Détection et prévention de Cheval de Troie Matériel (CTM) par des méthodes Orientées Test Logique

Papa-Sidy Ba

THÈSE

Pour obtenir le grade de

Docteur

Délivré par l'**Université de Montpellier**

Préparée au sein de l'école doctorale **I2S - Information, Structures, Systèmes**
Et de l'unité de recherche **UMR 5506 - Laboratoire d'Informatique, Robotique et Micro-électronique de Montpellier**

Spécialité: **SYAM - Systèmes Automatiques et Micro-électroniques**

Présentée par **Papa-Sidy Ba**

Détection et Prévention de Chevaux de Troie Matériels par des Méthodes Orientées Test Logique

Soutenue le 02 Décembre 2016 devant le jury composé de

M. Guy GOGNIAT	Pr	Université de Bretagne-Sud	Rapporteur
M. Paolo PRINETTO	Pr	Politecnico di Torino	Rapporteur
M. Bruno ROUZEYRE	Pr	Université de Montpellier - LIRMM	Directeur de thèse
M. Giorgio DI NATALE	DR CNRS	CNRS - LIRMM	Examinateur
Mme. Sophie DUPUIS	MCf	Université de Montpellier - LIRMM	Examinatrice
Mme. Marie-Lise FLOTTES	CR CNRS	CNRS - LIRMM	Examinatrice
M. Julien FRANCQ	Dr	Airbus Defence and Space- CyberSecurity	Examinateur
M. Jean-Baptiste RIGAUD	MCf	École Nationale des Mines de Saint Etienne	Examinateur

Avant propos

Remerciements

Contexte de la thèse

Cette thèse s'est déroulée dans le cadre du projet FUI#14 HOMERE (Hardware TrOjans : Menaces et robustEsse des ciRcuits intEgrés).
Ce projet a été financé par le gouvernement français (BPI-OSEO) et supervisé par la Direction Générale de l'Armement (DGA). C'est le fruit de la collaboration entre instituts de recherche (LIRMM, Mines de Saint Étienne, CEA LETI et Télécom ParisTech), d'industriels (Gemalto, Cassidian et SECURE-IC S.A.S.) et d'institutions gouvernementales (ANSSI et DGA).

Cette thèse s'est déroulée au LIRMM (Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier).
Le LIRMM est une unité mixte de recherche, dépendant conjointement de l'Université Montpellier (UM) et du Centre National de la Recherche Scientifique (CNRS). Il est situé sur le Campus Saint-Priest de l'UM.
Ses activités de recherche positionnent pleinement le LIRMM au coeur des sciences et technologies de l'information, de la communication et des systèmes.

Ainsi, de l'information aux systèmes, de la technologie à l'humain et aux usages, les activités de recherche du LIRMM concernent : la conception et la vérification de systèmes intégrés, mobiles, communicants, la modélisation de systèmes complexes à base d'agents, les études en algorithmique, bioinformatique, interactions homme-machine, robotique, etc.

Les travaux sont menés dans trois départements scientifiques de recherche, eux-mêmes organisés en « équipes-projet ».
Le département Microélectronique mène des recherches de pointe dans les domaines de la conception et du test de systèmes intégrés et microsystèmes en mettant l'accent sur les aspects architectures, modélisation et méthodologie.

Table des matières

Liste des figures

Liste des tableaux

Liste des acronymes

ASIC Application-Specific Integrated Circuit Standard. 14
ATPG Automatic Test Pattern Generation. 15

BEOL Back-End-Of-the-Line. 31, 84
BIST Built-In-Self-Test. 83

CMDS Classical MultiDimensional Scaling. 60
CMOS Complementary Metal Oxide Semi-conductor. 4
CMP Chemical Metal Polishing. 23

DCVSL differential cascade voltage switch logic. 84
DEFENSE Design-For-Enabling-Security. 74
DfHT Design for Hardware Trust. 15, 83
DfR Design-for-Reliability. 74
DoS Denial-of-Service. 14
DRE Destructive Reverse Engineering. 12
DSE Densité Spectrale d'Enrgie. 58

FEOL Front-End-Of-the-Line. 31, 84
FF Flip-Flop. 33, 95
FIA Fault Injection Attacks. 13
FPGA Field-Programmable Gate Arrays. 55

GDSII Graphic Database System II. 12

HPC Hardware Property Checker. 75

IP Intellectual Property. 18

LFSR Linear Feedback Shift Register. 83

MERO Multiple Excitation of Rare Occurence. 55
MISR Minimal Input Shift Register. 83
MODMOR Modified Dual Modular Redundancy. 15
MOS Metal Oxyde Silicium. 4

of outlier factor. 60

PCC Proof Carrying Code. 24

PSL Property Specification Language. 75

RSA Rivest Shamir Adleman. 79

RTL Registers Transfer Level. 14

SCA Side Channel Analysis. 23

SEM Scanning Electron Microscope. 23

SHADE Secure Heartbeat And Dual-Encryption. 75

SLL Super Long Lines. 48

SM Security Monitor. 28

STG State Transition Graph. 30

TFD Transformée de Fourier Discrète. 58

VHDL VHSIC Hardware Description Language. 79

Résumé et abstract

Résumé

Pour réduire le coût des Circuits Intégrés (CIs), les entreprises de conception se tournent de plus en plus vers des fonderies basées dans des pays à faible coût de production (outsourcing)… Cela a pour effet d'augmenter les menaces sur les circuits. En effet, pendant la fabrication, le CI peut être altéré avec l'insertion d'un circuit malicieux, appelé cheval de Troie Matériel (CTM). Ceci amène les vendeurs de CIs à protéger leurs produits d'une potentielle insertion d'un CTM, mais également, d'en assurer l'authenticité après fabrication (pendant la phase de test).
Cependant, les CTMs étant furtifs par nature, il est très difficile, voire impossible de les détecter avec les méthodes de test conventionnel, et encore moins avec des vecteurs de test aléatoires. C'est pourquoi nous proposons dans le cadre de cette thèse des méthodes permettant de détecter et de prévenir l'insertion de CTMs dans les CIs pendant leur fabrication.

Ces méthodes utilisent des approches orientées test logique pour la détection de CTMs aussi bien en phase de test (après fabrication du CI) qu'en fonctionnement normal (run-time).
De plus, nous proposons des méthodes de prévention qui elles aussi s'appuient sur des principes de test logique pour rendre difficile, voire impossible l'insertion d'un CTM aussi bien au niveau netlist qu'au niveau layout.

Mots clés : Cheval de Troie matériel ; Test logique ; Sécurité matérielle ; Circuits Intégrés

Abstract

In order to reduce the production costs of Integrated Circuits (ICs), outsourcing the fabrication process has become a major trend in the Integrated Circuits industry. As an inevitable unwanted side effect, this outsourcing business model increases threats to hardware products. This process raises the issue of untrusted foundries in which, circuit descriptions can be manipulated with the aim to insert malicious circuitry or alterations, referred to as Hardware Trojan Horses (HTHs). This motivates semiconductor industries and researchers to study and investigate solutions for detecting HTHs during testing and preventing HTHs insertion during fabrication.

However, considering the stealthy nature of HTs, it is quite impossible to detect them with conventional testing or random patterns. This motivates us to make some contributions in this thesis by proposing solutions to detect and prevent HTH insertion after fabrication.
The proposed logic testing based methods help to detect HTH during testing as well as during normal mode (run-time).
Furthermore, we propose prevention methods, which are also logic testing based, in order to make harder or quasi impossible the insertion of HTH both at netlist and layout levels.

Key words : Hardware Trojan Horses ; Logic testing ; Hardware security ; Integrated Circuits

Chapitre 1

Introduction générale

« *It always seems impossible until its done* »

Nelson Mandela

Sommaire

L'industrie Micro-électronique a connu des avancées significatives en termes d'intégration de fonctions complexes dans les circuits intégrés dont la taille est de plus en plus réduite. Cette avancée peut être illustrée par l'augmentation du nombre de transistors dans un processeur au fil des années (cf figure 1.1).

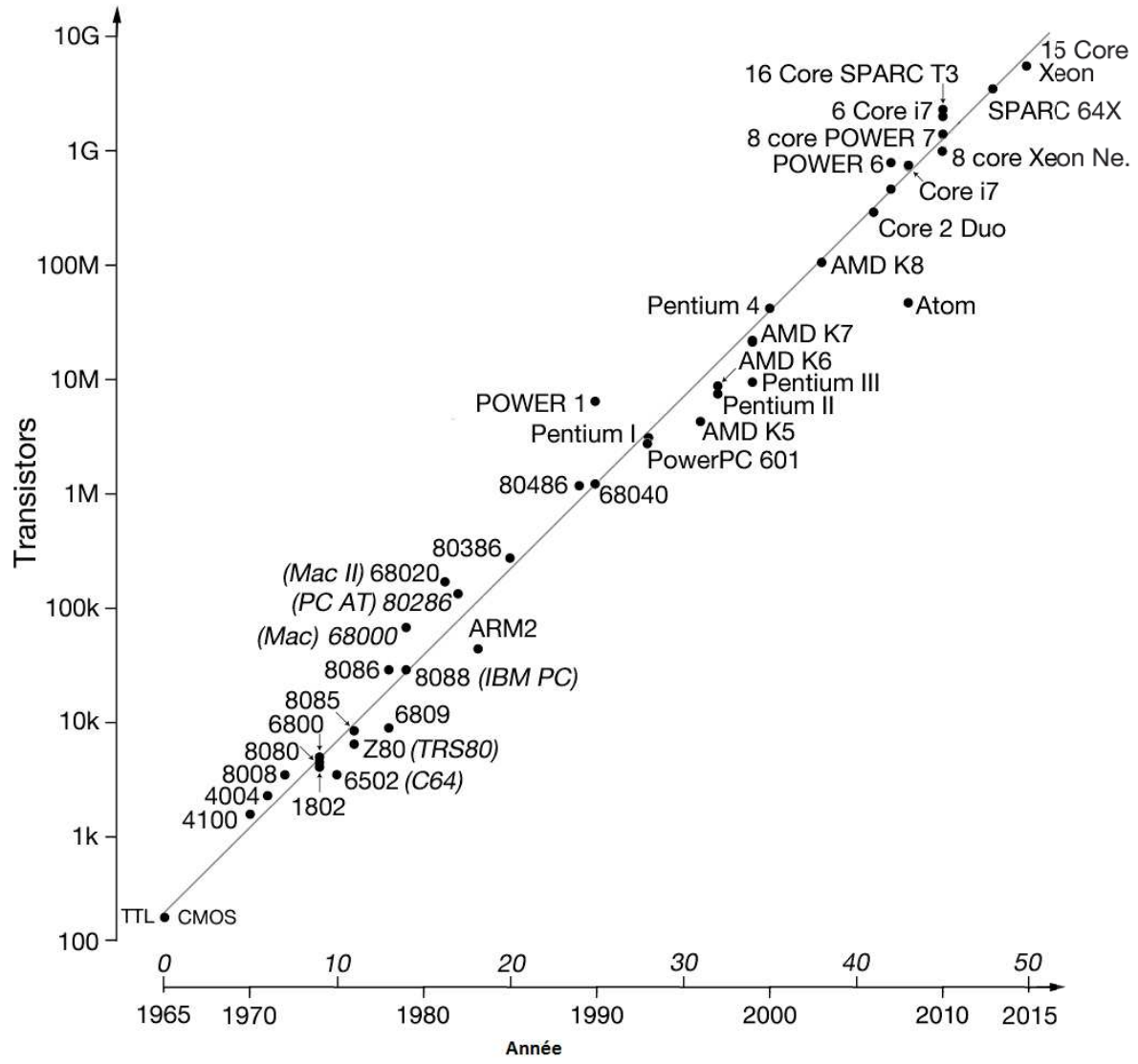


FIGURE 1.1 – Courbe d'évolution du nombre de transistors dans les processeurs [Sch16].

En effet, la miniaturisation des transistors Metal Oxyde Silicium (MOS) grâce aux avancées significatives de la technologie Complementary Metal Oxide Semi-conductor (CMOS) a permis l'augmentation de la densité d'intégration, la réduction de coûts de fabrication, la réduction du temps de transit des porteurs dans le canal des transistors et la réduction de la consommation. En atteste la diminution de la longueur de grille des transistors (voir figure 1.2).

Depuis 1950, la surface d'un transistor a été réduite d'un facteur supérieur à 10 000 [esp16]. Le coût des circuits diminue d'un facteur deux à peu près tous les 18 mois, accompagné par un accroissement important de leurs performances. En réduisant la taille et le coût, on peut alors rajouter de nouveaux services.

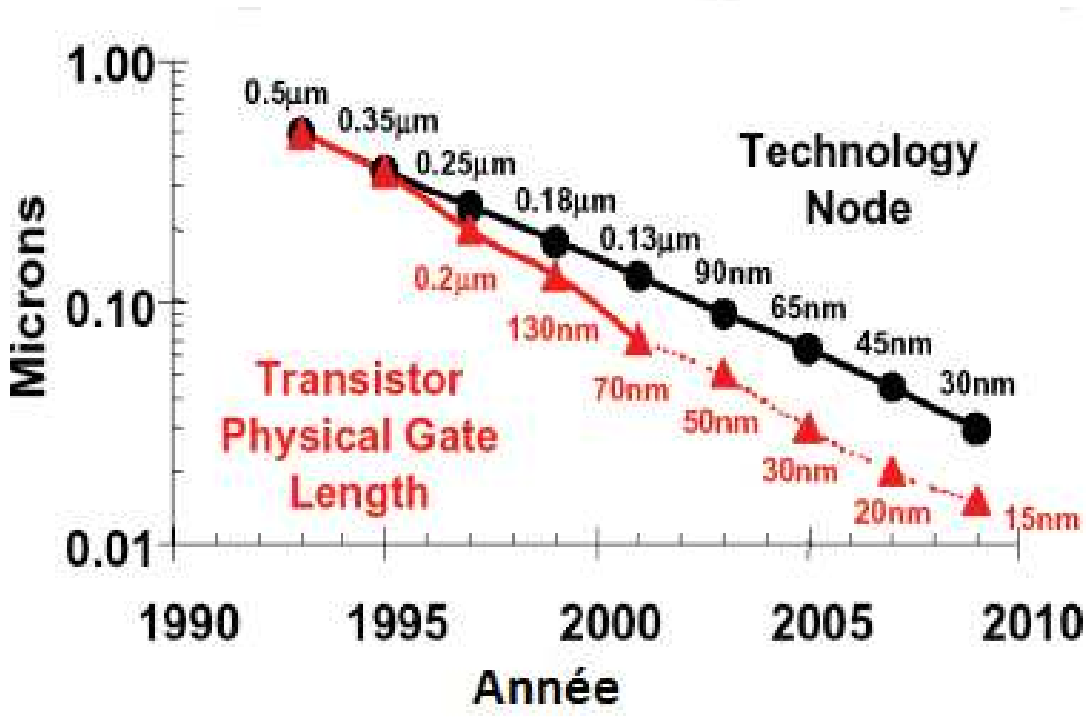


FIGURE 1.2 – Courbe d'évolution de la taille de grille des transistors [ROM04].

Ainsi, on observe une forte croissance du nombre de transistors par circuit depuis des années due à cette avancée. Ceci a permis un fort développement du nombre de fonctions embarquées dans les circuits pour des domaines variés (télécommunications, informatiques, aéronautiques et spatiales, militaires, etc.). Toutefois, le coût de fabrication augmente avec la diminution des tailles de transistor. Pour minimiser son impact, les fabricants ont adopté une stratégie consistant à délocaliser la fabrication dans des fonderies situées dans des pays à bas coût. Par ailleurs, la réduction du temps de conception (Time to Market) des CIs a amené les fabricants à sous-traiter la phase de conception. Ceci a favorisé l'émergence des fournisseurs de blocs IP.

Cette stratégie de délocalisation et la multiplication des fournisseurs de blocs IP soulèvent des questions sur la confiance que l'on peut attribuer aux circuits et à leur intégrité. En effet, les CIs peuvent faire l'objet d'attaques matérielles, en particulier l'insertion d'un circuit malicieux appelé Cheval de Troie Matériel (CTM), dont le but est de modifier les fonctionnalités initiales. C'est ainsi que depuis quelques années des questions concernant la présence de CTM sur les CIs ont été soulevées. Ceci a amené les chercheurs et les industriels à développer des méthodologies permettant de garantir l'intégrité des CIs.

Dans ce chapitre, nous allons aborder le flot de conception des CIs, les menaces potentielles et la confiance matérielle. Pour finir, nous introduirons les contributions que nous avons apportées dans le cadre de cette thèse, lesquelles seront détaillées dans des chapitres suivants.

1.1 Technologie CMOS

La technologie CMOS est utilisée pour la fabrication des circuits intégrés en utilisant des matériaux semi-conducteurs comme le silicium. Un semi-conducteur est un solide qui est isolant au zéro absolu et conducteur à la température ambiante. Dans un semi-conducteur, tout se passe comme si la conduction du courant était due à deux types de particules :

- Les électrons (comme dans un métal),
- Les trous, de charge opposée à celle de l'électron.

Dans un semi-conducteur parfaitement pur (semi-conducteur « intrinsèque ») la densité d'électrons n est égale à la densité de trous p : pour le silicium $n = p = 10^{10} cm^{-3}$. Un semi-conducteur dopé est un semi-conducteur dans lequel on a ajouté délibérément des très petites quantités d'impuretés bien choisies (typiquement 10^{12} à $10^{17} cm^{-3}$) qui modifient complètement les propriétés de conduction du matériau.
Il existe deux types d'impuretés :

- Les donneurs (impuretés pentavalentes dans le Silicium),
- Les accepteurs (impuretés trivalentes dans le Silicium).

Un semi-conducteur où les donneurs sont majoritaires est dit « semi-conducteur de type n » et un semi-conducteur où les accepteurs sont majoritaires est dit « semi-conducteur de type p ». Ceci nous amène à la fabrication des transistors MOS qui peuvent être soit de type N (NMOS) ou de type P (PMOS).

1.2 Transistor MOS

Comme nous l'avons vu précédemment, le type de transistor dépend du type de dopage dans le substrat. Un dopage de type N est un surplus de charges négatives dans le substrat (électrons) et un dopage de type P est un surplus de charges positives (trous). La figure 1.3 illustre ces deux types de transistor.
Un transistor MOS à enrichissement à canal N est une structure MOS sur un substrat de type P à laquelle on adjoint des zones de type N de part et d'autre de la capacité MOS de façon à pouvoir faire passer un courant dans une couche d'inversion d'électrons formée dans le substrat juste sous l'oxyde de grille. La capacité MOS se compose d'une première couche appelée « grille », la plupart du temps en Silicium poly cristallin très fortement dopée N ou P qui sert de contact électrique (le M de MOS). La deuxième couche est appelée « oxyde de grille » (le O de MOS), elle est généralement en Silice (SiO2) qui est réalisé par oxydation thermique. La troisième couche est appelée « substrat », (le S de MOS), elle est en silicium cristallin. Ce sera cette capacité MOS qui contrôlera, selon la polarisation qu'on lui applique, la création ou non d'une couche d'inversion dans le substrat mettant en contact électrique la source et le drain.

Ainsi, le transistor MOS se décompose en trois parties principales : l'électrode de grille, les électrodes de source et de drain et le canal de conduction entre ces deux dernières. La grille est polarisée par la tension V_g, le drain par la tension V_d et la source ainsi que le substrat sont reliés à la masse. Les tensions V_g et V_d permettent de contrôler le courant qui passe dans le canal. Ceci nous amène à déterminer les caractéristiques électriques du transistor MOS. On distingue trois régimes de fonctionnement :

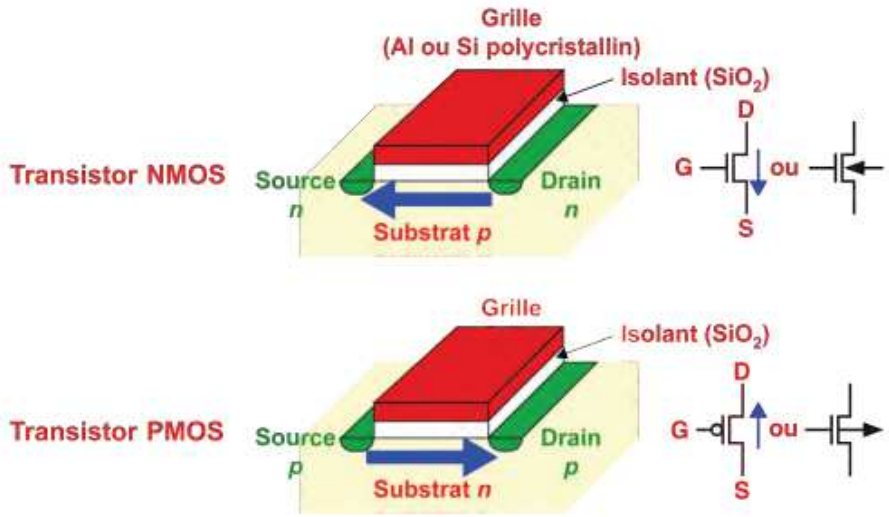


FIGURE 1.3 – Transistors NMOS et PMOS [ROM04].

- Le mode bloqué : Dans ce mode de fonctionnement, le courant drain source I_{ds} est nul.
- Le régime linéaire ou ohmique : Il est caractérisé par de faibles tensions de drain. Dans ce cas, le canal du transistor se comporte comme une résistance contrôlée par la tension grille-source V_{gs}.
- Le régime saturé : Pour une valeur de tension de drain V_d , le canal proche de V_{dsat}, le courant de drain reste constant avec l'augmentation de la tension de drain. C'est le régime principal de fonctionnement du transistor MOS.

Ainsi le transistor se comporte comme un interrupteur qui prend soit la valeur 0 soit V_{dd} (la tension d'alimentation).

1.3 Du transistor MOS aux cellules standards

Á partir des transistors, on peut construire des portes logiques qui sont utilisées dans le flot de conception VLSI des circuits intégrés comme cellules standards. Les portes logiques les plus utilisées sont les inverseurs qui nécessitent deux transistors MOS. La figure 1.4 illustre le schéma électrique d'un inverseur CMOS.

Pour les portes logiques, on ne considère que deux nivaux logiques '0' et '1'. Lorsque la tension drain-source V_ds est supérieure ou égale à V_dd, alors on considère que c'est un '1' logique. Lorsque Lorsque la tension V_ds est inférieure à V_dd on considère que c'est '0' logique.

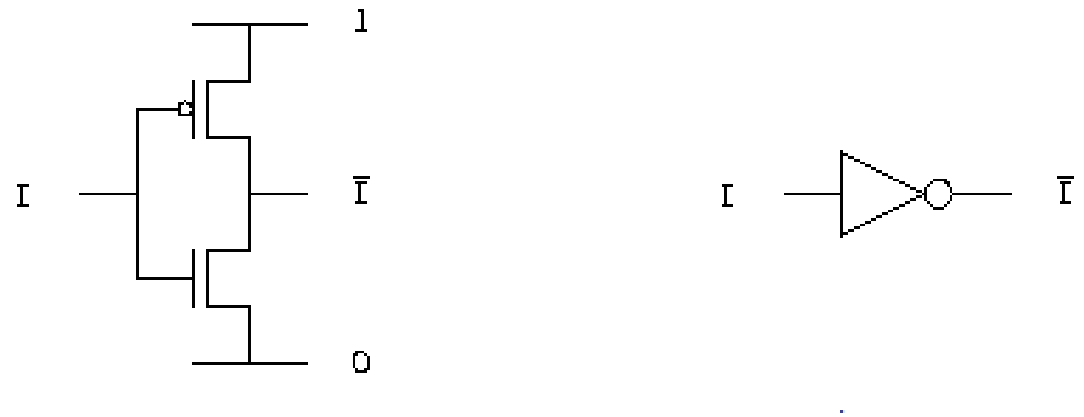


FIGURE 1.4 – Schéma électrique d'un inverseur CMOS.

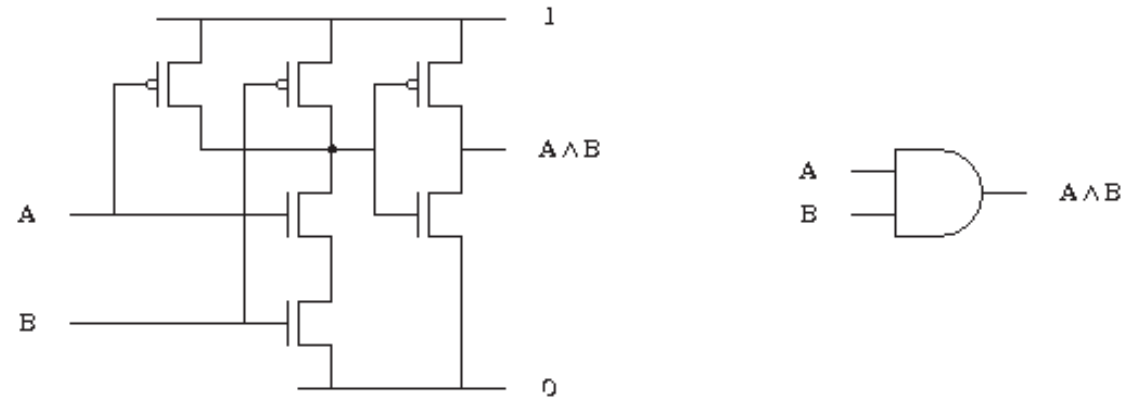


FIGURE 1.5 – Schéma électrique d'une porte AND CMOS à 2 entrées.

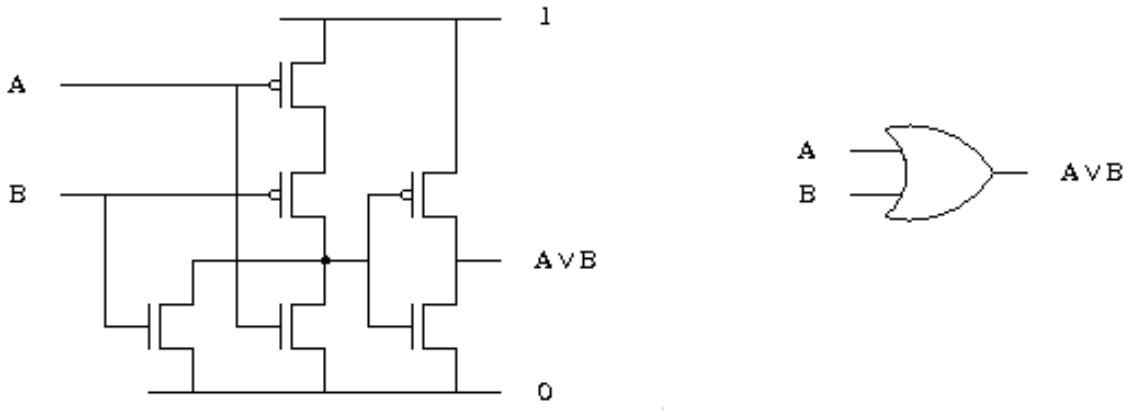


FIGURE 1.6 – Schéma électrique d'une porte OR CMOS à 2 entrées.

1.4 Flot de conception VLSI

Nous détaillons ci-après le flot standard qui est utilisé pour la fabrication de CIs à grande échelle. Il permet d'automatiser les différentes phases de conception d'un CI à partir d'une technologie ciblée. La figure 1.7 illustre ce flot.

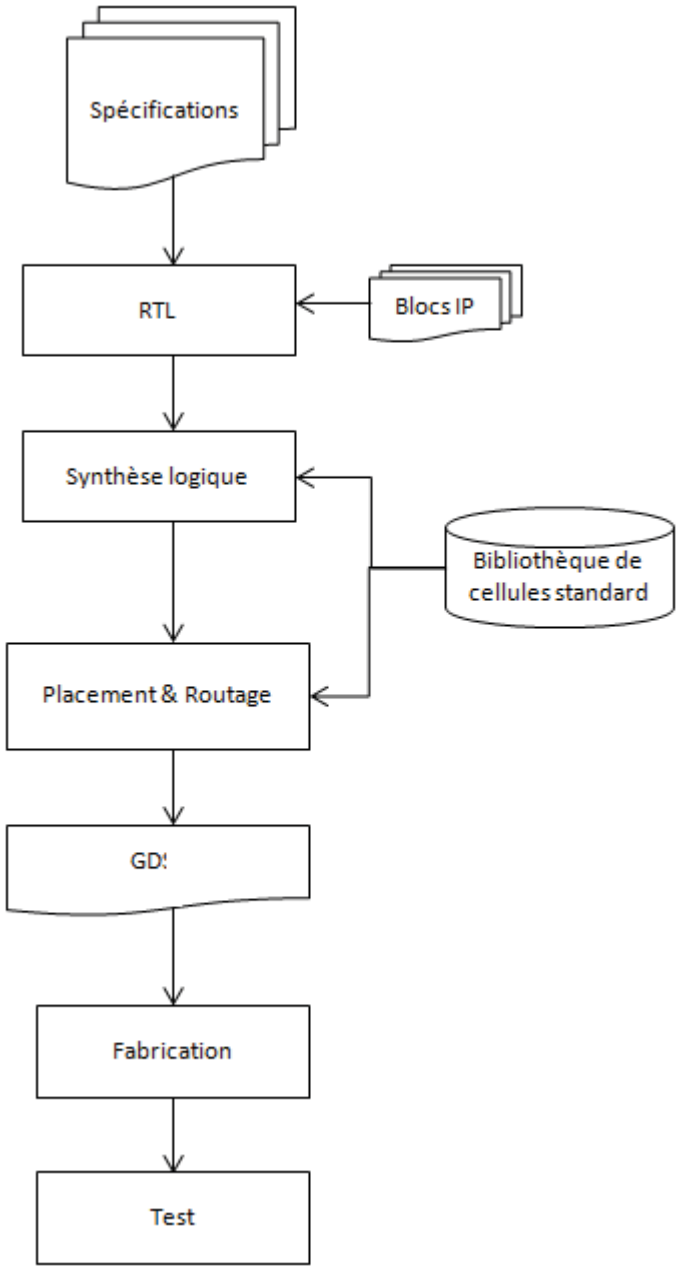


FIGURE 1.7 – Flot de conception VLSI.

1.4.1 Spécifications

Elles sont définies par le client. Elles décrivent les fonctionnalités du circuit ainsi que toutes les caractéristiques techniques que doit remplir le produit final. Ces caractéristiques vont ensuite être traduites en code RTL.

1.4.2 RTL

Á partir des spécifications, les langages RTL comme le VHDL ou le Verilog sont utilisés pour faire une description logique des différentes fonctions à implanter. Cette partie est souvent sous-traitée à des compagnies spécialisées dans le développement de blocs IP.

1.4.3 Synthèse logique

La description RTL est ensuite implantée dans une technologie cible pour générer une net-list qui correspond à la description au niveau porte des différentes fonctions du circuit en utilisant les cellules standards.

1.4.4 Placement et Routage

Une fois le circuit synthétisé et les différentes analyses réalisées, toutes les cellules standards sont placées sur la puce avant d'être inter-connectées (routage). Ceci permet d'avoir une description géométrique du circuit (GDSII). Une fois la vérification des interconnexions réalisée, le circuit est envoyé en fabrication.

1.4.5 Fabrication

Le GDSII est utilisé pour faire les dessin des masques du CI. La photolithographie est la technique utilisée pour transformer le dessin des masques en circuit physique (fabrication des transistors). C'est l'étape la plus complexe et celle qui nécessite beaucoup de moyens, c'est pourquoi beaucoup vendeurs de CI ne possèdent pas d'unités de fabrication et sous-traitent cette partie.

1.4.6 Test

Le test est réalisé avant la commercialisation du CI pour vérifier le bon fonctionnement des circuits. Des vecteurs de test sont appliqués aux entrées primaires et les valeurs aux sorties primaires sont comparées aux valeurs qui ont été générées avant fabrication. Ces tests permettent de détecter des défauts de fabrication.

1.5 Étapes vulnérables du le flot VLSI

Nous avons vu que la fabrication d'un CI à grande échelle (VLSI), passe par plusieurs phases. L'insertion d'un circuit malicieux (CTM) peut être réalisée durant certaines phases. C'est pourquoi une analyse de fiabilité de toutes les phases a été proposée. La figure 1.8 donne le niveau de confiance de chaque phase du flot VLSI. Les phases en vert sont considérées fiables. Les phases en rouges sont considérées critiques. Les phases en jaunes ne sont dignes que d'une confiance moyenne. Dans cette partie, nous détaillerons les phases considérées vulnérables (phases en jaune et en rouge).

1.5.1 Au cours de la conception

Cette phase est communément appelée Front-End. Elle regroupe plusieurs parties de la conception du CI. C'est durant cette phase que sont traduites les spécifications du client

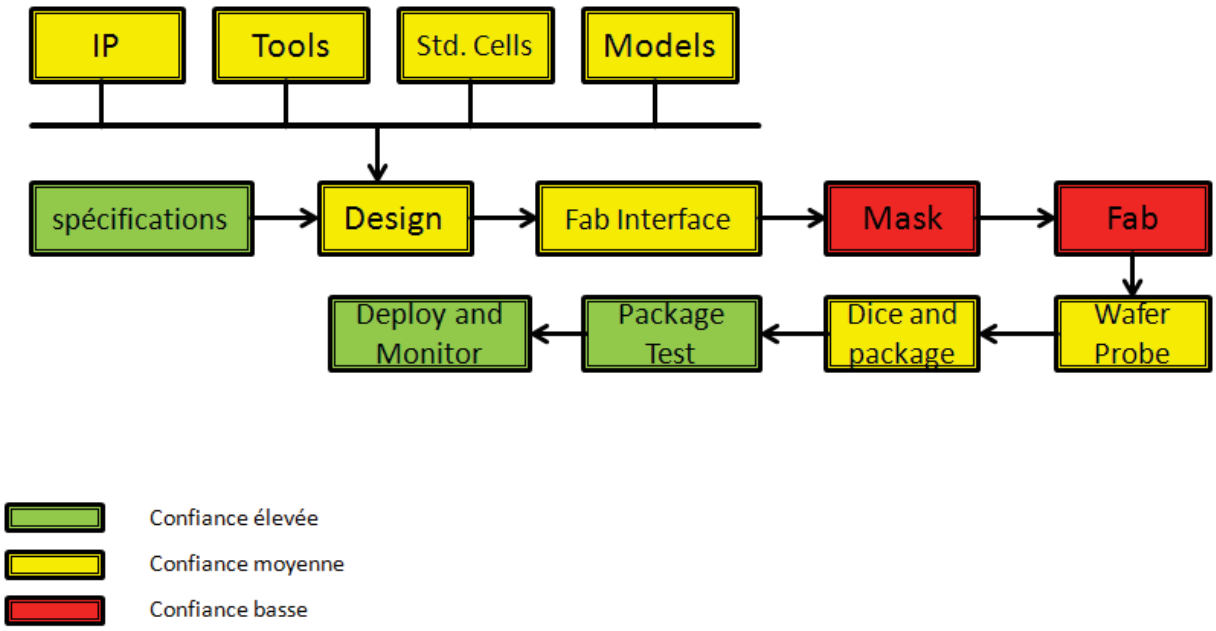


FIGURE 1.8 – Étapes vulnérables du flot de conception d'un CI [CNB09].

en code RTL puis en netlist. Pour finir, sont effectuées la vérification fonctionnelle et la validation.

Blocs IP

De nos jours, de plus en plus de compagnies confient la partie front-end à des entités tierces, notamment pour le développement de blocs spécifiques appelés IPs. Ces IPs peuvent contenir des fonctions malveillantes.

Cellules Standards

La traduction du code RTL en fonctions logiques repose sur l'utilisation de cellules standards, par exemple les portes AND vues plus haut. Ces cellules sont définies par des compagnies dont on ne connait pas le niveau de fiabilité et il n'est pas possible d'avoir un contrôle sur le processus de développement. Ceci soulève la question de la présence ou non de fonctions malicieuses.

Outils de CAO

Les outils de CAO sont utilisés durant tout le flot de conception (du RTL au test) dans le but d'automatiser les tâches et de réduire le temps de conception. Ces outils étant développés par des entités indépendantes, il est impossible de savoir s'ils sont fiables ou non.

1.5.2 Au cours de la fabrication

Cette phase est communément appelée Back-End. C'est dans cette partie qu'est réalisée le dessin de masque, la réalisation physique du circuit et la mise en boîtier. Il est donc évident qu'elle soit considérée comme étant la plus critique, car un attaquant aura toutes les informations pour implémenter un circuit malicieux. En effet, à partir du GDSII, il est possible de

reconstituer la netlist et donc les fonctionnalités du circuit. Ceci pose un réel problème de sécurité et de propriété intellectuelle.

1.6 Menaces potentielles

Plusieurs types de menaces ont été répertoriées dans la littérature.

- Les menaces sur la propriété intellectuelle : Des attaques telles que la rétro-ingénierie sont réalisées pour dérober le secret de fabrication et donc favoriser la contre-façon ou la surproduction.
- Les menaces sur l'intégrité des informations traitées par le CI : En effet, certaines attaques permettent de récupérer des informations secrètes telles que la clé de chiffrement d'un circuit cryptographique. Les attaques par canaux auxiliaires ou par fautes sont les plus connues.
- Les menaces sur la confiance matérielle : L'insertion d'un circuit malveillant dans le but de corrompre la fonctionnalité originale du circuit fait partie de ces menaces.

1.6.1 Attaques par rétro-ingénierie

Le principe d'une attaque par rétro-ingénierie (Destructive Reverse Engineering (DRE)) est de retrouver les différentes composantes du circuit (netlist ou Graphic Database System II (GDSII)) à partir du produit fini (le circuit intégré). Pour ce faire des méthodes optiques sont utilisées pour reconstituer les différentes couches de métal pour ensuite aller au niveau transistor, puis logique.

1.6.2 Attaques par canaux auxiliaires

Comme son nom l'indique, cette attaque est basée sur l'utilisation des différents paramètres du circuit afin de trouver une faille pour en extraire des informations cachées.

Attaques par sondage

Le principe d'une attaque par sondage est d'espionner l'activité électrique d'un composant électronique du circuit en positionnant une sonde suffisamment proche dudit composant. Dans la littérature scientifique, mesurer l'évolution de l'état d'une équipotentielle d'un bus (un « fil ») est un exemple typique d'attaque par sondage.

Cryptanalyse acoustique

Elle consiste à exploiter le bruit généré par le circuit. En effet, le circuit émet du bruit qui varie en intensité et en nature selon sa consommation et les opérations effectuées (typiquement des condensateurs qui se chargent ou se déchargent émettent un claquement facilement mesurable).

Analyse d'émanations électromagnétiques

Similaire à la cryptanalyse acoustique mais en utilisant le rayonnement électromagnétique ou ElectroMagnetic Analysis (EMA). Ce type d'attaque consiste à analyser les ondes électromagnétiques émises pour trouver des informations secrètes.

Attaque temporelle

Elle consiste à analyser le temps mis (ou le nombre de cycles d'horloge nécessaire) pour effectuer certaines opérations. Par exemple, l'attaque présentée dans [Koc96] permet de retrouver la clé de chiffrement de l'algorithme RSA. L'algorithme RSA consiste à calculer :
R = y^x mod n, avec x la clé de chiffrement. Ce calcul est effectué itérativement pour chaque bit de clé. Si le bit de clé vaut 1, le résultat partiel subit une multiplication, sinon il est inchangé. Ainsi en mesurant la durée de chaque itération, la clé peur être retrouvée.

1.6.3 Attaques par injection de fautes ou FIA

Les attaques par faute consistent à produire volontairement des erreurs dans le crypto-système. Ces attaques peuvent porter sur des composants matériels (crypto-processeurs) ou logiciels. Elles ont pour but de provoquer un comportement inhabituel des opérations cryptographiques dans le but d'en extraire des informations secrètes (par exemple, la clé de chiffrement).

Perturbations de l'alimentation ou de l'horloge

Des perturbations de l'alimentation ou du signal d'horloge externe peuvent perturber le fonctionnement du circuit. Typiquement en générant une période d'horloge inférieure au chemin critique, on induit des fautes de délai. C'est l'attaque la plus simple pour casser les codes des crypto-systèmes.

Injection de photons

Tous les circuits électroniques sont sensibles à la lumière du fait de l'effet photoélectrique. Le courant induit par les photons est utilisé pour provoquer une faute dans le circuit. Il est nécessaire de contrôler l'énergie du signal envoyé, la longueur d'ondes, la position et la durée.

Impulsion Magnétique

Une impulsion magnétique puissante près du silicium peut provoquer une faute. Le champ magnétique induit un courant local sur la surface du circuit qui peut générer une faute.

Température

La température de seuil du transistor évolue en fonction de la température. Des variations de température importantes peuvent provoquer un fonctionnement anormal du circuit et générer des fautes.

1.6.4 Insertion de CTM

C'est la modification ou l'altération des fonctionnalités du circuit par l'ajout ou la suppression de composants. Elle peut être réalisée à différents niveaux. L'insertion peut être faite durant la phase de conception (fournisseurs d'IP, librairie standard, outils de synthèse) ou durant la phase de fabrication (fonderies non fiables).

Ces menaces ont poussé les industriels et les chercheurs à trouver des moyens de garantir la confiance dans le matériel.

1.7 Confiance matérielle

Avec le niveau de complexité des CIs, le recours à des fournisseurs d'IP et à la délocalisation de la fabrication, le maître mot du développement futur des CIs est la confiance.

① Confiance dans la capacité des systèmes à résister à des attaques visant la confidentialité et l'intégrité des informations dont ils assurent le traitement.

② Confiance dans la traçabilité fournie par les systèmes quant à leur conception et leur origine.

③ Confiance dans la robustesse des systèmes aux variations de paramètres lors de la conception et de l'usage.

Pour assurer cette confiance, il faut donc développer des outils et des approches permettant de vérifier les propriétés de sécurité et de testabilité du CI. La sécurité matérielle consiste à étudier les attaques et contremesures pouvant être mises en oeuvre pour garantir la confidentialité, l'intégrité et l'authenticité des données manipulées.

① Confidentialité : les données ne sont pas accessibles à un individu ou une entité non autorisée.

② Intégrité : les données ne peuvent pas être modifiées de façon non autorisée.

③ Authenticité : les circuits sont ce qu'ils indiquent, ils ne peuvent pas être remplacés.

1.8 Contributions de cette thèse

Dans le cadre de cette thèse, les contributions portent sur la proposition d'approches permettant de détecter la présence de CTMs en phase de test et en-ligne, mais aussi de prévenir leur insertion pendant la fabrication.

1.8.1 Contribution 1 : Implémentation de CTMs dans le LEON2 et l'AES

Dans le chapitre 3, nous étudions la faisabilité de l'insertion de CTMs dans le flot Application-Specific Integrated Circuit Standard (ASIC) dans tous les niveaux d'abstraction (Registers Transfer Level (RTL), netlist et layout). Pour ce faire, nous utilisons comme benchmarks l'AES (un algorithme de chiffrement symétrique) et le LEON2. En fonction des types de CTMs définis dans la littérature, nous avons développé des CTMs en fonction de leur mécanisme d'activation et de l'effet souhaité.

- Pour l'AES, nous avons développé des CTMs qui ont un mécanisme d'activation combinatoire ou séquentiel. Pour ce qui est de l'effet, nous avons choisi d'implémenter des CTMs qui font fuir des informations (la clé), modifier des signaux internes (sabotage) et désactiver certaines fonctionnalités (Denial-of-Service (DoS)).
- Pour le LEON2, tous les CTMs que nous avons implémentés ont un mécanisme d'activation séquentiel. Ces CTMs ont pour but de désactiver certaines fonctionnalités, comme les interruptions timer (DoS) ou de dégrader les performances.

1.8.2 Contribution 2 : Détection de CTM en phase de test

La méthode que nous proposons dans le chapitre 4 [DBF+15], consiste à identifier les signaux sur lesquels un CTM potentiel pourrait être inséré puis générer les vecteurs de test permettant de les déclencher en comparant la réponse du circuit avec les réponses attendue. Les critères pris en compte pour identifier ces signaux sont censés refléter le mieux possible les choix d'un attaquant dans une fonderie.

- Sélection des signaux suspects : Trois critères sont considérés :
  - ① Un CTM furtif est une combinaison de signaux faiblement contrôlable. La première étape consiste à calculer la probabilité de chaque signal de valoir '0' ou '1'. Nous sélectionnons les signaux ayant des probabilités déséquilibrées.
  - ② Nous considérons que les signaux utilisés par le CTM ne doivent pas dégrader le délai du circuit et donc la fréquence de fonctionnement. La deuxième étape consiste donc à identifier les signaux qui ont une marge temporelle suffisante.
  - ③ Création des sous-groupes : Comme pour le délai, l'insertion de CTM ne doit pas modifier le layout pour éviter une détection optique. La dernière étape consiste donc à créer des groupes de signaux parmi ceux sélectionnés avec les deux premiers critères en fonction de leur proximité dans le layout.
- Génération des vecteurs de test : Une fois que tous les ensembles de signaux ont été générés, le vecteur de test permettant d'exciter chaque ensemble de signaux est déterminé. Un outil d'Automatic Test Pattern Generation (ATPG) est utilisé pour trouver ces vecteurs.

1.8.3 Contribution 3 : Détection de CTM en-ligne

Nous proposons dans le chapitre 5 une méthode de duplication de la netlist appelée Modified Dual Modular Redundancy (MODMOR) [PBD+16]. Cette approche consiste à générer deux netlists structurellement différentes mais fonctionnellement équivalentes. Cette architecture permet de comparer les sorties des deux répliques et détecter une anomalie (effet d'un CTM) en temps réel. De plus, les deux répliques étant structurellement différentes, il est très difficile voire impossible pour attaquant d'insérer des CTMs identiques dans les deux répliques.

1.8.4 Contribution 4 : Prévention d'insertion de CTM par remplissage du layout

Nous proposons également une technique basée sur le Design for Hardware Trust (DfHT) dans le chapitre 6 [BDP+16]. Elle consiste à créer un layout aussi dense que possible dans le but de prévenir l'insertion d'un CTM au cours de la fabrication. À partir d'un concept introduit dans [XT13], notre principale hypothèse est qu'un attaquant n'ira pas modifier les « standard cells » mais plutôt enlever les « filler cells » pour les remplacer par un CTM. Ces « filler cells » n'assurent aucune fonction logique, elles sont insérées dans les espaces vides du layout dans le but d'améliorer la densité. Nous proposons de remplacer les espaces vides avec des cellules standard à la place des filler cells. Ces cellules sont interconnectées pour former des fonctions supplémentaires. Le test de ces fonctions permet de vérifier leur non remplacement par un CTM.

1.8.5 Contribution 5 : Prévention d'insertion de CTM par insertion de points de test

Dans le chapitre 7, nous proposons une méthode facilitant la détection par test logique. L'objectif est de créer un circuit où tous les signaux sont contrôlables (probabilités équilibrées), ceci afin d'éviter qu'un attaquant puisse insérer un CTM furtif. Cela rendra de ce fait la détection par test logique plus efficace. Cette approche consiste à sélectionner les signaux ayant des probabilités déséquilibrées et ajouter de la logique combinatoire et séquentielle pour équilibrer leurs probabilités.

1.8.6 Plan du manuscrit

Dans le chapitre 2, nous donnons les différentes caractéristiques d'un CTM ainsi que l'état de l'art des méthodes permettant de prévenir leur insertion ou de les détecter. Dans les chapitres suivants, nous présentons chacune des contributions avant de donner les conclusions générales et perspectives.

1.9 Références

[BDP+16] Papa-Sidy Ba, Sophie Dupuis, Manikandan Palanichamy, Marie-Lise Flottes, Giorgio Di Natale, and Bruno Rouzeyre. Hardware trust through layout filling : a hardware trojan prevention technique. In *IEEE Computer Society Annual Symposium on VLSI*. ISVLSI, 2016.

[CNB09] Rajat Subhra Chakraborty, Seetharam Narasimhan, and Swarup Bhunia. Hardware trojan : Threats and emerging solutions. In *High Level Design Validation and Test Workshop*, pages 166–171. HLDVT, 2009.

[DBF+15] Sophie Dupuis, Papa-Sidy Ba, Marie-Lise Flottes, Giorgio Di Natale, and Bruno Rouzeyre. New testing procedure for finding insertion sites of stealthy hardware trojans. In *Design, Automation & Test Exhibition*, pages 776–781. DATE, 2015.

[esp16] espci. Transistors mos., 2016. https://cours.espci.fr/site.php?id=129&fileid=884.

[Koc96] Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer, 1996.

[PBD+16] Manikandan Palanichamy, Papa-Sidy Ba, Sophie Dupuis, Marie-Lise Flottes, Giorgio Di Natale, and Bruno Rouzeyre. Duplication-based concurrent detection of hardware trojans in integrated circuits. In *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices*, pages 1–4. TRUDEVICE, 2016.

[ROM04] Krunoslav ROMANJEK. *CARACTÉRISATION ET MODÉLISATION DES TRANSISTORS CMOS DES TECHNOLOGIES 50nm ET EN DEÇA*. PhD thesis, INPG, 2004.

[Sch16] Thomas Scherer. 50 years of moore's law., 2016. https://www.elektormagazine.com/articles/moores-law.

[XT13] Kan Xiao and Mohammed Tehranipoor. Bisa : Built-in self-authentication for preventing hardware trojan insertion. In *Hardware-Oriented Security and Trust*, pages 45–50. HOST, 2013.

Chapitre 2

Taxonomie et état de l'art de lutte contre les CTMs

« *After climbing a great hill, one only finds that there are many more hills to climb.* »

Nelson Mandela

Sommaire

2.1 Introduction

Comme nous l'avons mentionné auparavant, la fabrication des CIs dans des fonderies situées dans des pays à bas coût, le recours à des blocs IP conçus dans des entreprises tierces ainsi que la multitude d'outils de CAO, soulèvent des questions sur la confiance que l'on peut attribuer aux CIs. En effet, comme vu au chapitre précédent, un CTM peut être inséré durant durant la fabrication, ou dans un IP. Différentes applications peuvent être affectées par un CTM [WTP08] :

- Applications militaires : les systèmes de contrôle d'armes, les systèmes de communications, les systèmes de collecte d'informations, etc.
- Applications aéronautiques et spatiales : systèmes électroniques embarqués dans les navettes, les satellites, les avions, etc.
- Applications civiles critiques : systèmes de gestion d'informations commerciales confidentielles (banques, bourses, etc.), systèmes de gestion d'informations personnelles confidentielles (dossier médical, dossier financier, etc.).
- Applications courantes : tout système électronique qui gère des informations confidentielles.
- Intellectual Property (IP) destiné à la sécurité : les vendeurs de CI sans unité de fabrication, les vendeurs d'IP.
- Les moyens de transports : systèmes d'alerte, système de gestion d'informations sécurisées, etc.

Dans [Ade08; AK08; KTC+08], quelques mésaventures militaires sont mentionnées, que l'on soupçonne d'être dues à la présence d'un CTM. C'est pourquoi, des industriels ainsi que des universitaires ont proposé une classification et des méthodes permettant de lutter contre les CTMs. Dans ce chapitre, nous donnerons les caractéristiques des CTMs et ferons un état de l'art des méthodes de détection et de prévention des CTMs.

2.2 Taxonomie des CTMs

2.2.1 Définition de Cheval de Troie Matériel

Dans la littérature, un CTM est défini comme une altération intentionnelle d'un circuit intégré dans le but de modifier ses caractéristiques [WTP08; BHBN14a]. Un CTM est furtif par nature (c'est à dire qu'il se déclenche très rarement) et peut modifier certaines fonctionnalités du circuit. La figure 2.1 est un exemple de CTM.
Un CTM est constitué de deux parties distinctes :

- Le trigger : Mécanisme d'activation du CTM.
- La payload : Matérialisation de l'effet du CTM sur le circuit lorsqu'il est déclenché.

2.2.2 Caractéristiques d'un CTM

Les CTMs sont classés dans la littérature suivant leurs caractéristiques physiques, les caractéristiques du mécanisme d'activation (le trigger) et les caractéristiques du mécanisme d'action (la payload). La figure 2.2 donne la taxonomie d'un CTM.

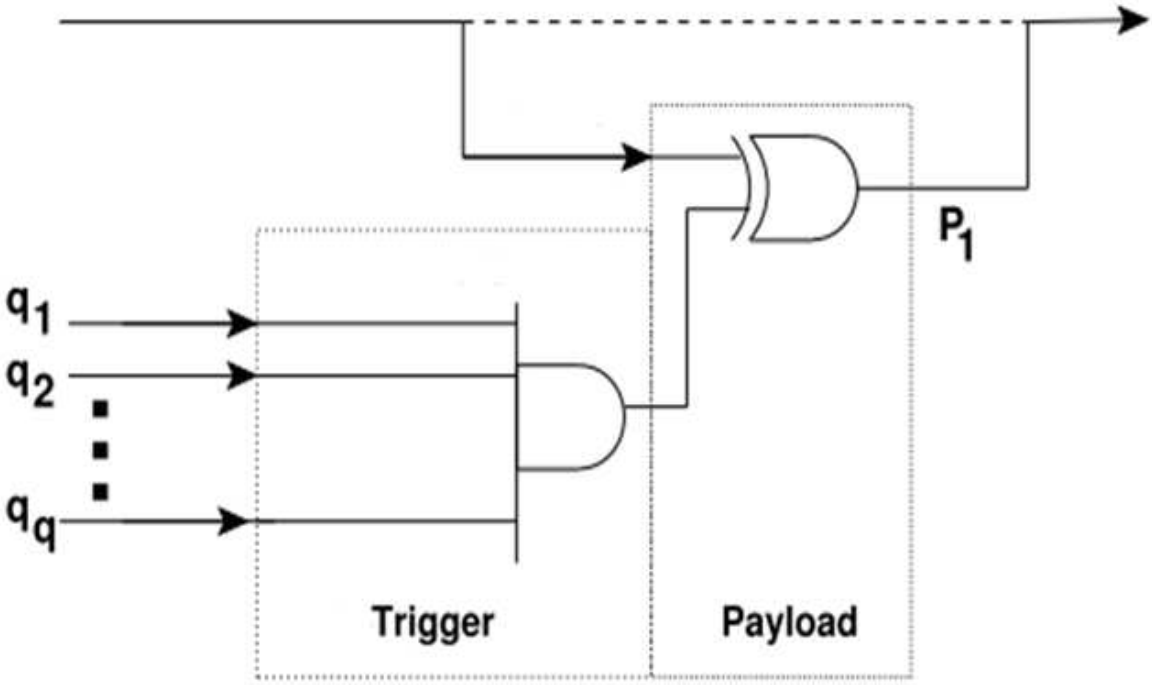


FIGURE 2.1 – Modèle de CTM.

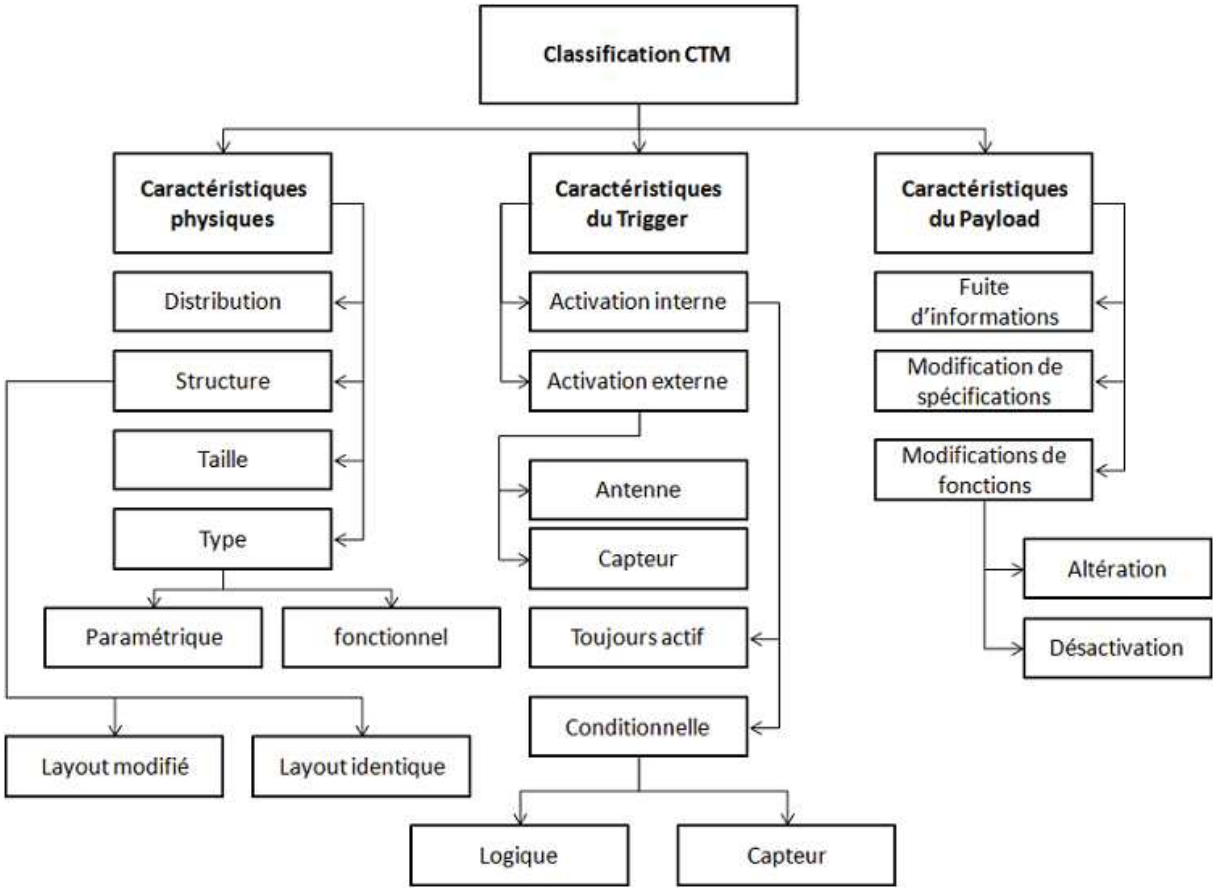


FIGURE 2.2 – Taxonomie d'un CTM [TSZ14].

2.2.2.1 Caractéristiques physiques

Les caractéristiques physiques d'un CTM peuvent être décomposées en quatre catégories :

Le type

Il y a deux types de CTM, paramétrique et fonctionnel. Un CTM fonctionnel est obtenu en ajoutant ou en supprimant de la logique dans le but de modifier la fonctionnalité du circuit. Á l'inverse un CTM paramétrique est conçu de façon à compromettre la fiabilité ou augmenter la probabilité d'un défaut fonctionnel (par une modification de la géométrie du layout par exemple).

La taille

Elle dépend du nombre de composants ajoutés ou supprimés pour créer le CTM. La taille peut être un facteur important pendant l'activation. Un petit CTM a une probabilité plus élevée d'être activé qu'un CTM de grande taille avec un nombre d'entrées plus important.

La distribution

Elle décrit la localisation du CTM dans le layout. Par exemple, une distribution serrée signifie que les composants du CTM sont proches sur le layout. Par contre une distribution relâchée signifie que les composants sont dispersés dans le layout. La figure 2.3 donne quelques exemples. La distribution des CTMs dépend des espaces vides dans le layout, ainsi l'attaquant peut être contraint de placer et router des petites portions du CTM dans différents espaces vides du layout.

La structure

Si l'attaquant modifie le layout pour insérer son CTM, cela peut entraîner un placement différent des composants du circuit, voire même modifier ses dimensions. Toute modification du layout peut modifier le délai ou la consommation et rendre plus facile la détection du CTM. Dans le but de minimiser la probabilité de détection du CTM, l'attaquant choisira un encombrement faible avec un CTM de petite taille.

2.2.2.2 Caractéristiques d'activation (Trigger)

Un attaquant, cherchera à rendre l'activation du CTM très difficile pour éviter une détection pendant la phase de test. De ce fait l'activation du CTM peut être considérée comme un « événement rare » d'un point de vue statistique. L'utilisation du terme « furtif » décrit le mieux ce phénomène. Il existe deux types de triggers, les triggers combinatoires et les triggers séquentiels.

2.2.2.3 Caractéristiques d'action (Payload)

Une classification des différentes catégories est illustrée sur la figure 2.2. On distingue les trois catégories suivantes :

- Modification de fonction : désigne les CTMs qui ont pour effet de modifier la fonction logique du circuit.

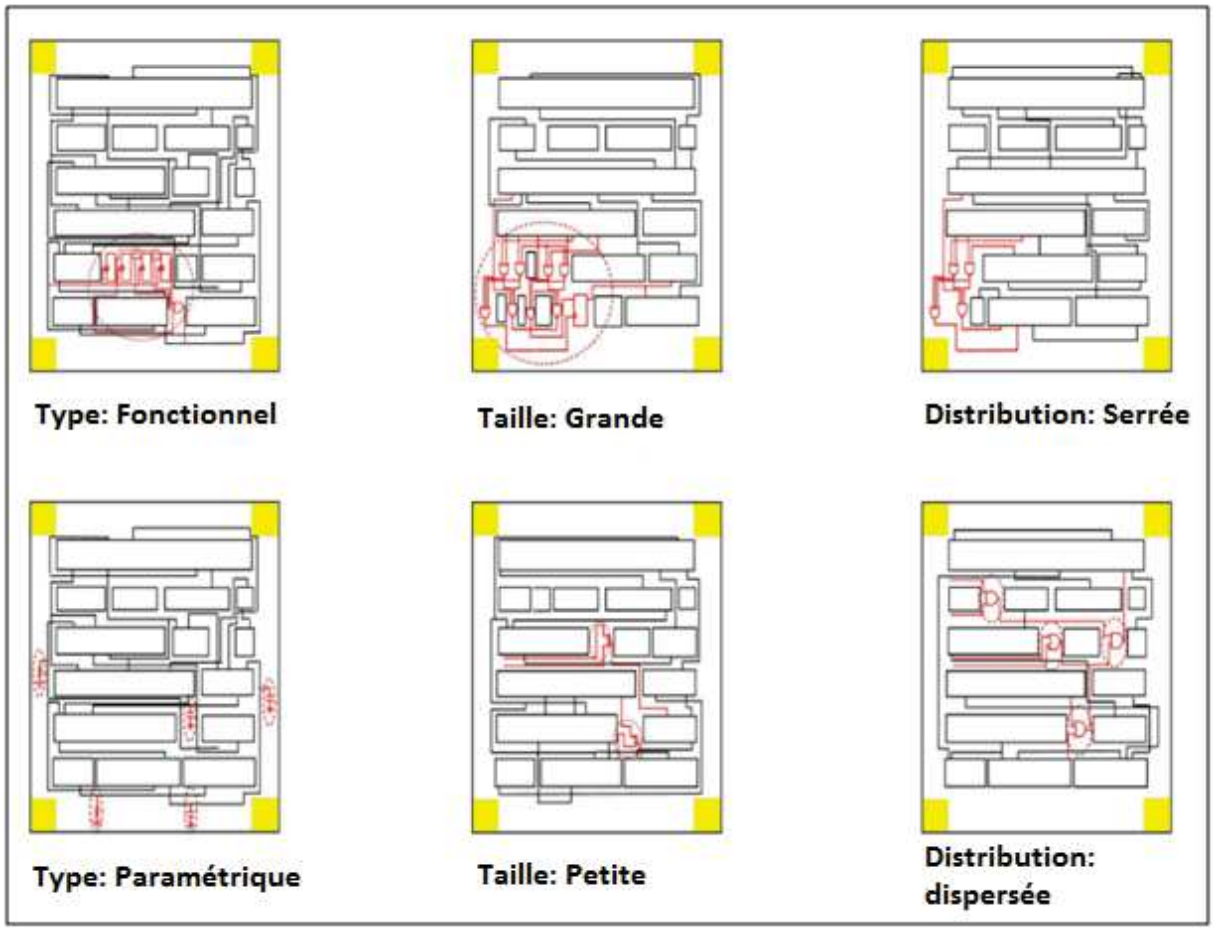


FIGURE 2.3 – Exemples de CTM avec différents caractéristiques [TK10].

- Modification de spécification : désigne les CTMs qui ont pour effet de modifier les propriétés paramétriques du circuit telles que le délai ou la consommation. Cette catégorie peut être considérée comme du sabotage, car entraînant une dégradation de performances de circuit.
- Transmission d'informations : désigne les CTMs qui ont pour effet de faire fuir des informations capitales du circuit.

2.3 Méthodes de détection

Du fait de la menace que représente l'insertion de CTMs dans les CIs, notamment en phase de fabrication, l'industrie des semi-conducteurs ainsi que le monde académique ont mené beaucoup de travaux ces dernières années pour essayer de trouver des méthodes permettant de détecter mais aussi de prévenir l'insertion de CTMs dans les CIs.
La figure 2.4 donne une classification des différentes méthodes de détection et de prévention que l'on retrouve dans la littérature.

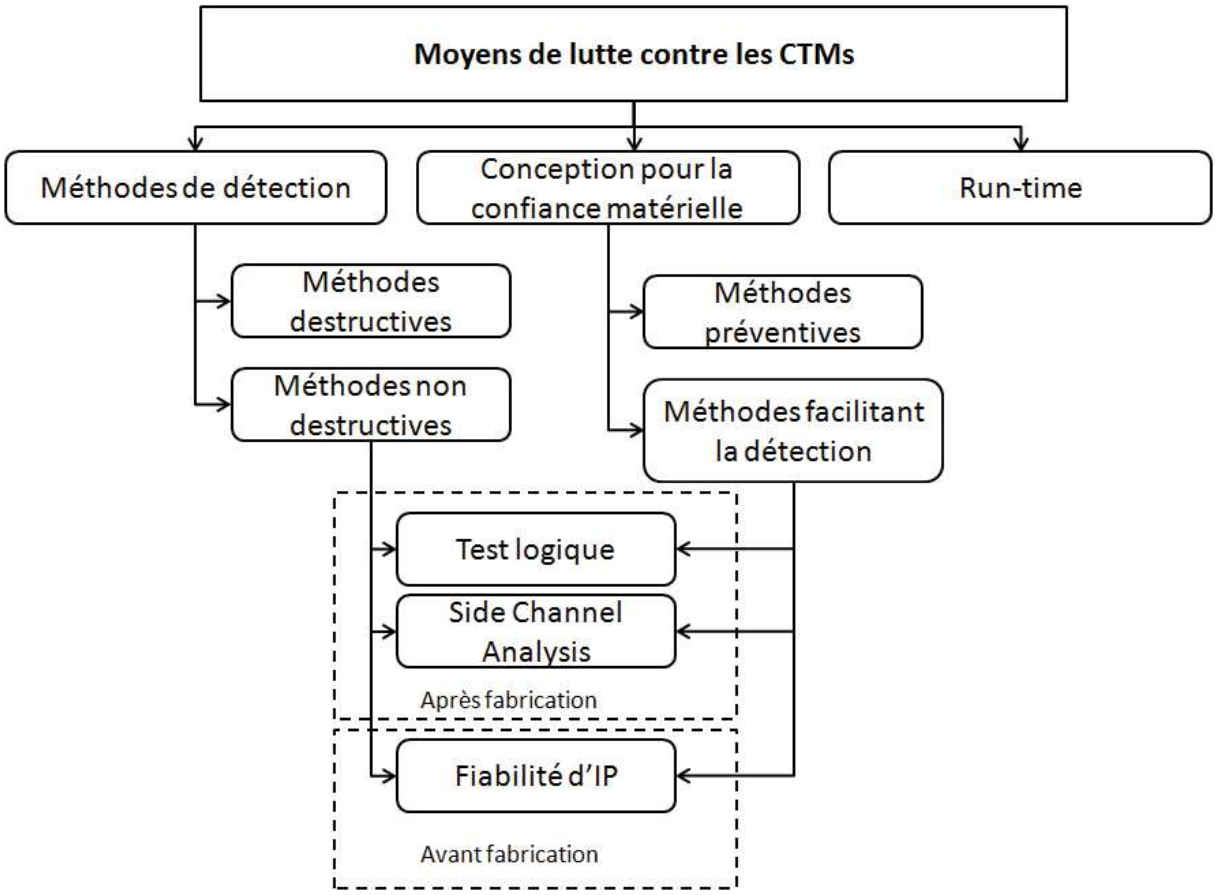


FIGURE 2.4 – Techniques de détection des CTMs [BHBN14b].

Il existe deux types de méthodes de détection : les méthodes destructives et les méthodes non destructives.

2.3.1 Méthodes destructives

Elles sont basées sur l'inspection visuelle et le reverse engineering. Elles consistent à inspecter physiquement le circuit pour vérifier qu'il n'y a pas eu d'attaque et qu'il est conforme à

un modèle de référence (appelé golden model).

Dans [KTK02; CLMFT15; Chi16], l'approche consiste à utiliser un échantillon du produit fabriqué pour dé-métalliser le circuit en utilisant un procédé chimique (Chemical Metal Polishing (CMP)), avant de faire une reconstitution d'image en utilisant une microscopie à balayage électronique (Scanning Electron Microscope (SEM)).
Cependant de telles approches sont très coûteuses et les résultats obtenus à partir d'un échantillon ne permettent pas de faire une extrapolation pour un lot entier.

2.3.2 Méthodes non destructives

On peut les diviser en trois catégories :

- Détection au niveau IP (avant fabrication),
- Détection en phase de test : Test logique, Side Channel Analysis (SCA) (après fabrication),
- Détection en-ligne ou run-time monitoring (après fabrication).

2.3.2.1 Au niveau IP

En général les blocs IP sont classés en trois catégories [TSZ14] :

- Les IPs "soft" sont décrits en VHDL ou Verilog et sont plus flexibles.
- Les IPs "firm" sont synthétisés avec une librairie spécifique.
- Les IPs "hard" sont fournis sous forme de GDSII.

Un CI peut être composé de plusieurs blocs IPs d'origines diverses. Ainsi, un CTM peut être inséré dans un bloc IP par le concepteur dans le but d'espionner les autres blocs IPs du CI. La détection de tels CTMs est très difficile car un modèle de référence n'est pas disponible. Ainsi les méthodes de détection consistent à vérifier les propriétés de l'IP et la fiabilité du code source [BHBN14b].

Dans [BH10], les auteurs proposent une méthode pour détecter et isoler un CTM furtif dans un bloc IP. Le principe de cette méthode repose sur quatre étapes :

- La première étape consiste à utiliser des vecteurs fonctionnels (vecteurs de test, vecteurs aléatoires, vecteurs de validation) pour activer et observer autant de signaux que possible du circuit suspect. Les signaux non activés par les vecteurs fonctionnels sont considérés comme des candidats suspects.
- Une fois les signaux facilement contrôlables éliminés dans l'étape précédente, un ATPG est utilisé pour filtrer les signaux restants. Les signaux non testables et non observables sont retirés de la liste des candidats suspects.
- La troisième étape consiste à faire une vérification d'équivalence (equivalence checking) en utilisant un SAT pour trouver un comportement anormal entre l'IP suspect et l'IP sain. La vérification d'équivalence consiste à comparer deux circuits structurellement différents pour assurer qu'ils sont fonctionnellement identiques. Les deux circuits reçoivent les mêmes vecteurs à leurs entrées primaires. Les sorties primaires des deux circuits sont connectées à une porte XOR et les sorties des portes XOR sont connectées à une porte OR pour dériver la sortie finale. Pour un vecteur donné, si les sorties diffèrent, la sortie de la porte OR sera '1', ce qui indique que les deux circuits

sont fonctionnellement différents. Si le SAT donne UNSAT, cela signifie que les deux circuits sont différents.

- L'étape finale permet d'isoler les régions infectées par le CTM. Chaque région est caractérisée par une porte et un rayon. Le rayon définit l'étendue de la portion de circuit couverte par la région. Chaque signal suspect est associé à la porte à laquelle il est connecté. Puisque le nombre de ces portes est petit, le nombre de régions est aussi petit. Le rayon de chaque région est progressivement élargi pour contenir toutes les portes dans la liste de candidats suspects. Le but est de trouver un petit nombre de portes susceptibles d'être les supports du CTM. Les résultats montrent que cette méthode est très efficace pour identifier un bloc IP altéré et isoler la partie du circuit contenant les portes du CTM. Cependant, cette méthode nécessite un bloc IP de référence pour faire la vérification d'équivalence.

Dans [LJM12], les auteurs proposent un nouveau protocole qui permet de valider la fiabilité des blocs IP achetés à des entités tierces. La figure 2.5 illustre le protocole d'interaction entre le client et le vendeur de blocs IP. La méthode repose sur le Proof Carrying Code (PCC) pour valider les propriétés de sécurité.
L'approche consiste à définir une liste de propriétés de sécurité que doit remplir le bloc IP. Ces propriétés sont ensuite traduites en propriétés mathématiques en utilisant le « theorem-proving language ». Ces propriétés mathématiques sont ensuite traduites en code HDL et ajoutées au bloc IP. Le langage Coq est utilisé pour modéliser le code HDL. Le client vérifie la conformité du bloc IP fourni en utilisant le « Coq langage interpreter ». La première limite de cette méthode se situe au niveau de la génération de liste de propriétés de sécurité, car elle ne peut pas être exhaustive. La deuxième limite est la représentation de ces propriétés en langage HDL et la modélisation du circuit en « theorem-proving language ».

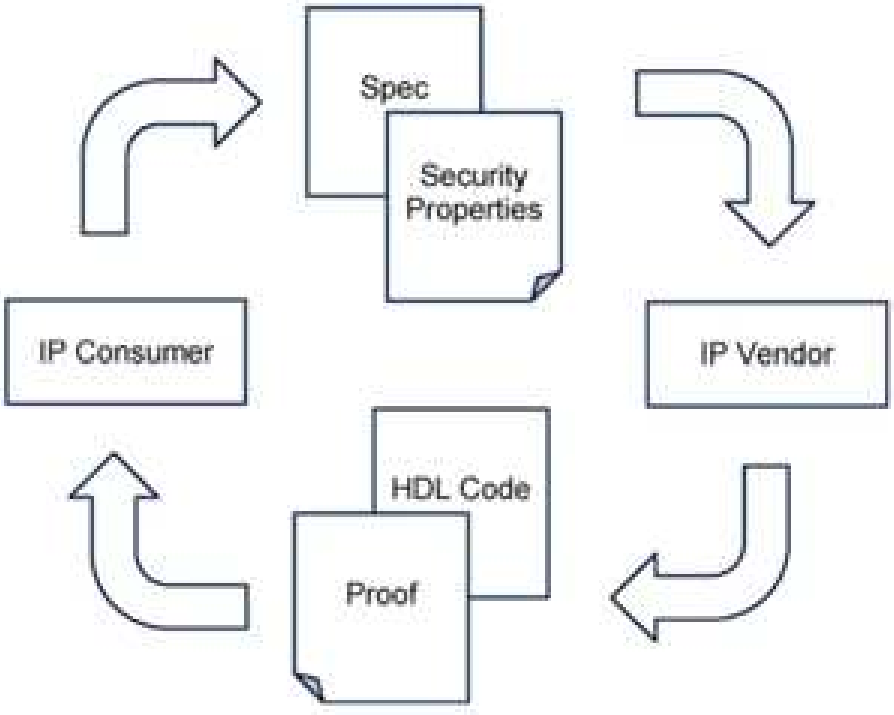


FIGURE 2.5 – Protocole d'acquisition été de transfert d'IP basé sur le PCC [LJM12].

2.3.2.2 Détection par test logique

Il s'agit de trouver les vecteurs de test permettant de déclencher et d'observer les effets du CTM sur les sorties primaires du circuit. Ce type de méthode est dédié aux CTMs de petite taille qui sont dits furtifs. Il est donc difficile voire impossible de détecter de tels CTMs avec des méthodes side channel.

À notre connaissance,la première méthode détection de CTM basée test logique a été présentée dans [WPBC08]. L'objectif est de trouver les sites potentiels pour attacher le trigger du CTM et générer les vecteurs de test en fonction de ces sites. La procédure d'identification des triggers potentiels et de génération de patterns se fait de la façon suivante :

- Identifier les signaux faiblement contrôlables avec une simulation exhaustive. Les triggers potentiels peuvent être des combinaisons de ces signaux.
- Identifier les signaux faiblement observables en utilisant un simulateur de fautes. Ces signaux peuvent être utilisés pour la payload du CTM.
- À partir d'un ensemble Q de triggers potentiels, et P Payloads potentiels, considérer Q × P CTMs potentiels.
- Utiliser un outil d'ATPG pour trouver les patterns pour chaque CTM potentiel.

Nous détaillerons la bibliographie de cette partie dans le chapitre 4.

2.3.2.3 Détection par analyse de canaux cachés (SCA)

Elles sont complémentaires au test logique car sont plus adaptées pour les CTMs de grande taille.

Analyse de consommation

Dans [RPT08], les auteurs proposent une méthode de détection de CTM basée sur l'analyse de consommation. Le principe repose sur l'analyse des mesures du courant transitoire IDDT obtenues à partir de plusieurs ports du circuit. les courants IDDT sont mesurés pour chaque séquence de test appliquée sur les entrées primaires du circuit. Les auteurs utilisent une technique d'analyse statistique robuste pour détecter la présence d'un CTM. Les mesures sont réalisées en premier sur le circuit sans CTM pour réduire les variations de process. Les données sont ensuite tracées sur un graphe de dispersion. La moyenne et la variance des données de chaque graphe de dispersion sont utilisées pour dériver une ellipse. La points contenus dans l'ellipse appartiennent au circuit sain. Les points qui se trouvent en dehors de l'ellipse appartiennent au CTM. La figure 2.6 illustre la détection d'un CTM à 3 entrées (T3).
Les résultats indiquent qu'il est possible de détecter des CTMs sans les activer. Cependant cette méthode de détection est sensible aux variations de process et nécessite aussi un modèle de référence.

Dans [BH08], les auteurs proposent un partitionnement du circuit en plusieurs régions pour faciliter la localisation d'un CTM dans un circuit. Ce partitionnement permet de détecter une variation de consommation introduite par l'insertion de CTM. Tout d'abord, le circuit est partitionné en plusieurs régions. Ensuite un pic de consommation est créé pour chaque région en stimulant le circuit avec des vecteurs qui maximisent l'activité de la région concernée tout en minimisant l'activité du reste du circuit. Si le CTM est présent dans l'une des

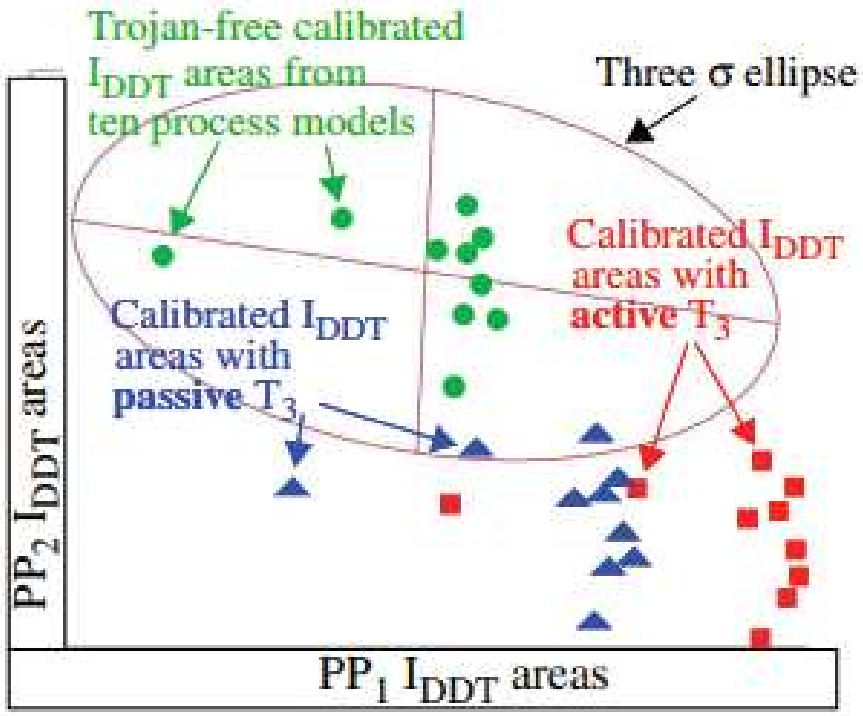


FIGURE 2.6 – Courbe de dispersion d'un circuit dain et d'un circuit avec CTM [RPT08].

régions, l'activité du circuit sain sera différente de celle du circuit infecté. La différence des courbes de puissances obtenues à partir des deux circuits (sain et infecté) permet donc de détecter la présence d'un CTM. Cependant l'efficacité de cette méthode dépend de la sélection des régions et des vecteurs de test pour maximiser la puissance de chaque région. Elle est sensible aussi aux variations de process et nécessite un modèle de référence.

Analyse de délai

Dans [JM08], les auteurs proposent une méthode de génération d'empreintes en utilisant les délais de chemins du circuit entier. Chaque délai de chemin représente une partie des caractéristiques du circuit. Ainsi des séries d'empreintes des délais de chemins sont générées pour détecter la présence d'un CTM.
La procédure de détection se fait en trois étapes :

- Des vecteurs de test avec un taux de couverture élevé sont appliqués sur les CIs et les délais des chemins sont collectés.
- En fonction des délais des chemins, des séries d'empreintes de délai sont générées.
- Les délais des chemins des circuits testés sont comparés aux empreintes de délai générées.

Cette méthode permet la détection de CTM insérés en fabrication avec une variation de process de ±7.5% et les empreintes utilisées pour caractériser le circuit sain sont plus précis que les courbes de consommation. La limite de cette méthode est la détection CTM à payload implicite qui introduit un délai très petit. Cela nécessite d'autres méthodes d'analyse et plus

d'empreintes. Elle est aussi sensible aux variations de process.

Dans [EZRR15], les auteurs proposent une méthode de détection de CTMs par la mesure de délais des chemins critiques du circuit. Pour mesurer de façon précise les délais des chemins critiques, les auteurs modulent la période d'horloge. Les délais de chemins mesurés sont comparés aux délais d'un modèle de référence pour détecter une variation induite par la présence de CTM. Cependant, la variation de process peut induire une variation de délai plus importante que celle induite par l'insertion de CTM. Pour minimiser l'effet de la variation de process, les auteurs proposent un modèle de délai qui prend en compte les variations inter-die, les variations de process intra-die et les conditions d'expérimentations. Cette méthode permet la détection de petits CTMs qui ont un impact très faible sur les délais de chemins. Cependant, tout comme les autres méthodes, elle nécessite un modèle de référence.

Analyse d'émanations électromagnétiques

Dans [NNB+14], les auteurs proposent une méthode de détection de CTM en utilisant les radiations électromagnétiques (EM) comme canal auxiliaire. Les radiations électromagnétiques apportent une meilleure représentation spatiale et temporelle que les mesures de consommation. De plus, l'impact d'un CTM sur les courbes EM est plus important que celui dû aux variations de process. Le principe consiste à comparer les courbes EM entre un lot de circuits sains et un lot de circuits infectés. La différence de ces courbes permet de détecter la présence d'un CTM. Tout d'abord, plusieurs mesures EM sont réalisées pour chaque circuit. Ensuite la moyenne des courbes EM est calculée pour chaque circuit. Enfin la différence des courbes par rapport à la moyenne est calculée pour chaque circuit.

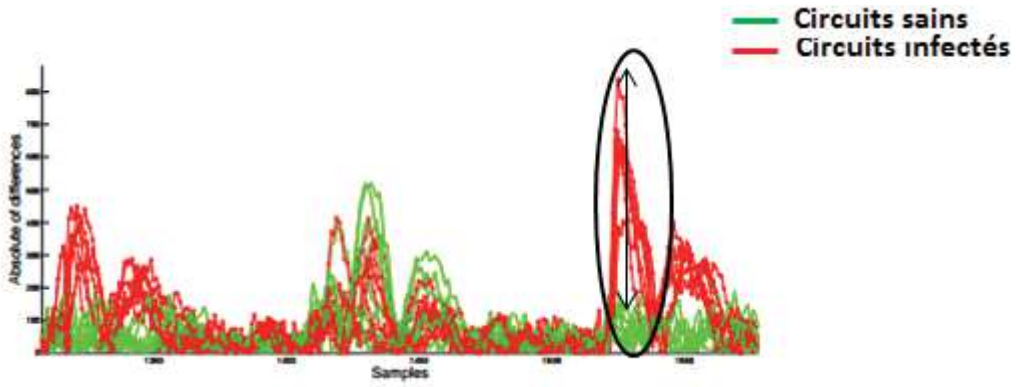


FIGURE 2.7 – Écart des courbes EM par rapport à la moyenne. [NNB+14].

La figure 2.7 illustre les courbes EM d'un lot de circuits sains et d'un lot de circuits infectés. Les courbes vertes représentent les différences des courbes EM par rapport à la moyenne pour le lot de circuits sains. Les courbes rouges représentent les différences des courbes EM par rapport à la moyenne pour le lot de circuits infectés. La différence entre les courbes vertes est due à la variation de process. L'écart entre les courbes rouges et vertes est plus important que l'écart dû à la variation de process. Ceci permet de détecter la présence d'un CTM. Les résultats montrent qu'en utilisant les radiations EM, la probabilité de détection est supérieure à 99% pour un taux de faux positif de 0.017% pour des CTMs dont la surface est supérieure à 1% de la surface du circuit.

Dans [CNB09], les auteurs donnent un tableau comparatif des avantages et inconvénients du test logique et du SCA (cf 2.1).

TABLEAU 2.1 – Avantages et inconvénients des approches test logique et SCA [CNB09]

	Approche test logique	**Approche SCA**
Avantages	(a) Efficace pour les petits CTMs (b) Insensible aux variations de process	(a) Efficace pour les CTMs gros (b) Génération de patterns facile
Inconvénients	(a) Génération de patterns complexe (b) Détection de CTMs gros difficile	(a) Sensible au bruit de process (b) Détection de petits CTMs difficile

Les techniques mentionnées ci-dessus ne peuvent pas garantir à 100% la détection de présence de CTM dans le circuit, ou peuvent ne pas être adaptées au type d'application visée. Par conséquent, il est nécessaire d'explorer d'autres alternatives comme le monitoring en temps réel (run-time).

2.3.2.4 Détection en-ligne

Ces méthodes peuvent être divisées en deux catégories : celles qui implémentent de la logique pour vérifier que certaines propriétés ne sont pas violées ou pour détecter un comportement anormal et celles qui utilisent la redondance.

Ajout de logique de sécurité

Dans [AB09; BAA+13], les auteurs proposent une approche basée sur l'ajout de logique reconfigurable pour surveiller en temps-réel les fonctionnalités en utilisant un moniteur de sécurité (Security Monitor (SM)) comme illustrée dans la figure 2.8.
Le SM est une machine d'états qui vérifie le comportement des signaux sélectionnés. La vérification peut être faite en parallèle avec le fonctionnement normal, et déclencher une alarme quand une anomalie est détectée (créée par un CTM), par exemple l'accès à une zone mémoire protégée.

Redondance

Dans [MWP+09], une approche basée sur des exécutions parallèles est proposée. Le principe consiste à répliquer le matériel de façon à avoir des versions différentes d'une même fonction et qui vont opérer en parallèle. Les résultats des différentes répliques vont être comparés à chaque cycle d'horloge pour voir s'il n'y a pas eu anomalie. Cette partie sera abordée plus en détail dans le chapitre 5.

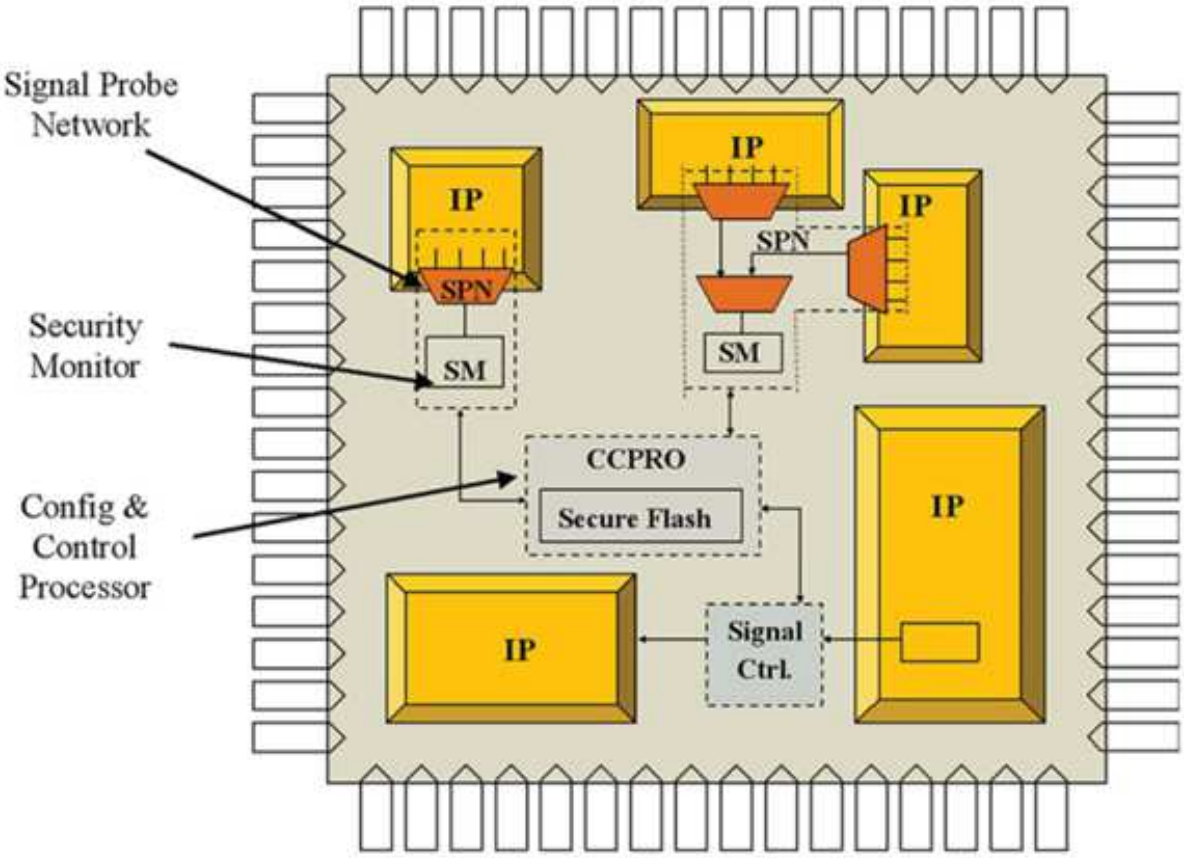


FIGURE 2.8 – Monitoring de CTM en utilisant une infrastructure reconfigurable [AL09].

2.4 Méthodes de prévention

Elles peuvent être divisées en deux catégories, les méthodes permettant de prévenir l'insertion de CTM au niveau netlist ou layout et les méthodes permettant de faciliter leur détection.

2.4.1 Méthodes permettant de prévenir l'insertion de CTMs

Elles peuvent être réparties comme suit :

- Les méthodes qui permettent de camoufler le circuit et d'en cacher les fonctionnalités et les propriétés structurelles pour éviter le reverse engineering.
- Les méthodes qui permettent de rendre difficile voire impossible l'insertion de composants dans le layout.

2.4.1.1 Au niveau netlist

Dans [CB11], les auteurs proposent une méthode de camouflage de la netlist en utilisant une clé pour modifier le graphe de transition de la machine à états (FSM) du circuit. Cette modification permet de faire fonctionner le circuit en deux modes distincts : le mode normal et le mode camouflé (obfuscated mode). Le mode normal est obtenu si une séquence spécifique de vecteurs est appliquée sur les entrées primaires. Cette séquence correspond à la clé. Ceci permet de cacher à l'attaquant les signaux qui sont dits « rares », de rendre difficile l'insertion de CTMs dits « furtifs » et d'avoir un taux de détection très élevé. La figure 2.9 donne la procédure de modification du graphe de transition.

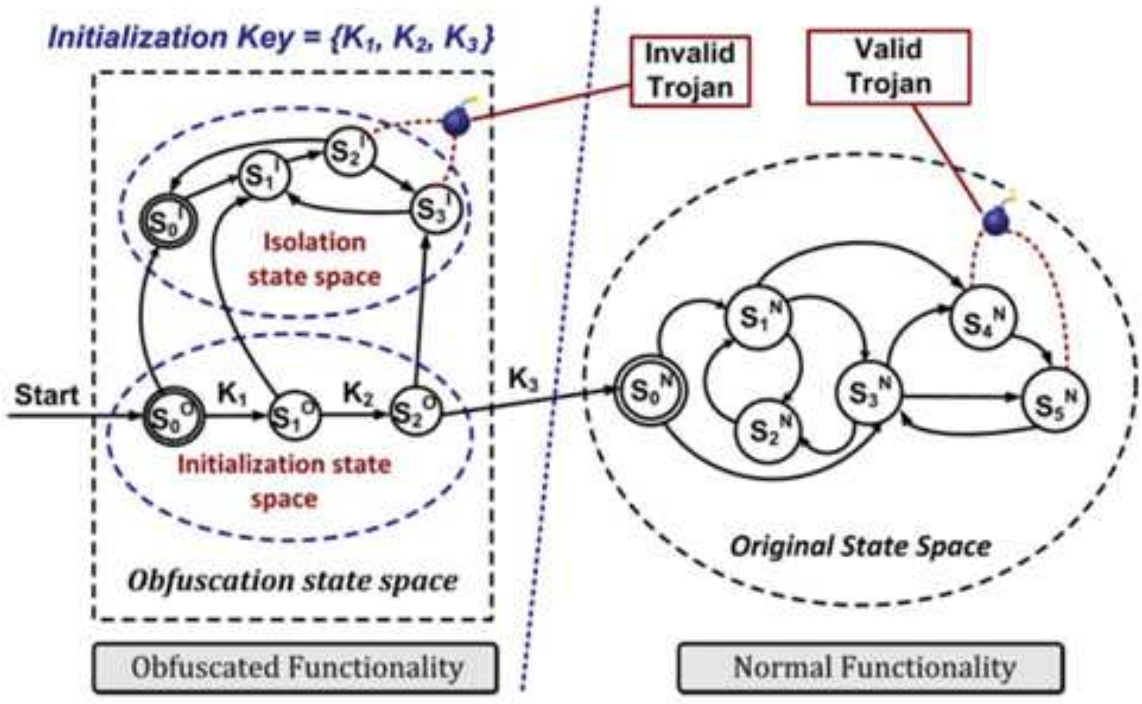


FIGURE 2.9 – Schéma d'obfuscation : modidfication du graphe de transition [CB11].

L'obfuscation est réalisée après deux modifications importantes du graphe de transition d'états (State Transition Graph (STG)).

- Ajout d'états ;
- Certains états, non justifiables dans le circuit original, sont utilisés en mode masqué.

En appliquant la bonne séquence de clé d'initialisation (k1, k2, k3), l'état initial du mode normal est activé et le circuit bascule en mode normal. Par contre si une mauvaise séquence de clé est appliquée le circuit passe en mode masqué.

2.4.1.2 Au niveau layout

Nous avons vu que la fabrication constitue une étape critique du fait des fonderies non fiables, car le layout peut être manipulé par un attaquant pour y insérer un CTM ou faire du reverse engineering. Donc il est nécessaire de développer des méthodes pour rendre le layout robuste face à de telles attaques.

Dans [XT13], les auteurs proposent une méthode qui permet de densifier le layout en remplissant les espaces vides avec des cellules fonctionnelles au lieu de cellules non fonctionnelles (filler cells). Nous aborderons cette méthode plus en détail dans le chapitre 6.

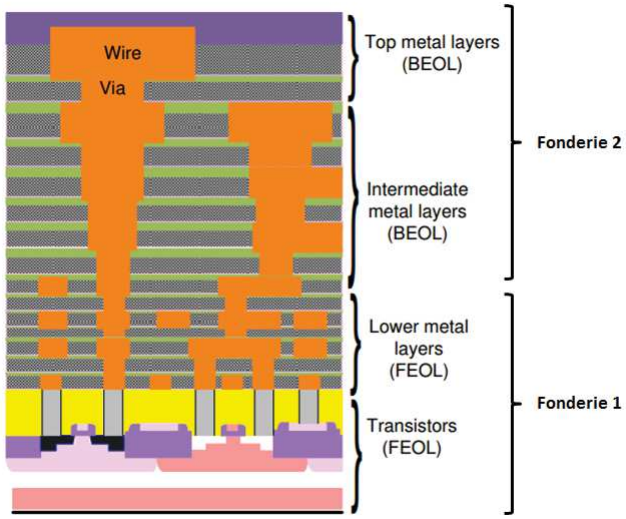


FIGURE 2.10 – Fabrication séparée [IEGT13].

Dans [IEGT13], les auteurs proposent de séparer la fabrication du CI dans plusieurs fonderies. Le principe consiste à diviser le GDSII en deux parties (Front-End-Of-the-Line (FEOL) et Back-End-Of-the-Line (BEOL)). Le FEOL contient les niveaux transistors et les premières couches de métal (cf figure 2.10). Le BEOL contient les couches de métal supérieures, c'est à dire les interconnexions. Ces deux parties sont fabriquées séparément dans des fonderies différentes pour cacher les fonctionnalités du circuit et pour éviter l'insertion d'un CTM. Ainsi,il est impossible au fabriquant du FEOL, d'insérer un CTM, car n'ayant pas les informations sur les interconnexions. De même, il est impossible au fabriquant du BEOL d'insérer un CTM, car n'ayant pas les informations sur les niveaux de métallisation.
Cependant, cette méthode est très coûteuse et difficile à réaliser.

2.4.2 Les méthodes facilitant la détection

Ces méthodes visent à faciliter la détection de CTM soit par des techniques SCA soit par test logique. Dans le premier cas, ces méthodes consistent à intégrer sur le circuit des composants de mesure de façon à faciliter la détection de variations de consommation ou de délais induits par le CTM. Pour la détection par test logique, ces méthodes consistent à augmenter la contrôlabilité des signaux rares en insérant des points de test dans le circuit.

2.4.2.1 Méthodes facilitant la détection par analyse de délai

Dans [LL08], les auteurs proposent une technique pour mesurer le délai sur un certain nombre de chemins combinatoires pour faciliter la détection de CTM par analyse de délai. La technique consiste à caractériser le délai en plaçant des registres à la fin de certains chemins combinatoires. La sélection de ces chemins est déterminée par le concepteur. La figure 2.11 illustre l'architecture de cette technique.

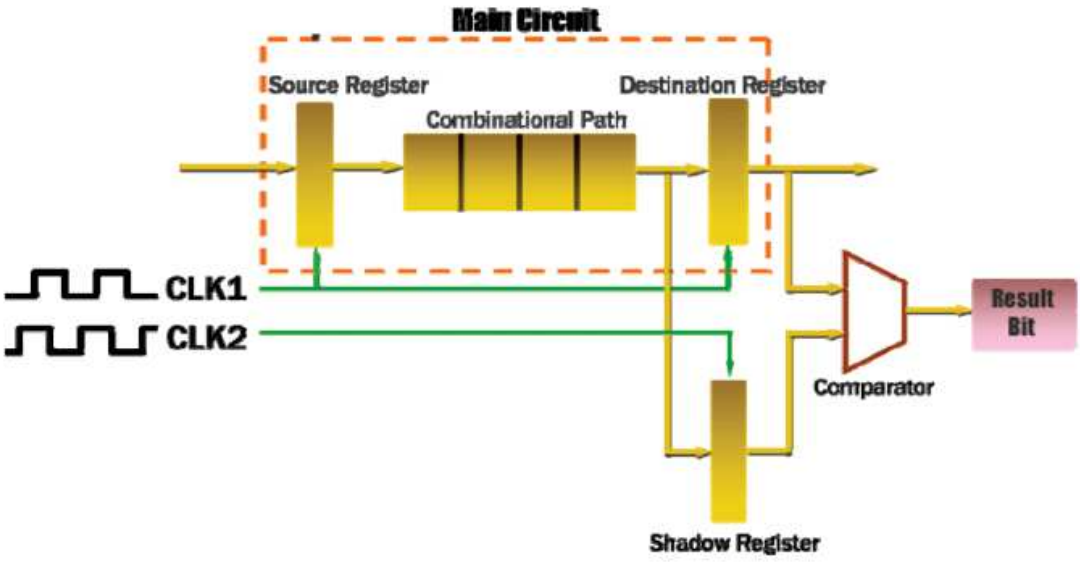


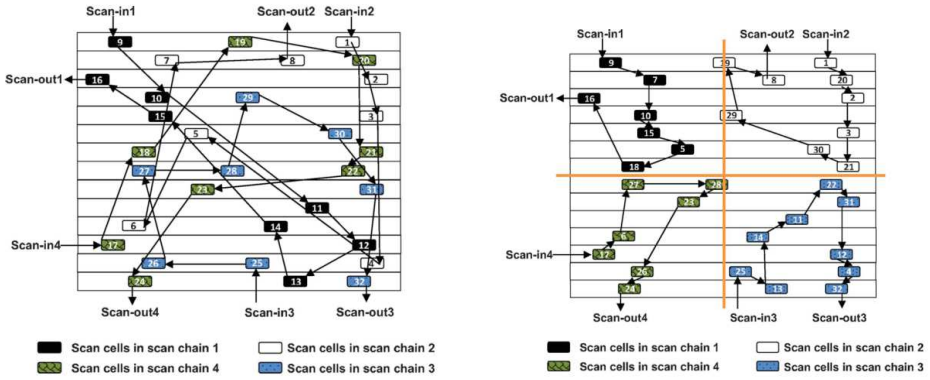
FIGURE 2.11 – Caractérisation de délai avec un registre [LL08].

En phase de test, le mécanisme de mesure de chemins de délai embarqué dans le circuit est utilisé pour obtenir une distribution statistique des chemins de délais. Ces mesures sont comparées à celles réalisées en phase de conception. Toute différence statistique substantielle signifie une altération. Ceci facilite même la détection de petits CTMs. Cependant les variations de tension et de température peuvent dégrader les mesures de délai et doivent donc être compensées. Les variations de tension ne sont pas un problème majeur, puisqu'elles sont transitoires mais la température est un problème majeur. Pour résoudre ce problème, les auteurs ajoutent un moniteur de température dans le circuit, dont la fréquence d'oscillation dépend de la température. L'efficacité de cette méthode dépend de la sélection des chemins à mesurer, et la logique ajoutée génère un surcoût en surface.

2.4.2.2 Méthodes facilitant la détection par analyse de consommation

Dans [ST12], les auteurs proposent une réorganisation du placement des Scan Flip-Flops et un partitionnement du circuit, dans le but de détecter plus facilement la contribution du

CTM dans l'augmentation de la consommation du circuit. Le principe repose sur le partitionnement du circuit en plusieurs régions et la réorganisation du placement des Scan Flip-Flops dans ces régions. La figure 2.12a montre le placement des scan FFs avant réorganisation pour le circuit s838 et la figure 2.12b montre leur placement après réorganisation.



(a) Placement des scan FFs avant réorganisation.

(b) Placement des scan FFs après réorganisation.

FIGURE 2.12 – Réorganisation des scan FFs dans le layout [ST12].

Les mesures de consommation obtenues pour chaque région sont comparées à celles obtenues pour un modèle de référence. L'impact du CTM dans la consommation d'une région est plus significatif que l'impact dans la consommation globale du circuit. Ceci permet d'augmenter l'efficacité des méthodes de détection de CTMs par analyse de consommation. Cependant, cette méthode nécessite aussi un modèle de référence.

2.4.2.3 Méthodes facilitant la détection par test logique

Dans [STP12], les auteurs proposent une méthode qui permet de faciliter la détection par le test logique, en insérant des Flip-Flop (FF) pour augmenter la probabilité de déclenchement des CTMs. Les CTMs qui sont furtifs utilisent une combinaison de signaux qui ont chacun une probabilité de déclenchement très faible (faible contrôlabilité), ce qui fait que la probabilité de cette combinaison est d'autant plus faible. En contrôlant ces signaux avec des FFs, on améliore leur contrôlabilité. Nous reviendrons plus en détails sur ce type de techniques au chapitre 7.

2.5 Conclusion

Dans ce chapitre nous avons introduit les CTM à travers leurs caractéristiques et les différentes moyens permettant de lutter contre leur insertion au niveau IP, netlist et layout. Nous avons vu que chaque méthode présente des limites et est adaptée à une classe de CTM bien définie. Ainsi, les méthodes orientées test logique sont plus adaptées aux CTMs de petite taille alors que les méthodes SCA sont plus efficaces pour les CTMs de grande taille. Dans le chapitre suivant nous allons implémenter des CTMs dans quelques circuits et analyser leur impact. Ensuite nous présenterons dans les chapitres qui suivent les méthodes de lutte contre les CTMs que nous avons développées.

2.6 Références

[AB09] Miron Abramovici and Paul Bradley. Integrated circuit security : new threats and solutions. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research : Cyber Security and Information Intelligence Challenges and Strategies*, page 55. ACM, 2009.

[Ade08] Sally Adee. The hunt for the kill switch. *iEEE SpEctrum*, 45(5) :34–39, 2008.

[AK08] Yousra Alkabani and Farinaz Koushanfar. Designer's hardware trojan horse. In *Hardware-Oriented Security and Trust*, pages 82–83. HOST, 2008.

[AL09] Miron Abramovici and PL Levin. Protecting integrated circuits from silicon trojan horses. *January/February*, 2009.

[BAA+13] Swarup Bhunia, Miron Abramovici, Dakshi Agrawal, Paul Bradley, Michael S Hsiao, Jim Plusquellic, and Mohammad Tehranipoor. Protection against hardware trojan attacks : Towards a comprehensive solution. *IEEE Design & Test*, 30(3) :6–17, 2013.

[BH08] Mainak Banga and Michael S Hsiao. A region based approach for the identification of hardware trojans. In *Hardware-Oriented Security and Trust*, pages 40–47. HOST, 2008.

[BH10] Mainak Banga and Michael S Hsiao. Trusted rtl : Trojan detection methodology in pre-silicon designs. In *Hardware-Oriented Security and Trust*, pages 56–59. HOST, 2010.

[BHBN14a] Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan. Hardware trojan attacks : threat analysis and countermeasures. *Proceedings of the IEEE*, 102(8) :1229–1247, 2014.

[BHBN14b] Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan. Hardware trojan attacks : threat analysis and countermeasures. *Proceedings of the IEEE*, 102(8) :1229–1247, 2014.

[CB11] Rajat Subhra Chakraborty and Swarup Bhunia. Security against hardware trojan attacks using key-based design obfuscation. *Journal of Electronic Testing*, 27(6) :767–785, 2011.

[Chi16] Inc. Chipworks. Semiconductor manufacturing - reverse engineering of semiconductor components, parts and process., 2016. http://www.chipworks.com.

[CLMFT15] Franck Courbon, Philippe Loubet-Moundi, Jacques JA Fournier, and Assia Tria. Semba : A sem based acquisition technique for fast invasive hardware trojan detection. In *European Conference on Circuit Theory and Design*, pages 1–4. ECCTD, 2015.

[CNB09] Rajat Subhra Chakraborty, Seetharam Narasimhan, and Swarup Bhunia. Hardware trojan : Threats and emerging solutions. In *High Level Design Validation and Test Workshop*, pages 166–171. HLDVT, 2009.

[EZRR15] Ingrid Exurville, Loie Zussa, Jean-Baptiste Rigaud, and Bruno Robisson. Resilient hardware trojans detection based on path delay measurements. In *Hardware Oriented Security and Trust*, pages 151–156. HOST, 2015.

[IEGT13] Frank Imeson, Ariq Emtenan, Siddharth Garg, and Mahesh Tripunitara. Securing computer hardware using 3d integrated circuit (ic) technology and split manufacturing for obfuscation. In *Presented as part of the 22nd USENIX Security Symposium*, pages 495–510. USENIX Security, 2013.

[JM08] Yier Jin and Yiorgos Makris. Hardware trojan detection using path delay fingerprint. In *Hardware-Oriented Security and Trust*, pages 51–57. HOST, 2008.

[KTC+08] Samuel T King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. Designing and implementing malicious hardware. *LEET*, 8 :1–8, 2008.

[KTK02] Jeffrey A Kash, James C Tsang, and Daniel R Knebel. Method and apparatus for reverse engineering integrated circuits by monitoring optical emission, December 17 2002. US Patent 6,496,022.

[LJM12] Eric Love, Yier Jin, and Yiorgos Makris. Proof-carrying hardware intellectual property : A pathway to trusted module acquisition. *IEEE Transactions on Information Forensics and Security*, 7(1) :25–40, 2012.

[LL08] Jie Li and John Lach. At-speed delay characterization for ic authentication and trojan horse detection. In *Hardware-Oriented Security and Trust*, pages 8–14. HOST, 2008.

[MWP+09] D McIntyre, F Wolff, C Papachristou, Swarup Bhunia, and D Weyer. Dynamic evaluation of hardware trust. In *Hardware-Oriented Security and Trust*, pages 108–111. HOST, 2009.

[NNB+14] Xuan Thuy Ngo, Zakaria Najm, Shivam Bhasin, Sylvain Guilley, and Jean-Luc Danger. Method taking into account process dispersions to detect hardware trojan horse by side-channel. In *Security Proofs for Embedded Systems*. PROOFS, 2014.

[RPT08] Reza Rad, Jim Plusquellic, and Mohammad Tehranipoor. Sensitivity analysis to hardware trojans using power supply transient signals. In *Hardware-Oriented Security and Trust*, pages 3–7. HOST, 2008.

[ST12] Hassan Salmani and Mohammad Tehranipoor. Layout-aware switching activity localization to enhance hardware trojan detection. *IEEE Transactions on Information Forensics and Security*, 7(1) :76–87, 2012.

[STP12] Hassan Salmani, Mohammad Tehranipoor, and Jim Plusquellic. A novel technique for improving hardware trojan detection and reducing trojan activation time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(1) :112–125, 2012.

[TK10] Mohammad Tehranipoor and Farinaz Koushanfar. A survey of hardware trojan taxonomy and detection. *IEEE Design and Test of Computers*, 27(1) :10–25, 2010.

[TSZ14] Mohammad Tehranipoor, Hassan Salmani, and Xuehui Zhang. *Integrated circuit authentication*. Springer, 2014.

[WPBC08] Francis Wolff, Chris Papachristou, Swarup Bhunia, and Rajat S Chakraborty. Towards trojan-free trusted ics : Problem analysis and detection scheme. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1362–1365. ACM, 2008.

[WTP08] Xiaoxiao Wang, Mohammad Tehranipoor, and Jim Plusquellic. Detecting malicious inclusions in secure hardware : Challenges and solutions. In *Hardware-Oriented Security and Trust*, pages 15–19. HOST, 2008.

[XT13] Kan Xiao and Mohammed Tehranipoor. Bisa : Built-in self-authentication for preventing hardware trojan insertion. In *Hardware-Oriented Security and Trust*, pages 45–50. HOST, 2013.

Chapitre 3

Implémentations de CTMs dans l'AES et le LEON2

« *Insanity : doing the same thing over and over again and expecting different results* »

Albert Einstein

Sommaire

3.1 Introduction

Dans ce chapitre, nous allons aborder l'insertion de CTM dans un design à des niveaux d'abstraction différents. En effet, un CTM peut aussi bien être inséré en phase de design qu'en fabrication, donc nous allons étudier la faisabilité en développant des CTMs de caractéristiques différents, et les insérer au niveau RTL, après synthèse et au niveau layout. Comme vu dans la partie taxonomie, les CTMs peuvent être caractérisés par leur mécanisme d'activation et leur mécanisme d'action. Le mécanisme d'activation peut être séquentiel ou combinatoire. Quant aux types d'action, ils peuvent être de plusieurs sortes, comme le déni de service ou la fuite d'information. Ce sont ces deux types que nous allons aborder dans ce chapitre.
Puisqu'il est question ici de sécurité, il semble pertinent de faire ces expériences sur des circuits qui seraient sensibles à une fuite d'information ou un déni de service. C'est la raison pour laquelle nous avons choisi l'AES et le LEON2 comme circuits d'expérimentations.

3.2 Implémentation dans l'AES

L'AES est un algorithme de chiffrement symétrique. Nous utilisons une implémentation 128 bits de l'AES. L'algorithme a pour entrée un bloc de 128 bits divisés en 16 octets de données. Il est basé sur la permutation et la substitution des données à chiffrer. Le chiffrement se fait en 4 opérations :

- AddRoundKey : chaque octet est combiné avec un block du round key en faisant un XOR.
- SubBytes : c'est une substitution non-linéaire où chaque octet est remplacé par un autre.
- ShiftRows : c'est une translation, où les trois dernières colonnes sont décalées de façon cyclique.
- MixColumns : c'est une opération de mélange, qui agit sur les colonnes, en combinant les quatre octets de chaque colonne.

La figure 3.1, tirée de [NGO16], illustre l'algorithme de l'AES.

3.2.1 Fuite d'informations

3.2.1.1 CTM1 : fuite d'information sur un port de sortie

Le CTM est activé quand le nombre de transitions sur un signal atteint une certaine valeur dans un court intervalle de temps. Une fois activé, le CTM copie la valeur d'un signal interne sur une sortie (voir figure 3.2). Pour choisir ce signal, nous avons calculé les probabilités pour chaque signal du circuit de valoir '0' et '1'. Nous avons ensuite sélectionné les signaux ayant les probabilités les plus équilibrées. Enfin nous avons effectué une simulation pour choisir parmi les signaux ayant les probabilités les plus équilibrées, celui qui commutait le plus fréquemment. Pour la condition de déclenchement du CTM, nous avons choisi 6 commutations sur 8 coups d'horloge consécutifs.
Le CTM a été codé au niveau RTL, puis synthétisé, et la description en portes obtenue a été insérée à la main dans l'AES.
Le trigger est un registre à décalage alimenté par le XOR entre l'état actuel du signal d'entrée et la valeur du même signal au coup d'horloge précédent. L'opération de XOR permet de

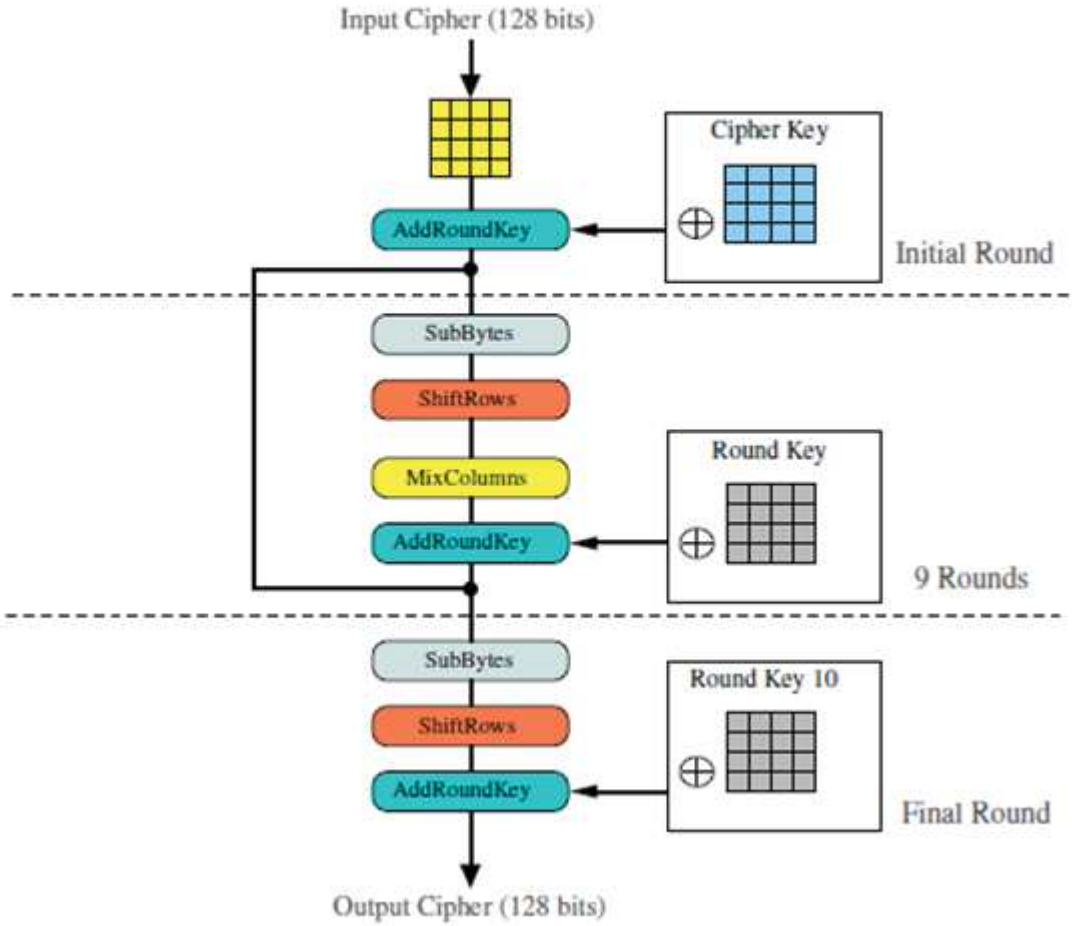


FIGURE 3.1 – Algorithme de l'AES [NGO16].

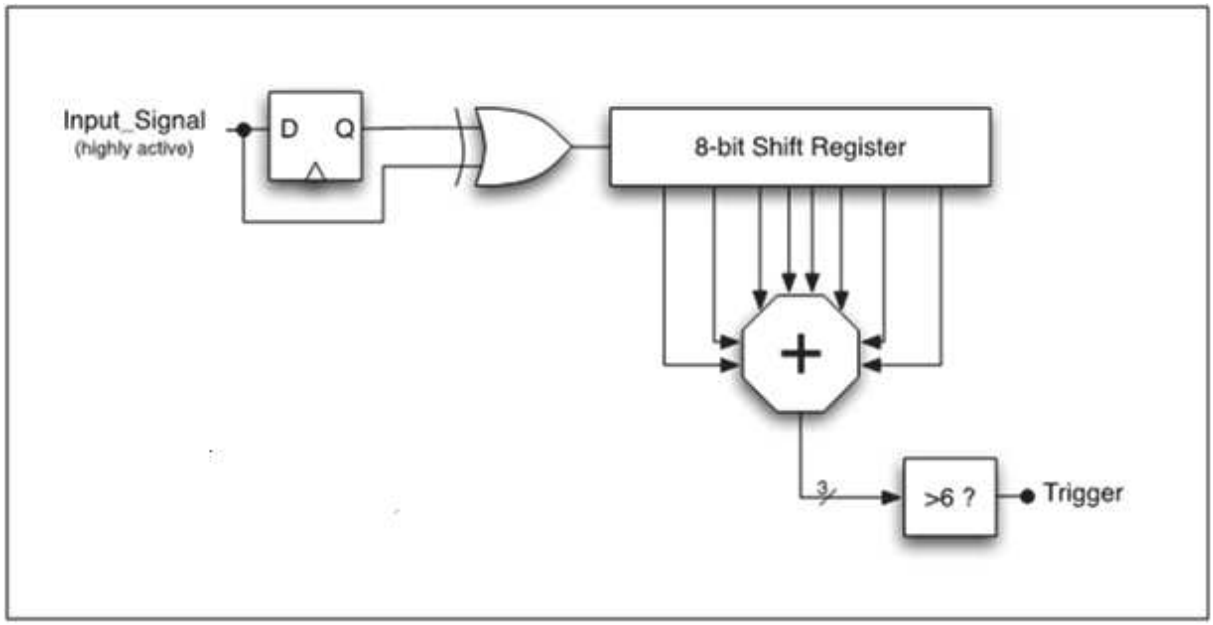


FIGURE 3.2 – CTM séquentiel : fuite d'informations sur port de sortie.

savoir s'il y a eu une transition (il est égal à '1' s'il y a eu une transition de '0' à '1' ou de '1' à '0'). Chaque bit du registre à décalage est additionné de façon à savoir si le signal en entrée a commuté au moins 6 fois durant les 8 derniers coups d'horloge.
La payload est un multiplexeur mettant un signal interne sur une sortie quand le trigger est déclenché.

3.2.1.2 Impact du CTM1 sur les performances du circuit

Le tableau 3.1 donne les performances du circuit en termes de délai, de consommation et de surface en présence du CTM1. Nous remarquons que le CTM impacte la consommation et la surface. Mais la hausse n'est pas significative.

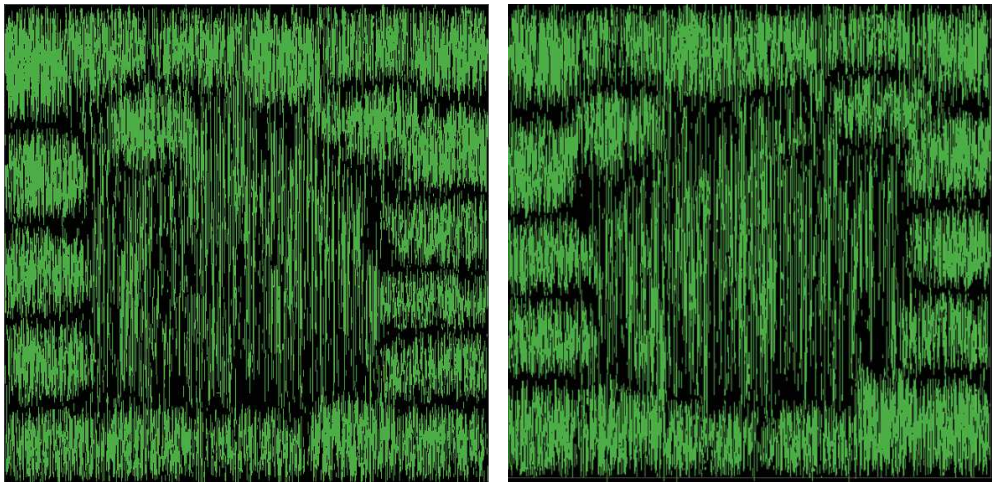
TABLEAU 3.1 – Impact du CTM1 sur les performances du circuit

Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	5.5923 ns	5.2840 ns	-5.5
Consommation statique	1.6836e-04 mW	1.5574e-04 mW	+7.5
Surface (μm^2)	39989.54992	43276.479412	+8.2

La figure 3.3 montre une comparaison entre la couche de métal 4 du circuit sain et celui avec CTM. La différence entre les deux est très nette. On remarque l'apparition de nouvelles lignes de connexion. La figure 3.4 illustre la couche de métal 5 du circuit sain et celui avec CTM. Nous remarquons une différence entre les deux layouts.

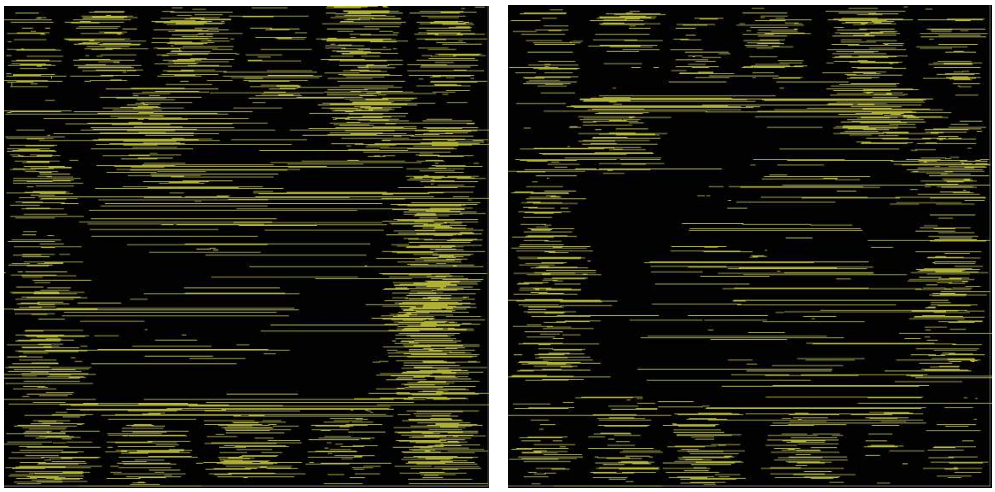
3.2.1.3 Discussions

Ce type de CTM assez volumineux, qui nécessite beaucoup de composants pour son implémentation, a un impact significatif sur certains paramètres du circuit comme la consommation, le délai ou la surface. Du fait de sa nature furtive (se déclenche très rarement), il est très difficile à détecter avec les méthodes de test logique. Cependant, il pourrait être détecté par



(a) Métal 4 de l'AES sans CTM (b) Métal 4 de l'AES avec CTM

FIGURE 3.3 – Comparaison des niveaux de métal 4 de l'AES avec et sans CTM1.



(a) Métal 5 de l'AES sans CTM (b) Métal 5 de l'AES avec CTM

FIGURE 3.4 – Comparaison des niveaux de métal 5 de l'AES avec et sans CTM1.

les méthodes de SCA par analyse de consommation [RPT08] ou par les méthodes de détection optique par analyse du layout [CLMFT15].

3.2.1.4 CTM2 : fuite d'information sur port de test

Le CTM ne peut être activé qu'en mode fonctionnel. Quand il est activé, il permet de « faire sortir » la valeur d'un signal interne sur le port de sortie de test (figure 3.5).
Pour ce CTM, nous avons effectué une synthèse de l'AES pour ajouter une chaine de scan et les signaux permettant de la contrôler (scanIn, scanOut et scanEnable). Nous avons choisi comme condition de trigger du CTM que le port d'entrée de données soit égal à h'4444'. Lorsque le CTM est activé, le signal de scanEnable est forcé à 1.

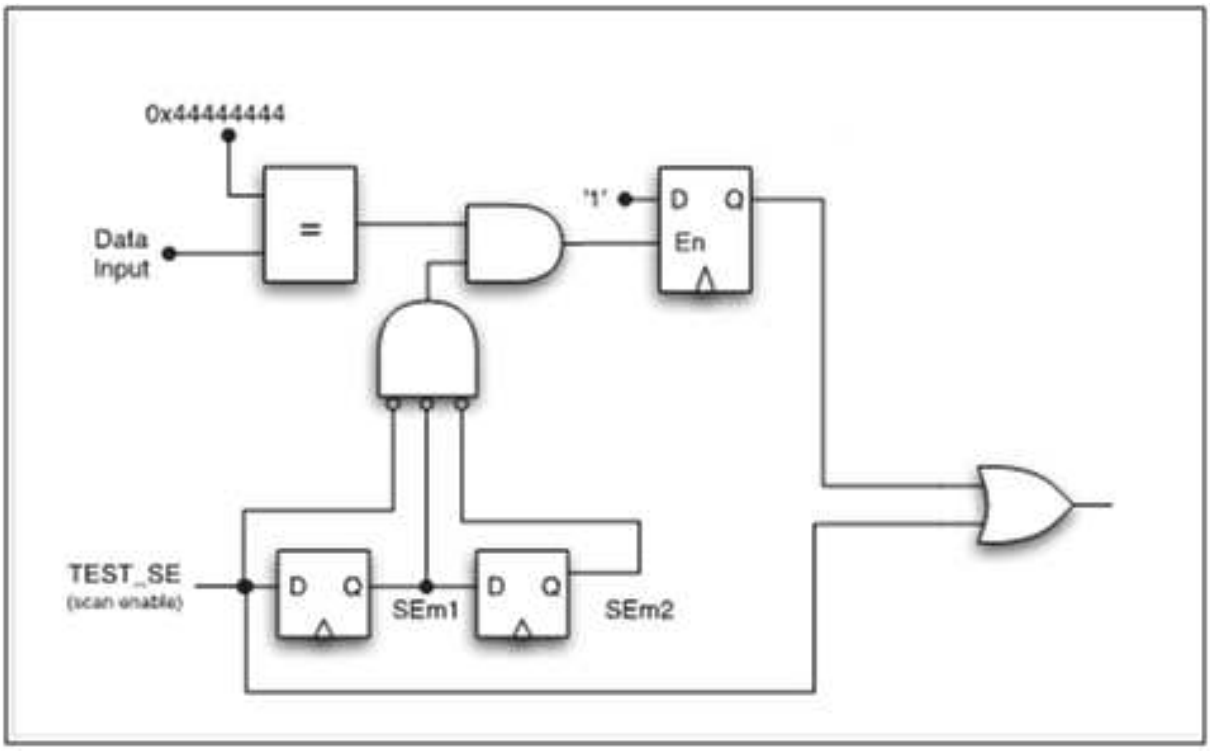


FIGURE 3.5 – CTM séquentiel : fuite d'information par le port de test.

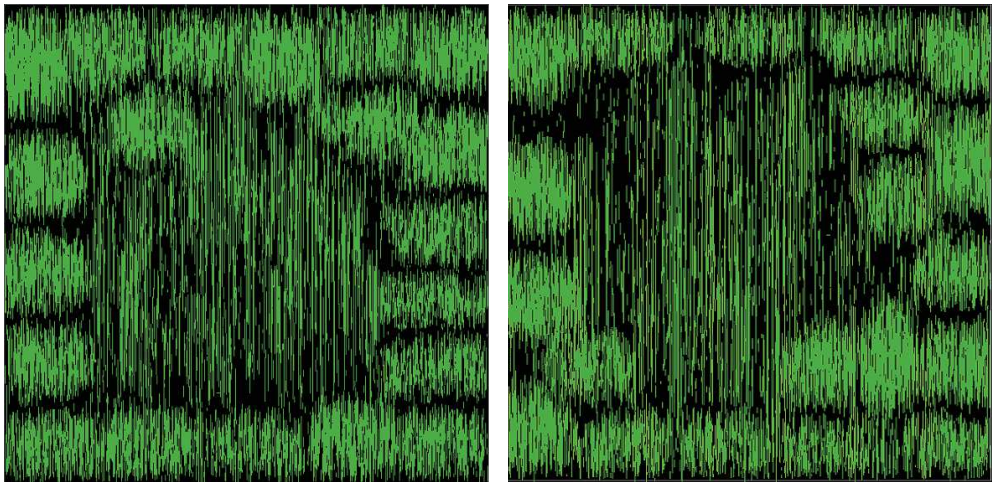
Le CTM a été codé au niveau RTL, puis synthétisé, et la description en portes obtenue a été insérée à la main dans la netlist de l'AES.
En mode test le scan enable est activé pendant N coups d'horloge pour entrer les vecteurs de test, puis il est remis à zéro. En vérifiant que le scan enable est inactif pendant 3 coups d'horloge, on est sûr d'être en mode fonctionnel.

3.2.1.5 Impact du CTM2 sur les performances du circuit

Nous voyons dans le tableau 3.2 qu'il n'y a pas de dégradation de performances en termes de délai. Mais la consommation et la surface sont impactées par la présence de CTM, avec une variation assez significative.
Sur la figure 3.6 et 3.7 nous remarquons que le layout (métal 4 et métal 5) est plus dense pour le circuit sans CTM. Ceci veut dire que le CTM impacte plus les niveaux de métal inférieurs.

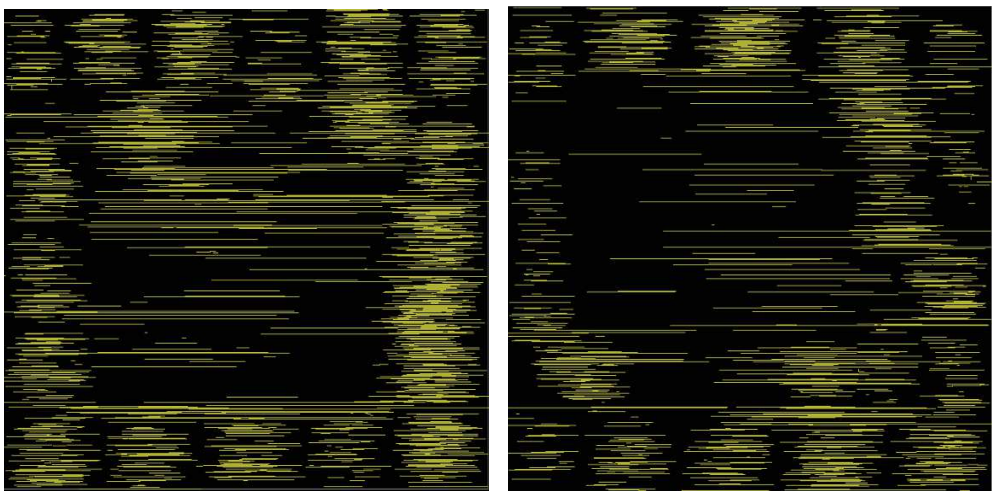
TABLEAU 3.2 – Impact du CTM2 sur les performances du circuit

Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	5.5923 ns	4.2016 ns	-24.8
Consommation statique	1.6836e-04 mW	1.5290e-04 mW	+9.1
Surface (μm^2)	39989.54992	43752.799423	+9.4



(a) Métal 4 de l'AES sans CTM (b) Métal 4 de l'AES avec CTM

FIGURE 3.6 – Comparaison des niveaux de métal 4 de l'AES avec et sans CTM2.



(a) Métal 5 de l'AES sans CTM (b) Métal 5 de l'AES avec CTM

FIGURE 3.7 – Comparaison des niveaux de métal 5 de l'AES avec et sans CTM2.

3.2.1.6 Discussions

Ce CTM est similaire au précédent en terme de taille. C'est pourquoi on remarque une dégradation de la consommation, de la surface et une modification du layout. Il ne peut pas être détecté avec les méthodes de test logique, car il est inactif quand le circuit est en mode test. Cependant, il pourrait être détecté par les méthodes de détection SCA par analyse de consommation ou par les méthodes de détection optique par analyse du layout.

3.2.2 Déni de service

3.2.2.1 CTM3 : modification d'un signal interne

Le trigger du CTM est un comparateur dont la probabilité de déclenchement est « très rare ». Lorsque le CTM est activé, le payload modifie la valeur d'un signal interne (figure 3.8). L'insertion de ce type de CTM ne peut pas être faite au niveau RTL étant donné qu'à ce niveau de description, on ne « raisonne pas en signaux et en portes ».

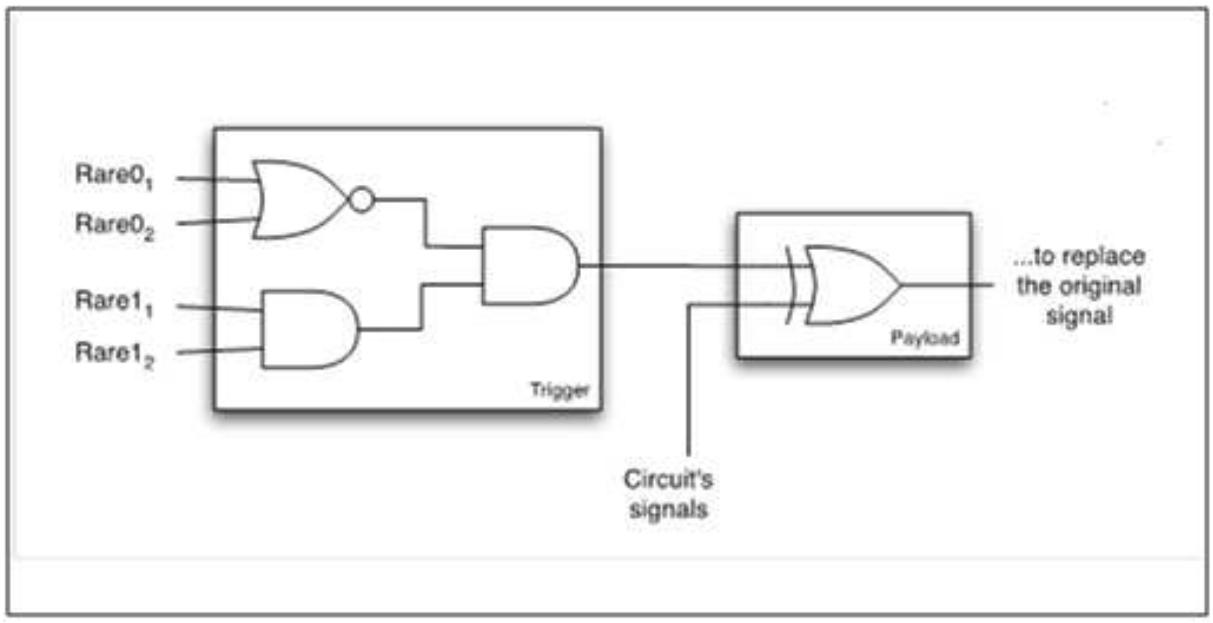


FIGURE 3.8 – CTM combinatoire : modification d'un signal interne.

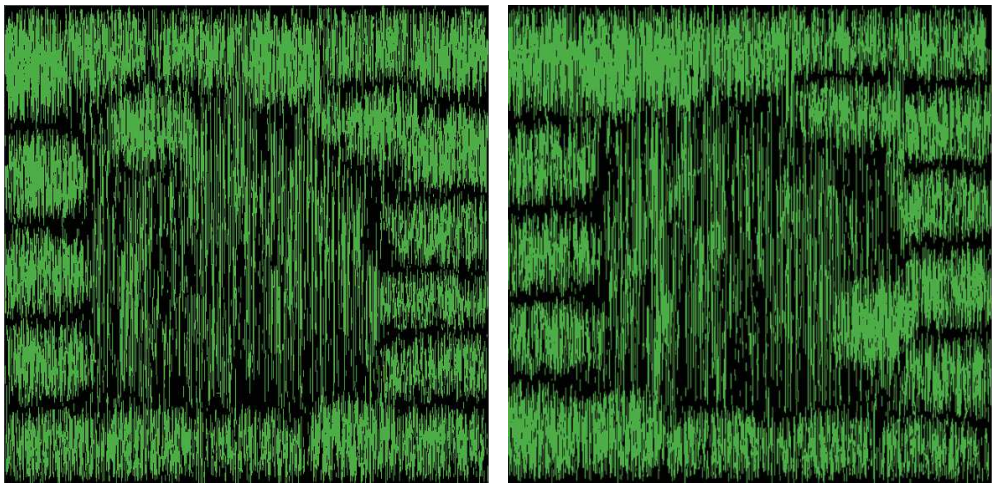
Nous avons utilisé notre outil permettant de calculer les probabilités de chaque signal de valoir '0' et '1', pour sélectionner 4 signaux qui sont faiblement contrôlables à '1' (P(1) = 6.10^{-11}). Le payload a été mis sur une entrée de bascule du registre de données. Lorsque le trigger est activé, le signal est inversé (par le biais d'une porte XOR). Le CTM a été incorporé à la main dans la description en portes à plat de l'AES.

3.2.2.2 Impact du CTM3 sur les performances du circuit

Nous voyons dans le tableau 3.3 que le CTM ne dégrade pas les performances en termes de délai et de consommation. Seule la variation de surface peut être exploitée.
La figure 3.9 illustre la couche de métal 4 du circuit sans CTM et celui avec CTM. On remarque des lignes de connexion supplémentaires sur le layout avec CTM dues à la présence du CTM. La figure 3.10 illustre la couche de métal 5 du circuit sans et avec CTM. Comme pour la couche 4, nous remarquons des lignes supplémentaires.

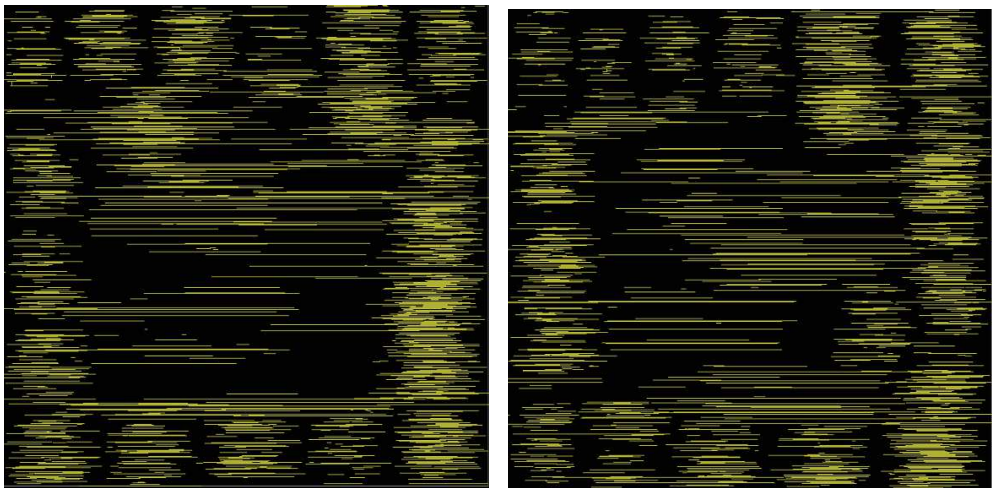
TABLEAU 3.3 – Impact du CTM3 sur les performances du circuit

Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	5.5923 ns	5.3354 ns	-4.6
Consommation statique	1.6836e-04 mW	1.6839e-04 mW	+0.017
Surface (μm^2)	39989.54992	42849.95934	+7.1



(a) Métal 4 de l'AES sans CTM (b) Métal 4 de l'AES avec CTM

FIGURE 3.9 – Comparaison des niveaux de métal 4 de l'AES avec et sans CTM3.



(a) Métal 5 de l'AES sans CTM (b) Métal 5 de l'AES avec CTM

FIGURE 3.10 – Comparaison des niveaux de métal 5 de l'AES avec et sans CTM3.

3.2.2.3 Discussions

Ce type de CTM ne peut pas être détecté avec les méthodes d'analyse de délai ou de consommation. Cependant, il pourrait être détecté par les méthodes de test logique [CWP+09] ou par les méthodes de détection optique par analyse du layout.

3.2.2.4 CTM 4 : modification d'un signal interne

Le trigger du CTM est un comparateur « séquentiel » dont la probabilité de déclenchement est « très rare ». Lorsque le CTM est activé, le payload modifie la valeur d'un signal interne (figure 3.11). L'insertion de ce CTM a été faite au niveau portes.

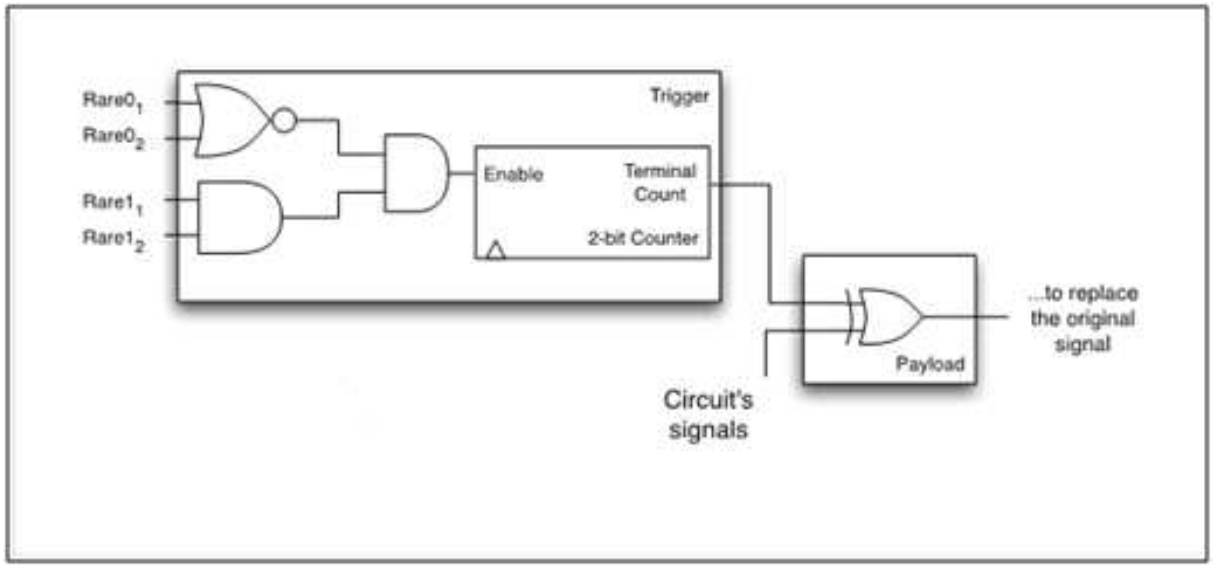


FIGURE 3.11 – CTM séquentiel : modification d'un signal interne.

Nous avons utilisé les mêmes signaux dits « rares » que pour le trigger du CTM précédent, et le même signal pour le payload. Le trigger composé par les 4 signaux rares a été connecté au port enable d'un compteur 2 bits. Il faut que la « condition rare » soit réalisée quatre fois pour que le payload se déclenche. Le compteur a été codé au niveau RTL puis synthétisé. La description en portes obtenue a ensuite été incorporée à la main dans la description en portes à plat de l'AES.

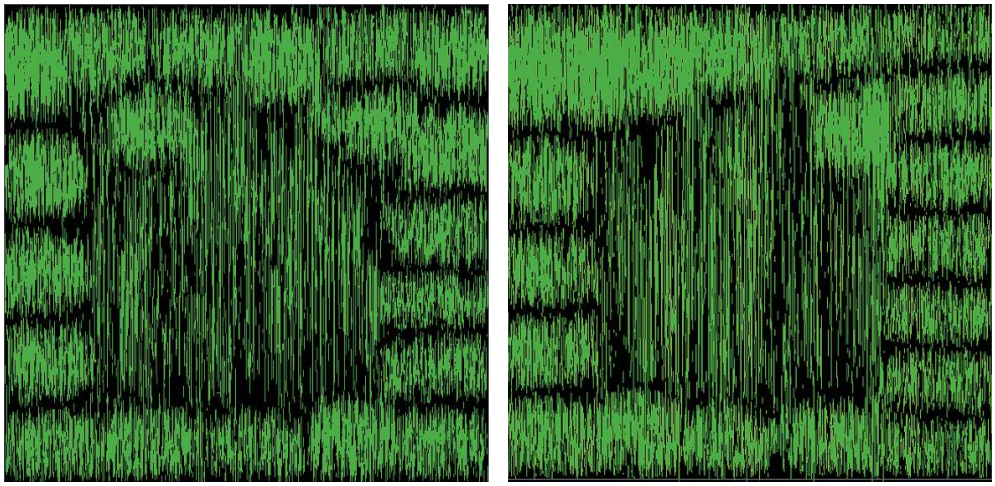
3.2.2.5 Impact du CTM4 sur les performances du circuit

Sur le tableau 3.4 Nous remarquons une légère variation de la consommation, ce qui peut être expliqué par la nature furtive du CTM. Par contre On note une variation de la surface, ce qui peut être expliqué par la présence du compteur qui nécessite des Flip-Flops.

TABLEAU 3.4 – Impact du CTM4 sur les performances du circuit

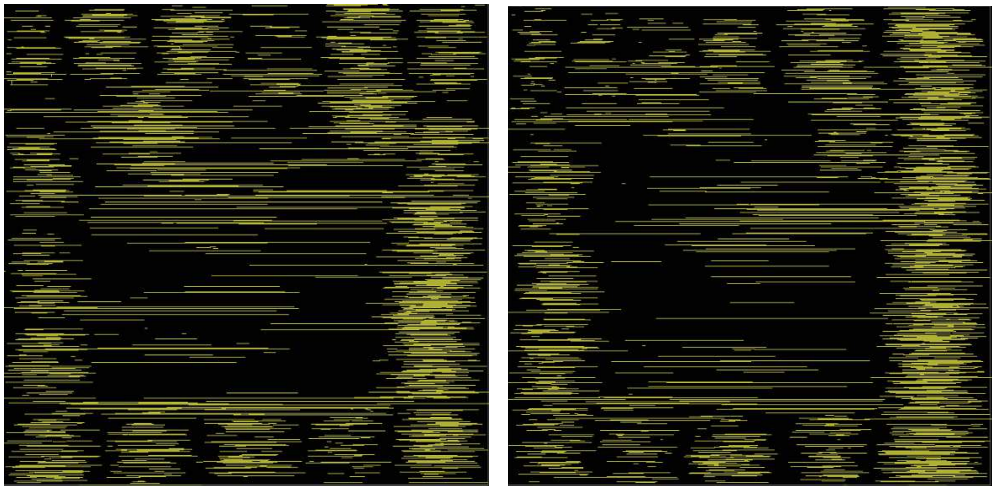
Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	5.5923 ns	5.5133 ns	-7.9
Consommation statique	1.6836e-04 mW	1.6839e-04 mW	+0.01
Surface (μm^2)	39989.54992	42887.39934	+7.2

Sur la figure 3.12 nous remarquons que les lignes de métal ne sont pas disposées de la même façon, ce qui traduit une modification du layout. Sur la figure 3.13 nous remarquons que le layout avec CTM est plus dense avec des lignes de métal supplémentaires.



(a) Métal 4 de l'AES sans CTM (b) Métal 4 de l'AES avec CTM

FIGURE 3.12 – Comparaison des niveaux de métal 4 de l'AES avec et sans CTM4.



(a) Métal 5 de l'AES sans CTM (b) Métal 5 de l'AES avec CTM

FIGURE 3.13 – Comparaison des niveaux de métal 5 de l'AES avec et sans CTM4.

3.2.2.6 Discussions

Du fait de sa nature séquentielle, ce CTM est très difficile à détecter avec les méthodes de test logique. Les méthodes de détection optique par analyse du layout semblent plus efficaces.

3.3 Implémentation dans le processeur LEON2

Le circuit cible est un processeur LEON2 d'origine l'ESA / GAISLER Reserch, modifié par Secure-IC. La figure 3.14 présente les différents blocs fonctionnels du circuit d'origine. Les

modifications apportées par Secure-IC consistent en des simplifications. La plus notable est la suppression de la MMU. C'est un processeur 32-bit. Il a 7-étages de pipeline et des caches de données et d'instructions séparées.

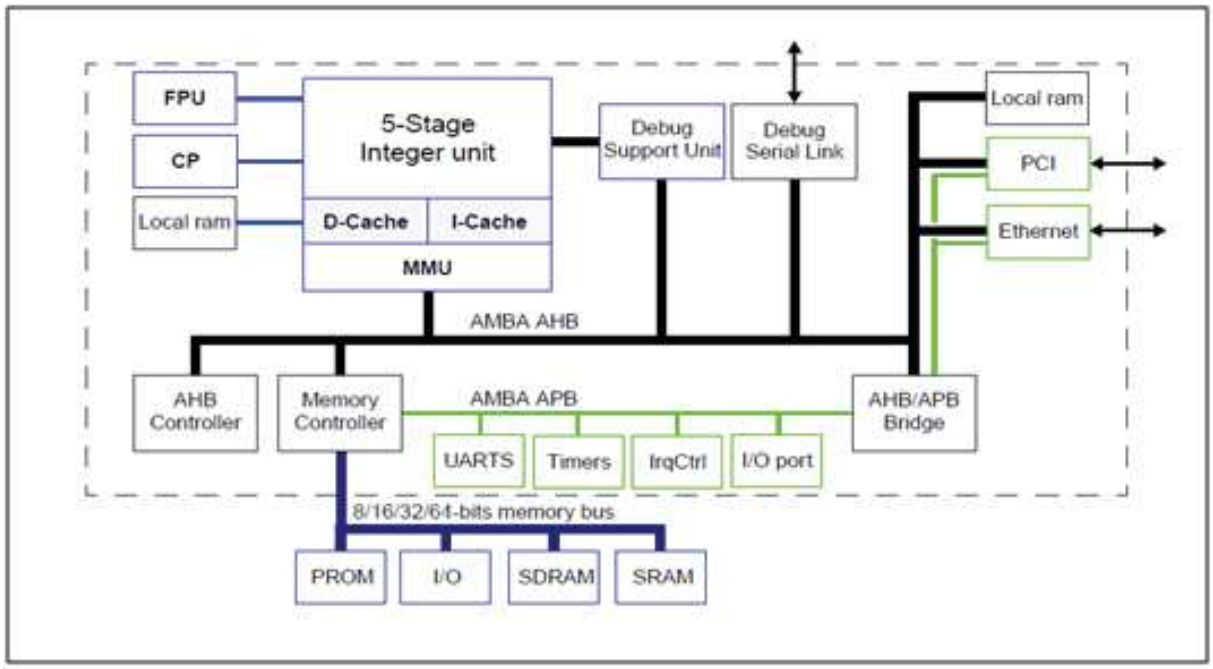


FIGURE 3.14 – Architecture du LEON2.

3.3.1 Déni de service

3.3.1.1 CTM5 : Désactivation du timer

Un compteur circulaire 32 bits est incrémenté à chaque changement d'état d'un signal interne (un bit du registre de données). L'activation du CTM ne se produit que lorsque le compteur se trouve dans une certaine plage [X Y]. Une fois déclenché, plus aucune interruption émanant du Timer ne parvient au contrôleur d'interruptions. Pour les simulations, nous avons choisi la plage [255 272], comme plage de déclenchement.

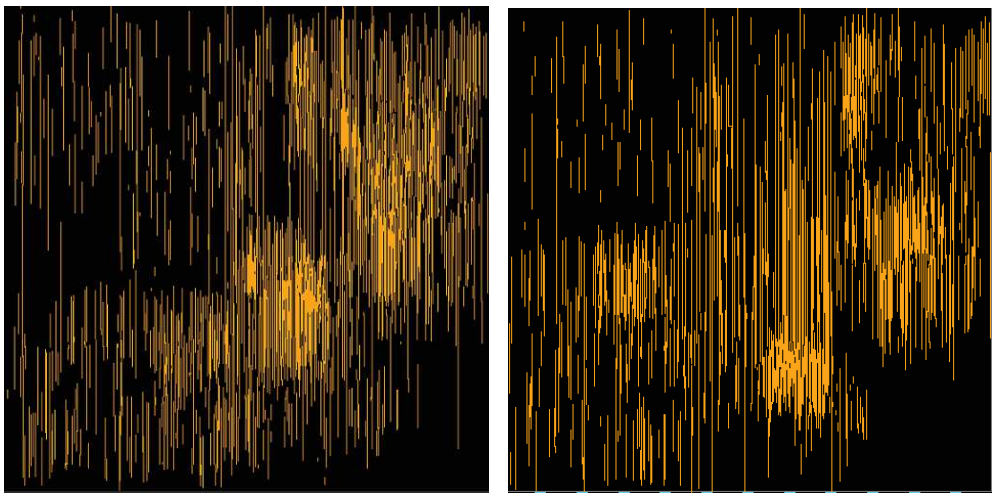
3.3.1.2 Impact du CTM5 sur les performances du circuit

Le tableau 3.5 montre un impact assez significatif en termes de délai et de consommation du CTM, ce qui peut être expliqué par le compteur 32 bits. Par contre l'impact en surface n'est pas significatif car le nombre de transistors ajouté par la présence de CTM reste marginal par rapport au nombre total de transistors du circuit qui est très volumineux.

TABLEAU 3.5 – Impact du CTM5 sur les performances du circuit

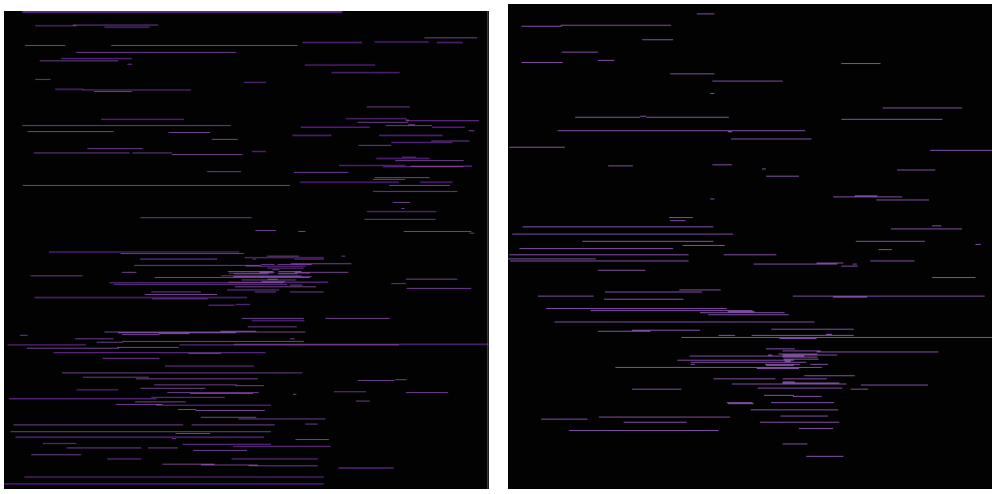
Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	4.5574 ns	4.9188 ns	+7.9
Consommation statique	1.7011e-03 mW	1.7754e-03 mW	+4.36
Surface (μm^2)	463223.196653	464067.756613	+0.18

Sur les figures 3.15 et 3.16 la différence est flagrante. On remarque que layout avec CTM est plus dense avec des lignes de métal supplémentaires introduites par l'insertion du CTM.



(a) Métal 6 du LEON2 sans CTM (b) Métal 6 du LEON2 avec CTM

FIGURE 3.15 – Comparaison des niveaux de métal 6 du LEON2 avec et sans CTM5.



(a) Métal 7 du LEON2 sans CTM (b) Métal 7 du LEON2 avec CTM

FIGURE 3.16 – Comparaison des niveaux de métal 7 du LEON2 avec et sans CTM5.

3.3.1.3 Discussions

Ce type de CTM est difficile à détecter avec des vecteurs de test, car il faut un test exhaustif sur un compteur 32 bits. Il est donc très difficile à détecter avec les méthodes de test logique. Cependant, il pourrait être détecté avec plus d'efficacité avec les méthodes de SCA par analyse de délai [JM08].

3.3.1.4 CTM6 : Désactivation des interruptions

Le déclenchement survient lorsqu'une combinaison particulière du vecteur des interruptions actives est détectée (IT 13 et 15 actives simultanément). Une fois le CTM déclenché,

même lorsque le processeur n'est pas en mode « superviseur » (bit Supervisor du registre %PSR à 0), les tentatives d'accès aux registres et instructions privilégiées ne provoquent pas d'exception (plus d'interruption privilégiée en mode « utilisateur »).

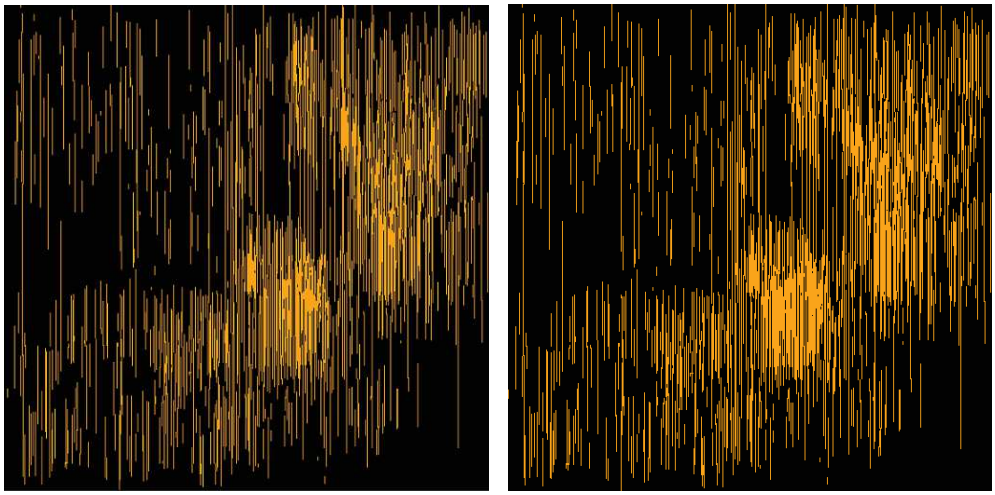
3.3.1.5 Impact du CTM6 sur les performances du circuit

Les résultats donnés dans le tableau 3.6 montrent un impact non significatif en termes de délai et de surface. La variation de consommation par contre, est exploitable.

TABLEAU 3.6 – Impact du CTM6 sur les performances du circuit

Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	4.5574 ns	4.5574 ns	0.0
Consommation statique	1.7011e-03 mW	1.7726e-03 mW	+4.2
Surface (μm^2)	463223.196653	465223.196653	+0.429

Les figures 3.17 3.18 ne montrent pas une différence flagrante au niveau layout. Ceci peut être expliqué par la taille du CTM, qui reste marginal par rapport à la taille du circuit.



(a) Métal 6 du LEON2 sans CTM (b) Métal 6 du LEON2 avec CTM

FIGURE 3.17 – Comparaison des niveaux de métal 6 du LEON2 avec et sans CTM6.

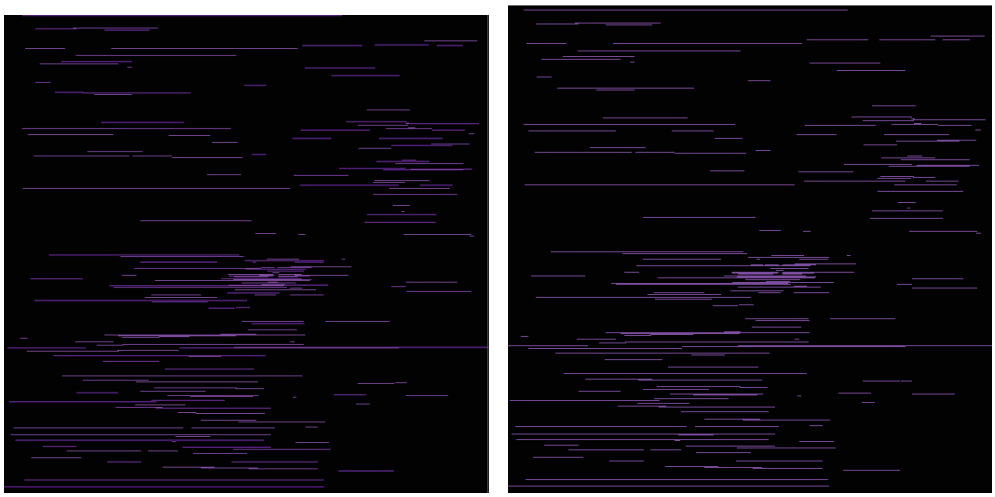
3.3.1.6 Discussions

En considérant l'impact du CTM sur les performances du circuit, la détection avec des méthodes optiques ou par analyse de délai ne serait pas efficace. Par contre les méthodes par analyse de consommation pourraient être efficaces. Il est aussi possible d'utiliser les méthodes orientées test logique, car le CTM est purement combinatoire.

3.3.2 Dégradation de performances

3.3.2.1 CTM7 : Exécutions retardées

Le déclenchement survient lorsqu'une opération unaire particulière couplée à une valeur précise dans un registre spécifique est soumise (Super Long Lines (SLL) %l0,5,%l1 avec l0 =



(a) Métal 7 du LEON2 sans CTM (b) Métal 7 du LEON2 avec CTM

FIGURE 3.18 – Comparaison des niveaux de métal 7 du LEON2 avec et sans CTM6.

0x0DDC0DED). Une fois déclenché, une certaine instruction nécessite plus de cycles d'horloge pour s'exécuter (opération ADD).

3.3.2.2 Impact du CTM7 sur les performances du circuit

Les résultats donnés dans le tableau 3.7 montrent une forte variation en délai du fait que les signaux du trigger se trouvent sur le chemin critique. La variation en consommation est aussi significative. Par contre le sur-coût en surface n'est pas très important.

TABLEAU 3.7 – Impact du CTM7 sur les performances du circuit

Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	4.5574 ns	5.3797 ns	+18
Consommation statique	1.7011e-03 mW	1.8150e-03 mW	+6.6
Surface (μm^2)	463223.196653	471747.119230	+1.9

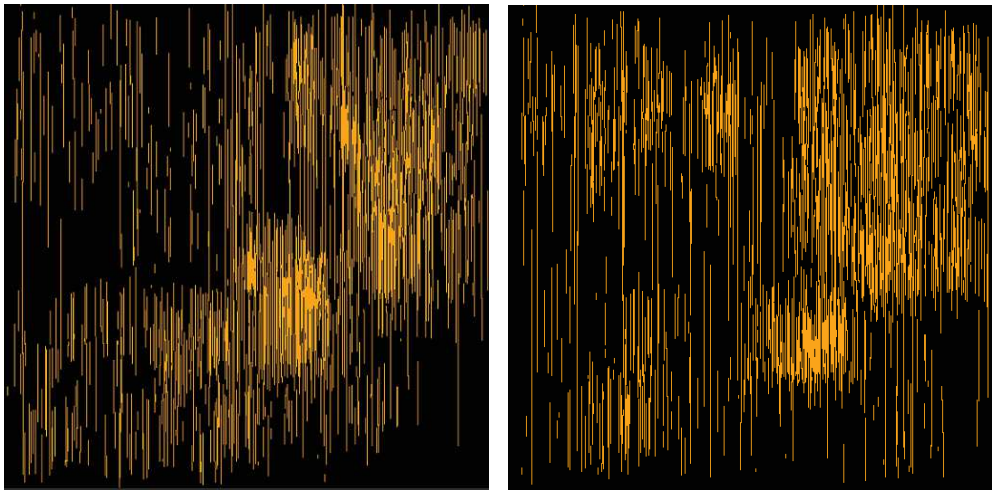
Sur les figures 3.19 et 3.20, nous remarquons une légère différence en termes de densité des couches de métal. On arrive quand même à distinguer des lignes de métal supplémentaires qui matérialisent la présence du CTM.

3.3.2.3 Discussions

Nous avons vu que le CTM impacte fortement sur le chemin critique du circuit, donc il serait assez facile de le détecter en utilisant les méthodes SCA par analyse de délai. Par contre la détection par test serait difficile car il faudrait être exhaustif sur 32 bits.

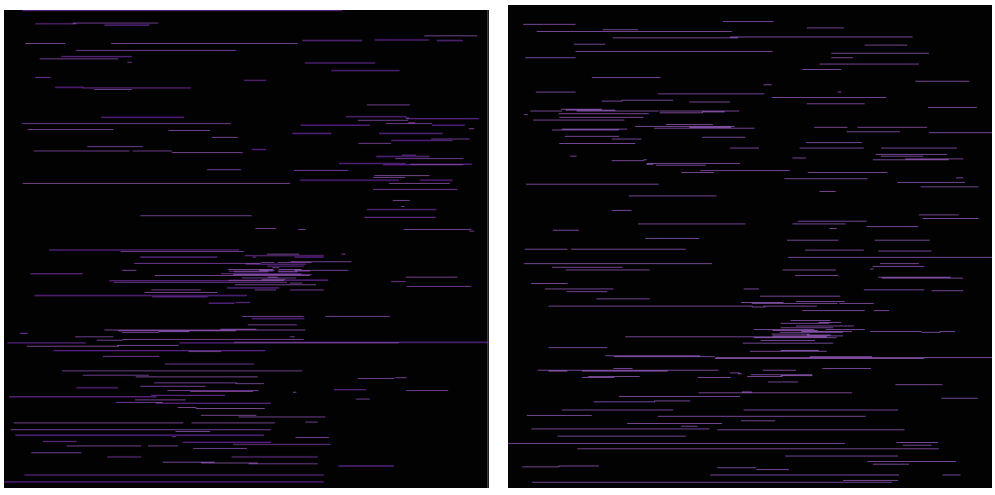
3.3.2.4 CTM8 : Surconsommation

Un compteur 32 bits est incrémenté à chaque changement d'état d'un signal interne. Le déclenchement de ce CTM se produit lorsque la valeur du compteur dépasse un certain seuil. Une fois déclenché, une surconsommation d'énergie est provoquée par une boucle auto-oscillante lors de l'exécution des opérations arithmétiques uniquement (ADD, SUB, ...).



(a) Métal 6 du LEON2 sans CTM (b) Métal 6 du LEON2 avec CTM

FIGURE 3.19 – Comparaison des niveaux de métal 6 du LEON2 avec et sans CTM7.



(a) Métal 7 du LEON2 sans CTM (b) Métal 7 du LEON2 avec CTM

FIGURE 3.20 – Comparaison des niveaux de métal 7 du LEON2 avec et sans CTM7.

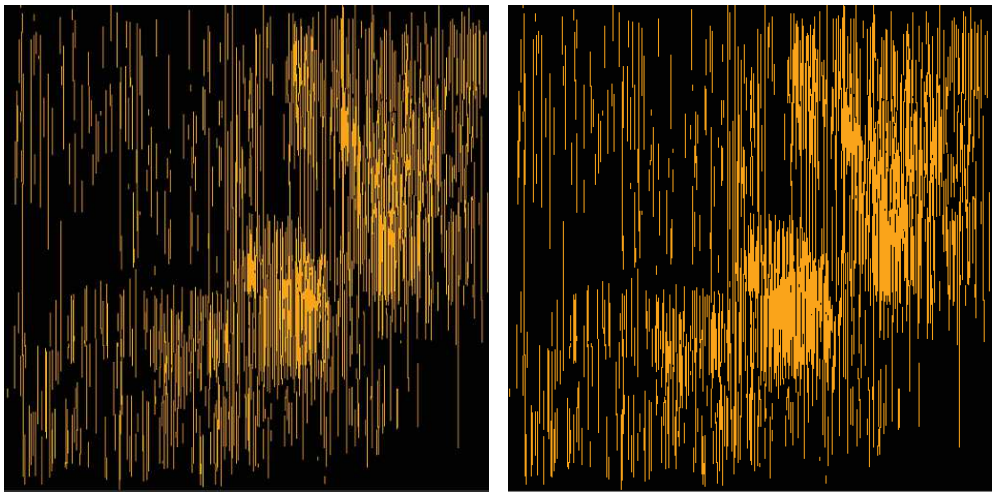
3.3.2.5 Impact du CTM8 sur les performances du circuit

Les résultats donnés dans le tableau 3.8 montrent une légère variation en termes de délai et de surface. L'impact sur la consommation est par contre non négligeable du fait du compteur 32 bits du trigger et de l'oscillateur du payload.

TABLEAU 3.8 – Impact du CTM8 sur les performances du circuit

Paramètres	Sans CTM	Avec CTM	Variations (%)
Chemin critique	4.5574 ns	4.6014 ns	+0.96
Consommation statique	1.7011e-03 mW	1.7726e-03 mW	+4.2
Surface (μm^2)	463223.196653	465223.196653	+0.429

Sur les figures 3.21 et 3.22 la différence n'est pas flagrante. En effet, on remarque une densité similaire entre les deux layouts au niveau des couches de métal 6 et 7. Ceci signifie que le CTM n'a pas modifié le layout.

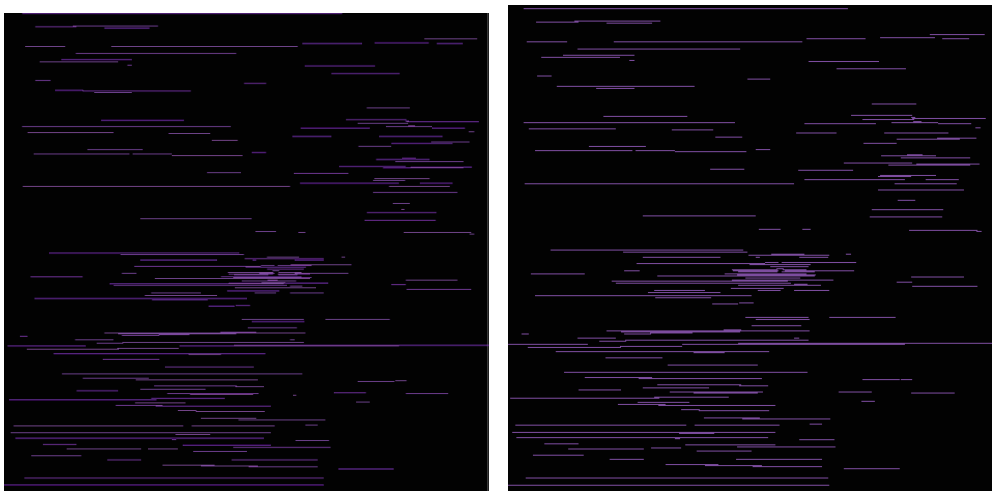


(a) Métal 6 du LEON2 sans CTM (b) Métal 6 du LEON2 avec CTM

FIGURE 3.21 – Comparaison des niveaux de métal 6 du LEON2 avec et sans CTM8.

3.3.2.6 Discussions

L'impact du CTM en termes de consommation peut être exploité en utilisant les méthodes de détection SCA par analyse de consommation. La détection par test serait inefficace du fait de la nature séquentielle du CTM et de sa taille. La détection optique n'est pas aussi adaptée, car le CTM ne modifie pas le layout.



(a) Métal 7 du LEON2 sans CTM (b) Métal 7 du LEON2 avec CTM

FIGURE 3.22 – Comparaison des niveaux de métal 7 du LEON2 avec et sans CTM8.

3.4 Conclusion

Dans ce chapitre, nous avons développé et implémenté quelques CTMs en fonction de leur taxonomie, comme expliqué dans le chapitre relatif à la taxonomie des CTMs. Les CTMs insérés au niveau netlist et RTL dans l'AES ont un impact sur la géométrie (layout) du circuit et on arrive à voir une différence entre un layout sans CTM et un layout avec CTM juste en comparant les couches de métal. Par contre, l'insertion au niveau layout a été beaucoup plus difficile, même en ayant toutes les informations (code RTL et netlist).

Nous avons constaté que le type de CTM (furtif) inséré dans l'AES n'a pas un impact important voire négligeable sur les performances du circuit en termes de délai et de consommation. C'est pourquoi ils sont plus adaptés à des méthodes dites de test logique, qui elles sont dédiées aux CTMs dits furtifs et ne nécessitent pas de modèle de référence en comparaison avec les méthodes optiques et les méthodes SCA. Par contre les CTMs insérés dans le LEON2 ont un impact considérable sur les performances du circuit (surface et consommation), donc les méthodes SCA sont plus adaptées pour leur détection.

Dans les chapitres suivants, nous proposons des méthodes orientées test logique qui permettent de détecter les CTMs dits furtifs, mais également des méthodes facilitant la détection et prévenant l'insertion de CTM durant la phase de fabrication.

3.5 Références

[CLMFT15] Franck Courbon, Philippe Loubet-Moundi, Jacques JA Fournier, and Assia Tria. Semba : A sem based acquisition technique for fast invasive hardware trojan detection. In *European Conference on Circuit Theory and Design*, pages 1–4. ECCTD, 2015.

[CWP+09] Rajat Subhra Chakraborty, Francis Wolff, Somnath Paul, Christos Papachristou, and Swarup Bhunia. Mero : A statistical approach for hardware trojan detection. In *Cryptographic Hardware and Embedded Systems-CHES*, pages 396–410. Springer, 2009.

[JM08] Yier Jin and Yiorgos Makris. Hardware trojan detection using path delay fingerprint. In *Hardware-Oriented Security and Trust*, pages 51–57. HOST, 2008.

[NGO16] Xuan Thuy NGO. *Prévention et Détection des Chevaux de Troie Matériels Dans les Circuit Intégrés*. PhD thesis, Telecom ParisTech, 2016.

[RPT08] Reza Rad, Jim Plusquellic, and Mohammad Tehranipoor. Sensitivity analysis to hardware trojans using power supply transient signals. In *Hardware-Oriented Security and Trust*, pages 3–7. HOST, 2008.

Chapitre 4

Détection de CTMs en phase de test

« *Every great and deep difficulty bears in itself its own solution. It forces us to change our thinking in order to find it.* »

Niels Bohr

Sommaire

4.1 Introduction

Nous proposons une méthode dédiée à la détection des CTMs numériques. Cette méthode est basée sur le test logique i.e. consistant à essayer d'activer les CTMs de façon à observer leur effet lors du test. La détection proprement dite est donc effectuée par comparaison des valeurs du circuit obtenues avec celles attendues, celles-ci étant obtenues par simulation logique du circuit. Les chevaux de Troie visés sont supposés :

- avoir été insérés après la phase de synthèse logique, i.e. soit dans le layout du circuit soit directement dans les masques de fabrication,
- avoir un effet logique sur le comportement du circuit.

Notre approche consiste à :

- identifier les signaux sur lesquels les CTMs ont pu être insérés,
- générer des vecteurs de test (i.e. des stimuli) aptes à les déclencher.

Les CTMs étant "furtifs" par nature i.e. :

- ils sont la plupart du temps inactifs, et sont déclenchés par des conditions internes au circuit très rares,
- leur insertion ne modifie pas les performances du circuit,
- ils ne se "voient" pas (modifient le moins possible le layout). Á contrario, ils pourraient être détectés par une méthode optique.

La procédure d'identification des sites potentiels repose donc sur les hypothèses suivantes :

- les signaux de déclenchement (trigger) ont une faible contrôlabilité [WPBC08a],
- l'insertion des portes logiques constituant le CTM n'a pas d'influence sur les performances du circuit (pas d'impact sur le chemin critique pour contourner la détection basée sur l'analyse de délai [JM08a]),
- les signaux de déclenchement sont proches dans le layout du circuit (à contrario, l'interconnexion de signaux éloignés, même si elle est possible, avait un fort impact sur le layout et serait donc certainement détectable par des méthodes d'inspection visuelle [CLMFT15a]).

En se basant sur l'hypothèse que le CTM ne devrait être activé que par des conditions très rares de façon à échapper à sa détection pendant la phase de test du circuit, la condition d'activation est supposée dépendre de signaux à faible contrôlabilité (cf. Figure 4.1). Notre méthode de détection est dédiée à ce type de CTMs, référencés comme « rare value type of triggering based CTMs » dans [WPBC08b]. Parmi les différents moyens de détecter les CTMs, nous adressons ici la technique de test logique, c.à.d, une technique consistant en des procédures de test pour déclencher le CTM et l'obtention des réponses du circuit non prévues initialement (erronées). Le but est donc de déclencher des CTMs potentiels grâce à une séquence de test spécifique.
Le problème de la détection des CTMs peut être modélisé comme le problème classique de la détection des fautes de collages dans un circuit : le but ici est de contrôler les entrées du trigger à la valeur rare et d'observer la sortie du payload.
Les concepts comme l'excitation, la propagation et justification de valeurs espérées, typiquement utilisées dans la génération de séquence de vecteurs de test, peuvent être utilisés dans le cadre de la génération de séquence de déclenchement de CTMs. Par exemple, sur le circuit de la Figure 4.2, une fois que les trois signaux de contrôle du trigger ont été identifiés,

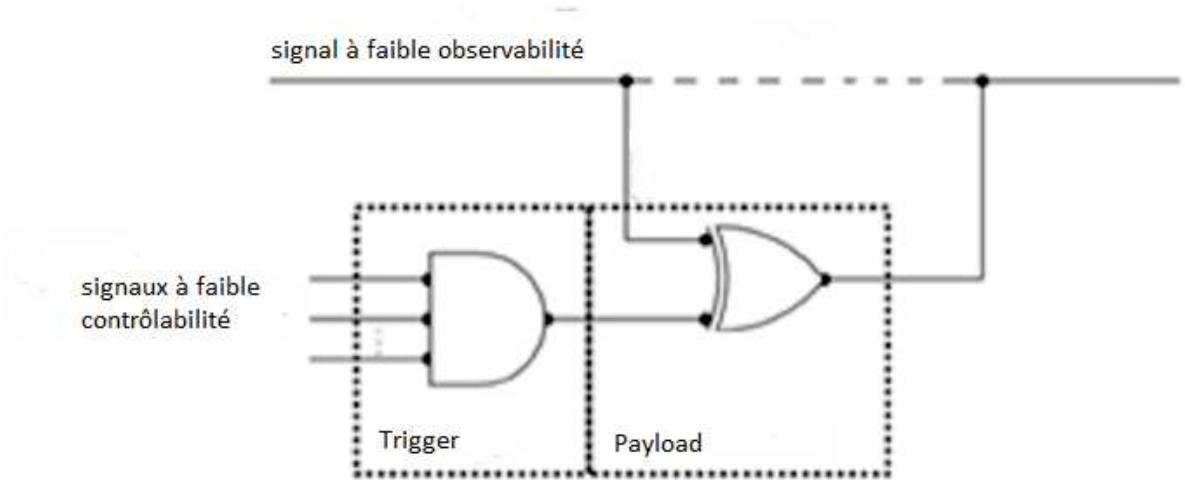


FIGURE 4.1 – Modèle de CTM basé sur une valeur rare [WPBC08b].

on doit trouver les vecteurs à appliquer aux entrées qui mettent la sortie du trigger à la valeur '1' pour déclencher le CTM. Le processus est similaire à celui utilisé pour la génération de vecteurs de test pour la faute de collage à zéro de la sortie du trigger : (1) excitation de la sortie du trigger à '1', (2) justification des trois entrées du trigger à ('1', '0', '1'), et (3) propagation de la valeur du trigger à la sortie du circuit. Il faut noter que dans le cas d'un CTM, la propagation de la valeur de sortie du trigger est supposée s'effectuer à travers la circuiterie ajoutée par l'attaquant dans le payload. La propagation de la valeur de sortie du trigger n'est donc pas un problème.

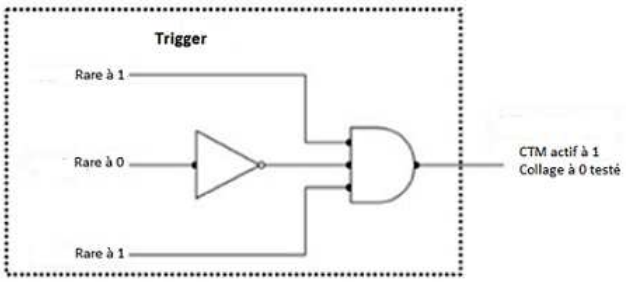


FIGURE 4.2 – Génération de pattern pour une combinaison de trigger particulière.

Toutefois, la génération de séquences de vecteurs pour le déclenchement des CTMs diffère de celles pour le test des pannes. Dans ce dernier cas, les données d'entrée de la génération sont d'une part une netlist complète du circuit (e.g. la description niveau portes logiques), et d'autre part une liste de fautes (telle sortie ou entrée de portes est collée à '0' ou à '1'). Á l'inverse, les CTMs étant par définition inconnus, aucune information ni sur le positionnement du CTM ni sa structure ne sont connus. Il faut donc une méthodologie spécifique de détection des CTMs qui visent à déterminer un ensemble de vecteurs de test, ensemble qui maximise la probabilité de déclencher le CTM, dont la structure n'est pas incluse dans la netlist à disposition.

4.2 Étude Bibliographique

Les CTMs supposés furtifs sont activés quand une condition très rare survient, ceci rend le test de production inefficace pour leur détection.
Le but du de la détection par test logique est de produire un nombre de patterns réduit qui maximise la probabilité d'activation des potentiels CTMs.

À notre connaissance, la première méthode de détection de CTM basée sur le test logique a été présentée dans [WPBC08a]. L'objectif est de trouver les sites potentiels pour attacher le trigger du CTM et générer les vecteurs de test en fonction de ces sites. La procédure d'identification des triggers potentiels et de génération de patterns se fait de la façon suivante :

- Identifier les signaux faiblement contrôlables avec une simulation exhaustive. Les triggers potentiels peuvent être des combinaisons de ces signaux.
- Identifier les signaux faiblement observables en utilisant un simulateur de fautes. Ces signaux peuvent être utilisés pour la payload du CTM.
- À partir d'un ensemble Q de triggers potentiels et P Payloads potentiels, considérer Q × P CTMs potentiels.
- Un outil d'ATPG est utilisé pour trouver les patterns pour chaque CTM potentiel.

Dans [CWP+09], les auteurs proposent une technique appelée Multiple Excitation of Rare Occurence (MERO). Elle permet de générer des patterns permettant d'exciter plusieurs signaux considérés comme rares. C'est une approche statistique qui permet de maximiser la probabilité de déclenchement du CTM tout en réduisant le nombre de patterns. Le concept de base est de détecter les signaux rares, sélectionner un groupe de trigger potentiel et déterminer un ensemble de vecteurs optimal permettant d'exciter chaque signal rare N fois. Cette méthode est conceptuellement similaire au *N-detect test*, qui est utilisé pour la génération de patterns pour les fautes de collage, où un ensemble de vecteurs de test est généré pour détecter chaque faute N fois avec N différents patterns, ce qui permet d'améliorer le taux de couverture.

Dans [LKC15], les auteurs proposent une méthode qui permet de couvrir de façon exhaustive tous k-sous-ensembles de signaux. Ils arrivent à détecter les CTMs qui utilisent moins de k bits pour le trigger. Ils donnent aussi une méthode pour déterminer k. Ils assument qu'un attaquant ne pourra utiliser qu'un petit sous-ensemble, k, sur les n signaux contrôlables pour le trigger du CTM. Si le trigger est composé de k bits, le but est d'appliquer le plus petit nombre de patterns de n bits, qui couvre les 2^k valeurs possibles. Pour générer plusieurs ensembles de test de n bits pour couvrir les k sous-ensembles, ils utilisent la méthode présentée dans [TW83]. Chaque ensemble de test est composé de 1 ou plusieurs ensembles de vecteurs à poids constant.

Dans [KSTJV15], les auteurs utilisent le test combinatoire pour la détection de CTM dans un Field-Programmable Gate Arrays (FPGA). Ils montrent que le test combinatoire permet d'exciter un CTM d'une certaine taille en couvrant toutes les combinaisons des entrées primaires avec un ensemble de vecteurs réduit. Les limites de cette méthode sont qu'elle n'est pas applicable aux CTMs de grande taille et aux CTMs séquentiels et qu'elle se focalise sur les triggers composés de signaux d'entrée.

4.3 Approche de détection proposée

La méthode proposée vise à déterminer :

- les sites les plus adéquats (d'un point de vue de l'attaquant) pour insérer le trigger d'un CTM,
- des vecteurs de test permettant de déclencher le dit trigger.

Les critères pris en compte pour déterminer les sites d'insertion reflètent les choix qui peuvent être faits par un attaquant dans une fonderie non digne de confiance. En d'autres termes, les CTMs sont supposés être insérés directement dans le layout du circuit, avant la fabrication et ce, de la façon la plus discrète possible, tant d'un point de vue fonctionnel que du point de vue du layout. Trois critères sont considérés pour identifier les triggers potentiels.

Le premier critère est, comme déjà mentionné, la contrôlabilité des signaux. Les signaux de faible contrôlabilité, et qui sont donc difficiles à positionner à une valeur logique requise, sont de bons candidats pour créer un CTM.

Le second critère est la marge temporelle des signaux. De façon à rendre le CTM le plus discret possible, l'attaquant doit introduire des portes supplémentaires sans avoir d'impact sur les performances du circuit (i.e. sur sa fréquence). Dans le cas contraire, le CTM pourrait être révélé par les méthodes de détection basées sur l'analyse des retards [JM08b; EFD+13]. Les signaux ayant une marge temporelle supérieure au délai de la porte la plus rapide de la bibliothèque de portes sont donc de bons candidats.

Le troisième critère est l'espace disponible dans le layout du circuit. De façon à cacher le CTM, l'attaquant doit insérer des portes supplémentaires dans des espaces vides de façon à ne pas (ou peu) modifier le initial placement des portes [XT13]. Dans le cas contraire, le CTM peut être découvert par des techniques d'analyse optique comme présenté dans [CLMFT15b]. Les signaux suspectés, en accord avec les deux premiers critères, d'être utilisables pour déclencher le CTM sont alors groupés en accord avec le troisième critère. Le groupement est effectué selon les positions relatives de ces signaux et de l'espace libre à proximité.

4.4 Sélection de signaux « rares »

Pour éviter que le CTM ne se déclenche souvent, un attaquant va choisir de construire le trigger avec des signaux qui ont une probabilité de déclenchement très faible. La combinaison de ces signaux donne une probabilité plus faible et rend de fait le CTM furtif. C'est pourquoi il est nécessaire de développer des outils permettant de déterminer la probabilité de chaque signal et d'identifier ceux qui ont les probabilités les plus déséquilibrées. Ce calcul de probabilité peut être fait en utilisant une analyse de testabilité ou une simulation. Dans ce chapitre nous allons présenter les deux approches.

4.4.1 Probabilité

La première étape est de déterminer les signaux activés rarement. Pour ce faire, une première possibilité est d'effectuer une simulation logique du circuit et de compter pour chaque signal

le nombre de fois où celui-ci vaut ’0’ et ’1’. Toutefois, une simulation exhaustive étant impossible, une simulation avec un sous-ensemble de stimuli peut conduire à des résultats erronés (et ce d’autant plus que les stimuli employés correspondent généralement à un fonctionnement standard du circuit). Nous proposons une alternative basée sur l’analyse de testabilité pour déterminer ces signaux peu contrôlables.

Nous avons choisi d’utiliser une approche similaire à celle utilisé en analyse de testabilité "COPE". Cette méthode est détaillée dans [CB09; DNDFR13a].
Notre approche consiste, pour chaque signal, à calculer les probabilités qu’il vaille ’0’ ou ’1’, les probabilités des entrées primaires étant supposées égales à 1/2. Les signaux de faible contrôlabilité (resp. à ’0’ ou ’1’) sont les signaux dont les probabilités d’être à ’0’ ou ’1’ sont très différentes. Nous parlerons dans ce cas de "signal déséquilibré". Les entrées primaires du circuit ainsi que les éléments de mémorisation (FFs) sont supposés pouvoir avoir indifféremment avoir la valeur ’1’ ou ’0’, i.e. leurs probabilités sont supposées équilibrées.

Les Figures 4.3 et 4.4 explicitent le processus de calcul. Sur la Figure 4.3, sont illustrés les calculs de la probabilité que la sortie S d’une porte vaille ’1’. Ces probabilités sont propagées dans le circuit comme illustré à la Figure 4.4. Bien évidemment ce calcul n’est valide que pour autant que les entrées soient indépendantes.

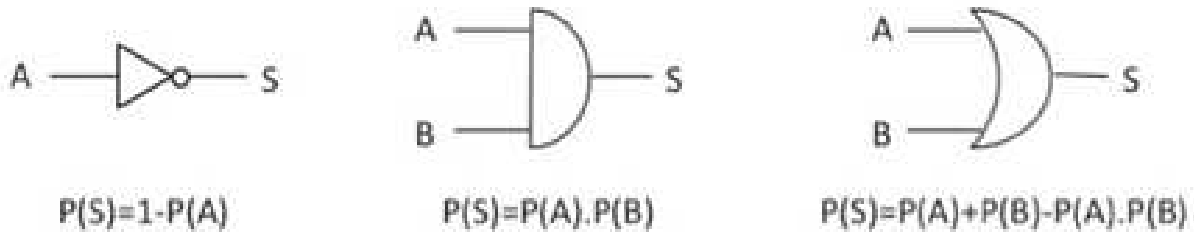


FIGURE 4.3 – Probabilités de la sortie d’une porte d’être à ’1’ en fonction des probabilités.

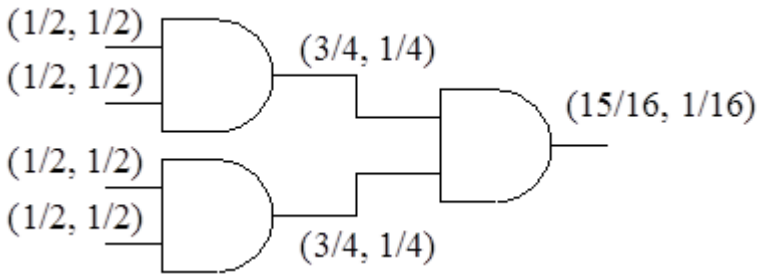


FIGURE 4.4 – Probabilités des valeurs ’0’ et ’1’ sur chaque signal.

Il faut noter que la méthode que nous avons développée pour calculer ces probabilités prend en compte les chemins divergents et reconvergents, i.e. les corrélations entre signaux d’entrée de portes [XT13], comme illustré par le calcul de probabilité de la Figure 4.5.
La probabilité de la sortie S de valoir "1" est :

$$P(S=1) = (\frac{1}{2} \times \frac{1}{2}) + (\frac{1}{2} \times \frac{1}{2}) - (\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}) = \frac{3}{8} \quad (4.1)$$

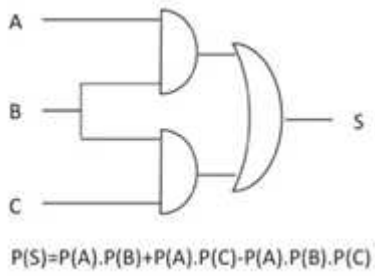


FIGURE 4.5 – Probabilités de la sortie d'une porte d'être à '1' en cas de divergence/reconvergence.

Pour cet exemple, la probabilité d'être à '1' du signal de sortie S est plus faible que celle d'être à '0', donc on peut parler de signal déséquilibré. Les détails de cette méthode sont donnés dans [CB09].

4.4.2 Commutation et Clustering

Une alternative à l'analyse de testabilité est de faire une analyse fréquentielle pour chercher les signaux qui ont un comportement fréquentiel différent de la distribution générale. Cette approche permet de trouver des points isolés, nommés « outliers », dans un ensemble de données, comme introduit dans [ÇM15]. Le principe consiste à faire une simulation et analyser le comportement fréquentiel de chaque signal et déterminer une similarité ou dissimilarité entre les signaux en fonction de leur comportement en fréquence (commutation). On utilise le clustering pour regrouper les signaux en fonction de leur similarité. Enfin on détermine analytiquement les signaux qui ne font partie d'aucun groupe (i.e qui commutent différemment des autres signaux).
Nous considérons que ces signaux peuvent être de bons candidats pour attacher un trigger de CTM. En plus des signaux isolés, nous considérons aussi les autres signaux qui font partie d'un groupe mais qui ont un nombre de points autour d'eux très faible comparée à la distribution générale.

4.4.2.1 Simulation et traitement du signal

La première étape est d'appliquer des vecteurs à l'entrée du circuit et récupérer la réponse. Les valeurs discrètes que nous récupérons pour chaque signal du circuit constituent leurs vecteurs. Les vecteurs appliqués à l'entrée peuvent être des patterns de validation, des patterns aléatoires, des patterns de test ou une combinaison des différents types. D'après nos expériences, il n'y a pas une grande différence en termes de signaux isolés entre ces différentes approches.

La deuxième étape est de déterminer la corrélation entre les signaux voisins (les entrées et sorties d'une porte). Pour ce faire, nous calculons le produit de convolution entre les entrées et la sortie.

La troisième étape est de réduire le nombre de coefficients obtenus avec le produit de convolution, pour faciliter le calcul par la suite. Puisque nous travaillons avec des signaux discrets et non périodiques, nous calculons la Transformée de Fourier Discrète (TFD), et en déduisons la Densité Spectrale d'Enrgie (DSE).

4.4.2.2 Convolution

Nous générons des vecteurs d'entrée à l'aide d'un outil d'ATPG. Ensuite nous faisons une simulation pour obtenir les réponses et créons des vecteurs en fonction des réponses successives. Considérons par exemple, le signal « a » qui a pour vecteur : v(a) = 10101010.
A partir de ce vecteur, nous déterminons le vecteur de commutation de « a » comme suit :
switch(a)[0] = v(a)[0]
switch(a)[i] = v(a)[i] xor v(a)[i-1]
Ensuite, nous pouvons calculer le produit de convolution entre chaque paire d'entrée sortie. Soit 2 signaux « a » et « b » qui pilotent une porte à entrées, le signal « g » étant la sortie de cette porte et SV_a, SV_b, SV_g étant respectivement les vecteurs de commutation des signaux a, b et g. le produit de convolution entre a et g est obtenu comme suit :

$$C_{ag}[n] = (SV_a * SV_g)[n] = \sum_{-\infty}^{+\infty} SV_a[m]SV_g[m+n] \tag{4.2}$$

4.4.2.3 Transformée de Fourier Discrète (TFD)

La TFD du produit de convolution entre a et g est le suivant :

$$S_{ag}(k) = \sum_{n=0}^{N-1} C_{ag}(n).e^{-2i\pi k\frac{n}{N}} \quad (0 \leq k \leq N) \tag{4.3}$$

4.4.2.4 Énergie

En utilisant le théorème de Parseval, on peut déduire la DSE à partir de la TFD (cf 4.3).

$$\phi_{S_{ag}}(k) = |S_{ag}|^2 \tag{4.4}$$

$$\phi_{S_{ag}}(k) = identité de Parseval \tag{4.5}$$

La DSE est l'aire occupée par les raies spectrales du signal obtenues avec la TFD. Chaque raie représente une composante fréquentielle.
Puisque nous essayons de trouver une similarité entre signaux voisins en fonction de leur comportement en fréquence (commutation), l'énergie nous apporte l'information sur la fréquence avec un nombre de coefficients minimal.

4.4.2.5 Clustering

Notre objectif est de trouver à partir d'un ensemble de données des points isolés appelés « outliers » qui peuvent être utilisés comme triggers pour des CTMs.
Pour ce faire, nous utilisons un algorithme de clustering. Le clustering est en effet le moyen de grouper un ensemble d'objets de telle manière que les objets d'un même groupe sont similaires entre eux. L'algorithme de clustering forme des groupes (clusters) de signaux ayant une activité similaire et les signaux qui n'appartiennent à aucun groupe (cluster) sont considérés comme « outliers ».

La première étape est de déterminer la corrélation entre tous les signaux en fonction de la corrélation entre signaux voisins (DSE). Pour ce faire, nous normalisons l'énergie en distance et en déduisons une matrice de distance carrée. Puisque notre algorithme travaille avec une

matrice de coordonnées, nous transformons notre matrice de distance en matrice de coordonnées. Chaque signal du circuit représente un point.
La dernière étape est de déterminer analytiquement les points isolés en calculant la distance entre chaque signal et le nombre de signaux dans son voisinage immédiat.

4.4.2.6 Normalisation de l'énergie en distance

Tout d'abord, nous transformons la netlist en graph orienté (énergie = poids). Ensuite, nous normalisons le poids et déterminons la similarité structurelle σ(u,v) entre deux signaux adjacents u et v, comme décrit dans [HSH+10] :

$$\sigma(u,v) = \frac{\sum_{x \in \Gamma(u) \cap \Gamma(v)} w(u,x).w(v,x)}{\sqrt{\sum_{x \in \Gamma(u)} w^2(u,x)}.\sqrt{\sum_{x \in \Gamma(v)} w^2(v,x)}} \tag{4.6}$$

Où Γ(u) représente le voisinage du signal u avec u inclus et tous les signaux adjacents ($w(u,u) = 1$).
Après avoir obtenu les valeurs de similarité pour tous les signaux, on définit la distance entre chaque signal et les signaux qui lui sont adjacents comme étant l'inverse de la similarité.
La distance entre deux signaux non-adjacents u et v est définie comme étant la somme des distances du chemin le plus court les reliant. Nous utilisons l'algorithme de Dijkstra [WF16] pour calculer de tels chemins.

$$distance(u,v) = \frac{1}{\sigma(u,v)} \tag{4.7}$$

Après avoir calculé la distance entre les signaux voisins, nous pouvons en déduire une matrice N*N, N étant le nombre total de signaux.

4.4.2.7 Transformation de la matrice de distance (N*N) en matrice de coordonnées (N*2)

Cette matrice est de taille (N*N), N étant le nombre total de signaux dans le circuit. Nous faisons une transformation dans le but d'avoir une matrice de coordonnées. Avec ces coordonnées, nous sommes en mesure d'avoir une représentation bi-dimensionnelle de tous les signaux en fonction de leur activité. Pour ce faire, nous utilisons la méthode Classical MultiDimensional Scaling (CMDS) décrite dans [Wan12]. Un exemple est donné en figure 4.6 d'une représentation 2-d du circuit c3540 après transformation et obtention de la matrice de coordonnées.
Chaque point représente un signal. Nous observons plusieurs groupes plus ou moins denses et des points plus ou moins isolés de ces groupes. De tels points sont appelés « outliers ».

4.4.2.8 Algorithme de détection d'outliers

Dans [BKNS99], une méthode permettant de trouver des outliers est présentée. Les outliers sont les objets déviant de la distribution générale d'un ensemble de points. En d'autres termes, être un outlier signifie ne pas faire partie ou être proche d'aucun groupe (cluster). Cet algorithme permet de déterminer un coefficient pour chaque signal en fonction de sa distance par rapport aux autres, le « outlier factor (of) » qui est défini comme suit :

FIGURE 4.6 – 2-d dataset de l'AES.

$$of_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts(o)}}{lrd_{MinPts(p)}}}{|\, N_{MinPts}(p)\,|} \tag{4.8}$$

Avec *lrd* (local reachability density), la densité locale d'un objet *p*. *MinPts* est le nombre minimum de points dans le voisinage de *p*.
Le « outlier factor » d'un objet p est le coefficient qui permet de déterminer si un signal est un outlier.

4.4.2.9 Normalisation des outlier factor *(of)*

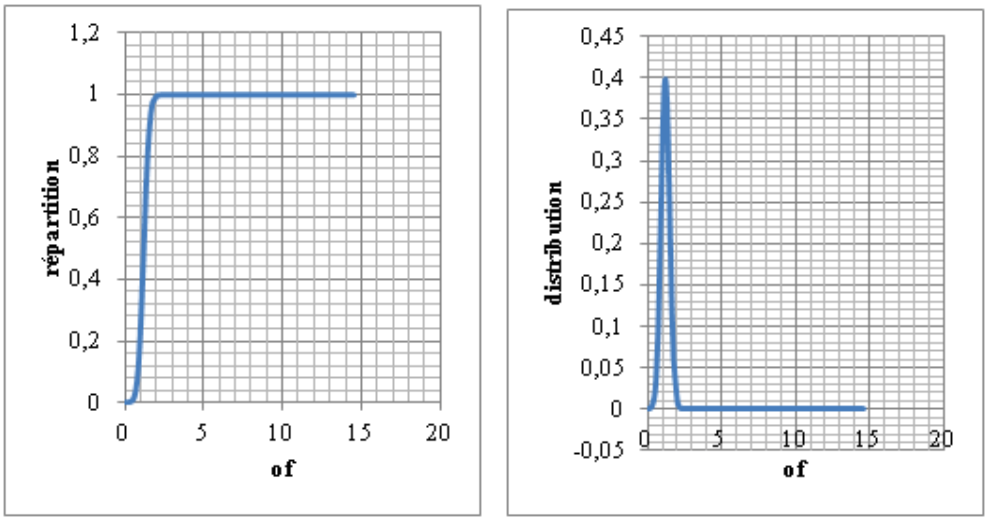
Notre objectif est d'avoir un *of* compris entre 0 et 1 pour chaque signal. Pour ce faire, nous utilisons la méthode présentée dans [KKSZ11] où il est montré que les *of* suivent une loi Normale de moyenne μ et d'écart type σ. Ainsi, nous pouvons utiliser la fonction de répartition et la fonction d'erreur *erf()* pour transformer les *of* en probabilité (P_o).
L'équation 4.11 montre comment normaliser les *of* en probabilité.

$$of \sim N(\mu, \sigma)$$

$$\mu = E(of) \tag{4.9}$$

$$\sigma^2 = E(of^2) - E(of)^2 \tag{4.10}$$

$$P_o = \frac{1}{2} \times \left(1 + erf\left[\frac{of_o - \mu}{\sigma \times \sqrt{2}}\right]\right) \tag{4.11}$$



(a) Fonction de répartition des of du c3540

(b) Distribution des of du c3540

FIGURE 4.7 – Distribution des of du circuit c3540.

La figure 4.7b illustre bien que les *of* suivent une loi normale. La fonction de répartition comme illustrée sur la figure 4.7a, nous permet de déterminer la probabilité de chaque signal d'être un outlier. Si un signal a une probabilité de 1, il est isolé par rapport aux différents

groupes de signaux. Si sa probabilité est proche de 1, il ne fait partie d'aucun groupe mais est proche d'un groupe.

Cette approche comparée à l'analyse de testabilité présente l'avantage de donner un nombre signaux rares réduit. Sa limite se situe au niveau des vecteurs simulation. En effet, les résultats peuvent varier selon les vecteurs utilisés.

4.5 Regroupement et réduction du nombre de signaux

Une fois que nous avons trouvé les signaux suspectés d'être les signaux de commande du CTM avec l'analyse de testabilité ou l'analyse fréquentielle, nous réduisons le nombre de signaux en faisant intervenir la marge temporelle et la proximité géométrique au niveau du GDSII.

4.5.1 Marge temporelle

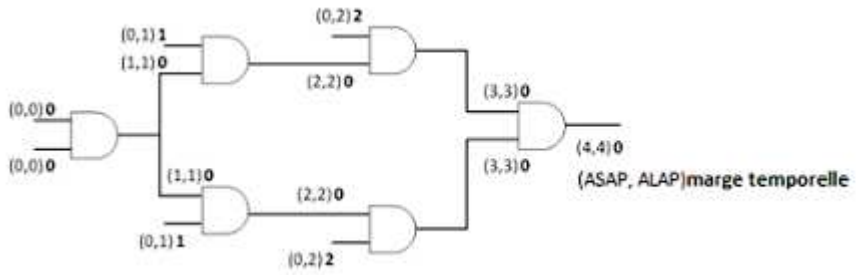


FIGURE 4.8 – Marges temporelles.

Comme nous l'avons mentionné en introduction, nous supposons que l'introduction d'un CTM ne modifie pas les performances du circuit. Or, l'ajout d'un CTM modifie le délai des chemins logiques sur lesquels le CTM est attaché par augmentation du fan-out des portes en amont. Eu regard de l'hypothèse ci-dessus, les signaux de déclenchement du CTM sont donc des signaux ayant une marge temporelle suffisante.
La seconde étape consiste donc à calculer la marge temporelle de chaque signal. En se basant sur les modèles de délai de portes fournis avec la bibliothèque de portes et les interconnections, une analyse temporelle est effectuée pour calculer ces marges :

- à partir des temps d'arrivée des signaux sur les entrées primaires, l'analyse propage les temps d'arrivée aux portes en aval (temps au plus tôt),
- à partir des temps d'arrivée requises aux entrées des bascules ou sur les sorties, une rétro-propagation est effectuée pour calculer les temps au plus tard.

La marge temporelle est la différence entre le temps d'arrivée du signal au plus tôt et le temps d'arrivée au plus tard [JM08b]. Le calcul des marges temporelles est montré dans la figure 4.8 (en supposant un délai unitaire pour chaque porte).
Pour obtenir une estimation aussi précise que possible de ces marges temporelles, ce calcul est effectué après placement/routage du circuit de façon à prendre en compte non seulement les délais des portes mais aussi les délais d'interconnexion. Par ailleurs, c'est ce qui reflète le mieux l'information que peut obtenir un attaquant à partir du GDSII.

4.5.2 Proximité géométrique

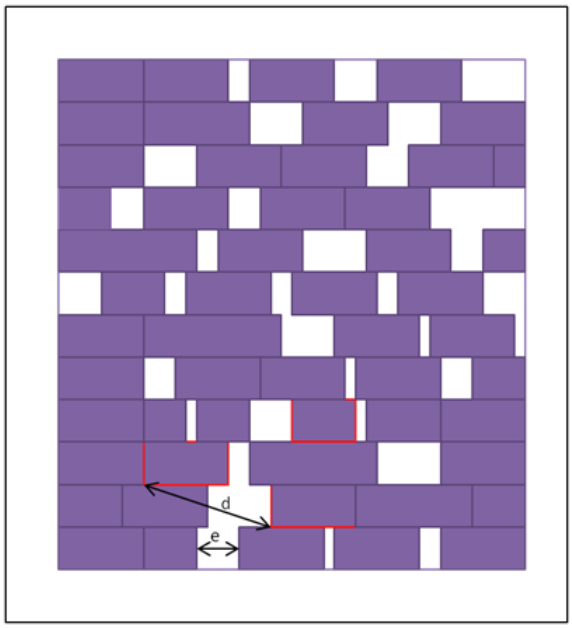


FIGURE 4.9 – Proximité géométrique.

On fait l'hypothèse que l'insertion d'un CTM ne doit pas modifier profondément la structure du GDSII et donc avoir un impact sur la surface. De ce fait les signaux qui sont utilisés pour le trigger du CTM doivent être proches. Nous regardons si les portes auxquelles les signaux suspects sont connectés sont proches entre elles mais aussi également si un espace vide dans le layout est disponible à leur proximité de façon à pouvoir insérer les portes du CTM. Pour ce faire, on définit tout d'abord une distance seuil. On calcule ensuite la distance d entre chaque porte et on la compare à la distance seuil. Si la distance est inférieure ou égale au seuil, on regarde s'il existe un espace vide à proximité, espace suffisamment grand pour qu'un attaquant puisse insérer une porte.
La figure 4.9 illustre le calcul de cette distance où e est la largeur de la plus petite porte de la bibliothèque.

4.5.3 Création des sous-ensembles de signaux

La dernière étape consiste à créer des sous-ensembles de signaux parmi les signaux sélectionnés précédemment. Chaque sous-ensemble est alors un groupe de signaux permettant de déclencher un CTM potentiel.
Le flot présenté sur la figure 4.10 représente notre première approche. Elle est basée sur l'hypothèse que le trigger du CTM est une combinaison de signaux rares. Pour rendre notre méthode plus robuste, nous proposons un deuxième flot présenté dans le paragraphe suivant.

4.5.4 Discussions et limites

Il faut noter que si le nombre de signaux identifiés est égal à M, cette méthode conduira à la production d'un nombre de groupes de signaux G égal à :

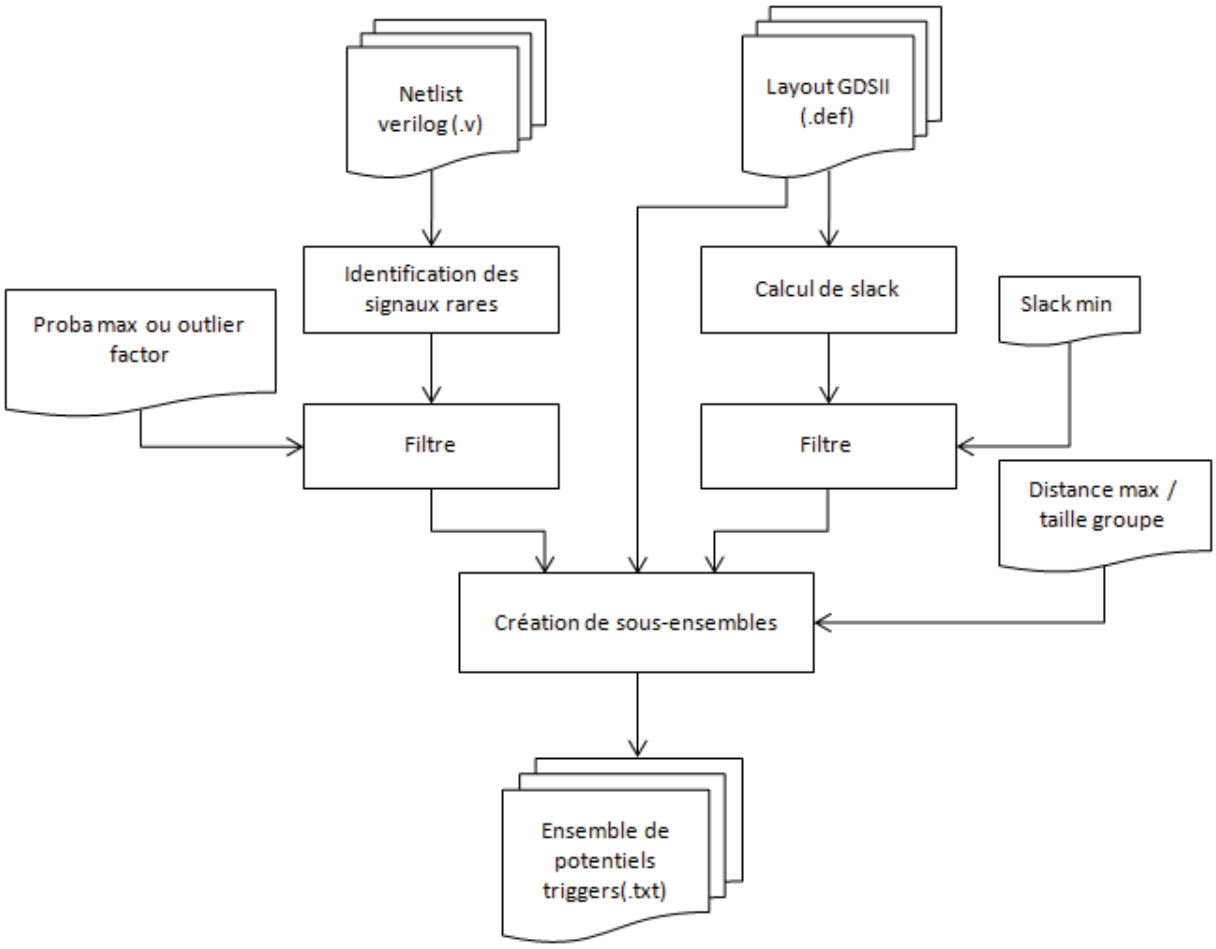


FIGURE 4.10 – Création des sous-ensembles.

$$G = \sum_{i=2}^{k} C_M^i \tag{4.12}$$

Il est donc évident que cette méthode n'est envisageable que pour identifier des CTMs dont le nombre de signaux servant au déclenchement est relativement limité, par exemple 5 signaux.
Dans le cas contraire, la détection par des méthodes Side-Channel est plus appropriée.

Il faut également noter qu'une condition de déclenchement rare sur les signaux n'est pas synonyme de la rareté de chacun des signaux. ceci est l'objet du paragraphe suivant.

4.6 Condition rare à partir de signaux non rares

À notre connaissance, toutes les méthodes de détection basées sur le test logique qui ont été proposées reposent sur la sélection de signaux rares en faisant une analyse de testabilité : on suppose que la condition de trigger est une combinaison de valeurs rares sur des signaux faiblement contrôlables [WPBC08b; CWP+09; DNDFR13a].

Toutefois, une condition de trigger rare sur plusieurs signaux n'est pas forcément synonyme de valeurs rares sur chacun des signaux. Par exemple, deux signaux peuvent avoir la probabilité 0,5 d'être à '0' (contrôlabilité maximale) alors que la combinaison (0,0) sur ces deux signaux n'arrive que très rarement. En ce sens, nous avons développé une seconde procédure d'identification des sites potentiels de trigger [DBF+15].

Les groupes de signaux de triggers sont tout d'abord construits en fonction de la marge temporelle et de la proximité des signaux sur le layout. La probabilité d'occurrence de chaque valeur de ce groupe de signaux est ensuite calculée. La procédure générale est illustrée sur la Figure 4.11.

La procédure de calcul des occurrences de valeurs de groupes de signaux se déroule comme suit. Soit par exemple le groupe constitué de 2 signaux S1 et S2 (proches sur le layout et dont la marge temporelle permet l'insertion de portes logiques supplémentaires).
Pour vérifier si l'une des 4 combinaisons de valeurs (0,0), (0,1), (1,0), (1,0) est rare, nous introduisons 4 portes logiques fictives dans la netlist et calculons les probabilités des signaux de sortie de ces portes. Ces probabilités reflètent les fréquences d'occurrences des jeux de valeurs de l'ensemble de S1, S2. Ceci est illustré à la Figure 4.12. Par exemple, la probabilité du signal O1 d'être à '1' représente la fréquence de la valeur (1,1) du groupe (S1, S2). De façon générale, la rareté d'une valeur d'un groupe de N signaux est donnée par la relation 4.13 :

$$R = 1 - P.2^N \tag{4.13}$$

Il faut noter que cette deuxième approche reste confrontée au même problème d'explosion combinatoire que l'approche précédente si le nombre réel de signaux de déclenchement du CTM est important.

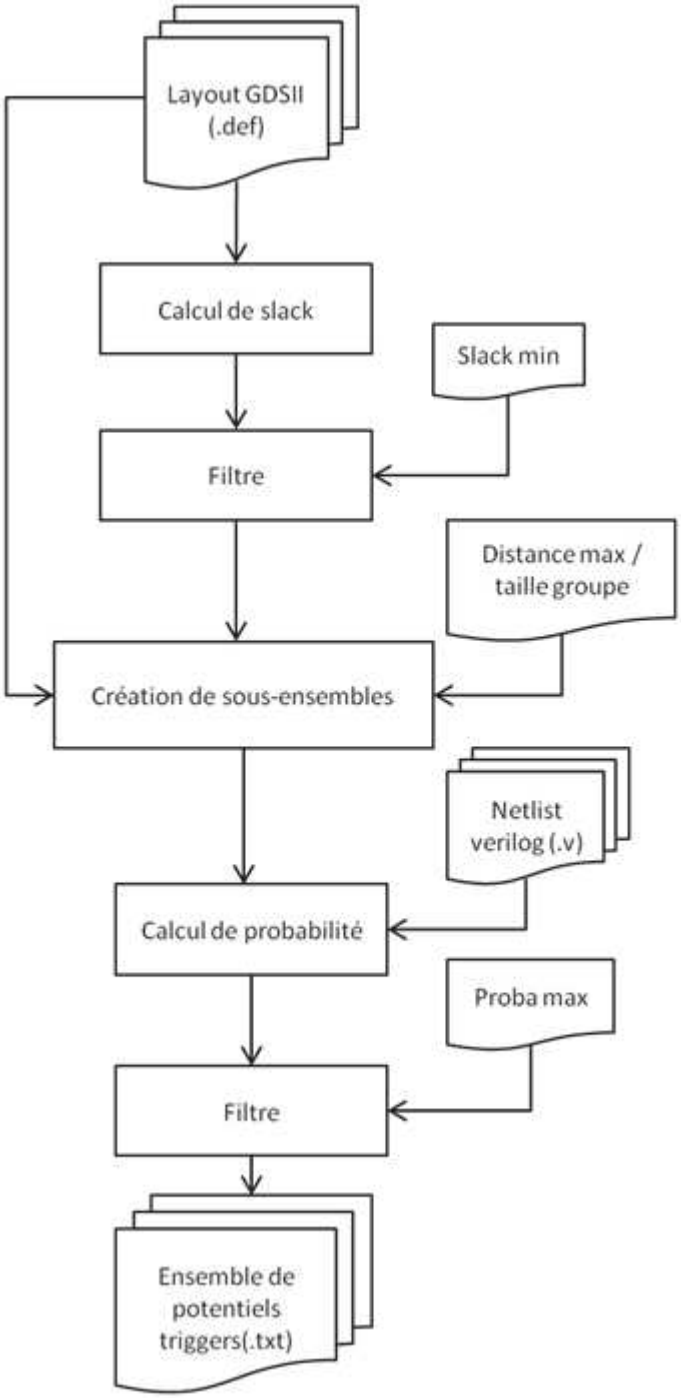


FIGURE 4.11 – Création des sous-ensembles, version "améliorée".

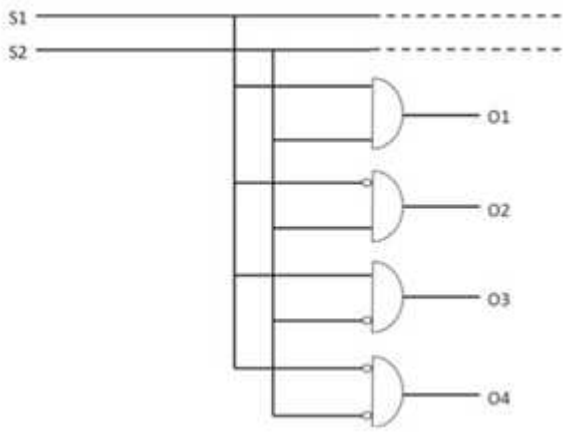


FIGURE 4.12 – Calcul des fréquences d'occurrences des jeux de valeurs d'un groupe de 2 signaux.

4.7 Génération des vecteurs de déclenchement

La dernière étape de la méthode consiste à trouver des vecteurs capables de justifier les valeurs rares pour chaque groupe de signaux déterminés précédemment. En comparant les réponses attendues à ces stimuli (réponses obtenues par simulation sur la netlist) et les réponses sur le circuit réel, on détecte la présence de CTMs. Pour déterminer les vecteurs de déclenchement, nous utilisons un outil d'ATPG.

4.7.1 Génération de fautes de collage

En partant de la liste de groupes de signaux pouvant être des triggers, on insère dans la netlist pour chaque groupe, une porte AND fantôme dont les entrées sont connectées aux signaux du groupe. Les signaux ayant la valeur rare '1' sont directement connectés aux entrées de la porte AND, ceux ayant la valeur rare '0' sont inversées avant d'être connectés à la porte AND. La sortie de cette porte est ensuite testée pour générer le vecteur permettant de justifier le collage à '0' (sa0), ce qui revient à avoir un '1' logique sur ce nœud.

Le vecteur de déclenchement est obtenu à l'aide de l'ATPG en cherchant un vecteur de test de la faute de collage à '0' de la sortie de la porte AND. Ce vecteur force la valeur '1' à la sortie de la porte AND et donc positionne les signaux de déclenchement à leur valeur rare, la seule combinaison qui produise un '1' sur la sortie de la porte AND.

4.7.2 Passage de circuit séquentiel en combinatoire

Il faut noter que les outils d'ATPG ne sont en général capables de générer un vecteur de test que si le circuit est combinatoire ou doté de chaines de scan. Pour contourner ce problème et pouvoir déterminer les vecteurs de déclenchement dans les circuits séquentiels, nous séparons la partie combinatoire de la partie séquentielle et déroulons la partie combinatoire dans le temps en l'instanciant n fois. La figure 4.13 illustre la procédure.

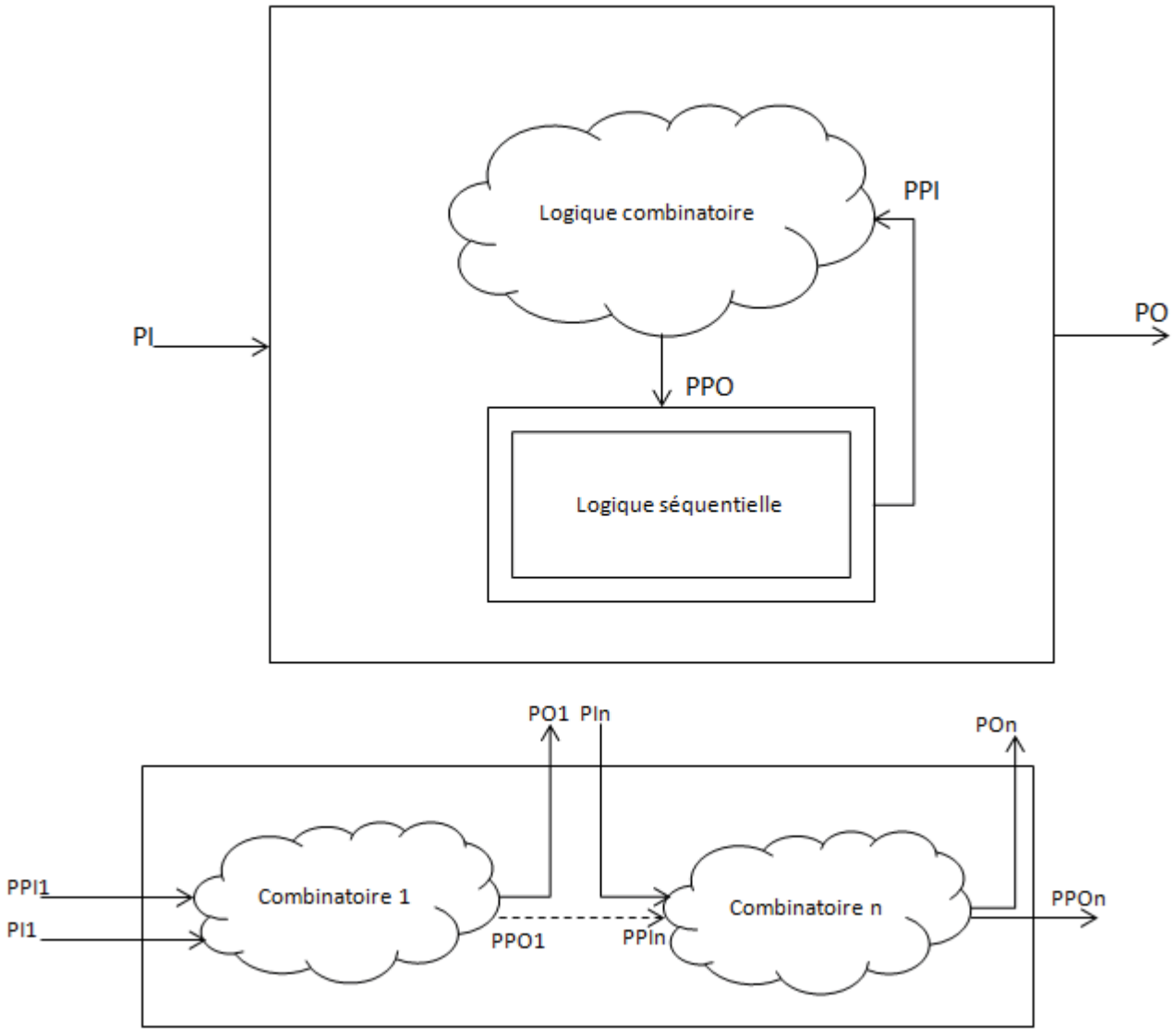


FIGURE 4.13 – Déroulement d'un circuit dans le temps.

4.8 Résultats

Nous avons évalué cette méthode sur des circuits ISCAS. Tout d'abord nous avons utilisé notre méthode pour trouver les triggers potentiels. Ensuite nous avons utilisé un outil d'ATPG commercial [Syn16] pour obtenir les vecteurs de test. Pour finir nous avons fait une simulation avec des vecteurs aléatoires pour vérifier la furtivité des triggers trouvés.

4.8.1 Sélection de triggers et génération de vecteurs avec la première approche

Pour chaque circuit, nous avons déterminé les triggers potentiels de la manière suivante :

- Calcul de probabilité de chaque signal et sélection des signaux ayant des probabilités déséquilibrées.
- Sélection des signaux ayant une marge temporelle positive parmi les signaux sélectionnés précédemment.
- Identification parmi les signaux précédents de ceux qui sont proches dans le layout.

Les résultats de cette expérience sont présentés sur le Tableau 5.1.

TABLEAU 4.1 – Recherche de triggers potentiels avec des signaux rares

Bench	Critères	Nombre de signaux	Nombre de triggers
c6288	Proba ≤ 0.1 Slack ≥ 0 Distance ≤ 20 µm	6	5
s13207	Proba ≤ 0.001 Slack ≥ 0 Distance ≤ 4 µm	48	11

Pour le circuit c3288, 6 signaux remplissent les deux premiers critères et nous avons formé 5 triggers à 2 entrées avec ces 6 signaux compte tenu de leur distance les uns par rapport aux autres. Pour le circuit s13207, 48 signaux remplissent les 3 critères, et nous en avons déduit 10 triggers à 2 entrées et 1 trigger à 3 entrées. Pour vérifier que les triggers sont furtifs, nous avons fait une simulation avec 10^6 vecteurs aléatoires pour chaque circuit.
De plus, pour chaque trigger, nous avons appliqué notre procédure pour générer le vecteur de test permettant d'exciter le trigger, prouvant qu'un tel vecteur existe.
Ces résultats de simulation sont présentés dans le Tableau 4.2. Chaque case donne le pourcentage de déclenchement des CTMs.

TABLEAU 4.2 – Résultats de simulation avec la première approche

Bench	0%	≤ 0.01%	≤ 0.1%	≤ 1%	≥ 1%
c6288	1	0	0	0	4
s13207	1	1	2	6	1

Nous observons que 4 triggers ne sont pas furtifs pour le circuit c6288. Ils sont déclenchés en moyenne 1.34% sur l'ensemble des vecteurs aléatoires appliqués. Ceci est logique car aucun signal rare n'existe dans ce circuit. Les 6 signaux utilisés pour les triggers ont les probabilités

les plus déséquilibrées (0.09 d'être à 0), ce qui n'est pas si rare. Le dernier trigger à 2 entrées est le plus furtif avec une moyenne de 0% sur l'ensemble des vecteurs. Pour le circuit s13207, 1 seul trigger n'est pas furtif avec un pourcentage de déclenchement supérieur à 1%. 1 trigger n'a jamais été déclenché durant la simulation et 9 autres ont une moyenne très faible, donc peuvent être considérés comme furtifs.

4.8.2 Sélection de triggers et génération de vecteurs avec la deuxième approche

Dans toutes les approches présentées dans la littérature, la sélection des signaux rares se fait en fonction de la contrôlabilité de chaque signal. Une condition rare est définie comme étant le résultat d'une combinaison de signaux rares (faiblement contrôlables). Un trigger dont les signaux ont des probabilités équilibrées serait facilement détectable par des séquences de test ou de vérification fonctionnelle[WPBC08c; WPBC08d; DNDFR13b]. Dans cette partie, nous présentons les résultats obtenus en utilisant la deuxième approche (cf 4.6) qui vise à déclencher de tels CTMs. Nous montrons ici qu'il est possible de créer une condition rare à partir de signaux contrôlables.

Dans les deux circuits nous avons construit des CTMs déclenchables par des signaux non rares. Pour ce faire, nous avons sélectionné les signaux qui ont une marge temporelle positive et qui sont proches dans le layout. Ensuite, nous avons créé des triggers avec ces signaux et avons calculé les probabilités de chaque trigger. Enfin nous avons éliminé les triggers qui ne sont pas rares. Les résultats sont présentés sur les Tableaux 4.3 et 4.4.

TABLEAU 4.3 – Recherche de triggers potentiels des signaux non rares

Circuit	Critères	Nombre de triggers
c6288	Slack ≥ 0 Distance ≤ 10 µm Nombre d'entrée du trigger = 2	274
	≤ 0.05 Proba ≥ 0	29
s13207	Slack ≥ 0 Distance ≤ 20 µm Nombre d'entrée du trigger = 2	159
	≤ 0.001 Proba ≥ 0	21

Dans le Tableau 4.3, nous observons que pour le circuit c6288 parmi les 274 triggers obtenus selon les deux premiers critères (marge temporelle et proximité dans le layout), seuls 54 ont une valeur dont la probabilité est inférieure à 0.05. Parmi ceux-ci, 25 ont une probabilité de 0, ce qui laisse 29 triggers potentiels. Pour le circuit s13207, parmi les 159 triggers, 46 ont une probabilité inférieure à 0.001 (dont 25 égale à 0), ce qui laisse 21 triggers potentiels.

Après avoir déterminé les triggers potentiels, nous avons lancé une simulation avec des vecteurs aléatoires pour vérifier si les triggers sont furtifs ou non. Les résultats sont présentés dans le Tableau 4.4.
Dans le Tableau 4.4, nous observons que pour le circuit c6288, parmi les 29 triggers, 5 n'ont jamais été déclenché durant la simulation. Pour le circuit s13207, 3 triggers n'ont jamais été

TABLEAU 4.4 – Résultats de simulation pour la deuxième approche

Bench	**0%**	**≤ 0.01%**	**≤ 0.1%**	**≤ 1%**	**≥ 1%**
c6288	5	0	0	0	25
s13207	3	3	10	4	1

déclenché et 3 autres ont été déclenché très rarement. Ces résultats confirment la rareté de plusieurs triggers créés à partir de signaux ayant des probabilités équilibrées.

À l'inverse des méthodes proposées dans la littérature, notre méthode est donc capable de détecter ce type de CTMs.

4.8.3 Limites

En raison de problèmes d'explosion combinatoire, cette méthode est plus particulièrement dédiée aux CTMs attachés à un nombre réduit de signaux.

4.9 Conclusion

Dans ce chapitre, nous avons présenté une procédure permettant d'identifier les sites d'un circuit où il est possible d'insérer un CTM furtif facilement dans une fonderie non fiable. La sélection de ces sites est basée sur l'hypothèse que le CTM est activé par des signaux avec des probabilités déséquilibrées, qui ne se trouvent pas sur des chemins critiques en termes de délai, doivent être proches dans le layout et proches d'un espace vide. Le calcul des probabilités est basé sur l'analyse fréquentielle (commutation) ou sur l'analyse de testabilité. Un ATPG est utilisé pour déterminer les vecteurs de test permettant d'exciter les triggers trouvés.
De plus, nous proposons une méthode qui permet de déclencher un CTM dont la condition de déclenchement, bien que rare, n'est pas composée de signaux rares.

4.10 Références

[BKNS99] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Optics-of : Identifying local outliers. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 262–270. Springer, 1999.

[CB09] Rajat Subhra Chakraborty and Swarup Bhunia. Security against hardware trojan through a novel application of design obfuscation. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, pages 113–116. ACM, 2009.

[CLMFT15a] Franck Courbon, Philippe Loubet-Moundi, Jacques JA Fournier, and Assia Tria. Semba : A sem based acquisition technique for fast invasive hardware trojan detection. In *European Conference on Circuit Theory and Design*, pages 1–4. ECCTD, 2015.

[CLMFT15b] Franck Courbon, Philippe Loubet-Moundi, Jacques JA Fournier, and Assia Tria. Semba : A sem based acquisition technique for fast invasive hardware trojan detection. In *European Conference on Circuit Theory and Design*, pages 1–4. ECCTD, 2015.

[ÇM15] Burçin Çakir and Sharad Malik. Hardware trojan detection for gate-level ics using signal correlation based clustering. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*, pages 471–476. EDA Consortium, 2015.

[CWP+09] Rajat Subhra Chakraborty, Francis Wolff, Somnath Paul, Christos Papachristou, and Swarup Bhunia. Mero : A statistical approach for hardware trojan detection. In *Cryptographic Hardware and Embedded Systems-CHES*, pages 396–410. Springer, 2009.

[DBF+15] Sophie Dupuis, Papa-Sidy Ba, Marie-Lise Flottes, Giorgio Di Natale, and Bruno Rouzeyre. New testing procedure for finding insertion sites of stealthy hardware trojans. In *2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 776–781. DATE, 2015.

[DNDFR13a] Giorgio Di Natale, Sophie Dupuis, Marie-Lise Flottes, and Bruno Rouzeyre. Identification of hardware trojans triggering signals. In *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices*. TRUDEVICE, 2013.

[DNDFR13b] Giorgio Di Natale, Sophie Dupuis, Marie-Lise Flottes, and Bruno Rouzeyre. Identification of hardware trojans triggering signals. In *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices*. TRUDEVICE, 2013.

[EFD+13] Ingrid Exurville, Jacques Fournier, Jean-Max Dutertre, Bruno Robisson, and Assia Tria. Practical measurements of data path delays for ip authentication & integrity verification. In *8th International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip*, pages 1–6. ReCoSoC, 2013.

[HSH+10] Jianbin Huang, Heli Sun, Jiawei Han, Hongbo Deng, Yizhou Sun, and Yaguang Liu. Shrink : a structural clustering algorithm for detecting hierarchical communities in networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 219–228. ACM, 2010.

[JM08a] Yier Jin and Yiorgos Makris. Hardware trojan detection using path delay fingerprint. In *Hardware-Oriented Security and Trust*, pages 51–57. HOST, 2008.

[JM08b] Yier Jin and Yiorgos Makris. Hardware trojan detection using path delay fingerprint. In *Hardware-Oriented Security and Trust*, pages 51–57. HOST, 2008.

[KKSZ11] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Interpreting and unifying outlier scores. In *SDM*, pages 13–24. SIAM, 2011.

[KSTJV15] Paris Kitsos, Dimitris E Simos, Jose Torres-Jimenez, and Artemios G Voyiatzis. Exciting fpga cryptographic trojans using combinatorial testing. In *IEEE 26th International Symposium on Software Reliability Engineering*, pages 69–76. ISSRE, 2015.

[LKC15] Nicole Lesperance, Shrikant Kulkarni, and Kwang-Ting Cheng. Hardware trojan detection using exhaustive testing of k-bit subspaces. In *The 20th Asia and South Pacific Design Automation Conference*, pages 755–760. DAC, 2015.

[Syn16] Inc. Synopsys. Tetramax., 2016. http://www.synopsys.com/Tools/Implementation/RTLSynthesis/Test/Pages/TetraMAXATPG.aspx.

[TW83] Donald T. Tang and Lin S. Woo. Exhaustive test pattern generation with constant weight vectors. *IEEE Transactions on Computers*, 100(12) :1145–1150, 1983.

[Wan12] Jianzhong Wang. Classical multidimensional scaling. In *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*, pages 115–129. Springer, 2012.

[WF16] Inc. Wikimedia Foundation. Algorithme de dijkstra, 2016. https://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra.

[WPBC08a] Francis Wolff, Chris Papachristou, Swarup Bhunia, and Rajat S Chakraborty. Towards trojan-free trusted ics : Problem analysis and detection scheme. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1362–1365. ACM, 2008.

[WPBC08b] Francis Wolff, Chris Papachristou, Swarup Bhunia, and Rajat S Chakraborty. Towards trojan-free trusted ics : Problem analysis and detection scheme. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1362–1365. ACM, 2008.

[WPBC08c] Francis Wolff, Chris Papachristou, Swarup Bhunia, and Rajat S Chakraborty. Towards trojan-free trusted ics : Problem analysis and detection scheme. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1362–1365. ACM, 2008.

[WPBC08d] Francis Wolff, Chris Papachristou, Swarup Bhunia, and Rajat S Chakraborty. Towards trojan-free trusted ics : Problem analysis and detection scheme. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1362–1365. ACM, 2008.

[XT13] Kan Xiao and Mohammed Tehranipoor. Bisa : Built-in self-authentication for preventing hardware trojan insertion. In *Hardware-Oriented Security and Trust*, pages 45–50. HOST, 2013.

Chapitre 5

Détection de CTMs en-ligne par duplication

« *The function of education is to teach one to think intensively and to think critically. Intelligence plus character - that is the goal of true education.* »

Martin Luther King, Jr.

Sommaire

5.1 Introduction

Nous avons vu que les méthodes de détection optique sont coûteuses et les résultats ne peuvent pas être appliqués au lot entier. Par ailleurs, la détection par test logique n'est pas efficace pour des CTMs dont le trigger est relié à un ensemble de signaux de grande taille. À l'inverse, les méthodes SCA sont efficaces pour la détection de CTMs larges mais sont sensibles aux variations de process et nécessitent un modèle de référence.

C'est pourquoi des méthodes complémentaires ont été proposées dans la littérature. Ces méthodes consistent à surveiller certains paramètres ou dupliquer le circuit pour détecter une anomalie en temps réel, par exemple l'effet d'un CTM sur les sorties primaires.
Une de ces techniques est le Design-for-Reliability (DfR).
Le DfR apporte plusieurs solutions orientées redondance pour les systèmes tolérants aux fautes : la redondance d'information, la redondance temporelle ou la redondance matérielle. Ces approches apportent la robustesse, puisque chaque traitement de données est vérifié grâce à une information supplémentaire, respectivement, les codes exécutés à partir de données traitées par le système, les données calculées à partir d'une seconde exécution différée par le même matériel, ou les données obtenues à partir d'exécutions supplémentaires par le matériel dupliqué.
La redondance matérielle repose sur la réplique de deux éléments physiques opérant en parallèle. Des entrées identiques sont appliquées à chaque réplique et des sorties identiques sont attendues. Ces sorties sont comparées en temps réel, et toute différence déclenche une alarme ou un processus de recouvrement.
Ce type de solution a été largement étudié dans le but d'augmenter la fiabilité des systèmes numériques [RH75; HCTS81; JW93; Pra96; MSM99; ZSM99; PR04].

Dans ce chapitre nous proposons une technique de détection de CTMs en-ligne basée sur une approche DfR.
Cette technique consiste à implanter le circuit original et sa réplique, fonctionnellement équivalente mais structurellement différente, et comparer leurs sorties primaires.
Les deux répliques reçoivent les mêmes données en entrée. Si leurs sorties différent, cela révèle un dysfonctionnement et éventuellement la présence d'un CTM.
Nous faisons l'hypothèse que deux CTMs identiques ne peuvent pas être insérés dans les deux répliques. Pour que cette hypothèse soit valide, il est donc nécessaire d'implanter deux répliques structurellement différentes.
Nous aborderons les moyens permettant de générer ces répliques et la façon de rendre cette technique robuste aux attaques.

5.2 Étude bibliographique

Pour palier les limites des méthodes de détection de CTM après fabrication, des méthodes de détection en temps réel ont été proposées dans la littérature.

La méthode proposée dans [AB09] consiste à ajouter de la logique re-configurable sécurisée, appelée Design-For-Enabling-Security (DEFENSE). Cette infrastructure est composée d'éléments distribués qui peuvent être reconfigurés dynamiquement pour vérifier la sécurité en-ligne et détecter un comportement anormal ou inattendu. La re-configuration per-

met d'implémenter un grand nombre de moniteurs de sécurité en faisant du multiplexage temporel. Des concepts similaires sont utilisés dans la fabrication de semi-conducteur pour la validation et le debug.

Dans [BNSZ09], les auteurs proposent une méthode appelée Secure Heartbeat And Dual-Encryption (SHADE). Elle permet de détecter et prévenir l'insertion de CTM dans un circuit. Deux couches d'encryption matérielle sont combinées pour apporter un environnement d'exécution sécurisé. Le système SHADE est complémentaire des méthodes de détection en phase de test et rend le circuit plus fiable. SHADE est composé d'une architecture composée de deux éléments, un contrôleur interne et externe, chacun vérifie le bon fonctionnement de l'autre.

Dans [MWP+09], les auteurs explorent une approche permettant de détecter les CTMs en-ligne en combinant le matériel avec un logiciel. Cette approche implique l'implémentation et l'exécution de différentes fonctionnalités équivalentes (obtenues avec des compilations différentes ou des algorithmes différents) en parallèle en comparant leurs résultats.

Dans [NGB+14; CDD+15], les auteurs proposent une technique qui permet de détecter les CTMs en-ligne mais aussi d'en prévenir l'insertion. La méthode consiste à encoder la partie séquentielle (les registres) du circuit avec des codes linéaires. Ensuite la partie combinatoire est mélangée avec la partie séquentielle encodée pour masquer les fonctionnalités du circuit. L'utilisation d'un paramètre d (la distance duale du code linéaire) permet d'assurer qu'un CTM connecté à moins de d registres ne sera pas efficace. En choisissant d assez grand, cela rend l'insertion de CTM difficile. Cette approche génère un surplus de surface entre 500% et 600%. Du fait du codage et du décodage des blocs dans le chemin de données, cette technique impacte aussi les performances du circuit.

Dans [NDG+15], les auteurs proposent une approche pour la vérification en-ligne. Le but est d'ajouter dans le circuit des modules de sécurité appelés Hardware Property Checker (HPC). Ils permettent de surveiller certaines propriétés du circuit en permanence. Si une de ces propriétés est violée, par exemple un accès non autorisé à une zone mémoire, une alarme est déclenchée. Des assertions Property Specification Language (PSL) sont utilisées pour implémenter ces module dans le code HDL du circuit.

5.3 Approche

Dans ce chapitre, nous proposons d'utiliser le concept de redondance matérielle dans le but détecter un CTM en-ligne.
Comme dit précédemment, notre approche repose sur la duplication du circuit et la comparaison des sorties. La Figure 5.1 illustre le concept de base. Dans le cas où un CTM a été inséré dans l'un des deux circuits, et que le CTM est activé durant le fonctionnement, les sorties diffèrent. Dans un tel cas, la sortie du comparateur peut être utilisée pour déclencher une alarme par exemple.
Si le circuit original C1 et sa réplique sont identiques, un attaquant pourrait être capable d'insérer le CTM dans les deux circuits, le rendant indétectable par la solution proposée.
Dans le but de prévenir un tel scénario, nous proposons une solution qui rend plus difficile voire impossible l'insertion de CTM dans les deux répliques.

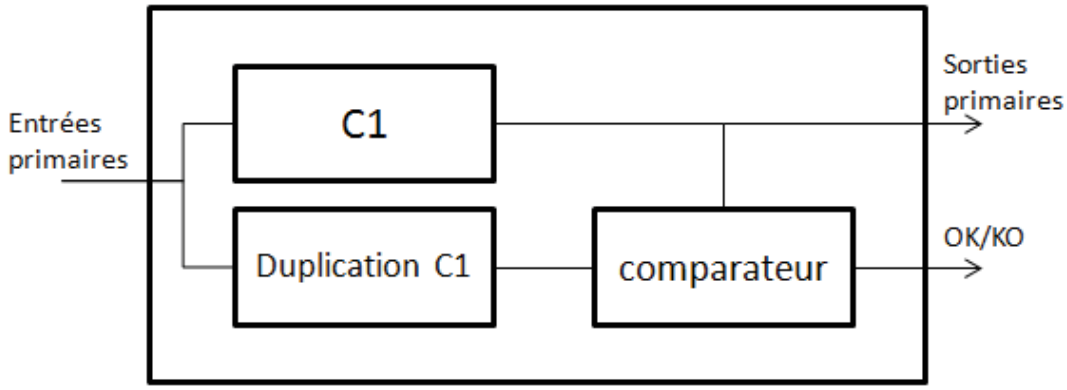


FIGURE 5.1 – Architecture générale.

Le principe de base de cette solution est axé sur l'implémentation de deux netlists différentes pour le circuit original et pour sa réplique fonctionnellement équivalente.
Des netlists alternatives d'une même fonction peuvent être obtenues de différentes manières :

- Utiliser un codage différent pour les états de la FSM,
- Utiliser un nombre de FSM différent pour chaque réplique,
- Utiliser des contraintes de synthèse différentes pour chaque réplique,
- Restreindre l'utilisation de certaines cellules de la librairie,
- Utiliser des outils de synthèse différents pour chaque réplique.

5.3.1 Génération de netlists différentes

Nous avons vu qu'il est possible de générer des netlists alternatives en codant différemment la FSM ou bien en utilisant un flot de synthèse différent.
Dans cette partie nous allons illustrer ces deux points.

5.3.1.1 Génération de netlists différentes par modification du flot de synthèse

Dans le flot de synthèse, on peut agir sur plusieurs paramètres pour générer des netlists différentes pour une fonction donnée.
En effet, nous pouvons agir sur les contraintes données à l'outil de synthèse. Ces contraintes sont la fréquence de fonctionnement du circuit, la consommation, etc. Ceci va pousser l'outil à ajouter des cellules dans certains chemins ou exclure l'utilisation de certaines cellules pour respecter les contraintes de temps ou de consommation.
Il est aussi possible d'agir sur la librairie de la technologie ciblée en définissant certaines restrictions (par exemple exclure les portes AND à 2 entrées). Ces portes vont être substituées par des portes plus grandes, entraînant une modification structurelle de la netlist.
Le dernier paramètre est l'outil de synthèse. Il existe plusieurs outils commerciaux permettant de réaliser l'opération de synthèse. Donc il est possible de synthétiser les deux répliques avec des outils différents.
La figure 5.2 donne un exemple de deux circuits fonctionnellement équivalents mais structurellement différents.

(a) Circuit original C1 (b) Duplication de C1

FIGURE 5.2 – Circuit original et sa duplication.

5.3.1.2 Génération de netlists différentes par modification de la FSM

Un autre moyen d'obtenir des netlists très différentes pour les deux répliques est d'implanter les FSMs en utilisant des codes différents.
Par exemple, considérons la FSM à 4 états suivante :

état1 → état2 → état3 → état4.

Si la FSM est codée en binaire naturel, nous obtiendrons :

état1 = 00, état2 = 01, état3 = 10 et état4 = 11.

Par contre, si nous codons cette même FSM avec un codage one-hot, nous obtiendrons :

état1 = 0001, état2 = 0010, état3 = 0100 et état4 = 1000.

Pour ce type de code, nous avons autant de FFs que d'états à savoir 4 pour cet exemple.

Les figures 5.3 et 5.4 illustrent l'implantation de cette FSM avec ces deux codes.
Nous remarquons que le nombre de portes logiques est plus important pour le codage one-hot. Ceci permet d'obtenir deux descriptions logiques différentes pour la même fonction.

De plus, si les cellules des deux répliques sont mélangées durant le placement et le routage, il est quasi impossible pour un attaquant d'identifier, à partir du layout, les signaux de chaque netlist pour insérer deux CTMs identiques dans les deux circuits.

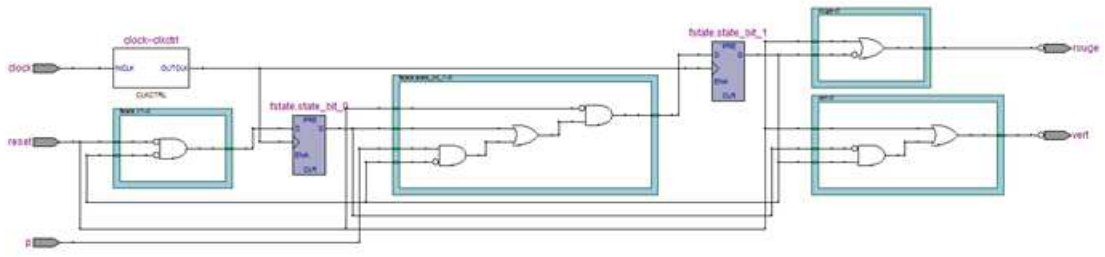


FIGURE 5.3 – Encodage binaire d'une FSM 4-états.

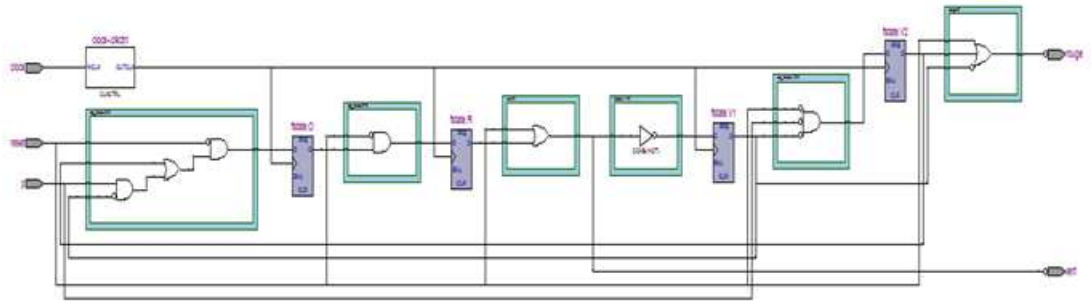


FIGURE 5.4 – Encodage one-hot d'une FSM 4-états.

5.4 Analyse d'attaque

Nous supposons qu'il y a 3 façons d'insérer un CTM, dont 2 permettent de créer un CTM "discret" dans un circuit où deux netlists différentes sont utilisées pour réaliser la même fonction :

- Si le CTM est inséré dans un des deux circuits, il va être détecté par le comparateur ;
- Puisque le layout a été produit en utilisant deux implémentations différentes pour la même fonction, il est très improbable que le même CTM puisse être inséré dans les deux versions. Par exemple, si le CTM est attaché à la connexion A de la netlist 1, il sera difficile de trouver la connexion équivalente à A dans la netlist 2 si elle existe ;
- Enfin, un attaquant peut insérer un CTM dans l'un des circuits et modifier la sortie du comparateur dans le but qu'il donne toujours OK.

Si le comparateur de sorties des deux répliques est corrompu, cette solution devient inefficace. Donc il faut trouver une stratégie pour le rendre robuste face une telle attaque.

5.5 Durcissement du système de monitoring

La détection de CTMs, dépend fortement de la sortie du comparateur et son module peut devenir le point faible de toute la solution. Pour cette raison, le bon fonctionnement du comparateur doit pouvoir être testé quitte à ajouter de la logique additionnelle.

Pour ce faire, une ou plusieurs entrées supplémentaires sont ajoutées et les duplicata sont modifiés en utilisant ces entrées. Le comparateur est testé hors ligne.
Cette modification est telle que :

- Pour certaines valeurs de ces entrées, la fonction de sortie des duplicata est identique. En fonctionnement normal, une de ces valeurs est appliquée aux entrées.
- Pour les autres valeurs, les fonctions des deux duplicata diffèrent pour certaines valeurs des autres entrées. Le comparateur doit alors répondre OK ou KO selon la valeur appliquée.

Ce principe est illustré à l'aide de l'exemple de la figure 5.5. L'entrée k a été ajoutée au duplicata.
En fonctionnement normal, k est mis à 0. Les sorties S_1 et S_2 sont identiques et égales à :

$$S = S_1 = S_2 = abc + abd + ef$$

Pour tester hors ligne le comparateur k est mis à 1.
S_1 est donc égale à :

$S_1 = abc + abd + ef$

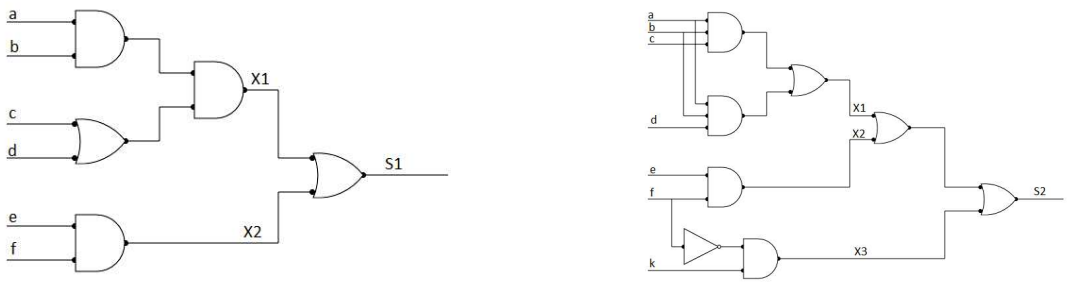
S_2 est égale à :

$S_2 = abc + abd + ef + \bar{f} = abc + abd + e + \bar{f}$

- Pour le vecteur (a,b,c,d,e,f) = (0,0,0,0,1,0) par exemple, $S_1 = 0$ et $S_2 = 1$. Le comparateur doit répondre KO.
- Pour le vecteur (a,b,c,d,e,f) = (1,1,1,1,0,1) par exemple, $S_1 = S_2 = 1$. Le comparateur doit répondre OK.

Remarques :

- Plusieurs entrées supplémentaires ($k_1, k_2, .., k_n$) peuvent être utilisées.
- La modification "idéale" consiste à ce que S_1 et S_2 soit distinctes pour la moitié des vecteurs d'entrée lorsque la valeur des k_i est différente de la valeur utilisée pour le fonctionnement normal.
- Ces modifications de la logique rendent encore plus difficile l'insertion d'un même CTM dans les deux versions.



(a) Circuit original C1. (b) Duplication de C1.

FIGURE 5.5 – Logique ajoutée pour tester le comparateur.

5.6 Résultats

L'approche présentée a été évaluée sur un circuit implémentant l'algorithme RSA (Rivest Shamir Adleman (RSA)). Nous avons utilisé la librairie ST65nm et les outils de Synopsys pour la synthèse, le placement et le routage.

L'algorithme RSA [RSA78] est largement utilisé dans les crypto-systèmes. Nous avons implémenté l'algorithme en VHDL (VHSIC Hardware Description Language (VHDL)).
Pour obtenir deux duplicata différents, nous avons décrits en VHDL l'algorithme RSA de deux façons distinctes. La version 1 utilise une machine à états monolitique alors que celle-ci est partitionnée en deux machines dans la version 2.
Les résultats sont présentés dans le Tableau 5.1
Les résultats de synthèse pour les deux versions sont donnés dans les deux premières colonnes. Nous constatons que la netlist de la version 2 contient plus de cellules combinatoires

TABLEAU 5.1 – Synthèse des résultats

	Vers1.	**Vers2.**	**Flot 1 :synthèse séparée**	**Flot 2 :synthèse combinée**
# comb. cells	1556	1605	3214	3137
# Sequ. cells	501	506	1008	1013
Comb. cell area (μm^2)	5833.8	5979.4	11990.1	11589.2
Non-comb. cell area (μm^2)	4691.9	4743.9	9446.3	9403.1
Total cell area (μm^2)	10525.8	10723.4	21436.4	20992.3
Total area after P&R (μm^2)	18110	18250	33782	33014

et séquentielles. Ceci montre une différence structurelle entre les deux netlists.
Le circuit entier a été synthétisé en utilisant deux flots.

- Dans le premier flot, les deux duplicata et le comparateur ont été synthétisés séparément. La netlist globale a été ensuite placée et routée. Le layout obtenu est présenté sur la Figure 5.6. La colonne flot 1 du Tableau 5.1 détaille les résultats de surface obtenus.
- De façon à rendre le layout encore plus difficile à analyser par un attaquant, nous avons utilisé un deuxième flot. Les descriptions RTL des deux duplicata et du comparateur sont combinées en un seul module. Ce module est ensuite synthétisé et placé/routé. Le layout est donné à la Figure 5.7. Nous observons que le layout généré à partir de ce flot est plus congestionné que celui du premier flot et occupe moins de surface (colonne flot 2 du Tableau 5.1). Ainsi, l'insertion d'un CTM dans un tel layout, où il est impossible de distinguer géographiquement les deux duplicata, devient encore plus difficile.

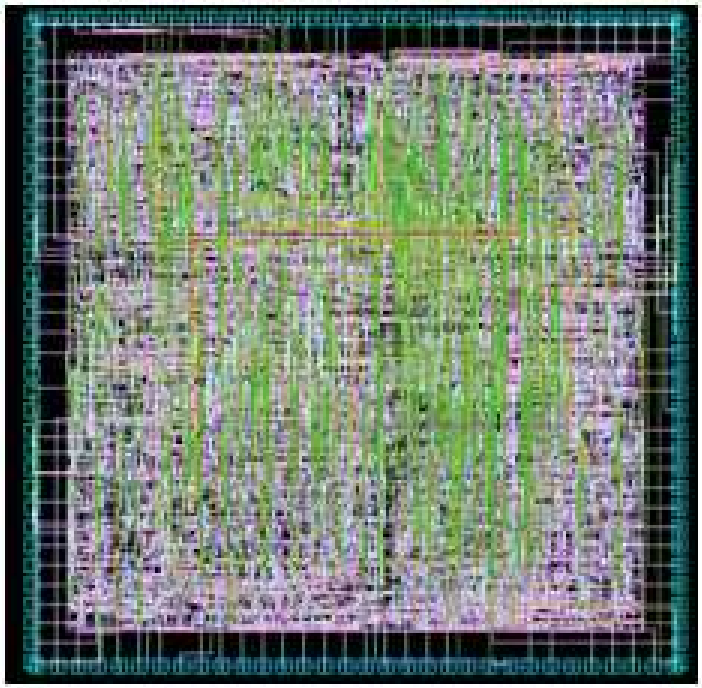


FIGURE 5.6 – Layout Flot 1.

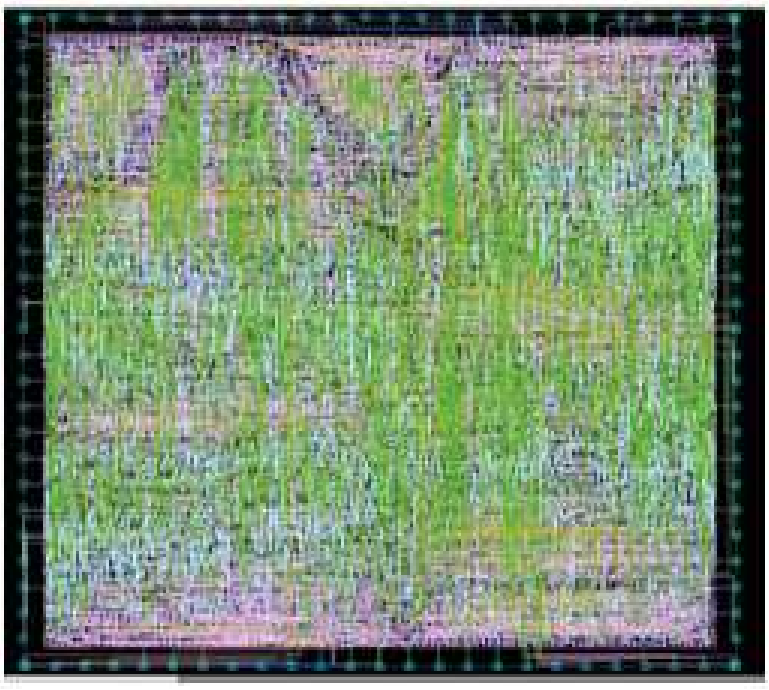


FIGURE 5.7 – Layout Flot 2.

5.7 Conclusion

La technique de détection en-ligne de CTMs proposée repose sur la duplication de la fonctionnalité des circuits. Cette solution de détection en-ligne concerne les insertions de CTM dans les fonderies. Le surplus en surface est d'un peu plus de 100% mais n'a aucun impact sur les performances du circuit en termes de délai.
De plus, elle permet la détection de fautes matérielles durant la durée de vie du circuit (fautes dues au vieillissement, impact de particules, etc...). Elle est complémentaire des méthodes de détection qui sont réalisées après la fabrication comme le test logique ou le SCA. Elle permet aussi de prévenir l'insertion de CTM en phase de fabrication avec un layout qui est difficile à exploiter pour un éventuel attaquant.

5.8 Références

[AB09] Miron Abramovici and Paul Bradley. Integrated circuit security : new threats and solutions. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research : Cyber Security and Information Intelligence Challenges and Strategies*, page 55. ACM, 2009.

[BNSZ09] Gedare Bloom, Bhagirath Narahari, Rahul Simha, and Joseph Zambreno. Providing secure execution environments with a last line of defense against trojan circuit attacks. *computers & security*, 28(7) :660–669, 2009.

[CDD+15] Claude Carlet, Abderrahman Daif, Jean-Luc Danger, Sylvain Guilley, Zakaria Najm, Xuan Thuy Ngo, Tliibault Porteboeuf, and Cédric Tavernier. Optimized linear complementary codes implementation for hardware trojan prevention. In *European Conference on Circuit Theory and Design*, pages 1–4. ECCTD, 2015.

[HCTS81] MY Hsiao, William C. Carter, James W. Thomas, and William R. Stringfellow. Reliability, availability, and serviceability of ibm computer systems : A quarter century of progress. *IBM Journal of Research and Development*, 25(5) :453–468, 1981.

[JW93] Niraj K Jha and S-J Wang. Design and synthesis of self-checking vlsi circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(6) :878–887, 1993.

[MSM99] Subhasish Mitra, Nirmal R Saxena, and Edward J McCluskey. A design diversity metric and reliability analysis for redundant systems. In *International Test Conference*, pages 662–671. IEEE, 1999.

[MWP+09] D McIntyre, F Wolff, C Papachristou, Swarup Bhunia, and D Weyer. Dynamic evaluation of hardware trust. In *Hardware-Oriented Security and Trust*. HOST, 2009.

[NDG+15] Xuan Thuy Ngo, Jean-Luc Danger, Sylvain Guilley, Zakaria Najm, and Olivier Emery. Hardware property checker for run-time hardware trojan detection. In *European Conference on Circuit Theory and Design*, pages 1–4. ECCTD, 2015.

[NGB+14] Xuan Thuy Ngo, Sylvain Guilley, Shivam Bhasin, Jean-Luc Danger, and Zakaria 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*, page 7. ACM, 2014.

[PR04] Irith Pomeranz and Sudhakar M Reddy. Concurrent on-line testing of identical circuits through output comparison using non-identical input vectors. In *19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 469–476. DFT, 2004.

[Pra96] Dhiraj K Pradhan. *Fault-tolerant computer system design*. Prentice-Hall, Inc., 1996.

[RH75] CV Ramamoorthy and Yih-Wu Han. Reliability analysis of systems with concurrent error detection. *IEEE Transactions on Computers*, 100(9) :868–878, 1975.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) :120–126, 1978.

[ZSM99] Chaohuang Zeng, Nirmal Saxena, and Edward J McCluskey. Finite state machine synthesis with concurrent error detection. In *International Test Conference*, pages 672–679. IEEE, 1999.

Chapitre 6

Prévention d'insertion de CTM par remplissage du layout

« *There is nothing like returning to a place that remains unchanged to find the ways in which you yourself have altered.* »

Nelson Mandela

Sommaire

6.1 Introduction

La dernière étape du flot de conception des CIs est la création du layout (description géométrique du fichier). Cette étape consiste à placer les cellules standards et les interconnecter (routage). Après placement des cellules standards, des espaces vides sont laissés entre elles pour éviter une congestion et des violations lors du routage. Ainsi, pour éviter une discontinuité au niveau des rails d'alimentation et avoir une densité uniforme du layout [Ich06], des cellules de remplissage appelées "filler cells" sont insérées dans ces espaces vides. Il faut noter que ces cellules ne remplissent aucune fonction logique. Leur remplacement ne pourrait pas être détecté lors du test et serait difficile à détecter avec une méthode optique [BDG+13]. Du point de vue de l'attaquant le remplacement des filler cells par des portes constituant un CTM semble être une bonne opportunité.

Dès lors il est primordial de développer des méthodes permettant de prévenir un tel scénario. Ainsi des techniques appelées DfHT ont été récemment dédiées à la prévention d'insertion de CTMs. Elles consistent à l'amélioration du design en ajoutant des fonctionnalités qui peuvent prévenir une possible altération du circuit original (insertion de CTM) pendant la fabrication [RSK14].

C'est pourquoi dans [XT13], les auteurs proposent une technique appelée BISA où des cellules combinatoires sont interconnectées et utilisées comme filler cells, de façon à créer un circuit combinatoire additionnel. En testant ce circuit additionnel (en plus du circuit original), il est possible de savoir si les filler cells ont été altérées. Le test de ce circuit additionnel est réalisé avec une architecture Built-In-Self-Test (BIST), où des Linear Feedback Shift Register (LFSR) et des Minimal Input Shift Register (MISR) sont implémentés dans les espaces dédiés aux filler cells. Cependant, cette méthode peut générer un surcoût en surface dans le cas où il n'y a pas suffisamment d'espaces vides pour implémenter les LFSR et MISR. De plus, le comparateur en sortie du MISR peut lui-même faire l'objet d'une attaque.

Dans ce chapitre, nous présentons une approche DfHT à coût nul en se basant sur le même principe que celui présenté dans [XT13]. Notre approche consiste à créer un layout aussi dense que possible, en hiérarchisant les espaces vides. Ces espaces vides sont ensuite remplis pour construire des fonctions combinatoires et des registres à décalage. Ainsi, toute modification intentionnelle du layout devient extrêmement difficile à réaliser par un éventuel attaquant (en comparaison à la méthode présentée dans [XT13], notre méthode permet de faire l'économie d'un TPG interne et d'un MISR).

6.2 Étude bibliographique

Étant donné les limitations des méthodes de détection, l'idée de modifier le flot de design en y ajoutant des fonctionnalités qui peuvent aider à détecter et/ou rendre difficile l'insertion de CTM est apparue. Les méthodes de DfHT proposées opèrent dans des niveaux d'abstraction différents, en fonction de l'objectif visé.

Dans [BH09], la contrôlabilité est améliorée grâce à l'inversion des potentiels électriques. L'inversion de tension sur une porte CMOS spécifique change en effet le comportement de la porte e.g. une porte NAND se comporte comme une porte AND, une porte NOR se comporte comme une porte OR. Par conséquent, l'inversion de tension sur une porte initialement peu

contrôlable à ’0’ rend la sortie de cette porte facilement contrôlable à ’0’. En utilisant les deux configurations de tension, cela permet de contrôler la sortie de la porte pour les deux valeurs. Cette approche affecte le placement et le routage puisque les portes avec la même profondeur logique doivent être connectées au même réseau d’alimentation, et les portes avec des niveaux différents doivent être connectées à un réseau d’alimentation différent.

La méthode présentée dans [DDY14], propose de modifier la librairie utilisée pour la synthèse du circuit. Les auteurs proposent une logique différentielle appelée differential cascade voltage switch logic (DCVSL). En mode normal le circuit synthétisé avec une DCVSL produit des valeurs complémentaires sur les sorties. L’insertion d’un CTM provoque une consommation anormale. Elle facilite la détection par analyse de consommation. Les auteurs obtiennent un taux de détection qui varie entre 66% et 98%.

Dans [IEGT13], les auteurs proposent de partitionner le processus de fabrication. Le principe consiste à diviser le GDSII en deux parties (FEOL et BEOL). Ces deux parties sont fabriquées séparément dans des fonderies différentes pour cacher les fonctionnalités du circuit et pour éviter l’insertion d’un CTM. Le FEOL contient les informations sur les niveaux de métallisation du circuit et le BEOL contient les informations sur les interconnexions. En cachant le FEOL au fabriquant du BEOL, il est impossible de déterminer les fonctionnalités du circuit et d’insérer un CTM.

Comme dit préablement, dans [XT13], les auteurs proposent de remplir les espaces vides avec des cellules fonctionnelles à la place des filler cells. Les ressources de routage limitées sont la cause du besoin d’élargissement du layout durant le placement et le routage pour apporter des lignes suffisantes pour résoudre la congestion de routage, ce qui résulte à la création d’espaces vides entre les cellules standard. Remplir ces espaces avec des cellules fonctionnelles et les connecter entre elles permet de densifier le layout et rendre l’insertion de CTM difficile voire impossible.

Ces méthodes ne sont pas coûteuses puisqu’elles ont un sur-coût en surface nul et sont basées sur la librairie ainsi que les outils de placement et routage standards. Bien que prometteuse en théorie, cette idée peut être limitée par les outils de placement et routage.

6.3 Approche

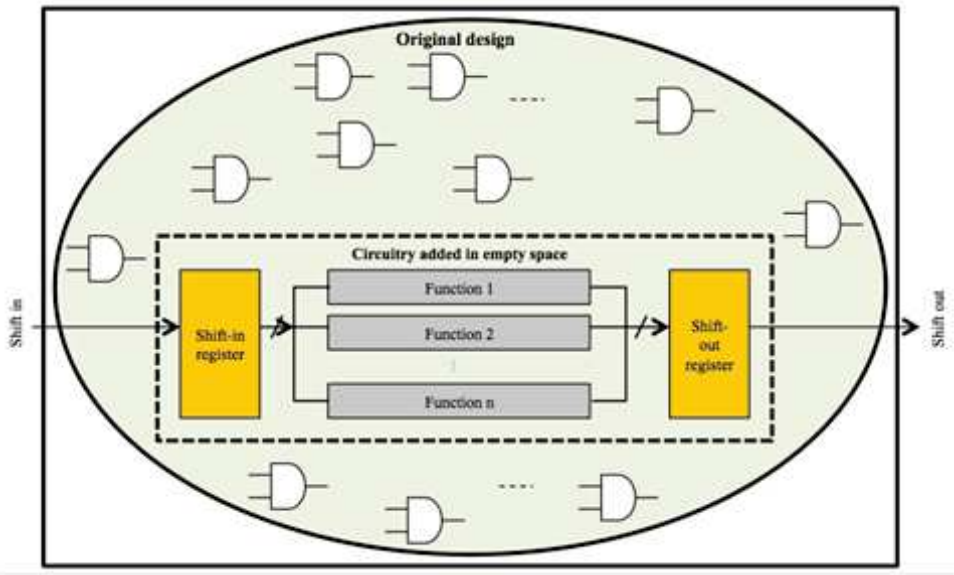
La figure 6.1a nous montre l’architecture de l’approche proposée au niveau layout. Le principe de cette méthode est de remplir les espaces vides et de créer des fonctions combinatoires. Ainsi un attaquant n’aura pas de place dans le circuit pour ajouter une fonction additionnelle tel qu’un CTM sans modifier la surface du circuit ou fortement modifier le placement et le routage. Les cellules standard additionnelles sont insérées et connectées ensemble pour former plusieurs fonctions combinatoires qui sont indépendantes du design original comme illustré sur la figure 6.1b.

Bien qu’il soit difficile pour un attaquant de différencier les fonctions additionnelles du circuit original, pour gérer le scénario pessimiste dans lequel l’attaquant réussit à remplacer les cellules correspondant à ces fonctions supplémentaires par un CTM, il est nécessaire de créer des fonctions testables. Pour tester ces fonctions, des registres à décalage sont utilisés pour appliquer des vecteurs d’entrée et d’analyser la réponse. Ces registres à décalage sont

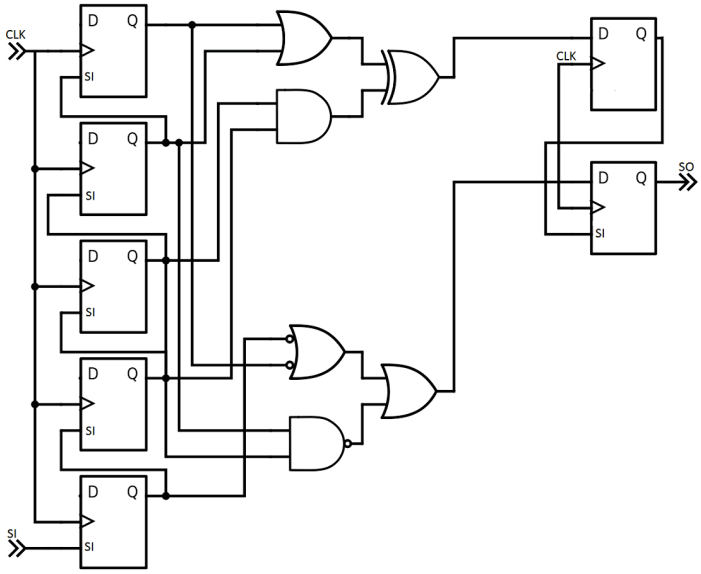
aussi implémentés dans les espaces vides.

Une fois le CI fabriqué, une phase de test est requise avant qu'il ne soit déployé. Outre le test conventionnel, assurant le bon fonctionnement de chaque CI, des vecteurs de test sont appliqués grâce aux registres à décalage, et les réponses de toutes les fonctions combinatoires ajoutées sont analysées sur la sortie shift-out.

Les registres à décalage sont pilotés par une horloge indépendante du circuit original. Cette horloge indépendante permet de désactiver les fonctions additionnelles quand le circuit est déployé, afin d'éviter une surconsommation de courant. Cette horloge n'étant utilisée que durant le test des fonctions ajoutées, elle peut être très lente. Ceci permet de relâcher toute contrainte temporelle lors du routage des fonctions additionnelles.



(a) Circuit original et circuit additionnel.



(b) Circuit additionnel.

FIGURE 6.1 – Architecture.

Le prochain paragraphe détaille le flot global, ainsi que les différentes étapes de la méthode.

6.3.1 Flot global

En partant d'un circuit placé (avec un taux d'occupation initial), le flot global de la procédure que nous proposons est le suivant :

- Calcul des espaces vides et des espaces vides critiques,
- Placement d'un nombre maximum de FFs qui peuvent entrer dans les espaces vides pour créer les registres à décalage,
- Placement des cellules combinatoires dans les espaces vides restants (prioritairement les espaces vides critiques),
- Interconnexion des cellules pour créer des fonctions logiques.
- Routage du circuit complet (le circuit original et les fonctions ajoutées).

Le but de cette procédure est d'obtenir le taux maximum d'occupation possible sans violation de routage. Le circuit est rempli au maximum avec des cellules additionnelles et si aucune violation ne survient durant le routage, on s'arrête. Sinon on décrémente le taux d'occupation jusqu'à ce que l'on n'ait aucune violation. Le flot global est illustré sur la figure 6.2.

6.3.2 Identification des espaces vides critiques

Comme mentionné plus haut, même avec une méthode de remplissage comme celle que nous proposons, atteindre un taux d'occupation de 100% n'est pas possible à cause des ressources de routage limitées. C'est pourquoi nous proposons de remplir en priorité ce que nous appelons les "espaces vides critiques".

Les espaces vides critiques sont ceux qui sont proches de signaux avec une marge temporelle grande. La raison est qu'un attaquant préférera insérer un CTM dans ce genre d'espace vide pour éviter d'utiliser des signaux qui se trouvent dans un chemin critique. Ceci va avoir un impact sur les chemins de délais du circuit, et sera détecté par les méthodes d'analyse de délai [DBF+15]. Dans notre flot, ces espaces vides critiques sont considérés comme prioritaires durant le remplissage.

6.3.3 Remplissage

Les FFs qui seront utilisées pour créer les registres à décalage sont insérées. Les FFs étant des cellules plus larges que les portes combinatoires, elles sont insérées en premier pour utiliser les plus grands espaces vides. Comme nous allons le voir dans les résultats expérimentaux, du fait de la large taille des FFs, le nombre de FFs inséré est souvent insuffisant par rapport au nombre des cellules combinatoires. Ainsi, dans cette étape, nous insérons autant de FFs que possible.

Ensuite, les espaces vides critiques restant sont remplis avec des cellules combinatoires, du plus grand au plus petit. Les espaces vides restants sont remplis de la même façon. Le choix des cellules combinatoires se fait en fonction de trois critères :

① Pour un espace vide donné, le choix est d'abord restreint à la plus grande cellule qu'on puisse y mettre (dans le but de limiter le nombre de cellules insérées).

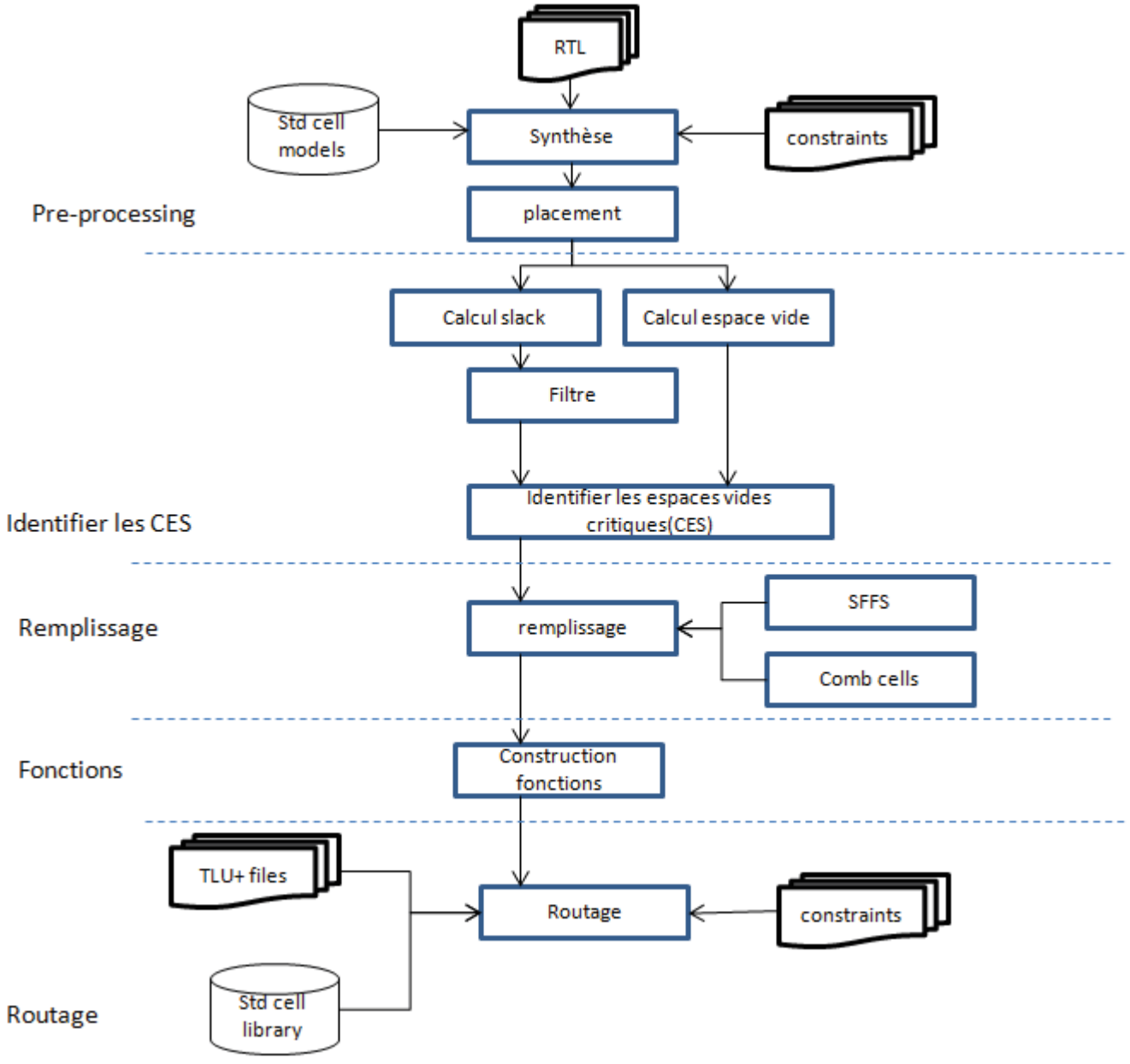


FIGURE 6.2 – Flot global.

② Ensuite, les cellules avec un grand nombre d'entrée sont prioritaires (car cela aide à réduire la profondeur logique des fonctions additionnelles comme décrit ci-dessous).

③ Enfin, les cellules avec une grande capacité de découplage sont prioritaires, parce que cela aide à compenser l'absence de filler cells ou de decap cells (DECAP [TV04]).

Le Tableau 6.1 illustre un exemple de choix de cellule pour le remplissage d'un espace vide.

TABLEAU 6.1 – sélection des cellules

Fonction	Largeur (μm)	Nombre d'entrée	Capacité de découplage (DECAP)
NAN3X38	4.6	3	0.67
AND4X25	4.6	4	0.44
AOI12X5	5	3	0.61
AOI21X35	5	3	0.61
OAI12X37	5	3	0.65
OAI21X37	5	3	0.65
OR4XX29	5	4	0.51

Par exemple, si nous considérons différentes cellules avec différents paramètres du Tableau 6.1, la porte OR4 est la plus convenable pour un espace vide de $5(\mu m)$. Il faut noter que les cellules avec un grand fan-out sont présentées dans le Tableau 6.1. On peut choisir de ne pas utiliser de telles cellules dans le but d'éviter des attaques de « resizing », ou de les utiliser (dans l'hypothèse qu'un attaquant ne sera pas capable de réaliser une telle attaque).

En outre, si un espace vide peut contenir seulement un inverseur, nous choisissons de le laisser vide : ce type d'espace ne sera pas utilisable pour attaquant. De plus, si nous choisissons d'insérer un inverseur et de l'utiliser dans la création des fonctions combinatoires, cela générera des contraintes de routage non nécessaires.

6.3.4 Construction des fonctions

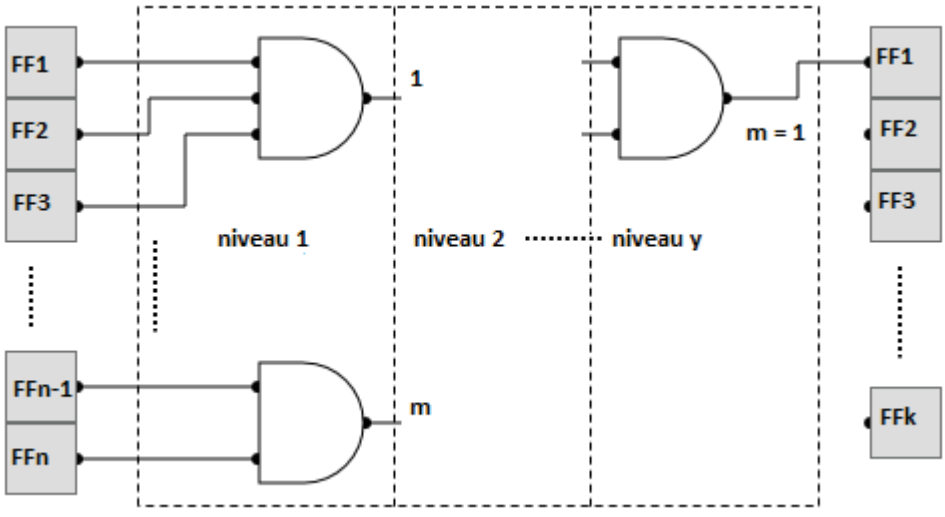


FIGURE 6.3 – Construction des fonctions.

Une fois ajoutées dans le layout, les cellules sont connectées de façon à obtenir une structure en arbre pour créer les fonctions, comme illustré sur la figure 6.3. La procédure est répétée jusqu'à avoir une fonction à une sortie. Dans le but de prévenir une congestion du routage, la première cellule de chaque fonction est choisie aussi proche que possible d'une FF et les autres cellules utilisées pour construire la fonction, sont choisies proches de la première.

Dans le but de prévenir le remplacement d'une fonction additionnelle par un CTM, une autre contrainte est prise en compte : ne pas créer de fonctions identiques. Ceci donnera à un attaquant l'opportunité de remplacer une des deux fonctions par un CTM (et connecter les deux sorties ensemble) sans que cela ne soit visible durant la phase de test.

6.3.4.1 Remarque

Comme mentionné auparavant, du fait de la taille occupée par les FFs, le nombre de FFs insérées peut être limité par rapport au nombre de fonctions combinatoires créées. Étant donné que le nombre d'entrée des fonctions est lié au nombre de FFs du registre à décalage d'entrée et le nombre fonctions au nombre de FFs du registre à décalage de sorties, l'objectif est de trouver la taille de fonction optimale qui peut produire le nombre optimal entre le nombre d'entrées et le nombre de sorties, résultant d'un nombre de FF réduit.

En pratique, nous répétons le flot global avec plusieurs nombres d'entrées, et déterminons en fonction du nombre de fonctions générées pour chaque itération, le nombre d'entrée optimal.

6.4 Résultats

Cette approche a été évaluée sur plusieurs circuits. Ces expériences ont été menées avec la librairie 65 nm de ST Microelectronics et les outils de synthèse, de placement et routage de Synopsys.

6.4.1 Analyse des résultats

Le Tableau 6.2 présente les résultats du remplissage du layout en termes de :

- Taux d'occupation initial et final (le taux le plus dense pour lequel il est possible de router sans violations),
- Consommation statique initiale et finale,
- Nombre de FF, nombre de fonctions (nombre d'entrées de chaque fonction) et le nombre d'espaces vides non remplis (doivent être remplacées par des filler cells) du fait du manque de FFs ou de violations de routage.

6.4.1.1 Consommation statique

La consommation statique est dégradée comme prévue, proportionnellement au nombre de cellules ajoutées : plus grand est le taux d'occupation initial, moindre est la détérioration. Rappelons que les fonctions supplémentaires sont connectées à une horloge indépendante, elles ne commutent pas durant le fonctionnement normal et de ce fait n'augmentent pas la consommation dynamique.

6.4.1.2 Taux d'occupation

Tout d'abord, il faut noter que des taux d'occupation élevés (au dessus de 90%) peuvent être atteints par notre méthode. Ensuite, il est intéressant de noter que le taux d'occupation final est généralement plus élevé avec un taux d'occupation initial qui n'est pas élevé. En effet plus grand est le taux d'occupation initial, plus grand est le risque de manquer de place pour loger les FFs nécessaires pour créer des fonctions supplémentaires.

Un layout du circuit s35932 est présenté dans la figure 6.4. Le circuit est placé avec un taux d'occupation initial de 75%. Les cellules en jaune sont celles ajoutées par notre outil avec un taux d'occupation final de 100% (i.e. tous les espaces utilisables ont été remplis).

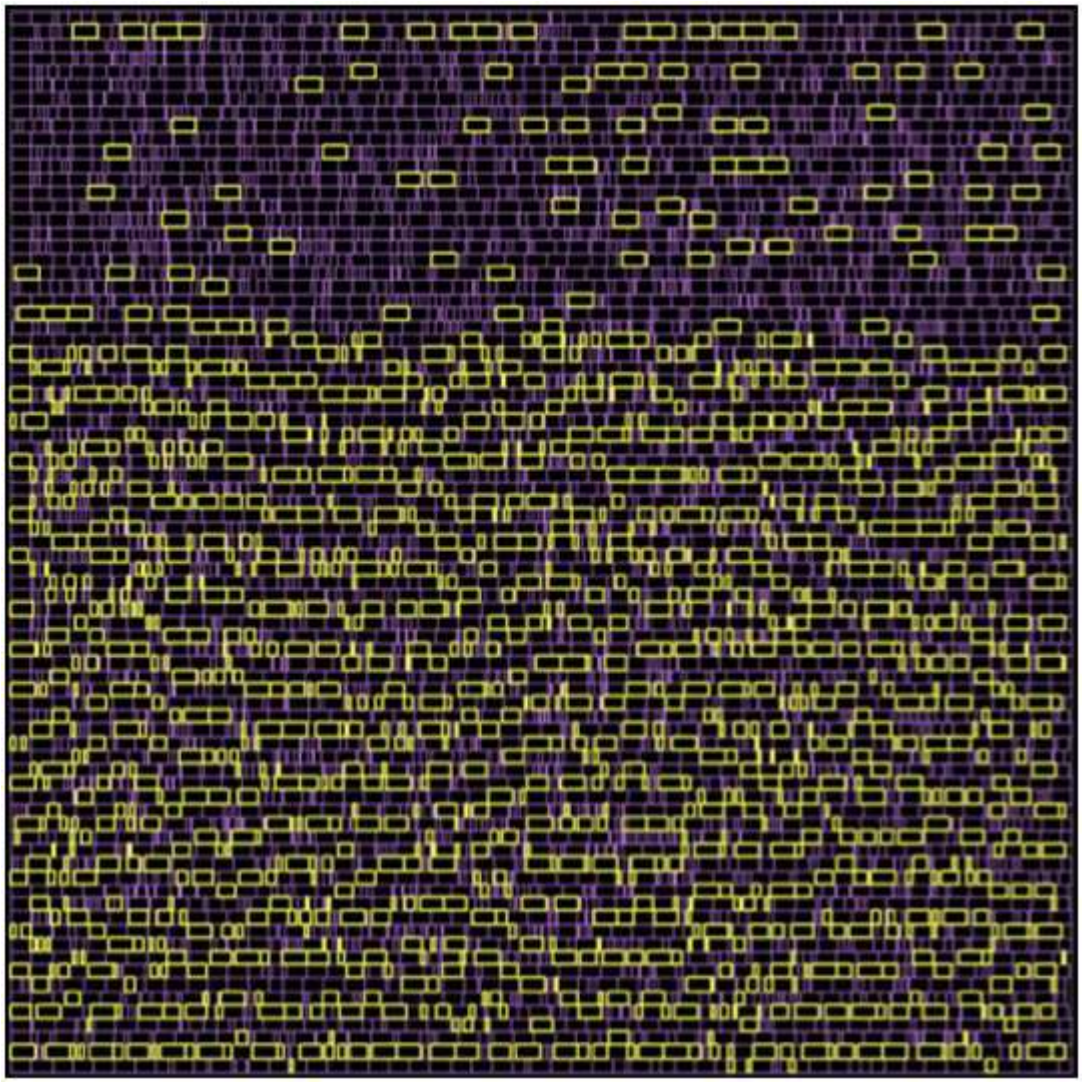


FIGURE 6.4 – Occupation du layout (cellules ajoutées en jaune).

Benchmark	Taux d'occupation initial	Consommation statique initiale (nW)	Taux d'occupation final	Consommation statique finale (nW)	Nb FFs	Nb entrées	espaces vides remplis (%)
AES	75	173.59	91	191.56 (+10%)	61	30 (31)	80
	80	173.95	90	185.13 (+6%)	27	12 (14)	73
	85	173.59	88	175.6 (+1%)	6	5 (1)	47
S13207	75	14.31	95	23.51 (+64%)	29	19 (10)	100
	80	13.42	90	18.41 (+37%)	13	5 (8)	63
	85	13.42	90	14.99 (+12%)	7	2 (5)	48
S35932	75	118.14	95	175.24 (+48%)	277	62 (26)	90
	80	117.91	91	160.8 (+36%)	231	11 (70)	63
	85	117.15	90	145.69 (+24%)	112	10 (34)	38
RSA	75	35.99	93	45.35 (+26%)	62	17 (23)	90
	80	35.92	93	43.26 (+20%)	53	46 (7)	87
	85	35.86	92	38.86 (+8%)	16	10 (6)	60
RS232	75	18.28	91	22.63 (+24%)	32	27 (5)	89
	80	18.29	91	20.95 (+15%)	19	15 (4)	82
	85	18.26	87	18.72 (+2%)	4	3 (1)	26
ARM4U	75	41.49	93	50.26 (+21%)	70	63 (6)	91
	80	41.48	91	46.98 (+13%)	36	16 (20)	75
	85	41.45	91	43.6 (+5%)	22	13 (8)	60

TABLEAU 6.2 – Remplissage du layout : procédure itérative

6.4.2 Discussions

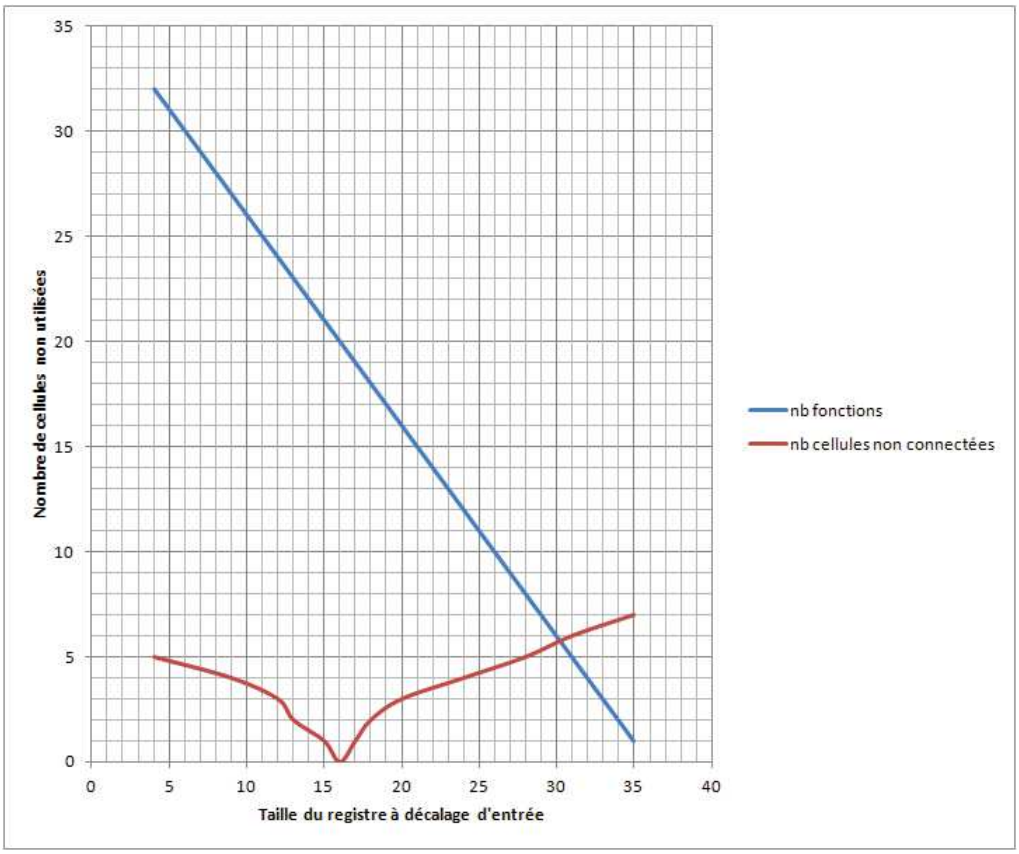


FIGURE 6.5 – Analyse du circuit ARM4U.

Comme mentionné plus tôt, le nombre de fonctions ajoutées est fonction de la taille du registre à décalage d'entrée. Ainsi, nous avons développé une procédure permettant de déterminer la taille optimale du registre d'entrée et obtenir un nombre minimal de cellules non connectées. Sur cet exemple, le circuit ciblé est le benchmark ARM4U, pour lequel le taux de remplissage initial a été fixé à 80%. Après placement, 36 FFs ont pu être insérées dans les espaces vides. Le registre d'entrée peut être composé de 35, 34,..., ou 4 FFs. Les espaces vides restants sont remplis avec des cellules combinatoires comme expliqué en 6.3.3 permettant d'obtenir un taux de remplissage de 91%. En appliquant de façon itérative la procédure de construction des fonctions, nous sommes en mesure de déterminer la taille optimale du registre d'entrée. Sur la figure 6.5, la solution laissant la moins d'espace inoccupé correspond à des fonctions à 16 entrées.

6.4.3 Limites

Dans le cas où le layout est trop dense pour insérer une FF, les fonctions combinatoires ne peuvent pas être connectées à des registres. Pour prendre en charge un tel scénario, nous avons rajouté une fonctionnalité qui consiste construire une unique fonction avec les cellules combinatoires placées dans les espaces vides et rajouter des entrées et des sorties primaires supplémentaires pour remplacer les registres à décalage.

Dans les résultats présentés dans le Tableau 6.3, nous arrivons à insérer des cellules combinatoires et créer une fonction combinatoire pour tous les circuits, jusqu'à même atteindre 100% de taux d'occupation pour deux circuits. Il faut noter que ce 100% ne signifie pas que le circuit a été entièrement rempli, mais que tous les espaces utilisables (suffisamment grands pour contenir une cellule) ont été remplis.

TABLEAU 6.3 – Remplissage du layout

Benchmark	Taux d'occupation initial	Taux d'occupation final	Nombre d'entrée
AES	93%	94%	19
S13207	99%	100%	8
S35932	99%	100%	30
RSA	94%	95%	9
RS232	95%	96%	9
ARM4U	96%	97%	10

6.5 Conclusion

Dans ce chapitre, nous avons présenté une méthode DfHT qui permet de créer un layout aussi dense que possible dans le but de prévenir une insertion de CTM par un attaquant dans une fonderie non fiable. Cette méthode consiste à remplir les espaces vides dans un layout par des cellules fonctionnelles à la place de filler cells. Ces fonctions additionnelles sont testables pour prévenir qu'un attaquant ne les remplace par un CTM. Une telle méthode peut générer de grosses contraintes pour le routage; nous avons expliqué comment minimiser ces contraintes additionnelles.

Les résultats expérimentaux montrent que des taux d'occupation très élevés peuvent être atteints, ce qui démontre la faisabilité de la méthode. Pour éviter une éventuelle dégradation du temps d'arrivée des données, un futur travail peut être mené pour créer un algorithme de routage ad-hoc, pour améliorer le routage du circuit global (circuit original et circuit additionnel). Si on arrive à améliorer les performances en termes de routage, on sera en mesure de monter en taux d'occupation et de rendre le circuit encore plus sûr.

Cette méthode permet de faciliter la détection par les méthodes de test logique ou de détection visuelle. En effet, en densifiant le layout il est très difficile de créer un CTM furtif, ou qui soit invisible dans le layout. Ceci va rendre la détection plus facile. De plus, cette méthode a l'avantage de présenter un coût nul.

6.6 Références

[BDG+13] Shivam Bhasin, Jean-Luc Danger, Sylvain Guilley, Xuan Thuy Ngo, and Laurent Sauvage. Hardware trojan horses in cryptographic ip cores. In *Workshop on Fault Diagnosis and Tolerance in Cryptography*, pages 15–29. FDTC, 2013.

[BH09] Mainak Banga and Michael S Hsiao. Vitamin : Voltage inversion technique to ascertain malicious insertions in ics. In *Hardware-Oriented Security and Trust*, pages 104–107. HOST, 2009.

[DBF+15] Sophie Dupuis, Papa-Sidy Ba, Marie-Lise Flottes, Giorgio Di Natale, and Bruno Rouzeyre. New testing procedure for finding insertion sites of stealthy hardware trojans. In *2015 Design, Automation & Test in Europe Conference Exhibition*, pages 776–781. DATE, 2015.

[DDY14] Wafi Danesh, Jaya Dofe, and Qiaoyan Yu. Efficient hardware trojan detection with differential cascade voltage switch logic. *VLSI Design*, 2014 :5, 2014.

[Ich06] Junji Ichimiya. Layout design method of semiconductor integrated circuit, and semiconductor integrated circuit, with high integration level of multiple level metalization, July 11 2006. US Patent 7,076,756.

[IEGT13] Frank Imeson, Ariq Emtenan, Siddharth Garg, and Mahesh Tripunitara. Securing computer hardware using 3d integrated circuit (ic) technology and split manufacturing for obfuscation. In *Presented as part of the 22nd USENIX Security Symposium*, pages 495–510. USENIX Security, 2013.

[RSK14] Jeyavijayan Rajendran, Ozgur Sinanoglu, and Ramesh Karri. Regaining trust in vlsi design : Design-for-trust techniques. *Proceedings of the IEEE*, 102(8) :1266–1282, 2014.

[TV04] Kris Tiri and Ingrid Verbauwhede. Place and route for secure standard cell design. In *Smart Card Research and Advanced Applications VI*, pages 143–158. Springer, 2004.

[XT13] Kan Xiao and Mohammed Tehranipoor. Bisa : Built-in self-authentication for preventing hardware trojan insertion. In *Hardware-Oriented Security and Trust*, pages 45–50. HOST, 2013.

Chapitre 7

Prévention d'insertion de CTM par insertion de points de test

« *If you talk to a man in a language he understands, that goes to his head. If you talk to him in his language, that goes to his heart.* »

Nelson Mandela

Sommaire

7.1 Introduction

Pour être furtif, un CTM doit être activé quand une condition très rare se produit. Une combinaison de signaux faiblement contrôlables permet d'avoir une telle condition. Ce type de condition est difficile à trouver avec des méthodes de détection basées sur le test logique (surtout quand le nombre de signaux utilisés pour le trigger est grand). Une façon d'augmenter l'efficacité de ces méthodes est de réduire le nombre de signaux faiblement contrôlables dans un circuit. Par ailleurs, cela rend la tâche de l'attaquant plus difficile.

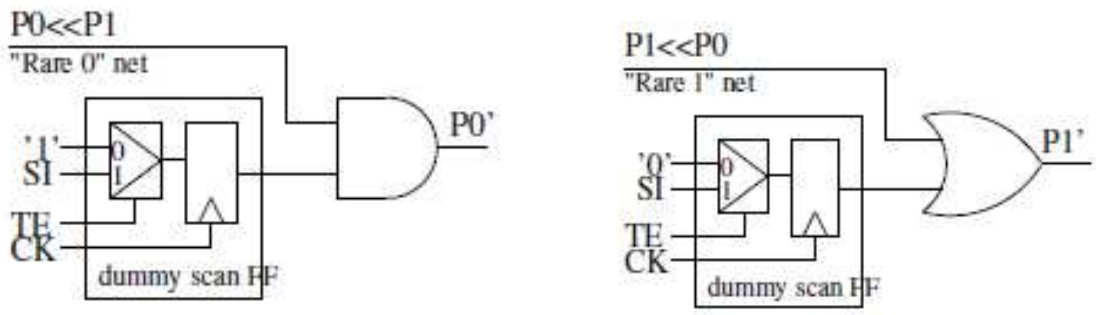
C'est ainsi que des méthodes permettant de faciliter la détection par test logique ont été proposées dans la littérature [STP12; XML+14; ZZTT14]. Elles consistent à ajouter des portes combinatoires ou/et des FFs pour améliorer la contrôlabilité des signaux. Ceci permet de limiter le choix d'un attaquant en termes de signaux pour le trigger, donc le forcer à créer des conditions moins rares et donc plus facilement détectables lors du test.

Cependant, le nombre de cellules ajoutées dans le circuit pour améliorer la contrôlabilité des signaux peut être une contrainte. En effet, dans le cas où le nombre de signaux ayant des probabilités déséquilibrées est important, cela nécessite beaucoup de cellules et génère un sur-coût en surface important. Les méthodes proposées jusqu'à présent font le compromis entre contrôlabilité et nombre de cellules insérées pour limiter le sur-coût en surface. Obtenir un taux de détection élevé avec un sur-coût en surface faible nécessite une sélection des points d'insertion optimisée.

C'est pourquoi dans ce chapitre, nous proposons une méthode qui permet d'avoir un circuit où tous les signaux sont contrôlables avec un nombre réduit de cellules additionnelles. Cette méthode consiste à insérer de façon intelligente des portes combinatoires et des FFs sur les signaux peu contrôlables (ayant des probabilités déséquilibrées). La sélection de tels signaux est réalisée de façon à impacter le plus grand nombre de signaux, ce qui réduit le nombre de portes à insérer. Nous détaillerons la méthodologie de sélection des signaux ainsi que la procédure d'insertion des cellules pour équilibrer ces signaux. Enfin, nous ferons une étude comparative entre les méthodes précédentes et notre approche.

7.2 Étude bibliographique

L'idée d'insérer des points de test pour faciliter la détection d'un CTM durant la phase de test a été introduite par [STP12]. Dans cette méthode, la probabilité de commutation des signaux est calculée. Cette probabilité est le résultat de la multiplication des probabilités d'être à '0' et à '1' des signaux (cf section 4.4.1) Les probabilités des signaux sont équilibrées grâce à l'ajout de portes (AND/OR) contrôlées par ce qu'on appelle des pseudo scan FF. Comme illustrée sur la figure 7.1, les portes AND permettent d'augmenter la probabilité du signal de valoir '0' et les portes OR la probabilité du signal de valoir '1'. Ces portes additionnelles sont contrôlées par des pseudo FFs alimentées par le port scan In en phase de test. En mode normal, la pseudo FF est collée à '0' ou '1' (en fonction du type de porte) pour ne pas modifier le fonctionnement du circuit. L'insertion de ces points de test se fait de façon itérative, en partant des signaux avec les probabilités les plus déséquilibrées. Tous les signaux des circuits ainsi modifiés ont des probabilités supérieures à un seuil défini par l'utilisateur.
Dans [XML+14], il est proposé d'insérer des portes de transmission au lieu de FFs pour alimenter les portes AND/OR additionnelles. Cependant, le contrôle indépendant de ces portes



(a) Point de test pour signal rare à 0.

(b) Point de test pour signal rare à 1.

FIGURE 7.1 – Structure du point de test.

de transmission n'est pas abordé. Une telle solution génère une augmentation de la contrôlabilité moindre par rapport à la méthode présentée dans [STP12].

Dans [ZZTT14], des pseudo FFs sont insérées couplées avec des MUXs au lieu de portes AND/OR. Cette solution permet d'avoir un moindre surcoût en surface, mais apporte une faible amélioration en termes de probabilités. Les points de test sont insérés en priorité sur les signaux avec une profondeur logique minimale. Ces points de test auront un impact significatif sur les signaux en aval. Ceci permet de réduire le nombre de points de test en comparaison à la méthode présentée dans [STP12].

Dans [DBDN+14], les auteurs proposent une méthode d'encryption logique permettant de prévenir l'insertion de CTM et de lutter contre la surproduction. Le but est d'encrypter un circuit avec une clé de façon à ce que la bonne fonctionnalité du circuit ne soit obtenue qu'en appliquant la bonne valeur de clé. Ceci permet donc de camoufler les probabilités réelles des signaux ainsi que la fonctionnalité réelle du circuit. Pour ce faire, des portes AND/OR sont insérées dans le circuit et une de leurs entrées est connectée à un bit de la clé. L'attaquant ne connaissant pas la clé, ne peut que supposer que ces bits d'entrée ont une probabilité de valoir 0 ou 1 égale à $\frac{1}{2}$. Par conséquent, tout calcul de probabilité sur les autres signaux donnera un résultat incorrect. Ainsi l'insertion d'un CTM potentiel pourra être plus facilement détectée. Cependant, le nombre de portes insérées est important par rapport au nombre de signaux impactés, d'où un sur-coût en surface élevé.

Ces méthodes présentent un sur-coût en surface élevé quand le nombre de signaux ayant des probabilités en dessous du seuil est important (seuil de probabilité faible). Pour réduire l'impact en surface chaque point d'insertion doit être choisi de façon à équilibrer un grand nombre de signaux. C'est pourquoi nous avons développé une méthodologie d'insertion de points de test optimisée qui permet d'obtenir un taux de détection en phase de test élevé pour un sur-coût en surface faible.

7.3 Approche proposée

L'approche que nous proposons consiste en l'insertion de portes combinatoires pilotées par des Scan FFs comme dans [STP12]. L'insertion se fait en priorité sur les signaux proches des entrées pour que l'impact se répercute sur les signaux en aval. Nous proposons une nou-

velle modification structurelle permettant d'avoir des probabilités plus équilibrées avec un sur-coût en surface moindre. Pour limiter le nombre de Scan FFs, nous avons choisi de fusionner certaines d'entre elles. Les différentes étapes de cette procédure sont détaillées dans les sections suivantes.

7.3.1 Sélection des points d'insertion et équilibrage de probabilités

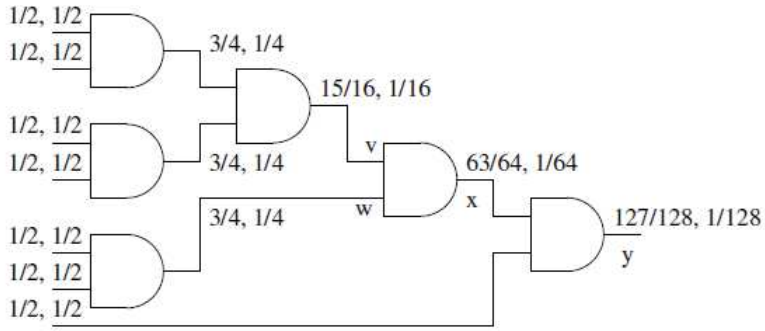


FIGURE 7.2 – Sélection de points d'insertion.

Tout d'abord les probabilités de tous les signaux sont calculées en utilisant la méthode décrite dans 4.4.1. En fonction du seuil de probabilité défini, les signaux à équilibrer sont déterminés.
Pour illustrer cela, considérons le circuit donné en figure 7.2 et un seuil de probabilité T = $\frac{1}{32}$, en dessous duquel tout signal S tel que P(S = 0) ≤ T ou P(S = 1) ≤ T sera considéré comme un signal déséquilibré. Par conséquent, un point de test sera inséré sur ce signal. Dans l'exemple ci-dessus, les signaux x et y sont donc déséquilibrés :

$$P(x = 1) < T \quad et \quad P(y = 1) < T$$

Le signal x étant plus proche des entrées primaires, l'insertion se fera sur ce signal en priorité. Pour améliorer les probabilités de x, deux solutions existent. La première solution consiste à insérer une porte sur un des signaux en amont de x à savoir sur v ou w. La deuxième solution consiste à insérer une porte sur le signal déséquilibré x.
Dans la suite nous analysons ces deux solutions et les comparons en termes d'équilibrage de probabilités et de nombre de signaux impactés.

7.3.1.1 Insertion de point de test sur un signal en amont

Pour illustrer ce choix, considérons l'exemple donné sur la figure 7.3. Le signal en amont de x ayant les probabilités les plus déséquilibrées est le signal v. Sa probabilité de valoir '1' est plus faible, alors une porte OR est insérée. L'insertion de cette porte OR permet d'augmenter les probabilités de ce signal. Les nouvelles probabilités de v sont donc : $P(v' =' 0') = \frac{15}{32}$ et $P(v' =' 1') = \frac{17}{32}$. Ceci permet également d'augmenter les probabilités des signaux en aval de v que sont x et y. Les nouvelles probabilités de x sont : $P(x' =' 0') = \frac{111}{128}$ et $P(x' =' 1') = \frac{17}{128}$. Celles de y sont : $P(y' =' 0') = \frac{239}{256}$ et $P(y' =' 1') = \frac{17}{256}$. Ainsi la modification des probabilités du signal v a entraîné une modification des probabilités des signaux en aval (x et y). Cependant, les probabilités de valoir '0' ou '1' restent fortement déséquilibrées pour les signaux x et y.

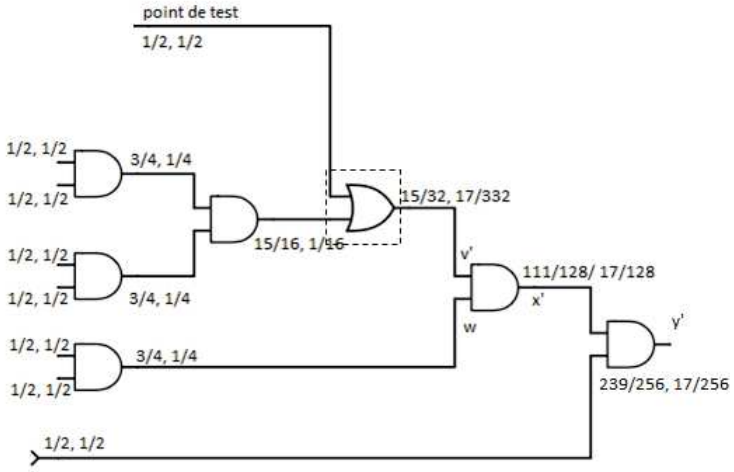


FIGURE 7.3 – Insertion de point de test sur un signal en amont de x.

Nous allons voir une deuxième solution, à savoir l'insertion d'un point de test sur le signal lui-même.

7.3.1.2 Insertion de point de test sur le signal

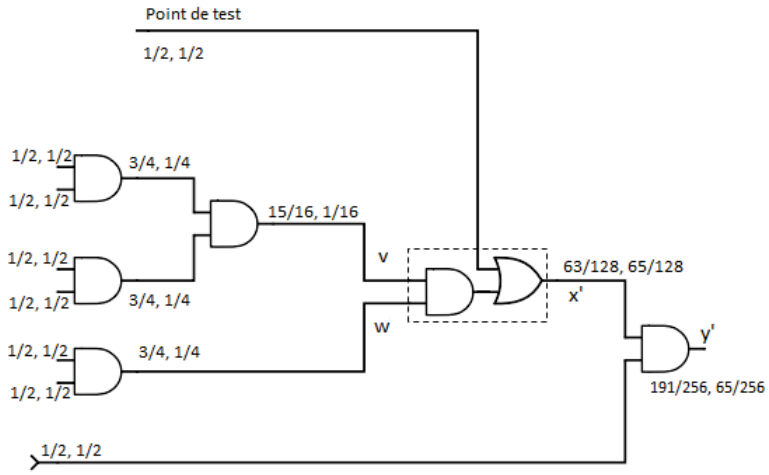


FIGURE 7.4 – Insertion de point de test sur le signal x.

L'objectif est d'équilibrer le signal x et le signal en aval, à savoir y. Sur l'exemple donné en figure 7.4, la porte AND à laquelle était connectée le signal x a été remplacée par une porte AND-OR à 3 entrées. Une porte OR est choisie parce que la probabilité de valoir '1' est plus faible. De manière générale, la porte à insérer doit intégrer la fonction de la porte à laquelle est connectée le signal à équilibrer et la fonction de la porte à insérer en fonction de la probabilité.
Après insertion de la porte ci-dessus, les nouvelles probabilités de x sont alors : $P(x' =' 0') =$

$\frac{63}{128}$ et $P(x' =' 1') = \frac{65}{128}$. Les probabilités du signal en aval à savoir y ont aussi été augmentées : $P(y' =' 0') = \frac{191}{256}$ et $P(y' =' 1') = \frac{65}{256}$. Les probabilités des signaux x et y sont plus équilibrées en comparaison à la première solution. Ainsi, l'insertion d'une porte à 3 entrées sur le signal déséquilibré montre deux choses :

① Des probabilités plus équilibrées.

② Un sur-coût en surface moindre en comparaison à l'ajout d'une porte OR en amont.

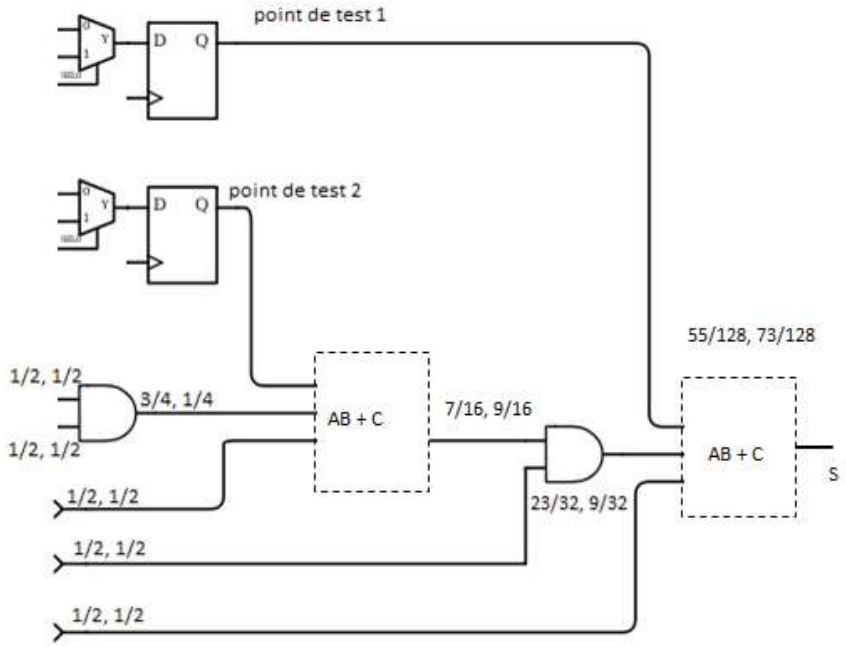
Par conséquent, nous avons choisi d'adopter la deuxième solution en priorité lorsqu'elle est possible, c'est à dire lorsqu'il existe une porte ayant la fonctionnalité nécessaire dans la bibliothèque.
Une fois que les portes combinatoires ont été insérées, il est nécessaire de les contrôler pour les forcer à leur valeur prioritaire lors du test, ceci dans le but de faciliter la détection de CTMs. C'est pourquoi, nous avons choisi de connecter une des entrées de chaque porte insérée à une Scan FF.

7.3.2 Ajout de Scan FFs

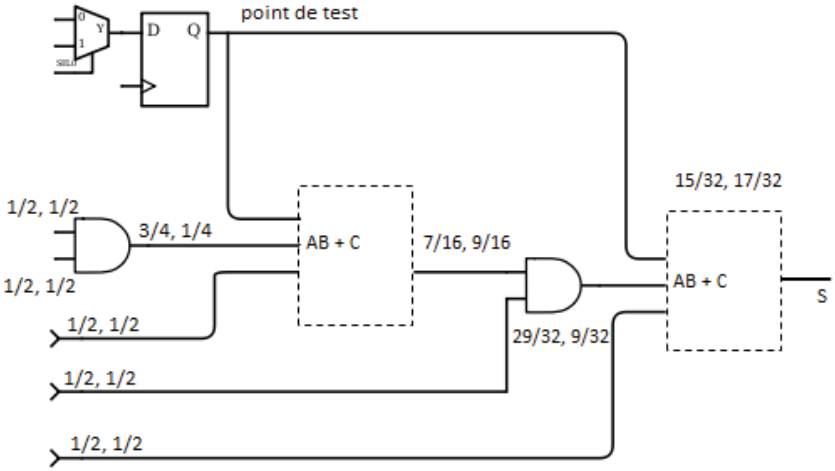
Les Scan FFs sont composées d'une FF et d'un multiplexeur à 2 entrées (voir figure 7.5a). Ce multiplexeur est piloté par le "Test Enable" permettant de choisir le mode choisir le mode test ou le mode normal. Il est donc possible en mode test de piloter les signaux déséquilibrés et de les forcer à leur valeur rare. L'ajout de Scan FFs pour contrôler les portes insérées ne dégradent pas les probabilités du signal équilibré.

7.3.3 Fusion des Scan FFs

Le sur-coût en surface peut être réduit en fusionnant certaines Scan FFs qui contrôlent les entrées des portes insérées. Cette fusion doit se faire sans détériorer les probabilités des signaux connectés à la porte contrôlée par la Scan FF. L'exemple donné sur la figure 7.5b montre une mauvaise fusion de Scan FFs. Les probabilités du signal S ont été dégradées après fusion. Avant fusion, les probabilités du signal S sont : $P(S = 0) = \frac{55}{128}$ et $P(S = 1) = \frac{73}{128}$ Après fusion, les probabilités du signal S deviennent : $P(S = 0) = \frac{15}{32}$ et $P(S = 1) = \frac{17}{32}$, soit $P(S = 0) = \frac{60}{128}$ et $P(S = 1) = \frac{68}{128}$. Nous remarquons une baisse de la probabilité de valoir 1 pour le signal S.



(a) Probabilités avant fusion.



(b) Probabilités après fusion.

FIGURE 7.5 – Exemple de fusion de Scan FF (portes ajoutées en pointillés).

7.4 Flot global

L'insertion d'un point de test est réalisée prioritairement sur les signaux qui sont proches des ports d'entrées. L'insertion d'un point de test sur ces signaux augmente également les probabilités de tous les signaux en aval. Pour ce faire, plusieurs phases d'insertion sont réalisées pour chaque niveau logique à partir des entrées primaires jusqu'aux sorties primaires. Les nouvelles probabilités sont calculées entre chaque phase. Une fois que tous les signaux dont les probabilités sont inférieures au seuil défini sont équilibrés, les Scan FFs sont ajoutées. Pour finir, la phase de fusion est réalisée.

7.4.1 Remarques

Nous avons choisi la solution qui consiste à insérer le point de test sur le signal déséquilibré. Cependant, l'insertion d'un point de test sur un signal en amont de ce signal peut être nécessaire si une porte à 3 entrées n'est pas disponible (quand la porte requise n'existe pas dans la librairie). Par ailleurs, un nombre maximum de portes insérées peut être défini par l'utilisateur, ceci dans le but de limiter encore plus le sur-coût en surface. Il faut également noter que notre procédure ne dégrade pas le délai des chemins critiques contrairement aux méthodes présentées dans [STP12; XML+14]. Les signaux ayant une marge temporelle faible ne sont pas considérés pour l'équilibrage de probabilités.

7.5 Résultats

Tout d'abord, nous avons évalué notre méthode en termes d'amélioration de probabilités. Ensuite, nous avons évalué l'apport de cette méthode pour la facilitation de la détection de CTM par test logique. Enfin nous avons fait une étude comparative entre notre méthode et les méthodes précédentes.

7.5.1 Équilibrage des probabilités

Nous avons évalué notre méthode sur plusieurs circuits ISCAS. Le Tableau 7.1 présente pour chaque circuit :

- Le seuil de probabilité choisi pour chaque expérience.
- le nombre de signaux ayant une probabilité en dessous du seuil.
- Le nombre de points de test ajoutés pour équilibrer tous les signaux dont les probabilités sont inférieures ou égales au seuil.
- L'augmentation moyenne des probabilités des signaux initialement déséquilibrés.

Il faut noter que pour tous les circuits, tous les signaux dont les probabilités initiales étaient inférieures ou égales au seuil ont été équilibrés. Nous observons également une augmentation moyenne de 0.38 en termes de probabilités pour les signaux initialement déséquilibrés. Ceci montre que tous les signaux sont désormais contrôlables car ayant tous des probabilités $\approx \frac{1}{2}$. De plus, le ratio nombre de signaux équilibrés sur nombre de points de test requis est grand. Par exemple, pour le circuit apex4, seuls 229 points de test ont permis d'équilibrer 683 signaux dont les probabilités initiales étaient inférieures au seuil (0.05). Ainsi nous obtenons pour ce circuit un ratio de ≈ 3 entre le nombre de points de test insérés et le nombre de signaux équilibrés. Ce ratio nous servira de métrique pour faire une analyse comparative entre notre méthode et celles présentées précédemment.

TABLEAU 7.1 – Insertion de points de test en fonction d'un seuil de probabilités

Benchmark	**Avant insertion de points de test**		**Après insertion de points de test**	
	Seuil de probabilité	Nombre de signaux avec probabilités ≤ seuil	Nombre de points de test insérés	Moyenne des probabilités modifiées
apex2	0.01	21	8	+0.39
	0.05	103	26	+0.40
apex4	0.01	58	29	+0.50
	0.05	683	229	+0.44
c432	0.1	7	4	+0.27
	0.2	53	31	+0.34
c1355	0.1	64	32	+0.25
	0.2	112	64	+0.31
c2670	0.001	28	11	+0.45
	0.01	33	12	+0.46
c5315	0.01	19	7	+0.42
	0.1	36	20	+0.44
c7552	0.01	45	17	+0.42
	0.05	77	49	+0.44
i4	0.05	40	16	+0.40
	0.1	100	40	+0.35
i7	0.1	20	10	+0.47
	0.2	172	52	+0.31
Moyenne				+0.38

7.5.2 Compromis entre gain en probabilité et sur-coût en surface

Plusieurs expériences ont été réalisées sur le circuit apex4 avec différents nombres de points de test (100, 150, 200 et 229) et un seuil de probabilité de 0.05. Les résultats de ces expériences sont détaillés dans le Tableau 7.2.
229 points de test sont nécessaires pour équilibrer les 683 signaux (soit l'ensemble des signaux dont les probabilités initiales étaient inférieures ou égales au seuil). Avec 100 points de test, nous arrivons à équilibrer 48.8% des 683 signaux et nous passons à 90.9% des signaux avec 200 points de test. Ainsi nous constatons que quand le nombre de points de test est multiplié par 2, le nombre de signaux équilibrés est multiplié en moyenne par ≈ 2.

TABLEAU 7.2 – Insertion itérative de points de test

Benchmark	**Avant insertion de points de test**		**Après insertion de points de test**	
	Seuil de probabilité	Nombre de signaux avec probabilités ≤ seuil	Nombre de points de test insérés	Nombre de signaux avec probabilités ≤ seuil restants
apex4	0.05	683	100	350
	0.05	683	150	175
	0.05	683	200	62
	0.05	683	229	0

7.5.3 Facilitation de la détection de CTM par test logique

Pour évaluer la résilience à l'insertion de CTMs des circuits ainsi modifiés avec points de test, nous avons inséré trois CTMs dans des circuits avec et sans points de test. En appliquant des vecteurs aléatoires sur les entrées primaires des circuits avec et sans points de test, nous sommes en mesure de vérifier si ces points de test permettent d'améliorer le taux de détection en phase de test. Le trigger du CTM est une porte AND à 3 entrées et la payload est une porte XOR (voir figure 7.6).

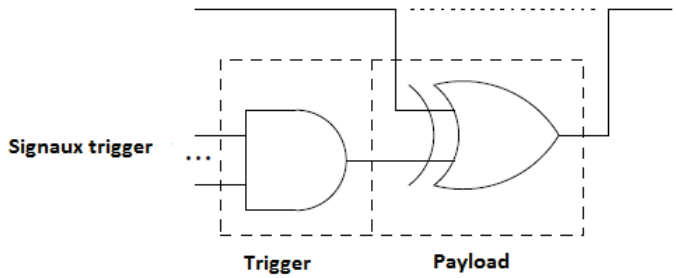


FIGURE 7.6 – Modèle de CTM basé sur une valeur rare.

Pour choisir les signaux d'entrée du trigger, nous avons calculé pour chaque circuit les probabilités de tous les signaux, et avons choisi ceux qui avaient les probabilités les plus déséquilibrées. Ensuite, nous avons utilisé un outil d'ATPG pour vérifier que les CTMs étaient effectivement déclenchables. Pour le circuit avec points de test, les Scan FFs ont été remplacées par des entrées primaires. Enfin, nous avons appliqué 10^6 vecteurs aléatoires sur les entrées primaires de chaque circuit.

TABLEAU 7.3 – Détection de CTM avec insertion de points de test

Circuit	% Activation CTM (circuits avec points de test)
c432	43.75
c1355	4.11
c7552	2.51
i7	6.30
apex4	0.018

Il faut noter que pour les 10^6 vecteurs aléatoires appliqués sur les circuits sans points de test, les triggers des CTMs n'ont jamais été déclenchés. À l'inverse, tous les triggers ont été activés plusieurs fois pour les circuits avec points de test. La deuxième colonne du Tableau 7.3 donne le rapport entre le nombre de fois que le trigger du CTM a été activé sur le nombre total de vecteurs appliqués (10^6). Ceci prouve la difficulté d'insertion d'un CTM furtif dans un circuit où tous les signaux ont des probabilités équilibrées.

7.5.4 Comparaison avec les autres méthodes

TABLEAU 7.4 – Comparaison avec les méthodes précédentes

Performances	Méthodes précédentes			Méthode proposée
	Méthode présentée dans [STP12]	Méthode présentée dans [XML+14]	Méthode présentée dans [DBDN+14]	
Nombre de signaux déséquilibrés	275	NC	604	683
Nombre de points de test insérés	100	NC	1024	229
Sur-coût en surface (%)	5.2	15	7.3	2.52

Pour montrer l'apport de notre méthode par rapport à celles présentées dans la bibliographie, nous avons fait une étude comparative en se basant sur le nombre de points de test insérés, le nombre de signaux dont les probabilités ont été équilibrées et le sur-coût en surface. Le Tableau 7.4 présente les résultats des méthodes précédentes ainsi que celle de notre méthode. Dans [STP12], les auteurs génèrent un sur-coût en surface qui varie entre 0.2 et 5.2%. Pour 100 points de test insérés, ils arrivent à équilibrer 275 signaux avec un sur-coût en surface de 5.2%. Ils obtiennent en moyenne un ratio nombre de signaux équilibrés sur nombre de points de test égal à 2.45 Dans [XML+14], les auteurs génèrent un sur-coût en surface de 15% . Ce pourcentage représente le rapport entre le nombre de transistors ajouté pour équilibrer les probabilités sur le nombre total de transistors du circuit. Dans [DBDN+14], les auteurs génèrent un sur-coût en surface de 7.3% pour 38% des signaux impactés. Parmi ces méthodes, celle présentée dans [STP12] semble obtenir de meilleurs résultats. Il paraît donc plus intéressant de comparer notre méthode à celle-ci. En termes de sur-coût en surface nous obtenons un taux qui varie entre 0.37 à 2.52% contre 0.2 à 5.2% pour leur méthode. En termes de ratio nombre de points de test sur nombre de signaux équilibré, nous obtenons en moyenne 2.98 contre 2.45 pour leur méthode. Par conséquent, notre méthode a un coût

en surface plus faible par rapport aux méthodes précédentes.

7.6 Conclusion

L'objectif de la méthode présentée est de faciliter la détection de CTM basée sur des valeurs rares en équilibrant les probabilités des signaux déséquilibrés. Ainsi un attaquant ne pourra pas exploiter les signaux faiblement contrôlables pour attacher un trigger.

De ce fait, un CTM inséré dans un circuit avec points de test est facilement détectable par des méthodes orientées test logique. En comparaison aux méthodes précédentes, nous proposons une insertion de points de test utilisant des portes à 3 entrées, ce qui permet d'obtenir de meilleurs résultats en termes d'équilibrage de probabilités avec un sur-coût en surface faible. Les expériences menées sur plusieurs circuits, montrent que la méthode proposée permet d'obtenir des probabilités équilibrées pour tous les signaux ayant des probabilités au dessous du seuil défini. De plus, l'insertion de CTMs dans ces circuits a montré qu'il est très difficile de créer une condition rare dans un circuit avec points de test.

7.7 Références

[DBDN+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 *IEEE 20th International On-Line Testing Symposium*, pages 49–54. IOLTS, 2014.

[STP12] Hassan Salmani, Mohammad Tehranipoor, and Jim Plusquellic. A novel technique for improving hardware trojan detection and reducing trojan activation time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(1) :112–125, 2012.

[XML+14] Hao Xue, Tyler Moody, Shuo Li, Xiaomeng Zhang, and Saiyu Ren. Low overhead design for improving hardware trojan detection efficiency. In *IEEE National Aerospace and Electronics Conference*, pages 379–383. NAECON, 2014.

[ZZTT14] Bin Zhou, Wei Zhang, Srikanthan Thambipillai, and JKJ Teo. A low cost acceleration method for hardware trojan detection based on fan-out cone analysis. In *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis*, page 28. ACM, 2014.

Chapitre 8

Conclusions générales et perspectives

« *Well done is better than well said.* »

Benjamin Franklin

8.1 Conclusions générales

Dans cette thèse, nous avons développé des méthodes permettant de détecter en phase de test et en-ligne, la présence de CTM dans un circuit intégré. Nous avons également proposé des méthodes permettant de prévenir l'insertion CTM durant la fabrication dans une fonderie.

Pour la détection en phase de test, les résultats montrent que le test de production et les vecteurs aléatoires ne sont pas efficaces pour détecter un CTM basé sur une condition rare (i.e. un CTM qui utilise comme trigger une combinaison de signaux faiblement contrôlables). De plus nous avons aussi montré qu'il est possible de construire un CTM furtif avec des signaux qui ne sont pas faiblement contrôlables. Ainsi la procédure que nous avons présenté permet de générer des vecteurs de test destinés à exciter des CTMs furtifs.

Pour la détection en-ligne, nous avons présenté une méthode basée sur la duplication matérielle. Cette méthode permet d'avoir un taux de détection très élevé avec un taux de faux positif nul. Elle permet aussi de prévenir l'insertion de CTM. En effet, les résultats montrent qu'il est très difficile voire impossible de distinguer les deux répliques à partir du layout pour insérer un CTM.

Nous avons présenté deux méthodes de prévention. Une méthode qui permet de densifier le layout pour empêcher qu'un attaquant puisse exploiter les espaces vides ou enlever les filler cells pour y mettre un CTM à leur place. Les résultats montrent qu'on arrive à atteindre pour certains circuits un taux de remplissage de 100%, c'est à dire qu'on arrive à remplir tous les espaces vides utilisables.

L'autre méthode consiste à insérer des points de test pour faciliter la détection en phase de test. Nous avons vu que les CTMs furtifs sont basés sur une condition rare, construits à partir d'une combinaison de signaux faiblement contrôlables. Les points de test que nous ajoutons permettent d'améliorer la contrôlabilité des signaux et rendre plus facile la détec-

tion. Les résultats présentés prouvent qu'il est difficile d'insérer un CTM furtif dans un circuit avec points de test.

8.2 Perspectives

Plusieurs méthodes de détection basées sur le test logique ont été proposées ces dernières années et ouvrent de nouvelles perspectives en termes de sélection de signaux de trigger et de génération de patterns spécifiques à la détection de CTM. Ainsi, pour la sélection des signaux pouvant être utilisés comme trigger, un algorithme pourrait être développé en utilisant le clustering. Ceci permettrait de prendre en compte les signaux rares et non rares avec un nombre de signaux réduit.

Pour la détection en-ligne, une outil permettant d'avoir deux netlists fonctionnellemnt équivalentes mais structurellement différentes est en train d'être développé. Ceci dans le but d'optimiser la procédure de duplication et d'éviter toute redondance de signaux équivalents. Pour la prévention d'insertion de CTM dans le layout, le taux de remplissage du layout pourrait être amélioré en optimisant le routage. Ainsi un algorithme permettant de faire un routage global avec les fonctions additionnelles pourraient être développé.

Nous envisageons également d'étudier la complémentarité entre les différentes approches (test logique, SCA et détection optique). En effet, nous avons vu dans le chapitre 2 que les méthodes SCA sont plus efficaces pour les CTMs de grande taille et les méthodes de test logique sont plus efficaces les CTMs de petite taille. En combinant les deux, un taux de détection très élevé devrait pouvoir être obtenu.

Une autre direction de recherche possible consisterait à étudier la complémentarité des méthodes de remplissage de layout avec les méthodes de détection optique.
En effet, celles-ci sont basées sur l'inspection de dernières couches de métal. Il serait intéressant d'étudier la possibilité pour un attaquant d'insérer et connecter un CTM sans modifier les dernières couches dans un CI dont le layout est très congestionné.

Contributions scientifiques

- **A novel hardware logic encryption technique for thwarting illegal overproduction and Hardware Trojans.**
  Sophie Dupuis, Papa-Sidy Ba, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre.
  IOLTS 2014 : *IEEE 20th International On-Line Testing Symposium 2014.*
- **New testing procedure for finding insertion sites of stealthy hardware trojans.**
  Sophie Dupuis, Papa-Sidy Ba, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre.
  DATE 2015 : *Design, Automation & Test in Europe Conference & Exhibition 2015.*
- **On the limitations of logic testing for detecting hardware Trojans horses.**
  Marie-Lise Flottes, Sophie Dupuis, Papa-Sidy Ba, Giorgio Di Natale, Bruno Rouzeyre.
  DTIS 2015 : *Design & Technology of Integrated Systems in Nanoscale Era 2015.*
- **Hardware Trojan prevention using layout-level design approach.**
  Papa-Sidy Ba, Manikandan Palanichamy, Sophie Dupuis, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre.
  ECCTD 2015 : *European Conference Circuit Theory and Design 2015.*
- **Duplication-based Concurrent Detection of Hardware Trojans in Integrated Circuits.**
  Manikandan Palanichamy, Papa-Sidy Ba, Sophie Dupuis, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre.
  TRUDEVICE 2016 : *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices 2016.*
- **Hardware Trust through Layout Filling : A Hardware Trojan Prevention Technique.**
  Papa-Sidy Ba, Manikandan Palanichamy, Sophie Dupuis, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre.
  ISVLSI 2016 : *IEEE Computer Society Annual Symposium on VLSI 2016.*
- **Using Outliers to Detect Stealthy Hardware Trojan.**
  Papa-Sidy Ba, Manikandan Palanichamy, Sophie Dupuis, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre.
  IVSW 2016 : *IEEE International Verification and Security Workshop 2016.*