, on peut observer qu'avec $\delta = 3$, le SR n'excède pas $50\,\%$. Cela montre un impact important des faux positifs. Cela peut être dû à un bruit de mesure plus élevé ou une synchronisation de piètre qualité. Cette figure montre également que $\delta = 6$ semble être pertinent pour ce type de CTM, car il offre le meilleur SR pour chaque taille de lot. On observe aussi que le t-test classique ($\alpha = 5\%$) permet d'obtenir un SR similaire à celui obtenu avec $\delta = 6$ et que les deux tests permettent d'atteindre un taux de succès de $98\,\%$, pour les tailles de lots disponibles, et ce pour une infection de 64 bits. Cependant, comme nous ne sommes pas parvenus à atteindre un taux de succès suffisamment élevé pour des CTMs plus petits, il apparait que l'analyse EM semble moins efficace qu'une méthode de détection basée sur des capteurs embarqués. Néanmoins, ce résultat confirme que le t-test adaptatif est également efficace quand il est appliqué à des mesures externes en particulier une analyse EM.

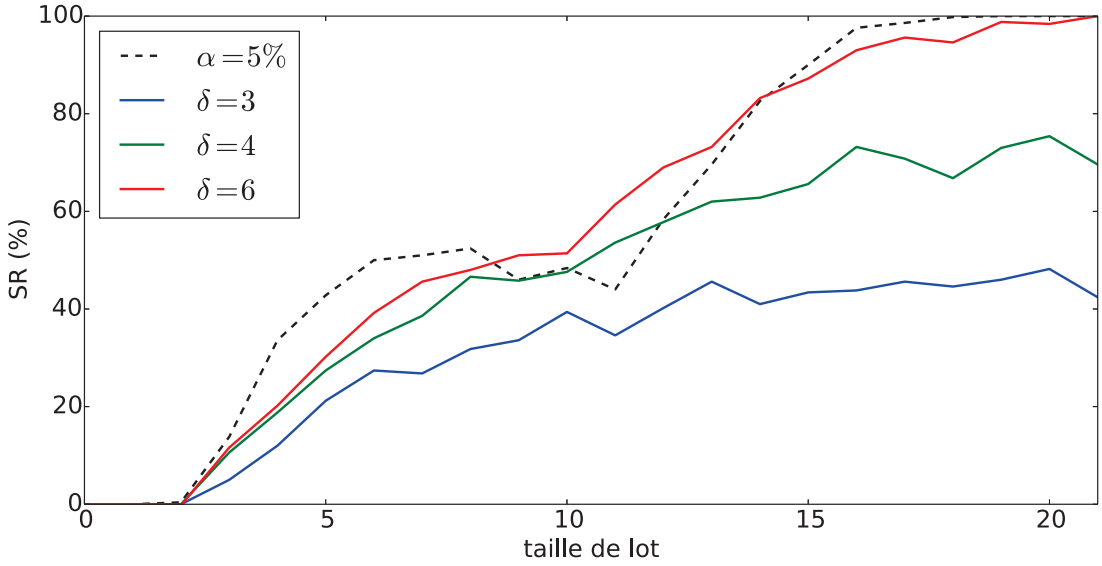

Figure 4.26: SR obtenue par une analyse EM (méthode de détection dans le domaine temporel): cas CTM 64 bits

4.5 Caractérisation

Dans cette section nous caractérisons la sensibilité des capteurs en fonction de la distance et de la taille du CTM. De plus, l'impact de la logique environnant l'infection sur la sensibilité des capteurs est aussi analysé.

4.5.1 Impact de la taille et de la distance des CTMs

Dans le but de déterminer la densité de capteurs nécessaire à un fort taux de couverture de la surface d'un CI, l'impact sur les capacités de détections de la distance séparant le RO du CTM a été étudié ainsi que l'impact de la taille (en bits) de l'infection. Nous avons analysé ces deux paramètres en implémentant plusieurs placement-routages avec différentes positions de LFSR. Pour chaque emplacement, quatre tailles différentes de LFSR ont été considérées: 8, 16, 32 et 64 bits. Le premier emplacement de LFSR considéré est celui d'un RO, la distance mesurée en slices est donc égale à zéro. Les autres emplacements sont notés $x-2$ et $x-4$ (resp. $y+2$ et $y+4$). Par exemple $x-2$ signifie que le LFSR est décalé négativement de deux slices selon l'axe X par rapport au RO lorsque $y+4$ signifie que le LFSR est décalé positivement de 4 slices selon l'axe Y.

Les résultats de détection obtenus avec le t-test sont donnés dans le tableau 4.2 pour un LFSR placé près du RO 48. Dans ce tableau t_{max} est la valeur maximale du t-test (en valeur absolue) obtenu avec les 60 ROs, t_{argmax} est l'index du RO pour lequel la valeur t_{max} est obtenue, et t_{critic} est la valeur considérée pour le t-test adaptatif (équation 3.19).

Tableau 4.2: Valeurs de t-test en fonction de la taille du LFSR et de la distance

Taille	8 bits			16 bits			32 bits			64 bits		
	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}
0, 0	0,08	1.26	48	0,11	1,6	48	0,08	3,85	48	0,08	4,79	48
x-2	0,11	0,62	48	0,13	0,09	6	0,14	1,06	48	0,07	2,59	48
x-4	0,10	0,07	6	0,10	0,11	26	0,09	2,01	48	0,13	2,58	48
y+2	0,13	0,08	6	0,12	0,6	48	0,08	3,01	48	0,08	2,56	48
y+4	0,10	0,38	48	0,11	0,65	48	0,08	2,92	48	0,12	2,86	48

Comme on peut l'observer, les valeurs de t-test obtenues décroissent très vite avec la taille du CTM considéré. Plus précisément, on peut observer qu'aussitôt la taille du CTM inférieure à 32 bits et le CTM placé loin du RO48 (RO le plus proche de l'infection), il devient plus difficile de le détecter. Dans ces conditions, t_{max} devient inférieur à t_{critic} et n'est pas obtenue pour le RO 48. Cependant, on peut observer que des petits CTMs sont toujours correctement détectés lorsqu'ils sont près du RO 48. En considérant ces résultats, nous pouvons conclure que la portée de détection de chaque RO est trop faible et qu'un nombre significatif de capteurs est nécessaire pour couvrir l'intégralité de la surface du CI. Cela pourrait ne pas être efficace selon les contraintes de conception.

4.5.2 Impact de la logique environnante

Tableau 4.3: t-test en fonction de la taille et la position du LFSR pour des ROs *compacts*

Taille	8 bits			16 bits			32 bits			64 bits		
	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}
en-dehors	0,08	1,26	48	0,11	1,6	48	0,08	3,85	48	0,08	4,79	48
au voisinage	0,10	3,4	19	0,15	3,52	19	0,23	2,46	19	0,06	2,88	19
dans	0,08	1,52	9	0,06	6,7	9	0,09	5,2	9	0,08	3,94	9

À ce stade, les expérimentations ont été effectuées avec une matrice de capteurs et des CTMs (LFSR) implémentés hors du bloc cible (AES), c.-à-d. qu'il n'y avait pas de logique implémentée autour du LFSR. On peut se demander si les capacités de détection de notre matrice de capteurs est affectée par l'influence statique de la logique environnante. Pour évaluer cette influence, le tableau 4.3 montre les valeurs de t-test obtenues pour un RO *compact* au plus proche de l'infection en fonction de la position relative de l'AES et de l'infection (en-dehors, au voisinage et dans l'AES). Pour ces trois positions, l'impact de la taille du CTM en nombre de bits a également été étudié. Comme on peut l'observer, dans tous les cas, la valeur de t-test est plus grande que t_{critic}, et donc le CTM est détecté. À partir de ces résultats, on peut supposer que la logique environnante n'a pas d'impact significatif sur l'efficacité de notre méthode de détection. En effet, pour chaque emplacement le placement du CTM peut ne pas être identique, ce qui peut impacter les résultats. Nous pouvons donc conclure que nous sommes en mesure de détecter la plus petite infection même si cette dernière est entourée de la logique de l'AES.

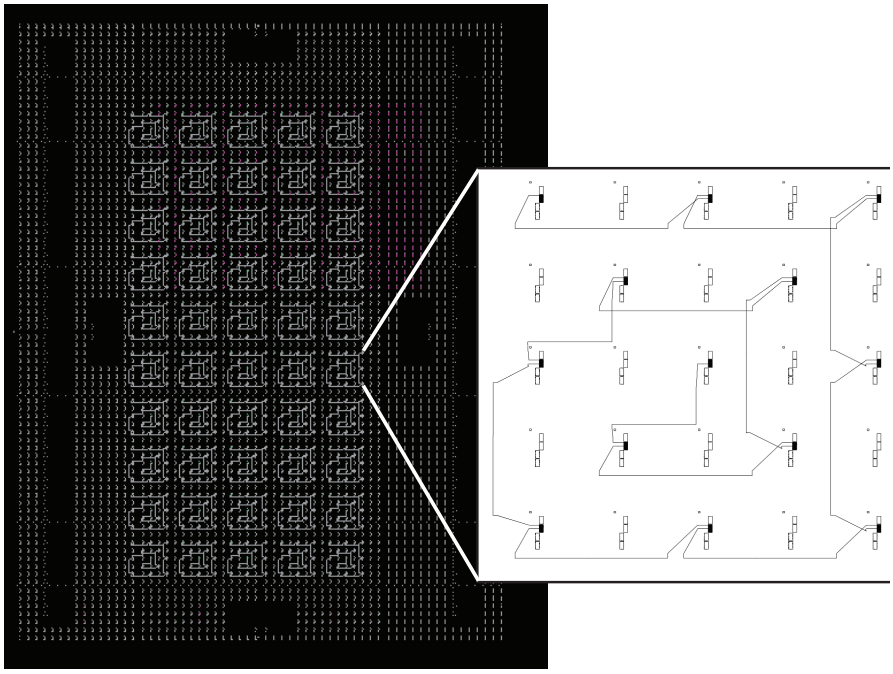

Figure 4.27: Implémentation du RO étendu.

4.6 Amélioration des ROs

Nous avons vu dans la section 4.5.1 que la granularité nécessaire pour couvrir la surface du circuit pouvait être importante avec l'implémentation du capteur proposé. En effet, les expérimentations précédentes, effectuées avec un RO *compact* de 5 étages, ont montrées que les ROs proposés ne permettent de détecter les impacts passifs seulement dans leur environnement immédiat. La portée de détection de ce type de RO était donc insuffisante pour détecter des infections petites et furtives. Pour améliorer la sensibilité spatiale de nos capteurs, nous proposons une nouvelle implémentation de RO.

4.6.1 Implémentation

L'idée pour améliorer la portée est d'étendre sa surface. La capacité de détection de ROs *étendus*, c.-à-d. couvrant chacun une surface plus importante du CI, a donc été analysée. Les ROs *étendus* comprennent 13 étages (à la place de 5), et sont composés d'une porte NAND et de 12 inverseurs, uniformément répartis dans un carré de 9 slices au lieu d'un rectangle de 2 par 3 pour les ROs *compacts*. Ces ROs *étendus* ont une période moyenne de $21\,\mathrm{ns}$ quand un RO *compact* possède une période de $13\,\mathrm{ns}$. La figure 4.27 représente les géométries d'implémentation du RO *étendu* ainsi que son intégration sur le circuit.

4.6.2 Impacts de la taille et de la distance des CTMs

Les résultats donnés dans cette section sont calculés à partir de 20 cartes. Ils sont obtenus avec exactement le même protocole que celui précédent à l'exception que seuls 50 ROs ont été implémentés à la place de 60. La surface supplémentaire induite par l'addition du matériel de détection embarqué est de $4,7\%$ des ressources du FPGA.

Tableau 4.4: t-statistique en fonction de la taille du LFSR et de sa distance au RO *étendu*

Taille	8 bits			16 bits			32 bits			64 bits		
	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}
0,0	0,21	0,55	35	0,47	0,87	35	0,48	1,49	35	0,39	4,79	35
x-2	0,17	0,45	35	0,47	0,67	35	0,44	1,70	35	0,39	3,86	35
x-4	0,11	0,90	35	0,13	0,78	35	1,16	2,27	35	0,4	3,41	35
y+2	0,14	0,34	35	0,46	0,90	35	0,43	2,06	35	0,42	4,41	35
y+4	0,15	0,72	35	1,18	1,57	35	0,44	1,94	35	0,71	5,02	35

Le tableau 4.4 montre les résultats de détection obtenus avec les ROs *étendus* en prenant en compte différentes distances entre le CTM et les ROs. En comparant ce tableau avec le tableau 4.2, on peut observer que malgré des valeurs de t-test plus faibles que celles obtenues avec les ROs *compacts*, les ROs *étendus* ont une meilleure portée de détection. En effet, alors que les cas (8 bits: x-4, y+2) et (16 bits, x-2, x-4, y+4) n'étaient pas détectés en utilisant les ROs *compacts*, ils sont maintenant détectés en utilisant les ROs *étendus*. Il semble donc que les placement-routages des ROs à une grande influence sur la couverture spatiale pour une densité de capteurs donnée.

4.7 Conclusion

Ce chapitre décrit les expérimentations servant à valider et caractériser les points exposés dans le chapitre 3. Pour cela, de nouvelles implémentations et de nouveaux CTMs ont été développés pour FPGA. Cela a demandé une phase d'automatisation permettant de créer différents placement-routages pour chaque paramètre testé tout en conservant un environnement de mesure inchangé. Cela permet de conserver le caractère furtif des infections émulées. De plus, plusieurs placement-routages ont été implémentés pour simuler des contrefaçons de type copies fonctionnelles.

À partir de là, les propositions du chapitre précédent ont pu être validées. Dans un premier temps, le modèle de variation des performances des structures CMOS a été confirmé. Puis la méthode de prise d'empreinte de lot indépendante des variations de procédé a été expérimentalement validée. En effet, nous avons confirmé que des empreintes de placement-routages identique sont similaires lorsque des empreintes provenant de placement-routages différents sont elles même différentes. Dans un second temps, les méthodes de détection de contrefaçon ont été vérifiées. Cela a été fait en utilisant un placement-routage simulant une contrefaçon pour la première méthode et

en utilisant un LFSR 64-bits pour émuler le CTM pour la deuxième méthode. Nous avons observé un fort impact du LFSR 64 bits sur notre signature pour le RO le plus proche de l'infection, cet impact est plus de trois fois supérieur aux valeurs non affectées.

Ensuite, les différents outils servant à mettre en valeur l'influence d'un CTM ont été expérimentalement évalués. Cela nous a amenés à sélectionner le t-test comme outil de prise de décision. La méthode de détection de CTM utilisant le t-test a été validée pour des mesures externes du champ électromagnétique. En effet, le LFSR de 64 bits impact notre empreinte de façon à obtenir une valeur de t-test supérieure à 3,8 pour un lot infecté alors que les valeurs reste inférieures à 1,5 pour un lot non infecté. Afin de diminuer le taux de faux positif lié au t-test, une amélioration de ce dernier a été proposée.

Pour tester l'efficacité du t-test et l'approche par lot dans les cas de mesures externes et internes, une métrique de taux de succès a été proposée. Elle a ensuite été utilisée pour analyser différentes tailles de CTM et différentes tailles de lot de CIs. En considérant les tailles de lot disponible, les taux de succès atteignent $100\,\%$ pour des infections de 64 et 32 bits. Cependant, les comportements observés mènent à penser que plus de CIs permettraient d'atteindre un taux de succès de $100\,\%$ avec des infections plus petites.

Enfin, une caractérisation de la granularité de détection offerte par notre réseau de capteurs et une analyse de l'impact de la logique environnante sur les capacités de détection ont été effectuées. Nous avons observé que le LFSR de 8 bits n'est pas toujours détecté lorsqu'il est situé à 4 slices de distance du RO le plus proche. Cependant, tous les emplacements de CTMs testés (relatifs à l'AES) on permit de détecter un CTM de 8 bits. Nous en avons donc déduit qu'une grande densité de capteur est nécessaire pour couvrir la surface du circuit, mais que la logique environnante n'a pas d'impact significatif sur la sensibilité du système. Cette portée limitée nous a amenés à proposer une nouvelle implémentation de capteurs dont nous avons montré qu'elle permettait d'augmenter la couverture spatiale. En effet, contrairement à la première implémentation cette-ci permet de détecter les CTMs de 8 bits à une distance de 4 slices. Cette approche semble donc pertinente pour atteindre une bonne couverture spatiale.

Conclusion et perspectives

Les premiers travaux cherchant à détecter l'impact physique d'un cheval de Troie matériel (CTM) datent de 2007. Le constat que l'on peut faire à ce jour est qu'aucune méthode n'atteint un taux de détection de $100\,\%$. De façon générale les méthodes proposées ne sont pas entièrement fiables et déployables industriellement. Nous déduisons à partir de ces constatations qu'une solution s'inscrivant dans un contexte industriel et ayant un taux de détection avoisinant les $100\,\%$ est nécessaire. Cela implique un test ayant un coût et une durée les plus réduits possible et qui puisse être ajoutés dans les processus de conception et de production actuels. Avec ces principes comme objectifs, le but de cette thèse est de proposer une méthode de détection de CTM et de contrefaçon. Cette méthode se veut embarquée, ce qui implique une structure de mesure intégrée et des traitements pouvant être implémentés sur une surface limitée.

Dans le premier chapitre, le domaine dans lequel s'inscrivent ces travaux est expliqué. La composition des circuits intégrés, les procédés de fabrication et les attaques physiques sur ces composants sont introduits. Les problèmes d'intégrité des circuits intégrés et leurs causes sont développés pour expliciter les risques associés. Cela nous amène à conclure qu'il est nécessaire de développer des solutions de détection de CTM et de contrefaçon en considérant les modes de production et de distribution adoptés par l'industrie. Par rapport à ce besoin, nous parcourons les solutions proposées dans l'état de l'art. Ce dernier met en avant la comparaison de grandeurs physiques entre des circuits de référence et des circuits testés pour la vérification d'intégrité. De plus, il montre que les solutions offrant une prise locale d'information sont les plus efficaces et que l'utilisation de capteurs embarqués, à cet effet, est pertinente. Des oscillateurs en anneau sont proposés dans plusieurs solutions en tant que capteurs embarqués. Nous choisissons donc de développer une méthode de détection basée sur un réseau de capteurs intégrés. Les oscillateurs en anneaux ayant montré leur efficacité, ils sont utilisés comme capteurs dans nos expérimentations. De plus, l'état de l'art, met en avant la principale difficulté lors de la vérification d'intégrité, les variations des procédés de fabrication. En effet, celles-ci créent dans les mesures des variations aléatoires difficiles à distinguer, leur impact pouvant être plus important que celui d'une altération du circuit. Dans le modèle

de variation utilisé dans l'état de l'art des variations allant jusqu'à $30\,\%$ sont considérées. Ces variations imposent des limites à la fiabilité des schémas de détection proposés.

Dans le chapitre 2, nous caractérisons les effets induits par l'implantation d'un CTM sur un circuit. Pour cela, nous avons utilisé des circuits reprogrammables de type FPGA. Nous développons un circuit de test pourvu d'une matrice d'oscillateurs en anneau nous permettant d'obtenir une cartographie de la tension d'alimentation sur la surface du circuit. Pour émuler la présence et l'activité d'un CTM, nous utilisons un LFSR. Celui-ci permet d'obtenir l'activité de commutation électrique souhaitée imitant l'activité d'un CTM. Ces expérimentations révèlent en valeur deux phénomènes principaux. Premièrement, l'activité de commutation électrique du CTM crée un impact dynamique. Celui-ci est très inférieur aux variations des procédés de fabrication et impact localement plusieurs capteurs. On observe un impact dynamique maximal, sur la période des ROs, de $10\,\mathrm{ps}$ pour des variations dues aux variations de procédé pouvant atteindre $500\,\mathrm{ps}$. Deuxièmement, nous observons un impact statique lié à l'ajout du LFSR sur le circuit. Celui-ci est plus de 10 fois supérieur à l'impact dynamique (pouvant atteindre $140\,\mathrm{ps}$) et est très local. Nous avons également observé la forte influence des modifications de placement-routages sur les valeurs des capteurs. L'implémentation d'un AES a un impact global équivalent à 3 fois l'impact du CTM et peut atteindre localement 6 fois cette valeur. Nous en en déduisons qu'une modification de la structure physique des circuits intégrés impact les valeurs retournées par notre structure de mesure.

Dans le chapitre 3, à partir des conclusions du chapitre 2, nous élaborons notre méthode de détection de CTM et de contrefaçon. Cette méthode est basée sur un nouveau modèle de variation des performances des structures CMOS. De plus, nous posons comme nouveau paradigme d'infection, que l'infection se fait à l'échelle d'un lot. Nous cherchons à déterminer l'infection d'un lot en étudiant une méthode d'extraction d'une empreinte caractéristique de ce lot. Cette empreinte est indépendante des variations des procédés de fabrication et est reliée à la structure physique des circuits constituant le lot. Notre méthode de vérification d'intégrité comprend deux cas. Dans le premier cas, la comparaison entre l'empreinte d'un lot de référence avec celle d'un lot testé est faite dans le but de détecter un CTM. Dans le deuxième cas, la comparaison entre l'empreinte d'un lot de référence et les mesures provenant d'un seul circuit est effectuée afin de détecter les contrefaçons. De plus, nous présentons un nouveau distingueur pour la prise de décision (circuit infecté ou sain). Celui-ci est basé sur le t-test. Il repose sur le principe qu'un CTM viable est nécessairement furtif, et par conséquent son impact temporel ou spatial est limité dans le temps et dans l'espace. Nous considérons que pour des mesures spatialement ou temporellement corrélées, la majorité d'entre elles correspondent au comportement d'un circuit sain. Cette méthode adaptative permet de réduire à son minimum le taux de faux positifs et réduire le risque de rejet de circuits sains.

Le chapitre 4 présente la validation expérimentale des principes énoncés dans le chapitre 3. Ainsi nous montrons la validité de notre modèle des variations des performances des structures CMOS, de la méthode d'extraction d'empreintes et des méthodes de détection de CTM et de contrefaçons. Ensuite, nous caractérisons les limites de détection de CTM. Pour cela, nous proposons une métrique de taux de succès en fonction de la taille des lots considérés. Une fois appliquée pour différentes tailles de CTM, nous observons que la taille de lot nécessaire pour détecter des CTMs augmente au fur et à mesure que la taille des CTMs diminue. En pratique, nous obtenons un taux de succès de $100\,\%$ pour des CTMs de 32 et 64 bits et la tendance observée indique que des tailles de lots plus importantes sont nécessaires pour détecter des infections plus réduites avec un taux de succès de $100\,\%$. Nous avons également caractérisé la granularité de capteurs nécessaire pour couvrir la surface du circuit. Nous démontrons que les oscillateurs en anneau implémentés ont une faible portée de détection, ce qui implique qu'un nombre important de capteurs est nécessaire pour couvrir le circuit. En effet, pour des infections de tailles inférieures à 16 bits, des CTMs situés à 4 slices de distance du RO le plus proche ne sont pas détectés. Suite à cela nous proposons et validons une nouvelle implémentation d'oscillateur en anneau permettant d'augmenter cette portée, et donc de réduire le nombre capteurs nécessaires. Cette nouvelle implémentation correspond à l'étalement des inverseurs constituant les oscillateurs en anneau afin que ceux-ci occupent une grande surface (une surface de 9x9 slices au lieu de 4x3 slices). Nos mesures montrent que ces capteurs ont une plus faible sensibilité, mais suffisante pour détecter les infections plus éloignées et détecter tous les cas d'infections considérés.

Il apparait donc que l'utilisation d'un réseau d'oscillateurs en anneau permet d'obtenir des informations relatives à la structure physique d'un lot de circuits et cela indépendamment des variations de procédé à condition que la taille du lot soit suffisante. Ces informations structurelles nous permettent de mettre en valeur une altération de circuit et donc de valider ou non leur intégrité. Les principaux résultats ont été soumis dans [Lec+16d]

Perspectives

Dans la perspective d'une utilisation industrielle de ce type de détection, plusieurs points restent à explorer. D'une part, les expériences présentées ont été effectuées sur FPGA. Il est donc nécessaire, préalablement à d'autres études, de confirmer les phénomènes observés dans le cadre de circuits dédiés (ASIC). Le développement d'un prototype ASIC, auquel nous participons, est en cour de développement. Celui-ci intègrera une structure de mesure et une structure émulant un CTM similaire à la nôtre. Ensuite, d'autres types de capteurs peuvent être recherchés afin d'augmenter la sensibilité du système. Cela permettrait de réduire le nombre de capteurs nécessaires pour assurer une large

couverture et de détecter des infections potentiellement plus petites. Cette étude doit être aussi être menée sur ASIC, car les possibilités des capteurs utilisables sur FPGA sont limitées. Les deux principales caractéristiques à considérer sont la sensibilité du capteur qui doit permettre de détecter les infections les plus petites (si elle est trop élevée elle sera d'autant plus sensible au bruit) et la portée de détection qui doit être suffisamment importante pour limiter le nombre de capteurs nécessaires pour couvrir la surface du circuit. Pour cela, l'intégration physique du capteur doit être considérée afin de déterminer à quels éléments du circuit il doit être connecté et sur quelle couche physique il doit être intégré. Enfin, des études doivent être menées sur l'insertion de ces capteurs dans le flot de conception et sur la réalisation des tests dans le flot de distribution et production. Les capteurs peuvent être intégrés de façon automatique ou manuelle à différents niveaux d'abstraction. De plus, les tests d'intégrité peuvent être effectués à différentes étapes, car notre méthode requiert peu d'infrastructure et de temps et s'adapte au bruit environnant.

Liste des figures

Liste des tableaux

Publications personnelles

[Lec+15] M. Lecomte, J. J. A. Fournier, and P. Maurine. « Thoroughly analyzing the use of ring oscillators for on-chip hardware trojan detection ». In: *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. 2015, pp. 1–6 (Cité à la page 59).

[Lec+16a] M. Lecomte, J. J. A. Fournier, and P. Maurine. « Granularity and detection capability of an adaptive embedded Hardware Trojan detection system ». In: *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2016, pp. 135–138 (Cité aux pages 87, 101).

[Lec+16b] M. Lecomte, J. J. A. Fournier, and P. Maurine. « On-chip fingerprinting of IC topology for integrity verification ». In: *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2016, pp. 133–138 (Cité aux pages 87, 101).

[Lec+16c] M. Lecomte, J. J. A. Fournier, and P. Maurine. « SYSTEM AND METHOD FOR SECURING AN ELECTRONIC CIRCUIT ». Pat. WO/2016/042144. patent: WO/2016/042144. 2016 (Cité à la page 59).

[Lec+16d] Maxime Lecomte, Jacques J A Fournier, and Philippe Maurine. « An on-chip technique to detect Hardware Trojans and assist counterfeit identification ». In: *IEEE transaction on VLSI* (2016). (Soumis) (Cité à la page 133).

Références

[Aar+10] Jim Aarestad, Dhruva Acharyya, Reza Rad, and Jim Plusquellic. « Detecting trojans through leakage current analysis using multiple supply pad IDDQs ». In: *IEEE Transactions on Information Forensics and Security* 5.4 (2010), pp. 893–904 (Cité aux pages 32, 37, 42).

[Abd+16] Karim M. Abdellatif, Christian Cornesse, Jacques Fournier, and Bruno Robisson. « New Partitioning Approach for Hardware Trojan Detection Using Side-Channel Measurements ». In: *Applied Reconfigurable Computing: 12th International Symposium, ARC 2016 Mangaratiba, RJ, Brazil, March 22–24, 2016 Proceedings*. Ed. by Vanderlei Bonato, Christos Bouganis, and Marek Gorgon. Cham: Springer International Publishing, 2016, pp. 171–182 (Cité aux pages 42, 44, 56).

[Ade08] Sally Adee. « The Hunt for the Kill Switch ». In: *IEEE Spectrum* 45 (2008), pp. 34–39 (Cité à la page 13).

[Aes] *Specification for the Advanced Encryption Standard (AES)*. Federal Information Processing Standards Publication 197. 2001 (Cité à la page 80).

[Ago+10] Michel Agoyan, Jean-Max Dutertre, David Naccache, Bruno Robisson, and Assia Tria. « When Clocks Fail: On Critical Paths and Clock Faults ». In: *Smart Card Research and Advanced Application: 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings*. Ed. by Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 182–193 (Cité à la page 8).

[Agr+07] Dakshi Agrawal, Selçuk Baktir, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar. « Trojan detection using IC fingerprinting ». In: *Proceedings - IEEE Symposium on Security and Privacy* (2007), pp. 296–310 (Cité à la page 31).

[An+14] Ting An, Hao Cai, and Lirida Alves De Barros Naviner. « Simulation study of aging in CMOS binary adders ». In: *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings* May 2014 (2014), pp. 51–55 (Cité à la page 20).

[Ba+15] Papa-Sidy Ba, Manikandan Palanichamy, Sophie Dupuis, et al. « Hardware Trojan prevention using layout-level design approach ». In: *2015 European Conference on Circuit Theory and Design (ECCTD)*. IEEE, 2015, pp. 1–4 (Cité aux pages 50, 51).

[Bal+15] J Balasch, B Gierlichs, and I Verbauwhede. « Electromagnetic Circuit Fingerprints for Hardware Trojan Detection ». In: *EMC 2015, IEEE* (2015) (Cité aux pages 38–40, 56, 60, 117).

[Bao+15] Chongxi Bao, Domenic Forte, and Ankur Srivastava. « Temperature Tracking: Toward Robust Run-Time Detection of Hardware Trojans ». In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34.10 (2015), pp. 1577–1585 (Cité à la page 33).

[Bar+10] A. Barenghi, G. M. Bertoni, L. Breveglieri, M. Pellicioli, and G. Pelosi. « Low voltage fault attacks to AES ». In: *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*. 2010, pp. 7–12 (Cité à la page 8).

[Bea+11] Mark Beaumont, Bradley Hopkins, and Tristan Newby. *Hardware Trojans – Prevention, Detection, Countermeasures*. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA547668. 2011 (Cité à la page 13).

[Bha+05] A. Bhavnagarwala, S. Kosonocky, C. Radens, et al. « Fluctuation limits amp; scaling opportunities for CMOS SRAM cells ». In: *IEDM 2005*. 2005, pp. 659–662 (Cité à la page 91).

[Bha+13] Shivam Bhasin, Jean Luc Danger, Sylvain Guilley, Xuan Thuy Ngo, and Laurent Sauvage. « Hardware trojan horses in cryptographic IP cores ». In: *Proceedings - 10th Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2013* Umr 5141 (2013), pp. 15–29 (Cité à la page 50).

[Bow+10] K. Bowman, C. Tokunaga, J. Tschanz, et al. « Dynamic variation monitor for measuring the impact of voltage droops on microprocessor clock frequency ». In: *CICC, 2010 IEEE*. 2010, pp. 1–4 (Cité à la page 43).

[Cao+14] Yuan Cao, Chip Hong Chang, and Shoushun Chen. « A cluster-based distributed active current sensing circuit for Hardware Trojan detection ». In: *IEEE Transactions on Information Forensics and Security* 9.12 (2014), pp. 2220–2231 (Cité aux pages 42, 43).

[CG13] Byeongju Cha and Sandeep K. Gupta. « Trojan Detection via Delay Measurements: A New Approach to Select Paths and Vectors to Maximize Effectiveness and Minimize Cost ». In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013* (2013), pp. 1265–1270 (Cité à la page 36).

[Cha+08] Rajat Subhra Chakraborty, Somnath Paul, and Swarup Bhunia. « On-demand transparency for improving hardware Trojan detectability ». In: *2008 IEEE International Workshop on Hardware-Oriented Security and Trust, HOST* (2008), pp. 48–50 (Cité à la page 46).

[Cha+09a] R. S. Chakraborty, S. Narasimhan, and S. Bhunia. « Hardware Trojan: Threats and emerging solutions ». In: *High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International*. 2009, pp. 166–171 (Cité à la page 15).

[Cha+09b] Rajat Subhra Chakraborty, Francis Wolff, Somnath Paul, Christos Papachristou, and Swarup Bhunia. « MERO: A statistical approach for hardware Trojan detection ». In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5747 LNCS (2009), pp. 396–410 (Cité aux pages 17, 24, 25).

[Cho+07] L.W. Chow, J.P. Baukus, and W.M. Clark. *Integrated circuits protected against reverse engineering and method for fabricating the same using an apparent metal contact line terminating on field oxide*. US Patent 7,294,935. 2007 (Cité aux pages 46, 47).

[Cou+15] Franck Courbon, Philippe Loubet-moundi, Jacques J A Fournier, and Assia Tria. « A high efficiency Hardware Trojan detection technique based on fast SEM imaging ». In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015* (2015), pp. 788–793 (Cité à la page 22).

[Cou15] Franck Courbon. « Partial hardware reverse engineering applied to fine grained laser fault injection and efficient hardware trojans detection ». Theses. Ecole Nationale Supérieure des Mines de Saint-Etienne, Sept. 2015 (Cité à la page 23).

[DAR07] DARPA. *"TRUST in Integrated Circuits (TIC) - Proposer Information Pamphlet*. 2007 (Cité aux pages 10, 11).

[Dav+13] Azadeh Davoodi, Min Li, and Mohammad Tehranipoor. « A Sensor-Assisted Self-Authentication Framework for Hardware Trojan Detection ». In: *IEEE Design & Test* 30.5 (2013), pp. 74–82 (Cité aux pages 35, 36).

[def05] U.S. Depatrement of defense. *Defense Science Board Task Force On High Performance Microchip Supply*. http://www.acq.osd.mil/dsb/reports/ADA435563.pdf. 2005 (Cité à la page 13).

[Deh+12] A. Dehbaoui, J.M. Dutertre, B. Robisson, et al. *Injection of transient faults using electromagnetic pulses -Practical results on a cryptographic system-*. Cryptology ePrint Archive, Report 2012/123. `http://eprint.iacr.org/2012/123`. 2012 (Cité à la page 8).

[Dev+08] S. Devadas, E. Suh, S. Paral, et al. « Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications ». In: *2008 IEEE International Conference on RFID*. 2008, pp. 58–64 (Cité à la page 53).

[Dup+13] Sophie Dupuis, Giorgio Di Natale, Marie-lise Flottes, et al. « Identification of Hardware Trojans triggering signals ». In: *First Workshop on Trustworthy Manufacturing and Utilization of Secure Devices* (2013) (Cité à la page 26).

[Dup+14] Sophie Dupuis, Papa Sidi Ba, Giorgio Di Natale, Marie Lise Flottes, and Bruno Rouzeyre. « A novel hardware logic encryption technique for thwarting illegal overproduction and Hardware Trojans ». In: *Proceedings of the 2014 IEEE 20th International On-Line Testing Symposium, IOLTS 2014* (2014), pp. 49–54 (Cité à la page 47).

[Exu+15] Ingrid Exurville, Loie Zussa, Jean Baptiste Rigaud, and Bruno Robisson. « Resilient hardware Trojans detection based on path delay measurements ». In: *Proceedings of the 2015 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2015* (2015), pp. 151–156 (Cité à la page 37).

[Fer+12] Andrew Ferraiuolo, Xuehui Zhang, and Mohammad Tehranipoor. « Experimental analysis of a ring oscillator network for hardware trojan detection in a 90nm ASIC ». In: *Proceedings of the International Conference on Computer-Aided Design - ICCAD '12* (2012), p. 37 (Cité aux pages 28, 29, 56, 84).

[Fpg] *Microblaze development kit spartan-3e 1600e edition user guide.* Federal Information Processing Standards Publication 197. 2007 (Cité à la page 60).

[Gen+13] Daniel Genkin, Adi Shamir, and Eran Tromer. *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*. Cryptology ePrint Archive, Report 2013/857. `http://eprint.iacr.org/2013/857`. 2013 (Cité à la page 9).

[Gui+14] Ujjwal Guin, Ke Huang, Daniel Dimase, et al. « Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain ». In: *Proceedings of the IEEE* 102.8 (2014), pp. 1207–1228 (Cité aux pages 11, 17, 53).

[He+15] Chunhua He, Bo Hou, Liwei Wang, Yunfei En, and Shaofeng Xie. « A failure physics model for hardware Trojan detection based on frequency spectrum analysis ». In: *IEEE International Reliability Physics Symposium Proceedings* 2015-May (2015), PR11–PR14 (Cité à la page 33).

[Hel+13] Clemens Helfmeier, Dmitry Nedospasov, Christopher Tarnovsky, et al. « Breaking and Entering Through the Silicon ». In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. CCS '13. Berlin, Germany: ACM, 2013, pp. 733–744 (Cité à la page 13).

[Hin] *HINT project*. www.hint-project.eu/ (Cité à la page 13).

[Jag+14] Meenatchi Jagasivamani, Peter Gadfort, Michel Sika, Michael Bajura, and Michael Fritze. « Split-fabrication obfuscation: Metrics and techniques ». In: *Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014* (2014), pp. 7–12 (Cité à la page 49).

[JJ08] Susmit Jha and Sumit Kumar Jha. « Randomization based probabilistic approach to detect trojan circuits ». In: *Proceedings of IEEE International Symposium on High Assurance Systems Engineering* (2008), pp. 117–124 (Cité à la page 24).

[JM07] R.W. Jarvis and M.G. McIntyre. *Split manufacturing method for advanced semiconductor circuits*. US Patent 7,195,931. 2007 (Cité à la page 49).

[Kar+10] Ramesh Karri, Jeyavijayan Rajendran, Kurt Rosenfeld, and Mohammad Tehranipoor. « Trustworthy hardware: Identifying and classifying hardware trojans ». In: *Computer* 43.10 (2010), pp. 39–46 (Cité aux pages 13, 15).

[Kar+15] Nima Karimian, Fatemeh Tehranipoor, Md. Tauhidur Rahman, Shane Kelly, and Domenic Forte. « Genetic Algorithm for hardware Trojan detection with ring oscillator network (RON) ». In: *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 2015, pp. 1–6 (Cité aux pages 28, 56).

[Kes+05] Ali Keshavarzi, Gerhard Schrom, Stephen Tang, et al. « Measurements and Modeling of Intrinsic Fluctuations in MOSFET Threshold Voltage ». In: *Proceedings of ISLPED 2005*. San Diego, CA, USA: ACM, 2005, pp. 26–29 (Cité à la page 91).

[Koc+11] Paul Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. « Introduction to differential power analysis ». In: *Journal of Cryptographic Engineering* 1.1 (2011), pp. 5–27 (Cité à la page 9).

[Koc+99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. « Differential Power Analysis ». In: *Advances in Cryptology — CRYPTO' 99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings*. Ed. by Michael Wiener. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397 (Cité à la page 9).

[Koc96] Paul C. Kocher. « Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems ». In: *Advances in Cryptology — CRYPTO '96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings*. Ed. by Neal Koblitz. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 104–113 (Cité à la page 9).

[KV14] Paris Kitsos and Artemios G. Voyiatzis. « FPGA Trojan Detection Using Length-Optimized Ring Oscillators ». In: *2014 17th Euromicro Conference on Digital System Design* (2014), pp. 675–678 (Cité aux pages 30, 84).

[Lam+11] Charles Lamech, Reza M Rad, Mohammad Tehranipoor, and Jim Plusquellic. « An Experimental Analysis of Power and Delay Signal-to-Noise Requirements for Detecting Trojans and Methods for Achieving the Required Detection Sensitivities ». In: *IEEE Transactions on Information Forensics and Security* 6.3 (2011), pp. 1170–1179 (Cité aux pages 29, 30).

[Lec+15] M. Lecomte, J. J. A. Fournier, and P. Maurine. « Thoroughly analyzing the use of ring oscillators for on-chip hardware trojan detection ». In: *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. 2015, pp. 1–6 (Cité à la page 59).

[Lec+16a] M. Lecomte, J. J. A. Fournier, and P. Maurine. « Granularity and detection capability of an adaptive embedded Hardware Trojan detection system ». In: *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2016, pp. 135–138 (Cité aux pages 87, 101).

[Lec+16b] M. Lecomte, J. J. A. Fournier, and P. Maurine. « On-chip fingerprinting of IC topology for integrity verification ». In: *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2016, pp. 133–138 (Cité aux pages 87, 101).

[Lec+16c] M. Lecomte, J. J. A. Fournier, and P. Maurine. « SYSTEM AND METHOD FOR SECURING AN ELECTRONIC CIRCUIT ». Pat. WO/2016/042144. patent: WO/2016/042144. 2016 (Cité à la page 59).

[Lec+16d] Maxime Lecomte, Jacques J A Fournier, and Philippe Maurine. « An on-chip technique to detect Hardware Trojans and assist counterfeit identification ». In: *IEEE transaction on VLSI* (2016). (Soumis) (Cité à la page 133).

[Lim+05] Daihyun Lim, J. W. Lee, B. Gassend, et al. « Extracting secret keys from integrated circuits ». In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 13.10 (2005), pp. 1200–1205 (Cité à la page 53).

[Lin+09] Lang Lin, Markus Kasper, Tim Güneysu, Christof Paar, and Wayne Burleson. « Trojan side-channels: lightweight hardware trojans through side-channel engineering ». In: *Cryptographic Hardware and Embedded Systems-CHES 2009*. Springer Berlin Heidelberg, 2009, pp. 382–395 (Cité à la page 12).

[LL07] Jie Li and John Lach. « Negative-skewed shadow registers for at-speed delay variation characterization ». In: *2007 IEEE International Conference on Computer Design, ICCD 2007* (2007), pp. 354–359 (Cité à la page 34).

[LL08] Jie Li and J. Lach. « At-speed delay characterization for IC authentication and Trojan Horse detection ». In: *HOST 2008*. 2008, pp. 8–14 (Cité aux pages 34, 35).

[LP12] Charles Lamech and Jim Plusquellic. « Trojan detection based on delay variations measured using a high-precision, low-overhead embedded test structure ». In: *Proceedings of the 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2012* (2012), pp. 75–82 (Cité à la page 35).

[Mak08] Yiorgos Makris. « Hardware Trojan detection using path delay fingerprint ». In: *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (2008), pp. 51–57 (Cité aux pages 17, 34, 35).

[Mau+12] Philippe Maurine, Karim Tobich, Thomas Ordas, and Pierre Yvan Liardet. « Yet Another Fault Injection Technique : by Forward Body Biasing Injection ». In: *YACC'2012: Yet Another Conference on Cryptography*. Porquerolles Island, France, Sept. 2012 (Cité à la page 8).

[Min+14] Xue Mingfu, Hu Aiqun, and Li Guyue. « Detecting Hardware Trojan Through Heuristic Partition and Activity Driven Test Pattern Generation ». In: *Communications Security Conference (CSC'14)* (2014) (Cité à la page 42).

[Moe+15a] S. Moein, J. Subramnian, T. A. Gulliver, F. Gebali, and M. W. El-Kharashi. « Classification of hardware trojan detection techniques ». In: *Computer Engineering Systems (ICCES), 2015 Tenth International Conference on*. 2015, pp. 357–362 (Cité aux pages 19, 20).

[Moe+15b] Samer Moein, Salman Khan, T Aaron Gulliver, and Fayez Gebali. « An Attribute Based Classification of Hardware Trojans ». In: (2015), pp. 351–356 (Cité aux pages 13, 15, 16).

[MS] M. K. MANDAL and B. C. SARKAR. « Ring oscillators: Characteristics and applications ». eng. In: *Indian journal of pure & applied physics* 48.2 (), pp. 136–145 (Cité à la page 27).

[MS99] B. Mathew and D. G. Saab. « Combining multiple DFT schemes with test generation ». In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18.6 (1999), pp. 685–696 (Cité à la page 24).

[Nar+10] S. Narasimhan, Dongdong Du, R.S. Chakraborty, et al. « Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach ». In: *HOST 2010*. 2010, pp. 13–18 (Cité à la page 43).

[Nej+15] Arash Nejat, David Hely, and Vincent Beroulle. « Facilitating Side Channel Analysis by Obfuscation for Hardware Trojan Detection ». In: *International Design & Test Symposium (IDT'15)* (2015), pp. 129–134 (Cité à la page 47).

[Noh+10] Karsten Nohl, Erik Tews, and Ralf-Philipp Weinmann. « Cryptanalysis of the DECT Standard Cipher ». In: *Fast Software Encryption* 6147 (2010), pp. 1–18 (Cité à la page 51).

[Now+14] Abdullah Nazma Nowroz, Kangqiao Hu, Farinaz Koushanfar, and Sherief Reda. « Novel techniques for high-sensitivity hardware trojan detection using thermal and power maps ». In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 33.12 (2014), pp. 1792–1805 (Cité aux pages 33, 70).

[PC07] Bipul C Paul and Toshiba Corp. « Circuit Failure Prediction and Its Application to Transistor Aging Mridul Agarwal Stanford University Intel Corporation Subhasish Mitra ». In: *25th IEEE VLSI Test Symposium (VTS'07)* (2007), pp. 277 –286 (Cité à la page 19).

[Pel+89] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers. « Matching properties of MOS transistors ». In: *IEEE Journal of Solid-State Circuits* 24.5 (1989), pp. 1433–1439 (Cité à la page 21).

[PQ03] Gilles Piret and Jean-Jacques Quisquater. « A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad ». In: *Cryptographic Hardware and Embedded Systems - CHES 2003: 5th International Workshop, Cologne, Germany, September 8–10, 2003. Proceedings*. Ed. by Colin D. Walter, Çetin K. Koç, and Christof Paar. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 77–88 (Cité à la page 8).

[QS01] Jean-Jacques Quisquater and David Samyde. « ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards ». In: *Smart Card Programming and Security: International Conference on Research in Smart Cards, E-smart 2001 Cannes, France, September 19–21, 2001 Proceedings*. Ed. by Isabelle Attali and Thomas Jensen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 200–210 (Cité à la page 9).

[Rad+08] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic. « Power supply signal calibration techniques for improving detection resolution to hardware Trojans ». In: *2008 IEEE/ACM International Conference on Computer-Aided Design*. 2008, pp. 632–639 (Cité aux pages 17, 31, 37, 43).

[Raj+11a] Jeyavijayan Rajendran, Vinayaka Jyothi, Ozgur Sinanoglu, and Ramesh Karri. « Design and analysis of ring oscillator based Design-for-Trust technique ». In: *Proceedings of the IEEE VLSI Test Symposium* (2011), pp. 105–110 (Cité aux pages 27, 60).

[Raj+11b] Jeyavijayan Rajendran, Vinayaka Jyothi, Ozgur Sinanoglu, and Ramesh Karri. « Design and analysis of ring oscillator based Design-for-Trust technique ». In: *Proceedings of the IEEE VLSI Test Symposium* (2011), pp. 105–110 (Cité à la page 45).

[Raj+12] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri. « Logic encryption: A fault analysis perspective ». In: *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2012, pp. 953–958 (Cité aux pages 47, 48).

[Raj+13] Jeyavijayan J V Rajendran, Ozgur Sinanoglu, and Ramesh Karri. « Is Split Manufacturing Secure? » In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)* Ic (2013), pp. 1259 –1264 (Cité à la page 49).

[Raj+14] Jeyavijayan Rajendran, Ozgur Sinanoglu, and Ramesh Karri. « Regaining trust in VLSI design: Design-for-trust techniques ». In: *Proceedings of the IEEE* 102.8 (2014), pp. 1266–1282 (Cité à la page 45).

[RL09] Devendra Rai and John Lach. « Performance of delay-based trojan detection techniques under parameter variations ». In: *2009 IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2009* (2009), pp. 58–65 (Cité à la page 35).

[Roy+08] J. A. Roy, F. Koushanfar, and I. L. Markov. « EPIC: Ending Piracy of Integrated Circuits ». In: *2008 Design, Automation and Test in Europe*. 2008, pp. 1069–1074 (Cité à la page 47).

[SA03] Sergei P. Skorobogatov and Ross J. Anderson. « Optical Fault Induction Attacks ». In: *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*. CHES '02. London, UK, UK: Springer-Verlag, 2003, pp. 2–12 (Cité à la page 8).

[Sah+15] Sayandeep Saha, Rajat Subhra Chakraborty, Srinivasa Shashank Nuthakki, Anshul, and Debdeep Mukhopadhyay. « Improved Test Pattern Generation for Hardware Trojan Detection Using Genetic Algorithm and Boolean Satisfiability ». In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9293. 2015, pp. 577–596 (Cité aux pages 25, 26).

[Sch+12] Alexander Schlösser, Dmitry Nedospasov, Juliane Krämer, Susanna Orlic, and Jean-Pierre Seifert. « Simple Photonic Emission Analysis of AES ». In: *Cryptographic Hardware and Embedded Systems – CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*. Ed. by Emmanuel Prouff and Patrick Schaumont. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 41–57 (Cité à la page 9).

[SD07] G. E. Suh and S. Devadas. « Physical Unclonable Functions for Device Authentication and Secret Key Generation ». In: *2007 44th ACM/IEEE Design Automation Conference*. 2007, pp. 9–14 (Cité à la page 53).

[Sol+] O. Soll, T. Korak, M. Muehlberghuber, and M. Hutter. « EM-based detection of hardware trojans on FPGAs ». In: *HOST 2014*, pp. 84–87 (Cité aux pages 37, 38, 40, 56, 117).

[Ste+14] Franco Stellari, Peilin Song, Alan J. Weger, et al. « Verification of untrusted chips using trusted layout and emission measurements ». In: *Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014* (2014), pp. 19–24 (Cité à la page 41).

[TJ11] R. Torrance and D. James. « The state-of-the-art in semiconductor reverse engineering ». In: *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*. 2011, pp. 333–338 (Cité à la page 9).

[Val+13] Jonathan Valamehr, Timothy Sherwood, Ryan Kastner, et al. « A 3-D split manufacturing approach to trustworthy system development ». In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32.4 (2013), pp. 611–615 (Cité aux pages 49, 50).

[Wan+08a] Xiaoxiao Wang, Mohammad Tehranipoor, and Jim Plusquellic. « Detecting malicious inclusions in secure hardware: Challenges and solutions ». In: *2008 IEEE International Workshop on Hardware-Oriented Security and Trust, HOST* (2008), pp. 15–19 (Cité à la page 15).

[Wan+08b] Xiaoxiao Wang, Hassan Salmani, Mohammad Tehranipoor, and Jim Plusquellic. « Hardware Trojan detection and isolation using current integration and localized current analysis ». In: *Proceedings - IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (2008), pp. 87–95 (Cité aux pages 32, 37).

[Wol+08] Francis Wolff, Chris Papachristou, Swarup Bhunia, and Rajat S. Chakraborty. « Towards trojan-free trusted ICs: Problem analysis and detection scheme ». In: *Proceedings -Design, Automation and Test in Europe, DATE* (2008), pp. 1362–1365 (Cité à la page 24).

[WP11] Sheng Wei and Miodrag Potkonjak. « Scalable consistency-based hardware trojan detection and diagnosis ». In: *Proceedings - 2011 5th International Conference on Network and System Security, NSS 2011* (2011), pp. 176–183 (Cité à la page 19).

[Xia+15] Kan Xiao, Domenic Forte, and Mark Mohammed Tehranipoor. « Efficient and secure split manufacturing via obfuscated built-in self-authentication ». In: *Proceedings of the 2015 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2015* (2015), pp. 14–19 (Cité aux pages 50, 51).

[XT+15] NGO Xuan Thuy, Najm Zakaria, Shivam Bhasin, Guilley Sylvain, and Danger Jean-luc. « Method Taking into Account Process Dispersions to Detect Hardware Trojan Horse by Side-Channel ». In: (2015) (Cité aux pages 40, 56, 60, 88).

[XT11] Xuehui Zhang and Mohammad Tehranipoor. « RON: An on-chip ring oscillator network for hardware Trojan detection ». In: *2011 Design, Automation & Test in Europe*. Vol. 1. IEEE, 2011, pp. 1–6 (Cité aux pages 27, 45, 56, 60).

[Zha+12] X. Zhang, N. Tuzzio, and M. Tehranipoor. « Identification of recovered ICs using fingerprints from a light-weight on-chip sensor ». In: *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*. 2012, pp. 703–708 (Cité aux pages 53, 54).

[Zha+13] Xuehui Zhang, Andrew Ferraiuolo, and Mohammad Tehranipoor. « Detection of trojans using a combined ring oscillator network and off-chip transient power analysis ». In: *ACM Journal on Emerging Technologies in Computing Systems* 9.3 (2013), pp. 1–20 (Cité aux pages 45, 84).

[Cha09] S. Chakraborty, R.S. and Bhunia. « Security against hardware Trojan through a novel application of design obfuscation ». In: *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on* (2009), pp. 113–116 (Cité à la page 46).

[Ngo14] Zakaria Ngo, Xuan Thuy and Guilley, Sylvain and Bhasin, Shivam and Danger, Jean-Luc and Najm. « Encoding the State of Integrated Circuits : a Proactive and Reactive Protection against Hardware Trojans Horses ». In: *Proceedings of the 9th Workshop on Embedded Systems Security* (2014), 7:1–7:10 (Cité aux pages 51, 52).

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2016LYSEM018

Maxime LECOMTE

ON-CHIP VOLTAGE MEASUREMENT SYSTEM FOR COUNTERFEITS AND HARDWARE TROJANS DETECTION

Speciality : Microelectronic

Keywords : Hardware Security, Hardware Trojan, Counterfeit, Ring Oscillator, FPGA

Abstract :

Due to the trend to outsourcing semiconductor manufacturing, the integrity of integrated circuits (ICs) became a hot topic. As hardware security is the root of trust and security, any potential malicious modification of an IC is a serious threat for the final product. This kind of malicious alteration is called Hardware Trojan and it can have different types of effects that range from the degradation of the ICs performances to the denial of service through the leak of sensitive information. In another hand, with the complexity of the semiconductor distribution market, ICs can be counterfeited. Several cases of recycling or remarking of ICs have been observed. The IC integrity verification has been studied in several recent researches. The main limit of the proposed techniques so far is that the bias, induced by the process variations, restricts their efficiency and practicality.

In this thesis we aim to detect HTs and counterfeits in a fully embedded way, it implies on-chip measurement and lightweight analysis to limit area impact of the solution. To that end we first characterize the impact of malicious insertions on a network of sensors. The measurements are done using a network of Ring oscillator, that is an oscillating structure whose the period depends on three main parameters (process variation, voltage and temperature). The malicious adding of logic gates (Hardware Trojan) or the modification of the implementation of a different design (counterfeits) will modify the voltage distribution within the IC.

Based on these results we present an on-chip detection method for verifying the integrity of ICs, in particular for detecting malicious add-ons like Hardware Trojans or counterfeited ICs. We propose a novel approach which in practice eliminates this limit of process variation bias by making the assumption that IC infection is done at a lot level. We introduce a new variation model for the performance of CMOS structures. This model is used to create signatures of lots which are independent of the process variations. A new distinguisher has been proposed to evaluate whether an IC is infected. This distinguisher allows automatically setting a decision making threshold that is adapted to the measurement quality and the process variation. The goal of this distinguisher is to reach a 100\% success rate within the set of covered HTs family. All the results have been experientially validated and characterized on a set of FPGA prototyping boards.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2016LYSEM018

Maxime Lecomte

SYSTEME EMBARQUE DE MESURE DE LA TENSION POUR LA DETECTION DE CONTREFAÇONS ET DE CHEVAUX DE TROIE MATERIELS

Spécialité: Microélectronique

Mots clefs : Sécurité matérielle, Chaval de Troie matériel, Contrefaçons, Oscillateur en anneau, FPGA

Résumé :

Avec la mondialisation du marché des semiconducteurs, l'intégrité des circuits intégrés (CI) est devenue préoccupante... La sécurité matérielle étant la racine de la confiance et de la sécurité, une potentielle modification malveillante d'un CI est une sérieuse menace pour le produit final. Cette catégorie de modification est appelée cheval de Troie matériel (CTM) et peut avoir différent type d'effet allant de la dégradation de performances CIs au déni de service en passant par la fuite d'information sensible. De plus avec la complexité de la distribution, les CIs peuvent être contrefait. Par exemple en recyclant des circuits utilisés ou en réinscrivant des circuits de qualité inférieure. L'intégrité des CIs a été étudiée dans plusieurs travaux récents. La principale limite des méthodes proposées jusqu'à maintenant est le biais induit par les variations des procédés de fabrication.

Cette thèse a pour but de proposer une méthode de détection embarquée de détection de CTM et de contrefaçons, cela implique des mesures internes et une analyse simple pouvant être embarquées sur une surface limitée. À cette fin, une caractérisation de l'impact des modifications malveillantes sur un réseau de capteurs embarqué a été effectuée. Le capteur utilisé est un oscillateur en anneau dont la période d'oscillation dépend de trois paramètres principaux (les variations de procédé, la tension et la température). L'addition malicieuse de portes logiques (CTM) ou la modification de l'implémentation du circuit (contrefaçons) modifie la distribution de la tension à la l'intérieur du circuit. À partir de cette caractérisation nous proposons une méthode embarquée de vérification d'intégrité des CIs.

Une nouvelle approche est proposée afin d'éliminer l'influence des variations des procédés. Nous posons tout d'abord que, pour des raisons de cout et de faisabilité, une infection est faite à l'échelle d'un lot de production. Un nouveau modèle de variation de performance temporelle des structures CMOS en condition de design réel est introduit. Ce modèle est utilisé pour créer des signatures de lots indépendantes des variations de procédé et utilisé pour définir une méthode permettant de détecter les CTMs et les contrefaçons.

Enfin nous proposons un nouveau distingueur permettant de déterminer si un CI est infecté ou non. Ce distingueur permet de placer automatiquement un seuil de décision adapté à la qualité des mesures et aux variations de procédés. Le but de ce distingueur est d'atteindre un taux de succès de 100 % pour l'ensemble des infections couvertes par notre méthodologie. Les résultats ont été expérimentalement validés sur un lot de cartes de prototypage FPGA.