Méthodologie et développement de solutions pour la sécurisation des circuits numériques face aux attaques en tensions

Kamil Gomina

Ecole Nationale Supérieure des Mines
SAINT-ETIENNE

NNT : 2014 EMSE 0751

THÈSE

présentée par

Kamil GOMINA

pour obtenir le grade de
Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : Microélectronique

METHODOLOGIE ET DEVELOPPEMENT DE SOLUTIONS POUR LA SECURISATION DES CIRCUITS NUMERIQUES FACE AUX ATTAQUES EN TENSION

Soutenue à Gardanne, le 11 septembre 2014

Membres du jury

Rapporteurs :	Régis LEVEUGLE	Professeur, TIMA, Grenoble
	Bruno ROUZEYRE	Professeur, LIRMM, Montpellier
Directeur de thèse :	Assia TRIA	Ingénieur de recherche, CEA-TECH, Gardanne
Président du jury :	Jean-Michel PORTAL	Professeur, IM2NP, Marseille
Examinateurs :	Jean-Baptiste RIGAUD	Maitre-assistant, ENSM-SE, Gardanne
	Philippe GENDRIER	Ingénieur, STMicroelectronics, Crolles
Invité	Philippe CANDELIER	Ingénieur, STMicroelectronics, Crolles

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

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

ABSI	Nabil	CR		CMP
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2		FAYOL
BASSEREAU	Jean-François	PR		SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BERGER DOUCE	Sandrine	PR2		FAYOL
BERNACHE-ASSOLLANT	Didier	PR0	Génie des Procédés	CIS
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR1	Informatique	FAYOL
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
BURLAT	Patrick	PR2	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DARRIEULAT	Michel	IGM	Sciences et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR1	Sciences et génie des matériaux	SMS
DESRAYAUD	Christophe	PR2	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FEILLET	Dominique	PR2	Génie Industriel	CMP
FEVOTTE	Gilles	PR1	Génie des Procédés	SPIN
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
HAN	Woo-Suck	CR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean-Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Frank	DR	Sciences et génie des matériaux	SMS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre			CMP
NORTIER	Patrice	PR1		SPIN
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	CR	Génie des Procédés	CIS
ROBISSON	Bruno			CMP
ROUSSY	Agnès	MA(MDC)		CMP
ROUSTANT	Olivier	MA(MDC)		FAYOL
ROUX	Christian	PR		CIS
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	MA(MDC)	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	MR(DR2)	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR1	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

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

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FEULVARCH	Eric	MCF	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	PU	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	PU	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 28/03/2014

Table des matières

Table des figures

Liste des tableaux

Introduction

La multiplication des objets connectés ces dernières années a engendré l'augmentation des données partagées. Ces objets connectés à Internet échangent de nombreuses informations et notamment des données personnelles ou confidentielles. Selon une étude de l'IDate [Idate 2013], 15 milliards d'objets étaient connectés en 2012 contre 4 milliards en 2010. Il y a donc un besoin d'échanges et de stockage de ces informations de façon sécurisée. D'autre part, la sécurité des circuits qui était majoritairement liée au domaine des cartes à puce tend à s'ouvrir vers d'autres applications grand public comme la télévision à péage ou encore la téléphonie mobile.

La sécurisation des données dans les SoCs est généralement basée sur la cryptographie. Ce procédé consiste à chiffrer un message à l'aide d'une clé pour que celui-ci ne soit pas lisible par des personnes non autorisées. Les algorithmes de cryptographie reposent sur des propriétés mathématiques qui complexifient l'accès au message initial sans la clé nécessaire. Néanmoins, l'implantation de ces algorithmes au niveau d'un circuit présente des failles qui compromettent la confidentialité ou l'intégrité des données.

Ces dernières années, plusieurs techniques d'attaque qui mettent en exergue ces faiblesses se sont développées. Les attaques peuvent être passives ou actives, si elles nécessitent ou non de perturber les opérations du circuit. Dans tous les cas, les failles créées ou exploitées proviennent de phénomènes physiques inhérents au fonctionnement du circuit. Ces vulnérabilités peuvent donc apparaître quelle que soit la fonction réalisée par le circuit, qu'il s'agisse ou non d'une opération de cryptographie. Pour garantir la sécurité de ces systèmes, il est nécessaire, d'une part, de comprendre les mécanismes mis en jeu lors d'une attaque. D'autre part, les concepteurs cherchent à réduire la taille des circuits intégrés pour gagner en performances et en autonomie. Il convient donc de se demander quel est l'impact de l'utilisation de technologies avancées sur la sécurité des systèmes.

Afin de limiter les vulnérabilités de ces systèmes sensibles, il faut mettre en place des mesures de protection dès la phase de conception. L'intégration de ces mesures doit être prise en compte très tôt dans le flot de conception des circuits, dès l'étape de spécification fonctionnelle, pour obtenir les certifications de sécurité nécessaires, délivrées par des organismes indépendants d'évaluation sécuritaire tels que les CESTI (Centres d'Évaluation de la Sécurité des Technologies de l'Information). Ces contremesures visent

généralement un type d'attaque. Le choix de la protection est donc primordial pour limiter son impact sur la surface et les performances du circuit.

Le travail présenté à travers ce manuscrit s'articule autour des attaques passives et actives en tension sur des circuits numériques conçus dans des nœuds technologiques avancés. Ces attaques sont particulièrement accessibles car les dispositifs nécessaires pour leur mise en œuvre sont peu coûteux. Les contributions de ce travail portent sur différents points :

- L'évaluation de vulnérabilités de circuits numériques vis-à-vis des attaques passives en tension durant la phase de conception.
- La compréhension des mécanismes d'injection de fautes induites par des attaques actives en tension.
- La conception et l'évaluation de solutions de détection d'attaques par impulsions sur la tension d'alimentation.

Le premier chapitre présentera les différentes techniques d'attaques sur les circuits. Il met en exergue l'ensemble des menaces visant les circuits et également les principales contremesures qui ont été proposées pour les protéger. Cette partie montre également les contraintes de conception numérique à considérer pour être en mesure de déployer des solutions dans un cadre industriel.

Dans le deuxième chapitre, les attaques par observation de l'alimentation du circuit sont étudiées en phase de conception. La signature électrique d'un circuit numérique sera analysée en développant un modèle électrique. Ce modèle sera utilisé pour évaluer aussi bien les circuits que les contremesures pour mieux anticiper les attaques.

Enfin, le dernier chapitre abordera les attaques par impulsions de tension d'alimentation. Les mécanismes d'injection de fautes seront analysés pour mieux comprendre les phénomènes impliqués. Par la suite, des contremesures seront étudiées puis conçues dans un flot numérique standard. Finalement, ces solutions seront évaluées sur silicium pour en déterminer les performances.

Chapitre 1

Menaces et sécurisation des circuits intégrés numériques

Sommaire

1.1 Aperçu des attaques visant les circuits numériques

Au cours de ces 20 dernières années, de nombreux travaux portant sur les méthodes d'attaques de circuits sont apparus dans la littérature, suivis de mise en pratique, pour faire prendre conscience de la menace qui pèse sur la sécurité des circuits intégrés. Si ces techniques ciblent souvent des circuits cryptographiques, elles peuvent être utilisées sur d'autres systèmes afin de compromettre leur sécurité. En fonction des moyens et des connaissances dont disposent les attaquants, IBM a proposé une classification de ceux-ci [Abraham 1991] :

- la première classe est constituée des attaquants astucieux (*clever outsiders*) qui disposent de moyens limités et de connaissances disponibles publiquement. Il s'agit généralement de personnes isolées mais qui, sur du long terme, peuvent devenir nombreux et former un groupe potentiellement dangereux.
- la deuxième classe (*knowlegeable insiders*) dispose de compétences spécifiques et de moyens plus importants. Les attaquants sont capables de comprendre un système complexe grâce aux équipements auxquels ils ont accès.
- la dernière catégorie regroupe les organisations financées (*funded organizations*). Ces groupes sont capables de rassembler des experts aux compétences complémentaires et disposent de moyens financiers très importants. Ils peuvent également faire appel à des personnes faisant partie des classes I et II. Les attaquants appartenant à cette catégorie font généralement partie d'agences gouvernementales, de grandes entreprises ou de groupes de crimes organisés.

Cette première partie rappelle les moyens d'attaques répertoriés dans la littérature. Ils sont souvent distingués en fonction de leur caractère intrusif ou destructeur. La description de ces catégories d'attaques sera l'objet du paragraphe suivant. Les principales contremesures développées pour lutter contre ces attaques sont ensuite répertoriées. Elles permettent de renforcer la sécurité des systèmes, qui peut être évaluée selon plusieurs critères, nous verrons l'exemple des critères communs. Par la suite, l'intégration de ces contremesures en suivant le flot de conception standard des circuits numériques sera analysée. Les résultats de cette analyse nous aideront à prendre en compte les contraintes dans le choix et l'implantation de ces solutions.

1.1.1 Attaques invasives

La première catégorie d'attaques est qualifiée d'invasive. Pour récupérer les informations sensibles, le circuit est désassemblé de son boîtier afin de pouvoir y déposer directement des sondes permettant d'observer les données. À l'aide de ces sondes, il est également possible de communiquer avec le circuit [Kömmerling 1999]. Une autre méthode consiste à réaliser la rétro-conception du circuit cible (*reverse engineering*). Elle est utilisée pour comprendre la structure du circuit afin d'apprendre ou d'émuler son comportement. Cette technique nécessite le même type d'équipements que ceux

utilisés pour la fabrication du circuit. Par conséquent, des équipements coûteux sont nécessaires pour mettre en place ces attaques. Il est aussi important de noter qu'elles peuvent prendre des jours voir des semaines pour être réalisées [Skorobogatov 2005]. À titre d'exemple, les différentes étapes de désassemblage d'une carte à puce sont données dans [Kömmerling 1999, Skorobogatov 2005]. Le plastique est d'abord chauffé jusqu'à ce

FIGURE 1.1 : Désassemblage du boîtier avec de l'acide nitrique [Kömmerling 1999]

qu'il devienne flexible, la puce est alors détachée en pliant la carte. Ensuite, la résine qui recouvre le silicium est dissoute dans de l'acide (figure 1.1). L'étape suivante est d'enlever la couche de passivation afin d'avoir accès aux niveaux de métaux les plus élevés. On peut dès lors établir des contacts sur les pistes métalliques à l'aide d'équipements spécifiques. Ce désassemblage permet également d'accéder aux couches internes et d'observer la structure de la puce qui peut servir pour de la rétro-conception (figure 1.2).

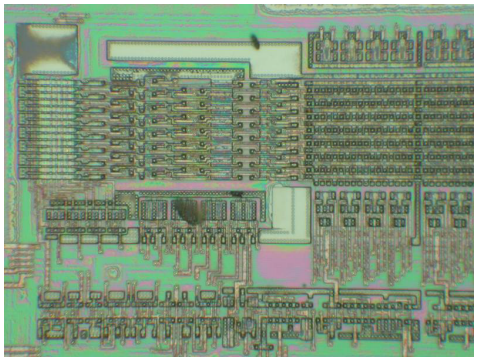


FIGURE 1.2 : Microcontrôleur Motorola MC68HC705C9A après gravure chimique [Skorobogatov 2005]

1.1.2 Attaques non invasives

Les attaques non invasives exploitent les informations inhérentes au fonctionnement du circuit comme le temps d'exécution, la consommation électrique, les émissions électromagnétiques, la température ou encore le bruit émis [Zhou 2005]. Ce sont en général des émissions non intentionnelles qui sont détournées en vue de mener une attaque. Ces informations peuvent être reliées statistiquement à l'activité, aux opérations internes ainsi qu'aux données manipulées dans le circuit. Certaines de ces attaques sont passives car elles ne nécessitent pas de perturber le circuit : on parle alors d'attaques par canaux auxiliaires. Les autres attaques sont donc actives, elles impliquent des modifications locales ou globales de l'environnement du circuit en jouant sur des paramètres tels que la tension d'alimentation, la fréquence de cadencement de l'horloge ou encore la température [Kim 2007]. Ces modifications peuvent engendrer la désactivation de solutions de protection du circuit ou entrainer une mauvaise opération.

Ces attaques ne nécessitent pas de préparation spécifique du circuit contrairement aux attaques invasives. Elles sont donc moins coûteuses à mettre en place et à réaliser. De plus, étant donné que la puce ne présente aucun dommage, il n'y a aucune trace pouvant attester que le circuit a été attaqué. Il s'agit donc d'une menace importante pour la sécurité des circuits. Toutefois, la récupération et l'analyse des données peut s'avérer longue avant de parvenir à l'information cherchée. Nous allons nous intéresser dans un premier temps aux attaques par canaux auxiliaires.

1.1.2.1 Attaques par canaux auxiliaires

Les opérations réalisées par un circuit prennent un certain temps, consomment du courant électrique, et émettent des radiations électromagnétiques : tous ces canaux sont des sources d'informations liées aux données en cours de traitement lors du fonctionnement d'un circuit.

Attaques temporelles. Le principe des attaques temporelles a été présenté par Kocher [Kocher 1996] : les opérations qui ont lieu dans un circuit dépendent des données d'entrée du système. Une mesure précise du temps de réponse peut révéler des informations a priori secrètes. Dans ce cas, l'attaquant dispose d'un ensemble de messages qui sont traités par le circuit avec les temps correspondant à chaque durée de traitement. Le principe de cette attaque est montré sur la figure 1.3. Plusieurs algorithmes peuvent être vulnérables à cette attaque. Cela résulte souvent de l'implantation logicielle des algorithmes : pour des raisons de gain en performances, les opérations suivant des conditions et des embranchements sont contournées. Certaines opérations ayant des temps de calculs non constants comme des multiplications ou des divisions sont effectuées [Dhem 2000]. Il en résulte des dépendances des temps de calculs vis-à-vis des données secrètes. Un exemple récent d'étude sur les attaques temporelles est donné dans

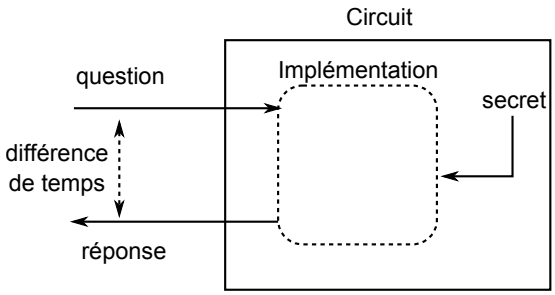


FIGURE 1.3 : Principe de l'attaque temporelle [Dhem 2000]

[Danger 2014]. L'attaque porte sur un microcontrôleur 8 bits exécutant un chiffrement AES 128 bits. Les auteurs montrent qu'il est possible de retrouver la clé de l'algorithme, en mesurant le temps d'exécution. L'idée est de considérer le carré du temps total d'exécution (2^{nd} *order timing attack*) car il dépend de la clé utilisée pour le calcul.

Attaques par analyse de consommation électrique. Cette attaque, présentée dans [Kocher 1999] se base sur la consommation électrique des circuits utilisant la technologie CMOS. Lorsqu'il y a une transition de la valeur de sortie d'une porte logique (de l'état haut vers l'état bas ou inversement), on peut mesurer une variation du courant consommé en fonction de la transition. Ce courant correspond à la charge et à la décharge de la capacité de l'étage logique suivant comme indiqué sur la figure 1.4. C_L représente la capacité de l'étage de sortie de l'inverseur. Les attaques par analyse

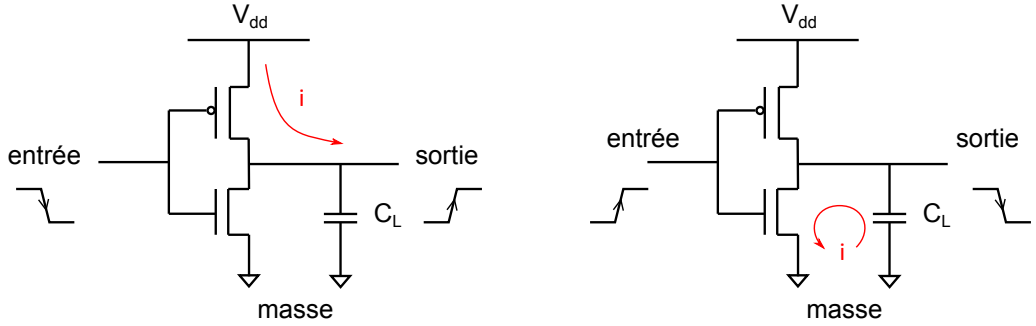


FIGURE 1.4 : Fonctionnement d'inverseur en technologie CMOS

de consommation consistent à mesurer, puis à analyser la consommation de courant par le circuit en fonction du temps lors de ses opérations. Leur principe repose sur la corrélation entre ce courant consommé et les données ou instructions qui sont traitées pour une opération donnée. [Mangard 2010].

Concrètement, les attaques par analyse de consommation nécessitent une mesure physique de la consommation électrique du circuit attaqué. Plusieurs méthodes permettent d'obtenir ces informations de consommation. La plus simple et la plus répan-

due est l'utilisation d'une résistance placée sur la ligne d'alimentation en série avec le circuit. En général, il s'agit d'une résistance de petite valeur, typiquement de 1 Ω à 50 Ω. Cependant, cette valeur doit permettre de mesurer la tension à ses bornes à l'aide d'un oscilloscope numérique. Un exemple d'expérimentation sera donné dans le chapitre 2. Les sections suivantes indiquent comment ces mesures peuvent être exploitées.

***Simple Power Analysis* (SPA)** Le but de l'analyse de courant simple est d'estimer directement à partir de la consommation mesurée quelle instruction particulière est exécutée à un moment donné et quelle est la valeur des signaux internes. L'analyse de courant simple peut révéler une séquence d'instructions et donc peut être utilisée pour détecter un algorithme de cryptographie [Mangard 2010] ou même d'identifier les différentes étapes de celui-ci : repérer des permutations dans l'algorithme DES (Data Encryption Standard), les différentes rondes de l'algorithme AES (Advanced Encryption Standard) ou encore des multiplications et des exponentiations (*square*) de l'algorithme RSA (Rivest Shamir Adleman) comme indiqué sur la figure 1.5. En regardant la forme et la durée du courant mesuré, les opérations de multiplication (*multiply*) et d'exponentiation (*square*) peuvent être repérées.

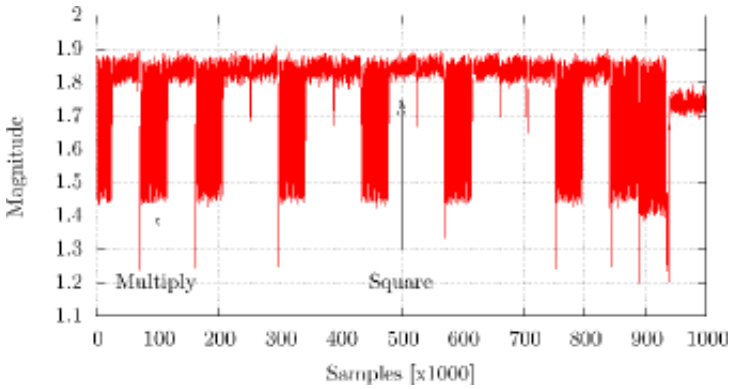


FIGURE 1.5 : Analyse de courant simple de courant sur l'algorithme RSA [Selmane 2010]

***Differential Power Analysis* (DPA)** L'analyse différentielle de courant [Kocher 1999] est une attaque plus sophistiquée que l'analyse simple. Elle est généralement utilisée pour des algorithmes de cryptographie. Elle se base sur l'analyse statistique pour isoler les signaux qui intéressent l'attaquant de l'ensemble du circuit. Cette méthode statistique permet d'identifier des différences infimes dans la consommation de courant qui peuvent être utilisées pour retrouver des bits d'une donnée secrète de façon individuelle. La puissance de l'analyse repose sur sa capacité à récupérer une information même si la relation entre la consommation et la donnée secrète est trop complexe pour être directement identifiée par une analyse simple de courant. À

noter aussi que les attaques DPA se focalisent sur un instant donné pour déterminer la dépendance des données et de la consommation, la dépendance de la consommation vis-à-vis du temps n'est pas primordiale. En effet, on se place à un instant donné pour étudier la différence de consommation pour différents vecteurs d'entrée [Mangard 2010].

Attaques électromagnétiques. Cette attaque a été présentée pour la première fois par Quisquater et Samyde [Quisquater 2000]. Tout mouvement d'une charge provoque un champ électromagnétique. Ainsi, l'utilisation d'une micro-antenne peut identifier le réseau de distribution d'alimentation et permettre d'établir un profil de consommation de la puce. Un attaquant peut donc observer ces émanations pour avoir des indications sur les données en cours d'utilisation. Comme pour l'analyse de consommation électrique, on peut distinguer l'analyse électromagnétique simple et différentielle. Cependant, l'analyse de consommation électrique mesure la totalité du courant du circuit tandis que l'analyse électromagnétique peut se focaliser sur des zones restreintes. Agrawal et al. [Agrawal 2003] suggèrent que les émanations électromagnétiques proviennent de deux sources :

- les émanations directes provenant de la circulation du courant dans la puce
- les émanations non intentionnelles provenant des effets de couplage entre les sous-blocs dus à leur proximité.

Selon les auteurs, les émanations non intentionnelles ont une meilleure portée et peuvent être mesurées jusqu'à 4.5 mètres. Même si cette attaque est non invasive, elle peut nécessiter une extraction de la puce de son boîtier pour obtenir une meilleure mesure.

Autres attaques par canaux auxiliaires. Il existe d'autres types d'attaques par canaux auxiliaires même si celles-ci sont moins développées dans la littérature.

Les attaques acoustiques consistent à analyser le son émis par le système pendant son fonctionnement. À l'instar des précédentes attaques répertoriées, il existe une différence de fréquences acoustiques émises par le circuit en fonction du calcul effectué. Cette technique a été utilisée dans [Genkin 2013] pour extraire la totalité des 4096 bits d'une clé de déchiffrement RSA. Le signal acoustique mesuré provient de la vibration des composants électroniques (bobines et condensateurs) dans le circuit de régulation de tension. Celui-ci parvient difficilement à fournir une tension d'alimentation constante au processeur à cause de la grande fluctuation de consommation causée par les différentes opérations effectuées par le CPU. Bien que cette attaque ait été réalisée dans des conditions favorables (choix des textes à déchiffrer, une seule machine en fonctionnement), elle met en avant l'utilisation d'un nouveau canal de mesure d'informations.

L'analyse de la lumière émise par un composant est une autre source d'informations qui peut être exploitée. Loughry et Umphress [Loughry 2002] ont décrit dans leurs travaux comment la lumière émise par la diode électroluminescente d'un ordinateur pouvait être corrélée aux informations qui étaient traitées.

1.1.2.2 Les attaques actives

Le deuxième groupe d'attaques non invasives concerne les injections de fautes. Les attaques en fautes exploitent les propriétés physiques des composants.

Attaques par modification de la température. Les informations sensibles utilisées par un circuit au cours de son fonctionnement ont besoin d'être stockées. Elles ne sont pas censées être divulguées à l'extérieur du circuit. Les mémoires volatiles sont utilisées dans ce but : l'information est perdue lorsque l'alimentation est coupée. Toutefois, à cause du phénomène de rétention de données, si le temps nécessaire à la lecture de la mémoire est plus petit que le temps de rétention, la volatilité de la mémoire peut être contournée. Pour une cellule SRAM, le phénomène de rétention est activé aux alentours de -20°C. Dès lors, un attaquant peut alors récupérer le contenu de la mémoire [Ali 2011a].

Dans [Skorobogatov 2009], il est montré que le réchauffement local d'une mémoire non volatile flash ou E^2PROM d'un microcontrôleur peut modifier son contenu de manière permanente. L'attaque a été utilisée pour effacer plusieurs bits de la mémoire.

D'autre part, les circuits sont spécifiés pour fonctionner dans une certaine gamme de températures. Au-delà de cette gamme, les temps de propagation sont modifiés au point de ne plus garantir le bon fonctionnement du composant. On peut retrouver la mise en évidence de ce phénomène dans [Dutertre 2010]. Des fautes ont été injectés dans un algorithme de cryptographie embarqué sur un *FPGA*. La température du circuit a été portée à 210°C pour obtenir les premières fautes.

Modification de la fréquence d'horloge. Le principe ici est d'augmenter la fréquence de l'horloge qui cadence les opérations du circuit. Cela ne concerne évidemment que les circuits synchrones. Au-delà d'une fréquence maximale, les opérations logiques n'ont pas le temps de s'effectuer avant le front d'horloge qui vient les mettre à jour. Cette méthode, bien qu'efficace, ne permet pas un contrôle spatial ou temporel des fautes injectées. En effet, en modifiant la fréquence d'horloge, les fautes sont potentiellement injectées à chaque cycle d'horloge, ce qui peut provoquer le non-fonctionnement complet du circuit attaqué.

Il existe une amélioration de cette méthode, qui consiste non pas à modifier le signal d'horloge en entier, mais à modifier la période d'un ou plusieurs cycles choisis par l'attaquant. Il s'agit ici d'impulsions créées sur le signal d'horloge afin d'en modifier temporairement la fréquence. Dans [Amiel 2006], il est mentionné que de tels raccourcissements de périodes d'horloge provoquent une modification d'un ou de plusieurs octets. L'idée est de diminuer progressivement la période d'un cycle jusqu'à ce qu'une faute apparaisse. Cette méthode d'injection de fautes a été présentée plus récemment dans [Agoyan 2010]. Il est montré qu'avec cette méthode, on est capable de modifier un seul

bit dans le circuit attaqué avec une bonne synchronisation temporelle et de manière reproductible.

Cependant, ces injections requièrent l'accès direct au signal d'horloge. Il n'est pas possible d'attaquer directement un circuit utilisant son propre système de génération d'horloge car la déconnexion de ce signal paraît difficile [Barenghi 2012]. D'autre part, les circuits numériques asynchrones ne possèdent pas de signal d'horloge, et peuvent donc s'avérer efficaces contre ce type d'attaque [Fournier 2003, Monnet 2006].

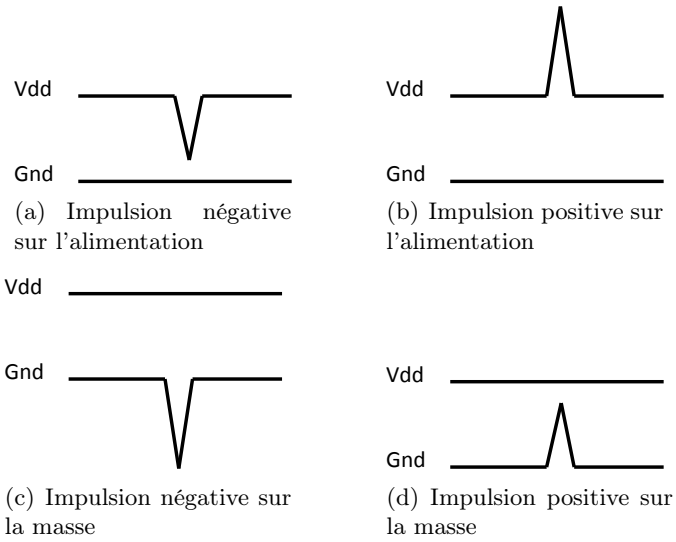
Modification de la tension d'alimentation. Les circuits intégrés sont conçus pour fonctionner dans une certaine gamme de tension. Cette tension dépend de la technologie des composants élémentaires CMOS. À titre d'exemple, pour les nœuds technologiques avancés actuels (28 nm), la tension d'alimentation nominale se situe autour de 1 V. Au-delà de la gamme opérationnelle de tensions, il n'est plus possible de garantir le bon fonctionnement du circuit. Dès lors, plusieurs types d'attaques peuvent être menées en jouant sur la tension d'alimentation. La sous-alimentation d'un processeur peut entrainer des interruptions ou des sauts d'instructions. On peut ainsi induire des fautes transitoires sur un ou plusieurs bits au fur et à mesure que la tension est diminuée. La baisse de tension modifie les propriétés temporelles des portes logiques en les ralentissant. Ainsi, lorsque les contraintes temporelles ne sont plus respectées, les premières fautes apparaissent [Zussa 2012]. L'influence de la baisse de la tension d'alimentation sur un circuit implémenté sur *FPGA* est montrée de manière expérimentale dans [Dutertre 2010]. Ce mode opératoire a été utilisé dans [Selmane 2008] sur une carte à puce conçue en technologie 130 nm embarquant un coprocesseur AES.

Comme pour les attaques par modification de la fréquence d'horloge, il est possible de modifier temporairement la tension d'alimentation en y créant une impulsion. L'avantage de cette méthode est d'avoir une meilleure synchronisation temporelle du moment où la faute est injectée. Il y a quatre principaux types d'impulsions qui peuvent être appliqués sur les rails d'alimentation d'un circuit (figure 1.6) :

- impulsion positive sur l'alimentation (1.6(b))
- impulsion négative sur l'alimentation (1.6(a))
- impulsion négative sur la masse (1.6(c))
- impulsion positive sur la masse (1.6(d))

Les attaques par impulsion en dessous de la tension nominale sont les plus répandues. Elles ont notamment été utilisées comme méthodes d'injection de fautes dans [Choukri 2005, Bar-El 2006] sur des circuits cryptographiques.

Les attaques par surtension qu'il s'agisse de modifications quasi-statiques ou d'impulsions sur l'alimentation restent relativement rares dans la littérature. Peu d'études se sont focalisées sur les effets de telles attaques. On peut néanmoins citer [Hutter 2009]. Des impulsions de 7 V ont été appliquées sur un tag RFID, ce qui a provoqué une écri-



(a) Impulsion négative sur l'alimentation

(b) Impulsion positive sur l'alimentation

(c) Impulsion négative sur la masse

(d) Impulsion positive sur la masse

FIGURE 1.6 : Les différents types d'impulsions sur les rails d'alimentation [Yanci 2008]

ture erronée dans la mémoire du tag. Toutefois, les mécanismes mis en jeu lors de ces attaques ne sont pas mentionnés et restent peu connus.

Attaques par impulsions électromagnétiques. Les attaques électromagnétiques ont d'abord été utilisées de manière passive, c'est-à-dire pour réaliser des attaques par canaux auxiliaires comme indiqué dans la section 1.1.2.1. Cependant, ce vecteur constitue également un moyen d'attaque qui a été étudié par plusieurs groupes de recherche. Quisquater et Samyde [Quisquater 2002] ont décrit l'usage d'une sonde pour appliquer un fort champ magnétique transitoire sur un microprocesseur. Ils ont utilisé le flash d'un appareil photo pour injecter une haute tension dans la bobine de la sonde. La forte tension entraine un champ magnétique qui à son tour provoque des courants de Foucault à la surface de la puce qui sont à l'origine des erreurs relevées. Dans [Schmidt 2007], les auteurs ont utilisé un éclateur à la place d'une sonde pour générer un arc électrique au-dessus du microcontrôleur exécutant l'algorithme de cryptographie asymétrique RSA. Grâce à la faute injectée, l'attaque réalisée a permis de factoriser le module de chiffrement de l'algorithme.

Plus récemment, les travaux [Dehbaoui 2012a] ont porté sur l'étude des fautes injectées par une sonde électromagnétique sur un microcontrôleur 8-bits en technologie 0.35 μm. En envoyant une impulsion de 50 V d'amplitude et d'une durée de 20 ns à différents moments de la 10^{e} ronde de l'AES, chacun des 16 octets de la ronde a pu être fauté. L'attaque permet donc une bonne synchronisation temporelle. Les mêmes auteurs dans [Dehbaoui 2012b] se sont intéressés aux types de fautes induites par des impulsions électromagnétiques sur *FPGA*. En regardant des chemins de propagation

témoins placés à différents endroits dans le circuit, ils ont observé des modifications de temps de propagation de manière localisée dans le *FPGA*.

1.1.3 Attaques semi-invasives

Skorobogatov et Anderson [Skorobogatov 2003] ont introduit une nouvelle technique pour combler l'écart qui existait entre attaques invasives et non invasives. Les attaques semi-invasives ont l'avantage d'être moins coûteuses à mettre en œuvre que celles invasives. De plus, elles sont facilement reproductibles comme les attaques non invasives. Ces attaques peuvent nécessiter de décapsuler la puce de son boîtier, tout en la gardant fonctionnelle. Elles sont essentiellement utilisées en vue d'injecter des fautes mais également servir de techniques d'attaque par observation.

Dans les attaques semi-invasives, les injections de fautes sont réalisées à l'aide de sources lumineuses. Dans [Skorobogatov 2003], le circuit attaqué a été décapsulé et placé sous un microscope. Ensuite, la lumière créée par un flash d'appareil photo a été concentrée sur le circuit avec une feuille d'aluminium. L'attaque a permis de modifier les bits ciblés de cellules SRAM. Un autre type de source lumineuse a été utilisé dans [Schmidt 2009]. Des radiations UV ont été appliquées sur différents microcontrôleurs. Ces irradiations ont provoqué l'effacement des mémoires EPROM et flash où étaient stockées les constantes utilisées dans les algorithmes cryptographiques. En fonction du temps d'exposition aux UV, la totalité des cellules des mémoires non volatiles peut être effacée. À noter que les mémoires EPROM sont des anciennes technologies qui ne sont plus utilisées notamment dans les circuits sécurisés.

Il est également possible d'utiliser des sources lumineuses plus précises pour se focaliser sur des zones particulières d'un circuit ou pour couvrir une surface plus petite. L'usage de sources lasers s'avère particulièrement adapté dans ce cadre. En effet, il est possible de diminuer la taille d'un spot laser jusqu'à 1 μm. En-deçà, interviennent des difficultés dues à la diffraction du faisceau [Mirbaha 2011]. Les attaques laser peuvent être particulièrement adaptées pour injecter des fautes avec une précision mono-bit ou mono-octet. Cette précision est utile pour réaliser des attaques de type DFA (*Differential Fault Analysis*) ou encore des attaques par modification de ronde sur des algorithmes cryptographiques comme l'AES [Mirbaha 2011]. L'injection de fautes par attaque laser repose sur l'effet photoélectrique [Roscian 2012]. Lorsque le silicium est illuminé par une source laser avec une énergie suffisante (plus grande que l'énergie de gap), des paires électrons-trous sont créées. En général, ces paires sont recombinées et le comportement du circuit n'est pas modifié. Néanmoins, lorsque l'illumination traverse la jonction PN polarisée en inverse, formée par le substrat de type P et la zone active de type N, les paires électrons-trous sont déplacées dans des sens opposés sous l'effet du champ électrique de la zone de charge espace (ZCE) et donnent naissance au courant photoélectrique transitoire (figure 1.7). Plusieurs paramètres vont influencer l'injection

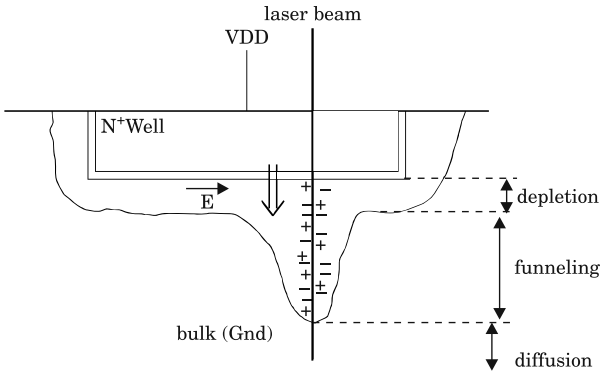


FIGURE 1.7 : Effet d'un faisceau laser sur une structure MOS [Roscian 2012]

de fautes d'un tir laser au-delà de la taille et de la puissance du laser. Le choix de la longueur d'onde, la position de tir (face avant ou face arrière du circuit), la distance entre la source laser et le circuit ou encore la durée du tir sont autant de paramètres qui font varier le résultat d'un tir laser [Sarafianos 2013, Roscian 2013].

Certaines techniques d'imagerie utilisent également l'illumination laser et l'effet photo-électrique. Deux de ces techniques ont été répertoriées dans [Skorobogatov 2005] :

- *Optical Beam Induced Current*
- *Light Induced Voltage Alteration*

Pour la première technique, le photo-courant est utilisé directement pour produire l'image du circuit : le port d'alimentation est connecté à un amplificateur de courant et les valeurs sont enregistrées avec une carte d'acquisition. Pour la seconde, pendant le scan de la surface de la puce, les changements de tension sont observés lorsqu'un courant constant est appliqué. Les photo-courants sont dépendants de l'état d'un transistor. Ces deux techniques ont permis de lire le contenu de la mémoire SRAM d'un microcontrôleur de manière non destructive.

Les attaques visant à compromettre la sécurité des circuits sont très variées et utilisent des principes différents. Certaines d'entre elles ne nécessitent pas de connaissances particulières sur l'implantation du système attaqué et peuvent être réalisées à faible coût. En effet, la plupart des attaques non invasives utilisent de simples équipements de laboratoire (générateur de tension, oscilloscope, générateur d'impulsions etc.) et sont donc relativement accessibles. La conception de circuits sécurisés doit tenir compte de ces attaques et intégrer des mesures de protection. En réponse à la menace qui pèse sur la sécurité de ces circuits et à la prolifération des attaques, de nombreuses contremesures ont été développées. Une revue des contremesures et de leur principe est donnée dans le paragraphe suivant.

1.2 Contremesures

Pour faire face aux attaques contre les circuits intégrés qui deviennent de plus en plus sophistiquées, les concepteurs ont dû développer des méthodes de protection visant soit à empêcher l'attaque, soit à la détecter. On peut distinguer les contremesures aux attaques invasives, passives et actives. Dans cette partie, les méthodologies générales de fonctionnement des contremesures sont présentées.

1.2.1 Protection contre les attaques invasives

Ces attaques physiques requièrent d'ouvrir le boîtier afin d'avoir accès aux interconnexions. La première contremesure est d'empêcher le désassemblage de la puce. Les boîtiers BGA (*Bold Grid Array*) peuvent rendre l'ouverture plus difficile : la puce doit être dessoudée et placée sur un adaptateur spécial (nécessite des équipements spécifiques et un personnel qualifié) [Skorobogatov 2005]. Toutefois, cette protection a un impact limité car le désassemblage peut être sous-traité ou externalisé afin de contourner cette difficulté. Le circuit est exposé à plus de lumière lorsque le boîtier est retiré. Par conséquent, un détecteur de lumière qui réinitialise ou détruit le circuit en cas de détection d'une quantité anormale de lumière peut être utilisé.

Une autre contremesure consiste à rendre plus difficile l'accès aux signaux importants, notamment en utilisant un placement routage automatique tout en aboutant les composants les uns contre les autres (*glue logic*). Le cryptage des bus de données contribue aussi à complexifier la récupération des données [Maingot 2009] Contre les techniques d'attaques par sondage (*microprobing*), l'ajout d'un bouclier (*shield*) passif, ou actif (des bits de valeurs aléatoires transitent sur le bouclier) assurent une protection supplémentaire.

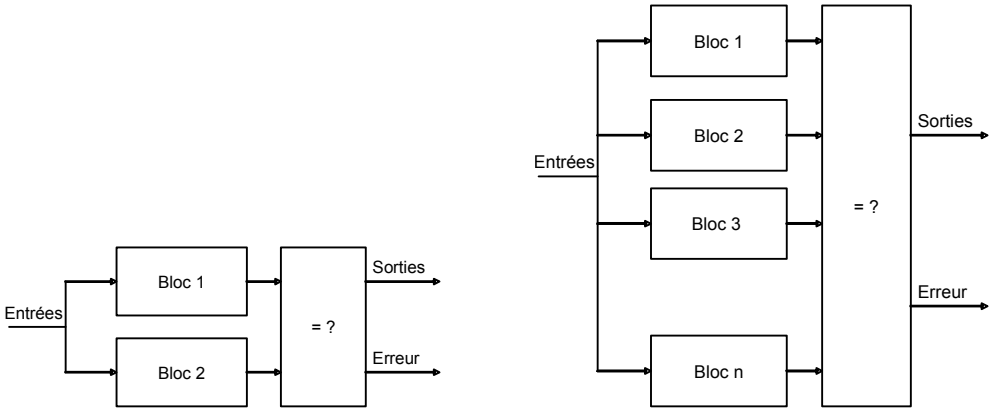
1.2.2 Protection contre les attaques actives

Les méthodes d'injection de fautes utilisent la modification de l'environnement du circuit pour le perturber. L'idée, la plus directe qui vienne à l'esprit, est d'empêcher la perturbation de cet environnement. Toutefois, il n'est pas possible d'empêcher systématiquement l'accès à tous les paramètres externes du circuit (température, tension d'alimentation, horloge, etc). En deuxième approche, il faudrait rendre robuste le circuit vis-à-vis des paramètres précédemment cités, c'est-à-dire diminuer fortement l'influence de ceux-ci sur le fonctionnement. Encore une fois, il est difficile d'atténuer complètement l'influence de ces paramètres car les sensibilités rencontrées sont propres au fonctionnement de la technologie CMOS et aux variations des procédés de fabrication. Il convient alors de chercher à détecter une faute et/ou de la corriger en utilisant des techniques appropriées.

Une méthode générique pour détecter les attaques en fautes repose sur la redondance. Le principe est de comparer des résultats provenant de deux ou plusieurs exécutions différentes afin de détecter une anomalie de calcul [Maingot 2009].

1.2.2.1 Redondance matérielle

La redondance matérielle consiste à réaliser la même opération de calcul sur deux ou plusieurs blocs et à comparer les résultats pour détecter une erreur comme l'indique la figure 1.8. À noter que s'il y a plus de trois blocs, il est possible de corriger l'erreur grâce à un vote majoritaire (figure 1.8(b)) [Bar-El 2006]. Il existe plusieurs variantes



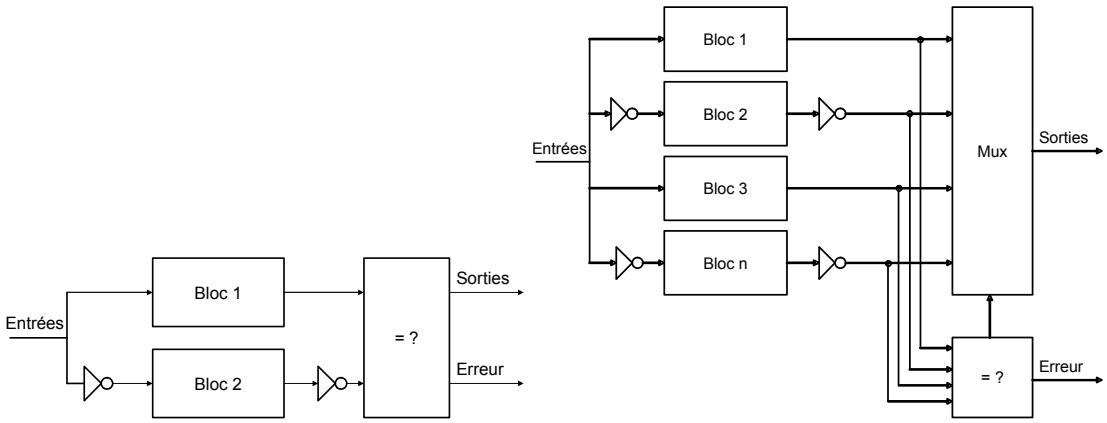
(a) Duplication simple avec comparaison (b) Duplication multiple avec comparaison

FIGURE 1.8 : Principe de la duplication simple et multiple

de duplication matérielle, faisant intervenir notamment des données complémentaires (figure 1.9(a)). Cette méthode permet une meilleure détection dans le cas où la même faute interviendrait dans les deux blocs de calcul. Une méthode de duplication hybride est présentée figure 1.9(b). En plus de la duplication multiple permettant un vote majoritaire, il est ajouté des blocs et des entrées complémentaires pour une meilleure couverture. À noter que lorsqu'un bloc fournit un mauvais résultat, il est exclu des prochains votes jusqu'à une réinitialisation du système.

1.2.2.2 Redondance temporelle

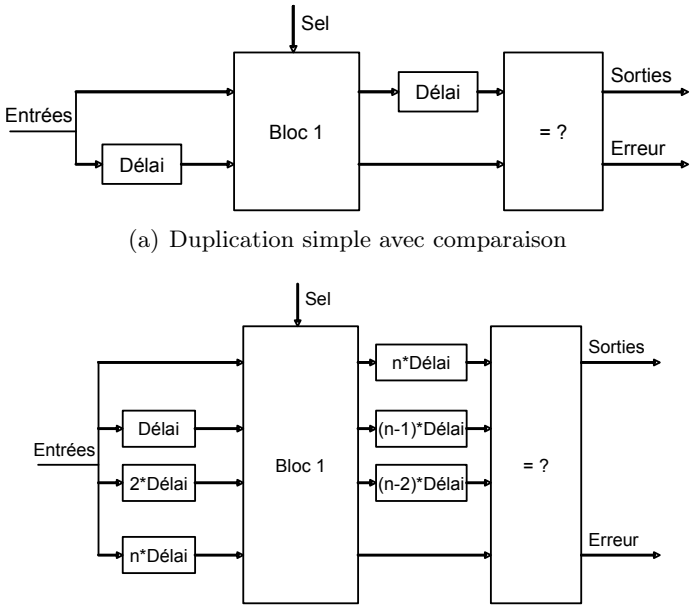
À l'image de la redondance matérielle, la redondance temporelle consiste à procéder deux fois à la même opération dans le même bloc ; mais cette fois-ci avec un décalage temporel [Bar-El 2006]. Le résultat de ces deux opérations est ensuite comparé pour détecter une erreur. La multiplication des délais de calculs entre les différentes opérations permet de détecter et aussi de corriger une erreur (figure 1.10). Plusieurs variantes de cette implantation sont possibles. Les données d'entrée peuvent subir des transforma-



(a) Duplication avec redondance complémentaire

(b) Duplication hybride

FIGURE 1.9 : Systèmes avancés de duplication



(a) Duplication simple avec comparaison

(b) Duplication multiple avec comparaison

FIGURE 1.10 : Principe de la redondance temporelle simple et multiple

tions (rotation, décalage, inversion etc.) en plus du décalage temporel avant de procéder au calcul à l'intérieur du bloc. Pour cela, le traitement réalisé à l'intérieur du bloc doit remplir certaines propriétés mathématiques pour réaliser la comparaison avec la sortie non modifiée du bloc. À noter également que les redondances temporelles et matérielles peuvent être combinées avec un surcoût à la fois en termes de performances et de surface.

1.2.2.3 Redondance d'information

La redondance d'information est une méthode communément utilisée en communication, notamment dans la transmission de l'information. L'idée est d'utiliser un codage particulier de l'information afin de savoir si la donnée a été modifiée ou non. L'information de départ est donc étendue pour y ajouter les bits de vérification qui permettent de détecter une erreur. Dès lors, deux types de code peuvent être distingués : les codes séparables et les codes non séparables. Les codes séparables contiennent deux parties : les bits correspondant à la donnée, que l'on peut qualifier de "bits utiles", et les bits de redondance qui servent à vérifier le code. Pour les codes non séparables, il n'est pas possible de distinguer ces deux parties : le mot contient à la fois les bits d'information et les bits de redondance. L'avantage des codes séparables est qu'ils permettent un décodage plus facile.

Le principe de vérification des données est présenté sur la figure 1.11. Les sorties

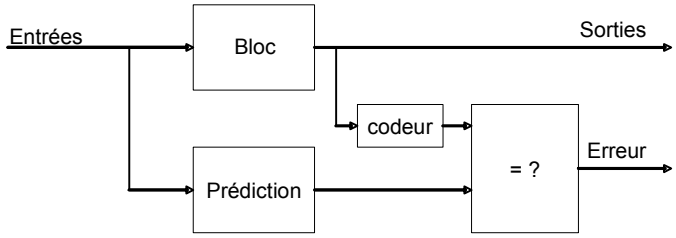


FIGURE 1.11 : Principe de fonctionnement d'un code détecteur d'erreur [Maingot 2009]

du bloc de calcul sont encodées à l'aide du bloc de codage pour obtenir les bits redondants. Le bloc de prédiction permet de calculer les bits redondants de la sortie à partir des bits redondants des entrées. Ces deux résultats sont ensuite comparés pour détecter une erreur. Le choix du code utilisé est déterminant pour l'efficacité de la protection en termes de capacité de détection, de facilité d'implantation et de coût. Dans [Maingot 2006], différents codes sont comparés en prenant en compte le pourcentage de détection, le surcoût surface, corrélation entre les données manipulées et la consommation électrique : la parité simple, la parité complémentaire, la double parité et le code de Berger. La parité complémentaire donne un bon compromis entre l'efficacité de la détection de faute et la vulnérabilité à l'analyse de consommation.

Les contremesures présentées dans cette partie sont utilisées contre plusieurs types d'attaques en fautes quelle que soit la méthode d'injection. Comme on peut le constater, ces solutions peuvent être très coûteuses en surface notamment en cas de duplications multiples. Même si elles s'avèrent efficaces, ces contremesures ne peuvent pas être utilisées dans toutes les applications à cause de leur coût additionnel en surface élevé. Il existe toutefois des solutions moins coûteuses qui sont spécifiques aux injections de

fautes par modification de fréquence d'horloge ou de tension d'alimentation. Ces méthodes seront étudiées et discutées dans le chapitre 3.

1.2.3 Protection contre les attaques par canaux auxiliaires

Les contremesures protégeant des attaques par canaux auxiliaires sont généralement utilisées dans des circuits cryptographiques. Elles peuvent être regroupées en deux catégories : les techniques de masquage et les techniques de dissimulation.

L'idée de base du masquage est de rendre aléatoires les valeurs intermédiaires des opérations effectuées par le circuit à protéger. Le fait de changer les valeurs intermédiaires va modifier par conséquent la consommation électrique qui sera différente des valeurs non masquées. Les différences de consommation pourront toujours être observées cependant elles ne sont plus liées aux valeurs sans masquage. Le masquage ne modifie pas directement la consommation électrique mais modifie les données qui elles, vont modifier la consommation.

Les solutions de dissimulation, contrairement au masquage, s'attaquent directement à la consommation électrique du circuit sans pour autant modifier les données. Le but recherché est de modifier la consommation de telle sorte qu'un attaquant puisse difficilement trouver une dépendance vis-à-vis des données. Pour cela, il est possible de concevoir le circuit de telle sorte que chaque opération requiert approximativement la même quantité d'énergie. La deuxième solution est de rendre aléatoire la consommation électrique. Cependant dans la pratique, la dépendance des données ne peut pas être complètement effacée ; il subsiste une certaine dépendance vis-à-vis des données [Mangard 2010].

Les techniques usuelles de masquage et de dissimulation sont présentées dans le chapitre 2. Il est important de noter que ces techniques par leur principe ajoutent un coût en termes de complexité, de surface et de consommation qui peut empêcher leur usage en fonction de l'application.

1.2.4 Bilan

La multiplication des attaques et des menaces sur les circuits a poussé les concepteurs à développer des contremesures pour protéger leurs circuits. Ainsi, de nombreuses solutions de protections existent, et sont étudiées au même titre que les attaques sur les circuits. Toutefois plusieurs contremesures comme les méthodes de duplication restent encore trop coûteuses pour leur intégration dans les applications où entrent en jeu les contraintes de réduction des coûts. Dans un contexte industriel, il est nécessaire de développer des solutions garantissant à la fois une facilité d'intégration, une complexité et un coût surface/consommation limité. Le but est de mettre en place des solutions de protection intégrées dans des SoCs en technologie avancée (40 nm et 28 nm). Dans

ce cadre, il n'est pas possible de dupliquer tous les blocs de calcul. Il convient donc d'identifier les vulnérabilités en fonction de leur importance pour ensuite proposer des contremesures adaptées.

Déterminer le niveau de sécurité d'un produit est essentiel pour certaines applications liées au paiement par exemple. Il existe des critères d'évaluation permettant de qualifier le niveau de sécurité d'un produit. Les critères communs (CC) font partie des normes d'évaluation des systèmes.

1.3 Exemple de critères d'évaluation : les critères communs

Les critères communs regroupent un ensemble de normes (ISO/IEC 15408) [ENISA 2005] pour la certification de systèmes d'information. Le but est de pouvoir assurer que la sécurité d'un système répond à des critères précis d'évaluation, qui ont été testés par des laboratoires certifiés. Cela prouve que le processus de spécification, de développement et d'évaluation a été conduit de manière standardisée, rigoureuse et répétable dans l'environnement de fonctionnement du système. Ces normes ont été à l'origine définies pour les systèmes informatiques, mais ont depuis été étendues aux cartes à puce et aux circuits intégrés. Les CC sont reconnus au niveau international et permettent aux concepteurs de systèmes de faire reconnaître la qualité de leurs produits dans le monde.

1.3.1 Cible de sécurité et profil de protection

La méthodologie d'évaluation des Critères Communs s'applique sur une partie des fonctionnalités d'un système, celle-ci est appelée cible de sécurité (ST : *Security Target*). Pour éviter qu'il n'y ait qu'une infime partie du système qui ne soit soumise aux tests, il a aussi été prévu la notion de Profil de Protection (PP : *Protection Profile*) [Pelkins 2007].

Un profil de protection est définissable comme des exigences de sécurité qui sont créés par un utilisateur ou une communauté d'utilisateurs, auquel tout produit doit se soumettre pour être certifié. Ainsi, le commanditaire doit rédiger une ST qui délimite le périmètre du système qui va être soumis aux tests de certification. Si le produit possède déjà un ou plusieurs profils de protection, ceux-ci seront inclus dans la cible de sécurité.

La cible de sécurité inclut les menaces auxquelles le produit en cours de certification doit se confronter, mais également son comportement pour les contrecarrer et le ou les profils de protections auxquels on se réfère. La surface d'évaluation est définie en choisissant parmi les éléments fonctionnels de la partie 2 de la méthodologie d'évaluation des CC [Criteria 2012a], ceux qui correspondent aux fonctionnalités à tester [Pelkins 2007]. Il est décidé du niveau de certification que l'on souhaite atteindre, ce qui détermine les éléments d'assurance qu'il faut inclure dans les tests. Dans ce cadre, il y a également une

description très précise de la cible d'évaluation qui sera soumise aux différents tests : la TOE (*Target Of Evaluation*).

L'ensemble des tests va porter sur la cible d'évaluation décrite dans la ST, qui doit être au moins aussi large que les profils de protections auxquels la cible se réfère.

1.3.2 Exigences d'assurance de sécurité

Un produit peut être certifié Critère Commun sur l'un des sept niveaux d'évaluation (EAL1 à EAL7 : *Evaluation Assurance Level*). EAL1 est le niveau le moins coûteux des Critères Communs et définit des tests de fonctionnement en boîte noire : on s'assure que le produit se comporte conformément à ce qui est écrit dans sa documentation. Il ne nécessite pas la présence du concepteur. Le niveau EAL2, a contrario, requiert la coopération du concepteur pour fournir les informations sur l'architecture du produit et le résultat des tests menés conformément à la description de la documentation. EAL7 est le niveau le plus élevé. Il impose des spécifications formelles c'est-à-dire que celles-ci sont exprimées dans un langage syntaxique basé sur des concepts mathématiques. Les tests sont également réalisés de manière formelle.

La méthodologie d'évaluation des CC décrit des classes d'assurance qui définissent comment sont conduits les tests sur la TOE.

1.3.3 Analyse des vulnérabilités et potentiel d'attaque

Parmi les classes d'assurance, il y a l'analyse de vulnérabilités qui traite des menaces qu'un attaquant est en mesure de découvrir et qui permettent l'accès non autorisé à des données ou à des fonctions [Criteria 2012b]. Dans [Criteria 2013], il est défini le potentiel d'attaque sur une carte à puce ou un circuit intégré selon plusieurs critères : le temps passé pour réaliser l'attaque, le niveau d'expertise de l'attaquant, la connaissance de la TOE, l'accès à la TOE, l'équipement nécessaire ainsi que la nécessité d'avoir un échantillon ouvert et la facilité d'accès à cet échantillon. Le tableau 1.1 donne le niveau de complexité des équipements qui sont utilisés pour mener une attaque en les répartissant en 3 catégories : standard, spécialisé et sur-mesure. Il est notable que les outils entrant en jeu pour réaliser des attaques non invasives sont des outils standard. La mise en œuvre de ces attaques en est donc facilitée. Le potentiel d'attaques est ensuite évalué grâce au tableau 1.2.

En additionnant les colonnes correspondant à l'identification et à l'exploitation de l'attaque on obtient la valeur du potentiel d'attaque. Cela permet de donner une évaluation de la vulnérabilité face à une attaque, comme le montre le tableau 1.3.

Pour des caractérisations sécuritaires du système, il convient de prendre d'abord en compte des attaques ayant des valeurs faibles puis d'aller vers des valeurs plus élevées de potentiel d'attaque. De ce fait, les attaques peu coûteuses à mettre en place (en temps et en équipement) et nécessitant peu de connaissances sur le système sont à privilégier.

TABLEAU 1.1 : Catégorisation des outils d'attaque [Criteria 2013]

Tools	Equipment
UV-light emitter	Standard
Flash light	Standard
Low-end visible-light microscope	Standard
Climate chamber	Standard
Voltage supply	Standard
Analogue oscilloscope	Standard
Chip card reader	Standard
PC or work station	Standard
Signal analysis software	Standard
Signal generation software	Standard
High-end visible-light microscope and camera	Specialized
UV light microscope and camera	Specialized
Micro-probe Workstation	Specialized
Laser equipment	Specialized
Signal and function processor	Specialized
High-end digital oscilloscope	Specialized
Signal analyzer	Specialized
Tools for chemical etching (wet)	Specialized
Tools for chemical etching (plasma)	Specialized
Tools for grinding	Specialized

Une grande partie des attaques non invasives peut rentrer dans cette catégorie. C'est pour ces raisons que nous allons nous focaliser sur ces attaques.

Le but sera d'une part d'évaluer qualitativement et quantitativement l'effet des attaques sur un circuit et d'autre part, de développer des contremesures et de les intégrer dans la conception globale du système. La conséquence sera d'augmenter la valeur du potentiel d'attaque sur la TOE. Pour cela, il est nécessaire de voir comment peut s'inscrire l'apport des solutions de protection dans le flot de développement du circuit avec les contraintes qui sont impliquées.

TABLEAU 1.2 : Évaluation du potentiel d'attaque [Criteria 2013]

Factors	Identification	Exploitation
Elapsed time		
< one hour	0	0
< one day	1	3
< one week	2	4
< one month	3	6
> one month	5	8
Not practical	*	*
Expertise		
Layman	0	0
Proficient	2	2
Expert	5	4
Multiple Expert	7	6
Knowledge of the TOE		
Public	0	0
Restricted	2	2
Sensitive	4	3
Critical	6	5
Very critical hardware design	9	NA
Access to TOE		
< 10 samples	0	0
< 30 samples	1	2
< 100 samples	2	4
> 100 samples	3	6
Not practical	*	*
Equipment		
None	0	0
Standard	1	2
Specialized (1)	3	4
Bespoke	5	6
Multiple Bespoke	7	8
Open samples (rated according to access to open samples)		
Public	0	NA
Restricted	2	NA
Sensitive	4	NA
Critical	6	NA

TABLEAU 1.3 : Évaluation de la vulnérabilité pour les critères communs [Criteria 2013]

Range of values	TOE resistant to attackers with attack potential of:
0-15	No rating
16-20	Basic
21-24	Enhanced-Basic
25-30	Moderate
31 and above	High

1.4 Intégration de contremesures dans le flot de conception

Les systèmes numériques actuels intègrent des fonctions de plus en plus complexes. Il a donc fallu développer des méthodes qui permettent d'uniformiser au maximum la conception de circuit, et ainsi de diminuer le temps de développement. En fonction des fonctionnalités à intégrer, plusieurs méthodes de conception peuvent être envisagées, notamment en ce qui concerne la flexibilité, les performances et le coût de développement du circuit.

1.4.1 Méthodes de conception

1.4.1.1 Utilisation de processeurs

La première solution pour réaliser un système ayant une fonction donnée est d'utiliser un microprocesseur standard. Ces types de circuit embarquent généralement de la mémoire vive (RAM) ou de la mémoire non volatile (flash ou E²PROM). Parmi ces exemples on peut citer les processeurs PIC de *Microchip*, AT91SAM de *Atmel* ou encore STM32 de *STMicroelectronics* qui offrent plusieurs types de gammes de fréquences de fonctionnement, de tailles de mémoire ou encore de signaux d'entrées/sorties de type analogiques [Microchip 2014, STMicroelectronics 2014, Atmel 2014]. L'avantage de cette méthode est qu'elle offre beaucoup de flexibilité.

1.4.1.2 Logique programmable

Il est possible que certains paramètres comme la performance ou la dissipation énergétique d'un microprocesseur ne répondent pas aux exigences d'un système donné. Il existe alors des solutions alternatives aux microprocesseurs à base de circuits programmables. Ces solutions restent plus rapides à développer que des circuits dédiés. Les FPGA (*Field Programmable Gate Arrays*) utilisent des circuits à haute densité pour concevoir des circuits intégrés qui pourront être reprogrammés au cours de leur utilisation. Une architecture simplifiée de *FPGA* est présentée sur la figure 1.12. Les blocs de logique programmable (CLB : *Configurable Logic Bloc)* permettent de réaliser les fonctions logiques. Une matrice de routage relie les différents CLB. La définition des connexions entre les blocs logiques est assurée par les interconnexions programmables. Les avantages principaux que fournissent les *FPGA* sont leur coût modéré et leur grande flexibilité. Ils peuvent servir également de prototypage avant de lancer la fabrication en grande série de circuits dédiés.

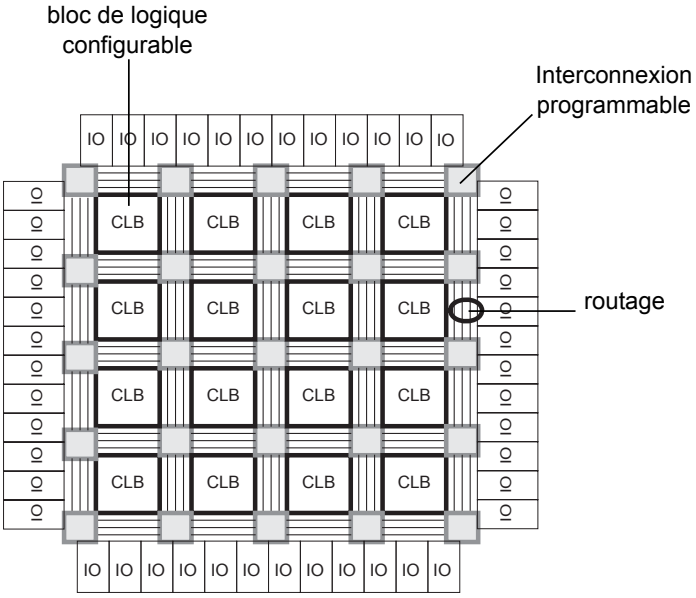


FIGURE 1.12 : Structure simplifiée d'un FPGA

1.4.1.3 Conception automatisée à base de cellules standard

La conception à base de cellules standard utilise une bibliothèque de portes logiques prédéfinies. Les cellules sont placées dans la position appropriée, et leurs interconnexions sont routées. Cette méthode de conception permet d'obtenir des circuits ayant à la fois une consommation plus faible, des performances accrues et une taille réduite par rapport aux *FPGA*. Cependant les coûts de développement sont plus importants notamment pour la production des masques [Weste 2010]. Le développement de la synthèse logique et des outils de placement et routage ont contribué à l'utilisation de cellules standard pour la conception de circuit.

Les fondeurs et les fabricants de bibliothèques fournissent des cellules avec une large gamme de fonctions et de différentes tailles. Elles peuvent inclure :

- des fonctions logiques élémentaires (portes OU, ET, NON, OU-exclusif etc.) ou complexes
- des portes logiques de mémorisation (verrous, bascules)
- des mémoires (SRAM, ROM, CAM etc.)

Généralement, les cellules standard ont une hauteur fixe, ce qui permet de les connecter directement aux rails d'alimentation V_{dd} et Gnd (figure 1.13).

Les cellules sont ainsi disposées les unes contre les autres sur plusieurs rangées en fonction de la taille du circuit. L'étape de routage permet ensuite de réaliser les connexions entre les cellules.

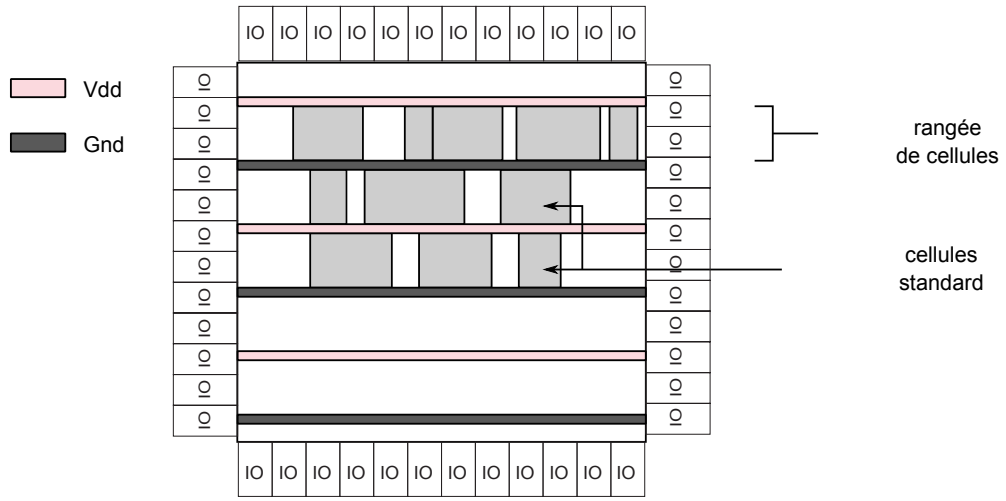


FIGURE 1.13 : Disposition des cellules standard dans un circuit

1.4.1.4 Conception sur-mesure (*full-custom*)

Comme son nom l'indique, cette méthode de conception permet de développer un circuit en ayant la possibilité de modifier chacun des transistors qui le compose. Basé sur le schéma électrique, le circuit est dessiné au niveau transistor. Cela permet de contrôler les différents paramètres des transistors de même que leur placement.

Pour des circuits numériques très haute performance et pour la plupart des circuits analogiques, la conception sur mesure est préférée [Xiu 2007]. En effet, il n'y a pas de langage de description efficace disponible actuellement pour modéliser les blocs analogiques à cause des caractéristiques inhérentes à ce type de circuits.

Ainsi, la méthode de conception sur-mesure permet de concevoir et de dessiner chaque transistor manuellement. Pour cette raison il est possible d'obtenir des circuits plus denses et potentiellement plus rapides en comparaison avec les autres méthodes de conception. Toutefois, il en résulte un temps de développement plus long et un coût plus important également.

Les solutions de conception abordées dans cette étude se basent sur l'utilisation de cellules standard. Cette méthode offre un bon compromis entre le temps et le coût de développement ainsi que de bonnes performances. Néanmoins, ce mode de conception impose un flot de développement qui est présenté dans le prochain paragraphe.

1.4.2 Contraintes imposées par le flot numérique

Le flot de conception numérique désigne l'ensemble des étapes qui permettent de passer des spécifications d'un système à sa fabrication. Ces étapes comprennent aussi les phases de vérification fonctionnelle. La figure 1.14 donne les différentes étapes de la

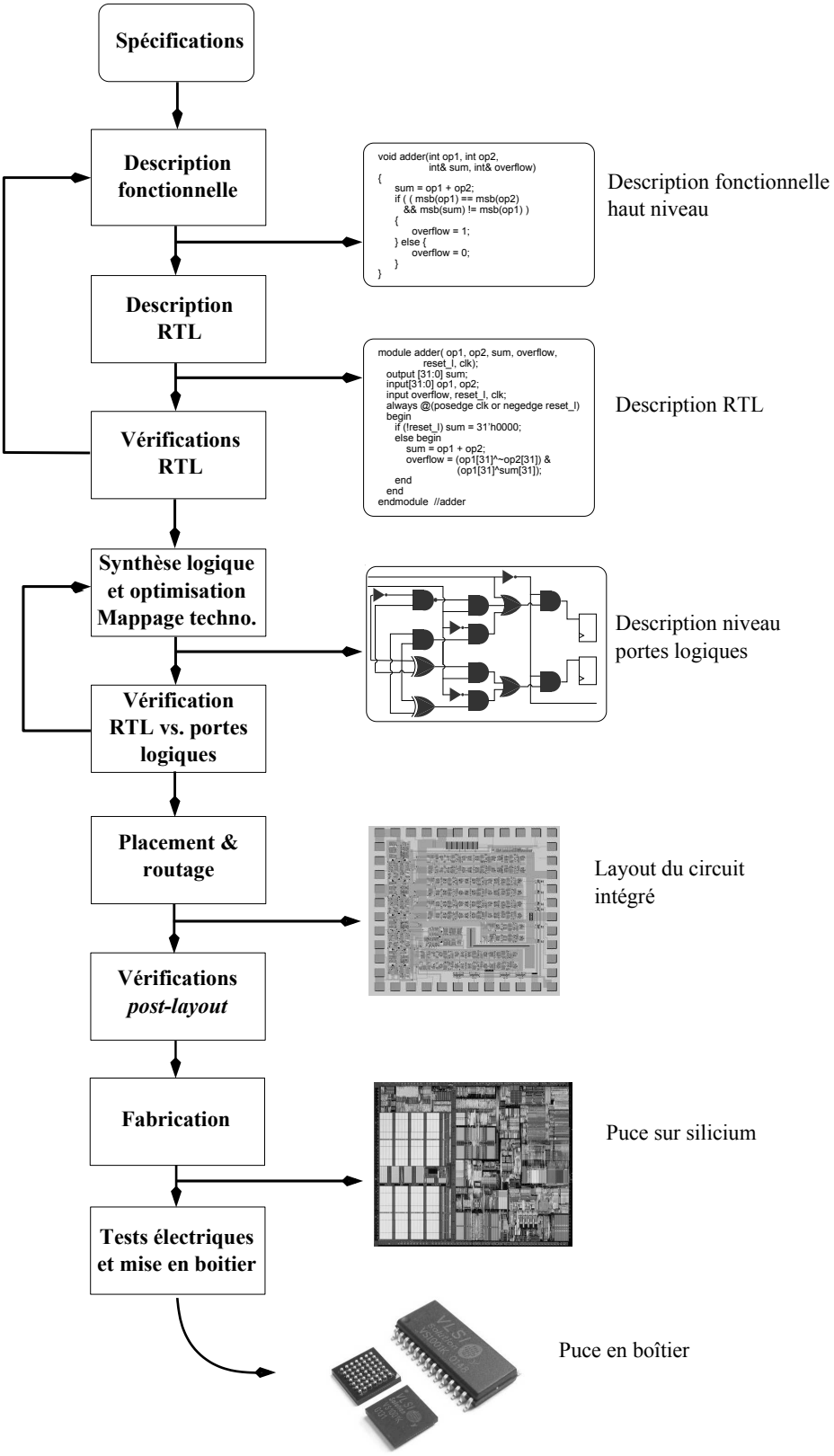


FIGURE 1.14 : Flot de conception numérique

conception d'un système numérique. Les données de départ sont fournies par les spécifications du système à concevoir. Les spécifications regroupent les différentes fonctionnalités du produit, les conditions de fonctionnement, les performances (vitesse, consommation etc.). Certaines contremesures peuvent être confondues avec des spécifications fonctionnelles, par exemple en termes de consommation afin de réduire la signature électrique. Sinon, les contremesures possèdent leurs propres spécifications, en termes de fonctionnalité ou de capacité de détection d'erreurs par exemple. La difficulté réside dans le fait d'anticiper tous les types d'attaques auxquelles le système doit faire face, et donc de prévoir et d'intégrer l'ensemble des contremesures nécessaires, notamment pour une question de coût final de l'application qui doit rester limité. Dans ce cadre, les critères communs peuvent apporter une solution en termes de classification d'attaques et d'analyse de vulnérabilité afin de définir les principales attaques à couvrir.

Une fois que les spécifications sont définies, il est possible de décrire les fonctionnalités avec un haut niveau d'abstraction. Cela peut se faire à l'aide d'un langage de programmation par exemple. L'étape suivante consiste à définir les fonctionnalités avec un modèle comportemental. Ce modèle est décrit à l'aide d'un langage de description matériel (HDL : *Hardware Description Language*) comme le VHDL ou le verilog. Lorsque les fonctionnalités sont décrites, une première étape de vérification consiste à réaliser des simulations fonctionnelles pour valider la description comportementale (validation RTL : *Register Transfer Level*).

Lorsque la description RTL est vérifiée, l'étape de synthèse logique permet de transformer la description fonctionnelle au niveau RTL en implantation matérielle à l'aide de portes logiques. Cette étape inclut l'optimisation de la logique pour atteindre les performances souhaitées en termes de vitesse et de surface en utilisant les portes présentes dans la bibliothèque de cellules standard. Durant cette phase, des contraintes de vitesse de fonctionnement sont introduites pour veiller à ce que le circuit fonctionne à une certaine fréquence conformément aux spécifications. Ces contraintes peuvent avoir des impacts sur le nombre de cellules utilisées et donc sur la surface totale du circuit. Il s'agit de trouver le compromis qui répond aux spécifications du système.

À la fin de la synthèse, le circuit est défini comme un ensemble de fonctions logiques élémentaires (ET, OU, NON, OU exclusif etc.) et d'éléments de mémorisation (de type bascules et verrous).

À ce stade, il faut faire particulièrement attention car les outils de synthèse sont optimisés pour simplifier au maximum les fonctions logiques. Par conséquent, les fonctions sécuritaires de redondance ajoutées durant la description RTL peuvent être supprimées dans le cadre d'une optimisation. C'est pourquoi l'ensemble des vérifications fonctionnelles doit tenir compte de ce phénomène afin de relancer l'étape de synthèse pour corriger ce problème si nécessaire (avec la modification de la description RTL

correspondante). Il existe plusieurs outils de synthèse logique, parmi lesquelles *Design Compiler* de Synopsys ou *RTL Compiler* de Cadence.

Le modèle synthétisé doit maintenant être vérifié. L'objectif de cette vérification est de garantir l'équivalence entre la description faite au niveau RTL du circuit et de la description au niveau portes logiques. Si la description a été correctement réalisée, il ne doit pas y avoir d'erreurs introduites pendant la phase de synthèse logique.

Une des stratégies de vérification consiste à faire tourner les bancs de test définis pour la description RTL, et à vérifier que les mêmes signaux de sortie sont produits par le modèle comportemental et la description niveau portes logiques. Une autre stratégie utilise un logiciel de vérification formelle qui compare l'équivalence logique des deux descriptions. Les outils de vérification formelle se développent progressivement, ils permettent de démontrer mathématiquement que deux descriptions ont exactement la même fonction booléenne. En comparaison, les méthodes par simulation permettent uniquement de valider les fonctions pour les vecteurs d'entrée choisis. Des outils comme *Conformal Equivalence Checker* de Cadence, *Formality* de Synopsys ou encore *Formal-Pro* de Mentor Graphics peuvent être utilisés pour faire de la vérification formelle.

À cette étape, il peut être nécessaire de réaliser une analyse de timing, une notion qui ne rentrait pas en compte dans la description comportementale. Elle peut se faire avec un simulateur : les temps de propagation réels des cellules sont utilisés pour vérifier les différentes fonctionnalités du circuit, méthode dite de rétro-annotation (*back-annotation*). D'autre part, une méthode plus rapide permet de dresser la liste complète des chemins entre les entrées et les sorties des portes ainsi que les temps de propagation qui leur sont associés. Cette méthode est appelée analyse statique de timing (STA : *Static Timing Analysis*). L'outil *ETS* développé par Cadence et PrimeTime de Synopsys sont des exemples d'analyseurs de timing.

À noter que c'est dans la phase de synthèse que sont insérés les éléments d'aide à la testabilité. Deux techniques peuvent être utilisées : la première est l'insertion des bascules observables qui peuvent être configurés dans un état donné (*scannable registers*). La deuxième appelée test intégré in-situ (BIST : *Built-In Self Test*) modifie les registres pour autoriser des tests à l'intérieur du circuit. La technique généralement utilisée dans les circuits numériques est l'utilisation de chaînes de scan (*scan chains*) car elles permettent une meilleure observabilité de l'ensemble des bascules du circuit. La technique permet notamment de mettre toutes les sorties des registres dans un état donné ou de récupérer les valeurs présentes à la sortie à un instant donné. Cela peut évidemment poser des problèmes de sécurité, sachant que l'accès aux données du système est possible à tout moment. Il existe des solutions pour protéger l'accès aux chaines de scan comme l'insertion de logique entre les registres, la déconnexion des chaines après fabrication, ou encore le brouillage (*scrambling*) du chaînage [Hely 2006, Joaquim Da Rolt 2012].

Des outils comme *Encounter True-Time ATPG* de Cadence ou *Tetramax* de Synopsys permettent de générer les vecteurs de test pour les chaînes de scan.

L'étape finale d'implantation d'un circuit consiste à transformer la description structurelle en une implantation physique. Une phase manuelle de disposition préalable des modules communicants entre eux peut être requise pour une optimisation du placement (*floorplanning*). Les cellules qui constituent le circuit sont ensuite placées de manière automatique par l'outil de placement routage. L'automatisation de cette implantation est rendue possible grâce à l'utilisation de cellules de hauteur constante, ce qui permet de les disposer les unes à côté des autres dans des rangées comme le montre la figure 1.13. L'algorithme de placement cherche alors à optimiser les temps de propagation à l'intérieur du circuit.

Après le placement des cellules les différents signaux doivent être routés. Il s'agit d'utiliser différents niveaux de métaux pour connecter les ports des cellules entre eux. Dans la plupart des cas, il est suffisant de minimiser la longueur totale des fils utilisés pour réaliser les connexions. Il faut ensuite s'assurer que les contraintes temporelles de fonctionnement sont respectées. Le routage est généralement divisé en deux phases : le routage global qui associe une liste de régions de circuit à chaque fil sans définir la géométrie exacte de ce fil, et le routage détaillé où la géométrie du fil dans la région qui lui est définie est trouvée [Xiu 2007]. *Encounter* de Cadence ou *IC compiler* de Synopsys sont des exemples d'outils commerciaux de placement routage.

Avant la fabrication, il y a une dernière étape de vérification, qui regroupe à la fois les vérifications physiques et fonctionnelles. Pour assurer que le dessin de masques est correct, il existe un ensemble de règles qui est propre à chaque fondeur de circuits intégrés. Le dessin des masques (*layout*) du circuit doit répondre à ces règles pour éviter des erreurs de fabrication. L'ensemble des règles à respecter est consigné dans un document appelé DRM (*Design Rule Manual*) et cette étape de vérification est appelée DRC (*Design Rule Checking*). Une deuxième vérification physique permet de s'assurer que les connexions, les fonctionnalités des composants ainsi que leur taille, telles que définies au niveau portes logiques sont les mêmes que celles du dessin des masques, il s'agit du LVS (*Layout Vs. Schematic*).

Une fois l'étape de placement routage terminée, il faut extraire les éléments parasites du circuit qui vont sensiblement modifier ses performances. Les éléments extraits sont généralement des capacités et des résistances dues au routage, mais peuvent contenir également des inductances. Un fichier contenant les éléments parasites associés à chaque fil dans le dessin des masques est obtenu. Après la phase d'extraction des parasites, on retrouve toutes les étapes de vérification fonctionnelle effectuées après la synthèse logique, comme l'analyse de timing, mais aussi l'analyse de consommation électrique, toujours pour vérifier la conformité des spécifications du circuit.

Le circuit ayant passé toutes les vérifications, la base de données associée peut être

envoyée en fonderie pour fabrication, généralement au format GDSII ou plus récemment au format OASIS [White 2013]. Les derniers tests se feront alors sur les plaquettes de silicium avant la mise en boîtier.

L'insertion de contremesures intervient dans le flot de développement principalement à trois étapes. La première est dans la description fonctionnelle au niveau RTL. Les contremesures qui nécessitent de modifier l'architecture du système (changement de la taille des données, modification des chemins de données ou de contrôle, etc.) doivent être intégrées dans la description comportementale du système. On peut citer par exemple les contremesures de masquage qui requièrent un chemin de données particulier, ou la redondance d'information qui augmente la taille des données en intégrant une partie redondante. Ces modifications interviennent dès le début du flot et suivent donc exactement les mêmes procédures de vérification.

La deuxième étape qui introduit des modifications est la synthèse logique. Pour la fonctionnalité de certaines contremesures, il peut être nécessaire d'appliquer des contraintes de temps de propagation sur l'ensemble des chemins du circuit. Ces contraintes sont également appliquées pour le placement routage. Cette approche sera détaillée dans le chapitre 3.

Enfin, la troisième étape est celle de l'implantation physique (placement et routage). Il est possible d'intégrer des cellules particulières de la bibliothèque de composants standard comme des capacités de découplage qui vont apporter des fonctionnalités supplémentaires. L'étude de ce type de contremesures est abordée dans le chapitre 2. Certaines contremesures peuvent également nécessiter une disposition particulière des cellules afin d'assurer une fonctionnalité. Le flot permet de contraindre le placement d'une ou d'un ensemble de cellules selon les besoins du concepteur du circuit. Ce type de contraintes s'avère intéressant notamment pour des solutions à base de redondance, les fonctions dupliquées nécessitant d'être séparées spatialement.

Il est important de faire toutes les étapes de vérification lorsque les contremesures sont insérées dans un circuit existant car les performances requises en termes de vitesse et de consommation peuvent ne plus être atteintes. Cela est d'autant plus important lorsque ces solutions doivent être intégrées dans des produits dans le cadre d'un développement industriel.

1.4.3 Contraintes industrielles de développement

Avec les menaces qui se font persistantes sur les circuits présents sur le marché, les concepteurs doivent être en mesure de proposer rapidement des solutions adaptées à l'évolution de ces attaques. Les améliorations passent par l'intégration de contremesures, qui doivent avoir un impact limité sur le coût final du système. De plus, la tendance aujourd'hui est à la réduction du temps entre la conception d'un produit et sa mise sur le marché (*time to market*). Cela doit inclure bien évidemment l'ensemble des tests

fonctionnels une fois le circuit fabriqué. Toutes ces contraintes doivent être prises en compte dans les solutions de protection à adopter pour un système donné.

En effet, le cycle de développement du système protégé doit être court, ce qui implique l'utilisation de contremesures peu complexes à développer. D'autre part, il faut s'assurer que les protections apportées soient les mêmes d'un circuit à l'autre (critère de reproductibilité). À cela, s'ajoute un autre critère important qui est la testabilité des contremesures. Il est possible de réaliser des mesures pour les attaques passives, néanmoins celles-ci peuvent être coûteuses en temps de test. De même, les tests de fonctionnalité des protections contre les attaques actives peuvent d'une part s'avérer longs à effectuer (toutes les configurations possibles) et d'autre part, certains types de fautes peuvent endommager ou faire vieillir le circuit. Ces considérations sont à prendre en compte pour le choix d'une contremesure adaptée à un type d'attaque.

Une fois que les fonctions testées sont conformes aux attentes, il faut s'interroger aussi sur l'évolution du comportement global du circuit dans le temps. Si l'exemple d'un détecteur d'erreurs est considéré, il est important de savoir comment le ou les seuils de détection vont changer au cours du temps et si ceux-ci vont dégrader les performances du circuit en se déclenchant trop tôt. Pour avoir une dégradation uniforme, il faudrait veiller à ce que l'ensemble du système soit utilisé en même temps, ce qui éviterait un déséquilibre dans le vieillissement des composants.

1.5 Conclusion

Cette partie a rappelé les vulnérabilités et les menaces qui pèsent sur les systèmes actuels. La multiplication des méthodes d'attaque rend difficile la conception de systèmes entièrement robustes. Une approche cohérente serait de s'intéresser d'abord aux attaques les moins coûteuses à mettre en place, puis d'aller progressivement vers des attaques nécessitant des connaissances plus fines et du matériel plus onéreux. La méthodologie d'évaluation des critères communs fournit des informations permettant de calculer la résistance d'un système face à une menace en définissant le potentiel d'attaque. Ces critères orientent en premier lieu vers les attaques non invasives parce qu'elles utilisent des équipements standard et qu'elles n'exigent que peu de connaissances sur l'implantation du circuit à attaquer.

Il existe dans la littérature des méthodes de protection qui peuvent s'avérer efficaces contre les attaques par canaux auxiliaires mais aussi contre les attaques actives. Néanmoins, il faut considérer les contraintes dues à l'utilisation du flot numérique : celui-ci limite le type de cellules utilisables (uniquement celles définies dans la bibliothèque de composants), le placement et routage des cellules sont automatisés etc. À cela s'ajoutent la complexité d'intégration, le coût surface ainsi que l'impact sur les performances du système qui doivent être limités dans les applications à coût modéré. Elles doivent malgré tout être robustes vis-à-vis des attaques non invasives. Il faut donc être

en mesure de proposer des solutions de protection à bas coût et applicables facilement dans un cadre industriel. Le but est de pouvoir implanter ces solutions dans un SoC, dans des nœuds technologiques avancés en particulier 40 nm et 28 nm. C'est dans ce cadre qu'interviennent les contraintes liées au coût global du système.

Les études qui vont suivre s'intéressent aux attaques non invasives et plus particulièrement visent la tension d'alimentation des circuits. Il s'agit d'une entrée primaire de tout circuit qui s'avère accessible pour un attaquant. Ce vecteur permet à la fois d'observer le comportement d'un système mais aussi de modifier celui-ci en y induisant des fautes. Deux objectifs principaux sont identifiés. Le premier consiste à analyser les mécanismes qui permettent les attaques par le biais de la tension d'alimentation dans le cadre d'une *IP* sécurisée dans un *SoC* ou dans un *SoC* complet en technologie avancée. Le second est de trouver des contremesures matérielles efficaces qui puissent être intégrées dans la conception d'une *IP* ou d'un *SoC* en suivant le flot de conception numérique.

Le chapitre suivant s'articule autour de la consommation électrique à travers la tension d'alimentation. Le but est de pouvoir analyser la signature électrique en phase de conception et de proposer des solutions contre les attaques par analyse de consommation peu coûteuses dans un souci d'intégration et de développement industriel.

Chapitre 2

Attaques passives en tension

Sommaire

La première partie de ce travail s'intéresse à la modélisation de la consommation électrique d'un circuit numérique synchrone. Cette analyse a pour but de déterminer la quantité d'information qu'on est capable de recueillir en analysant la signature de la consommation électrique d'un circuit en fonction du temps. L'approche vise donc à évaluer la robustesse d'un circuit face à des attaques de type canaux auxiliaires autour de la tension d'alimentation. Pour mieux anticiper ces fuites d'information, il faut introduire en avance de phase des contraintes permettant de les minimiser. Cela passe par la prise en compte de la signature de consommation dès la phase de spécification et également l'intégration de mesures de protection.

Nous allons voir comment la consommation est évaluée dans le flot de conception standard, puis les différentes composantes de la consommation électrique ainsi que les paramètres qui la modifient. Dans un deuxième temps, on s'intéressera à la signature de la consommation électrique c'est-à-dire les traces telles qu'elles seront recueillies par une personne effectuant des mesures extérieures. Un modèle est ainsi développé pour obtenir une telle signature électrique. Par la suite, ce modèle sera utilisé pour étudier l'impact de contremesures sur les attaques par analyse de consommation. La dernière section conclura ce chapitre en donnant des perspectives pour les prochaines études à mener.

2.1 Consommation des circuits

Les évolutions technologiques ont pour objectif de développer des circuits de plus en plus performants. Ce gain de performance s'accompagne aussi d'une optimisation de la consommation électrique des circuits. Pour y parvenir, il est nécessaire de savoir comment calculer la puissance consommée et de connaitre les paramètres intervenant dans ce calcul.

2.1.1 Étude théorique de la consommation d'un circuit numérique

2.1.1.1 Différents types de consommation

Les mécanismes de dissipation de puissance sont généralement divisés en deux catégories : la dissipation statique et la dissipation dynamique. La consommation dynamique est uniquement présente lorsque le circuit est opérationnel (transition des signaux) tandis que la consommation statique est prépondérante lorsque le circuit est inactif ou en veille.

Puissance statique. Un composant CMOS idéal ne présente aucune puissance statique dissipée en l'absence de transitions des signaux d'entrée. C'est un des avantages de la logique CMOS. Toutefois, pour les systèmes réels, il subsiste un courant qui circule entre le nœud d'alimentation et la masse. Il s'agit du courant de fuite. Ce courant prend

une place de plus en plus importante dans les nœuds technologiques avancés notamment en 40 nm et en deçà. En effet, pour parvenir à des circuits de plus en plus denses et performants, la réduction de la taille des composants s'est accompagnée de celle de la tension de seuil des transistors. Pour des raisons de fiabilité, la tension d'alimentation a été également abaissée.

Cela a pour conséquence une réduction de la puissance consommée mais aussi une augmentation du courant de fuite [Roy 2003]. Pour les composants nanométriques, le courant de fuite est dominé par la fuite en dessous du seuil (I_{sub}), la fuite grille-oxyde par effet tunnel I_G et la fuite de la jonction PN polarisée en inverse (I_D) [Abdollahi 2004]. Ces trois mécanismes de courants de fuite sont illustrés sur la figure 2.1. Le courant

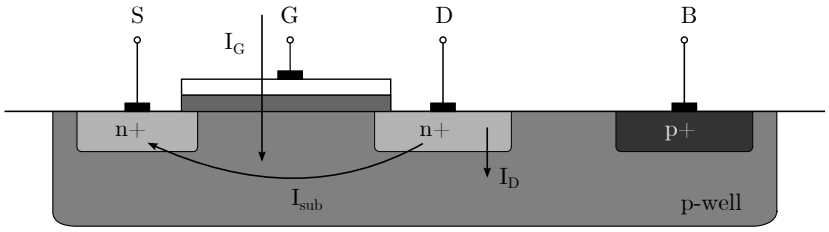


FIGURE 2.1 : Illustration des courants de fuite sur un transistor NMOS

statique $I_{statique}$ est défini comme la somme des courants de fuite :

$$I_{statique} = I_{sub} + I_G + I_D \tag{2.1}$$

Un circuit dissipe donc une certaine puissance en l'absence de transitions des signaux de sortie. Le circuit est alors polarisé dans un état donné. Cette puissance statique dissipée est également appelée puissance de fuite. La puissance statique et le courant de fuite sont liés par la relation :

$$P_{statique} = I_{statique} \cdot V_{DD}. \tag{2.2}$$

$P_{statique}$ est la puissance statique dissipée par le circuit, $I_{statique}$ le courant de fuite et V_{DD} la tension d'alimentation du circuit.

Puissance dynamique. Elle est due au courant qui circule lorsque les transistors transitent d'un état passant à un état bloqué ou vice-versa. Deux effets sont ici mis en jeu :

- la charge et la décharge des capacités associées aux nœuds internes des composants (courant de transition)
- un courant qui circule entre l'alimentation et la masse quand les transistors de type P et de type N sont simultanément passants. Lorsqu'une transition intervient sur le signal d'entrée V_{in}, pendant un court instant les transistors P et N sont passants. Ce courant dépend notamment de la pente du signal d'entrée V_{in}.

Ces deux courants sont représentés sur la figure 2.2, i_{tr} étant le courant de transition et i_{sc} le courant de court-circuit. La fréquence à laquelle se font les transitions, les temps de montée et de descente des signaux d'entrée ont des effets directs sur les pics de courant enregistrés. La puissance moyenne dissipée (P_{dyn}) lors d'une transition du

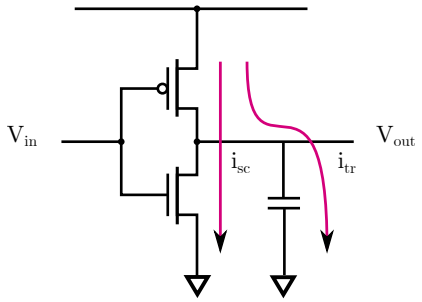


FIGURE 2.2 : Courant de court-circuit et de transition dans un inverseur CMOS

signal de sortie est donnée par la formule [Chandrakasan 1995]

$$\begin{aligned} P_{dyn} &= P_{tr} + P_{sc} \\ &= \alpha \cdot C_L \cdot V_{DD}^2 \cdot f_{clk} + i_{sc} \cdot V_{DD} \end{aligned} \tag{2.3}$$

P_{tr} et P_{sc} sont respectivement la puissance de transition et la puissance dissipée due au court-circuit, C_L la capacité de charge de l'étage de sortie, f_{clk} la fréquence de fonctionnement du circuit et V_{DD} la tension d'alimentation. Le paramètre α appelé facteur d'activité, représente le nombre moyen de transitions de l'état bas vers l'état haut sur un nœud pendant une période. C'est lors de cette transition que le circuit consomme du courant depuis la source d'alimentation. Le signal d'horloge par exemple a un facteur d'activité de 1 car les deux transitions ont lieu à chaque cycle.

La puissance totale consommée par un circuit est donc définie par :

$$P_{totale} = P_{dyn} + P_{statique} \tag{2.4}$$

À noter que si la puissance dynamique représentait la majeure partie de la puissance dissipée, la part de la puissance statique tend à augmenter avec la réduction de la taille des composants. Sur la figure 2.3 est représentée la densité de puissance en fonction de la longueur de grille des composants CMOS. On peut remarquer que pour les technologies autour de 30 nm et en dessous, la puissance statique devient aussi importante que la puissance dynamique. Cette observation est importante pour la réalisation de circuits consommant peu d'énergie. On va s'intéresser maintenant aux paramètres physiques qui modifient la consommation d'énergie.

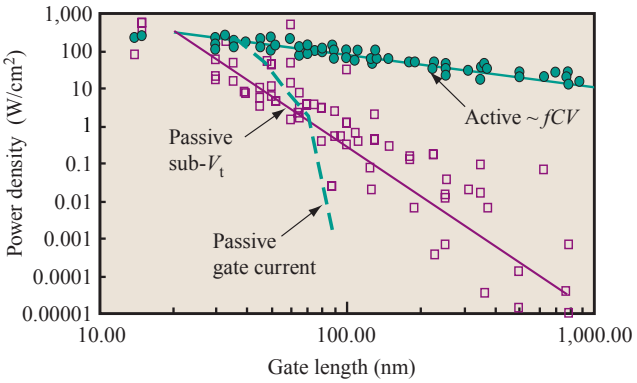


FIGURE 2.3 : Densité de puissance des composants CMOS en fonction de la taille de grille [Haensch 2006]

2.1.1.2 Influence des paramètres physiques

Capacités et résistances parasites. Les capacités dans les circuits CMOS sont présentes à plusieurs niveaux : les capacités dues au routage, les capacités parasites entre les différents niveaux de métaux et également les capacités internes aux transistors.

Comme vu précédemment avec l'équation 2.3, les capacités internes interviennent dans l'expression de la puissance dynamique. Plus les capacités sont importantes, plus la puissance consommée augmente.

Les lignes de métaux permettant de connecter l'alimentation des composants ont une certaine résistivité. Lorsque le courant traverse ces lignes de métaux, il y a donc une tension non nulle qui s'établit à leurs bornes. Le potentiel développé fait donc chuter la tension qui arrive au niveau des composants. L'*IR-drop* représente la chute de tension sur les rails d'alimentation qui est due à la résistance des lignes entre le point d'arrivée de la tension et le nœud considéré dans le circuit [Saxena 2003]. Cette différence de potentiel peut entraîner des fluctuations de temps de propagation et dans la transition des signaux, et ainsi affecter les performances du circuit et augmenter la puissance consommée [Lin 2001].

Paramètres de fabrication. Les transistors MOS possèdent trois zones de fonctionnement qui dépendent de la tension appliquée à leurs bornes :

- la zone bloquée où le courant circulant dans le transistor est égal au courant de fuite 2.1.
- la zone linéaire qui correspond à la formation du canal : le courant entre le drain et la source I_{DS} augmente avec la tension V_{DS}, définie par l'équation :

$$I_{DS_{lin}} = \mu_{0,n} C_{ox} \frac{W}{L} (V_{GS} - V_t - V_{DS}/2) V_{DS}, \tag{2.5}$$

  pour un transistor de type N.

– la zone de saturation où on assiste à un pincement du canal, le courant I_{DS} ne dépend plus de la tension V_{DS} :

$$I_{DS_{sat}} = \mu_{0,n} C_{ox} \frac{W}{L} (V_{GS} - V_t)^2. \tag{2.6}$$

Dans ces équations, $\mu_{0,n}$ représente la mobilité des électrons, C_{ox} la capacité de grille du transistor, W et L respectivement la largeur et la longueur de la grille, et V_t la tension de seuil. On peut constater que l'augmentation de la largeur ou la diminution la longueur de grille a pour conséquence une augmentation du courant I_{DS} et donc une augmentation de la consommation.

L'autre paramètre intervenant dans l'équation donnant le courant à travers le transistor est la tension de seuil V_t. Cette tension dépend des paramètres technologiques de fabrication et également de la tension entre source et le substrat du transistor (V_{BS}). La tension de seuil peut être modélisée selon l'équation [Sedra 2010] :

$$V_t = V_{t0} + \gamma \left(\sqrt{\phi_s + V_{SB}} - \sqrt{\phi_s} \right) \tag{2.7}$$

V_{t0} est la tension de seuil lorsque la source est au potentiel du substrat, ϕ_s est un paramètre physique qui dépend du niveau de dopage et γ un paramètre de fabrication donné par

$$\gamma = \frac{\sqrt{2qN_A\epsilon_s}}{C_{ox}} \tag{2.8}$$

où q est la charge de l'électron, N_A est la concentration de dopage du substrat de type P et ϵ_s la permittivité du silicium. Le choix des paramètres de fabrication va donc directement influencer la quantité de courant traversant les transistors et ainsi modifier la consommation électrique. Pour des circuits consommant très peu d'énergie, l'augmentation de la tension de seuil aura pour effet de diminuer le courant tiré par le transistor et donc de réduire la consommation. Inversement, avec une tension de seuil plus faible, le courant I_{DS} augmente, ce qui permet d'obtenir des circuits plus performants. Par la suite, nous verrons comment évaluer la consommation globale au niveau d'un circuit numérique.

2.1.2 Évaluation de la consommation

La puissance consommée par un circuit est la somme des puissances consommées par chacun des transistors du circuit. Ainsi, pour obtenir la consommation totale, il faudrait obtenir les courants dynamiques et les courants de fuite de tous les transistors présents dans le circuit. S'il est envisageable d'obtenir ces courants par simulation électrique pour quelques dizaines de transistors, il est difficile voire impossible de simuler électriquement un circuit numérique comprenant plusieurs dizaines de milliers de transistors. La solution du flot numérique consiste à se placer au niveau des portes logiques. On va alors utiliser les caractérisations individuelles des portes logiques implantées dans

le circuit. Grâce aux caractérisations des cellules dans un état polarisé et en présence de transitions, il est possible d'analyser les informations de consommation qui nous intéressent.

Dans un premier temps, nous présenterons les différentes analyses de puissance qui sont utilisées. Ensuite, nous verrons comment générer les informations de puissance consommée par le circuit.

2.1.2.1 Types d'analyse

Analyse de la consommation moyenne. Le premier type d'analyse consiste à calculer la consommation moyenne du circuit sur un intervalle de temps donné (analyse statique). La probabilité pour chaque événement ainsi que la contribution de chacun d'entre eux dans le calcul de la puissance moyenne des portes logiques sont estimés. L'analyse requiert un fichier qui comporte l'activité du circuit. Généralement, des fichiers de type SAIF sont utilisés pour cette analyse. Il s'agit d'un format développé par *Synopsys* mais qui est supporté et utilisé par plusieurs outils d'aide à la conception de circuits. Ces fichiers contiennent le nombre de transitions sur chaque signal du circuit, ainsi que la durée pendant laquelle celui-ci se trouve dans un état logique haut ou bas, dans un état indéterminé ou en haute impédance [Xil 2009]. La génération du fichier SAIF se fait lors de l'étape de simulation fonctionnelle logique du circuit en cours de test. Ce type d'analyse privilégie la performance à la précision.

Analyse de la consommation en fonction du temps. Le deuxième type d'analyse permet d'évaluer la puissance consommée en fonction du temps. Pour chaque événement, l'énergie correspondante est obtenue à partir de la bibliothèque de caractérisation des portes logiques et permet de construire la courbe de puissance consommée. Cette analyse nécessite un type de fichier différent de l'analyse statique : ici il s'agit d'un fichier VCD. Le format VCD est un standard défini par la norme IEEE Standard 1364-1995 [IEEE 1996]. Ce fichier contient les événements intervenant sur les signaux à chaque étape de la simulation. Il est généré à l'étape de simulation fonctionnelle tout comme le fichier SAIF. Le fichier VCD donne l'information sur la valeur d'un signal à un instant donné et également le moment où une transition s'effectue sur ce signal. À titre de comparaison, cette dernière information est absente des fichiers SAIF, qui ne contiennent qu'une information cumulée des fichiers VCD ce qui explique aussi que les fichiers SAIF soient moins volumineux que les VCD. Contrairement à l'analyse statique, la précision est le critère le plus important ici.

Pour notre étude, on a besoin de la puissance consommée en fonction du temps, c'est donc l'analyse dynamique qui sera utilisée par la suite.

2.1.2.2 Flot d'évaluation

Bibliothèque de caractérisation des cellules. Comme indiqué dans le paragraphe 2.1.2, les informations de consommation d'un circuit dans le flot numérique sont obtenues à partir de la caractérisation des portes logiques. Ces informations sont stockées dans des fichiers qui constituent la bibliothèque de caractérisation des cellules d'une technologie donnée (fichiers *liberty*). Le format de stockage sous forme de fichiers .lib (*liberty*) est un standard utilisé par de nombreux fournisseurs de composants semiconducteurs [ope 2014]. Chaque fichier contient les informations sur les caractéristiques et les fonctions des cellules standard fournies dans la bibliothèque. Parmi les caractéristiques, on retrouve le nom de la cellule, les entrées et les sorties de la cellule ainsi que leur charge, sa surface et les chemins depuis les ports d'entrées vers les ports de sortie sur lesquels sont définis les temps de propagation et l'énergie consommée [Ope 2013]. Pour les temps de propagation, les délais d'une entrée à une sortie et la pente du signal de sortie correspondant sont caractérisés pour des arcs donnés comme une fonction de la charge et de la pente du signal d'entrée. De la même façon, on caractérise la puissance dynamique pour un arc en fonction de la transition du signal d'entrée de la charge du signal de sortie. La consommation est alors modélisée sous forme d'un tableau à trois dimensions comme représenté sur la figure 2.4. La puissance statique quant à elle, ne

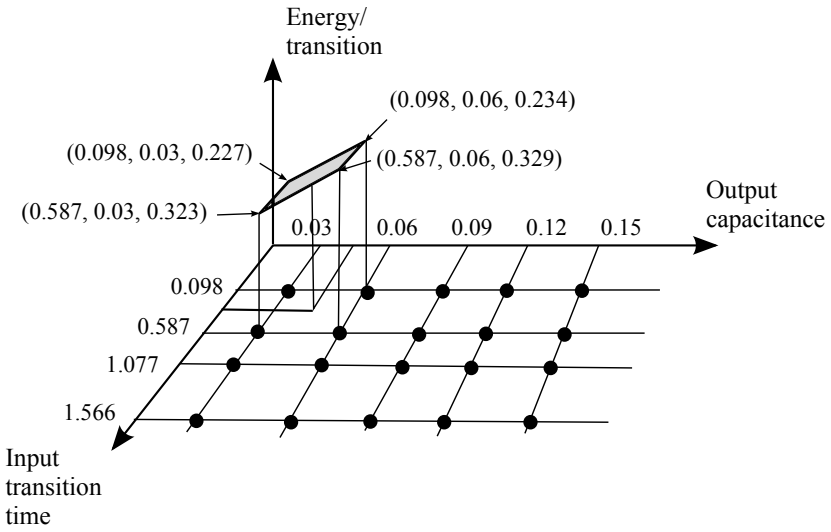


FIGURE 2.4 : Table de puissance dynamique de la sortie d'une cellule [Ope 2013]

dépend que de la polarisation de la cellule.

Génération de courbe de puissance. L'analyse dynamique de la puissance consommée par un circuit numérique utilise les données des fichiers *liberty*. Le flot de génération de la courbe de puissance s'appuie sur l'activité du circuit au cours du temps. Ce flot

est décrit sur la figure 2.5. L'outil utilisé pour l'analyse est *PrimeTime PX* développé

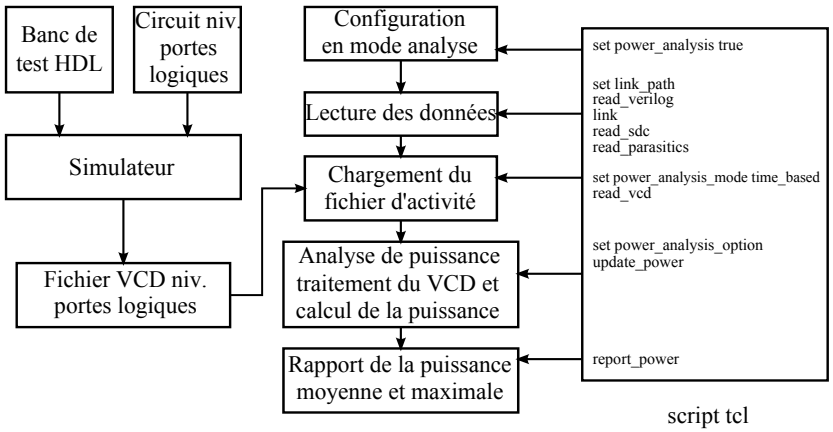


FIGURE 2.5 : Flot d'analyse dynamique de puissance

par Synopsys. La première étape consiste à générer le fichier d'activité (fichier VCD) nécessaire à l'analyse. Il est important de noter que pour obtenir des résultats les plus proches possible de l'implantation finale, il faut se placer après l'étape de placement et routage dans le flot de conception. Cela permet d'avoir le circuit tel qu'il sera fabriqué et surtout de disposer des éléments parasites sur les signaux (capacités et résistances) dus au routage. À ce stade, la liste des interconnexions (*netlist*) du circuit au niveau portes logiques (après routage) ainsi que le banc de test du circuit sont à disposition. En simulant le système, le fichier VCD requis par *PrimeTime PX* est obtenu. Le simulateur logique utilisé est Cadence *NCSim*. Dans l'outil *PrimeTime PX*, les données d'entrée de l'analyse à savoir les bibliothèques de caractérisation des cellules standards utilisées dans le circuit sont chargées, le circuit lui-même ainsi que les résistances et capacités parasites sur les signaux. Ensuite, le fichier d'activité issu de la simulation va servir à calculer la puissance totale dissipée par le circuit à partir de chaque transition enregistrée. Finalement, la courbe de puissance correspondant à l'intervalle de temps de simulation reporté dans le fichier VCD est obtenue. Un exemple de courbe de puissance est donné sur la figure 2.6. La puissance est donnée pour l'intervalle de temps spécifié lors de la génération du fichier VCD, ici entre 2750 et 3050 ns. Les pics de consommation enregistrés correspondent aux transitions des signaux sur les fronts montants et descendants de l'horloge du système. Le circuit évalué fonctionne à 10 MHz, on peut effectivement remarquer que la durée entre deux pics de consommation est de 50 ns, ce qui correspond bien à la demi-période de l'horloge.

À partir de cette courbe de consommation, il est déjà possible de voir des différences de consommation en fonction des données traitées ou de l'opération du système. Il est notamment possible d'identifier les points sur la courbe qui sont susceptibles d'être exploités pour récupérer des informations sur les données internes.

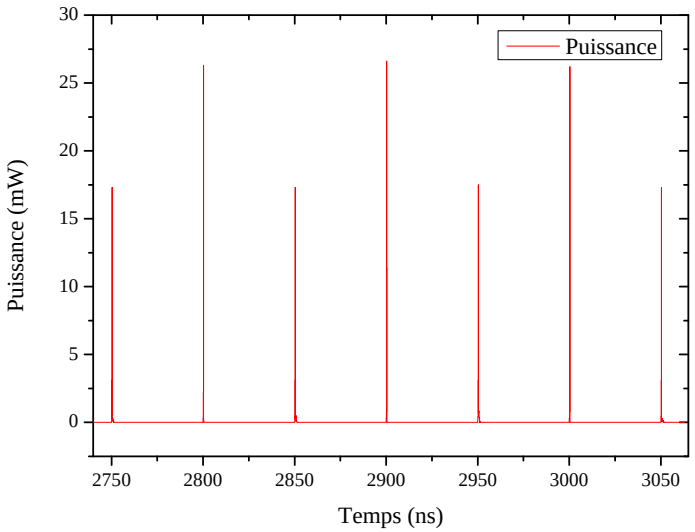


FIGURE 2.6 : Courbe de puissance obtenue avec *PrimeTime PX*

2.1.3 Bilan

Grâce à la méthodologie d'évaluation basée sur l'exploitation des fichiers *liberty* (.lib), il est possible d'obtenir à la fois la puissance moyenne et la puissance en fonction du temps d'un circuit numérique comportant plusieurs milliers de portes logiques. Cette méthode prend en compte les consommations statiques et dynamiques de l'ensemble du circuit, tout comme celles d'un sous-circuit si on ne s'intéresse qu'à une partie de celui-ci. Toutefois la méthode utilisée pour générer les courbes présente des limitations. L'utilisation des fichiers *liberty* ne permet pas d'avoir la capacité globale de la grille d'alimentation, c'est-à-dire la capacité vue depuis l'extérieur du circuit entre l'alimentation et la masse. Cette capacité de grille a pour effet de changer la forme d'onde du courant mesuré, et donc potentiellement de modifier des valeurs de comparaison entre les courbes acquises. Il est important de prendre en compte cette capacité pour la signature électrique. Il faut donc trouver un moyen de l'extraire et de l'inclure dans la consommation du circuit.

Dans le paragraphe suivant, on va analyser comment obtenir une signature de la consommation telle qu'elle serait obtenue lors d'un test électrique. En plus de la consommation, on va s'intéresser à la grille d'alimentation et aux paramètres externes au circuit.

2.2 Signature de consommation électrique en phase de conception

La signature électrique d'un circuit est considérée ici comme la mesure du courant obtenue grâce à un oscilloscope. L'objet de cette partie consiste à réaliser un modèle pour pouvoir évaluer la signature électrique d'un circuit en phase de conception.

2.2.1 Intérêt d'évaluer la signature

Les attaques par canaux auxiliaires, notamment par analyse de consommation, se font par des tests sur silicium. L'analyse est donc réalisée en fin de développement, et si une ou plusieurs failles sont identifiées, il ne sera pas possible de modifier le circuit déjà fabriqué. Il peut ainsi s'écouler un long moment entre la découverte d'une faille, et la correction de celle-ci. Le but est donc de réduire ce temps de réaction en ayant la possibilité d'effectuer ces caractérisations sécuritaires pendant la phase de conception. En effet, la détection de vulnérabilités du circuit en amont dans le flot de conception permet d'anticiper le développement de contremesures. L'idée est de pouvoir introduire au moment de la définition des spécifications du circuit des contraintes de conception visant à limiter la signature électrique. Ces contraintes seront vérifiées tout au long du flot. Avec cette approche, la robustesse d'un circuit vis-à-vis d'attaques par analyse de consommation peut être augmentée avant tout test électrique.

2.2.2 Niveaux hiérarchiques à considérer pour la signature

Les circuits dont on cherche à évaluer la signature peuvent être de différents types : sous bloc numérique, analogique, une *IP* mixte ou un *SoC* entier. Chaque type de circuit a une caractérisation propre qui permet d'obtenir sa consommation. Le tableau 2.1 donne un aperçu des besoins pour évaluer la signature d'un circuit en fonction du niveau de considération. Comme déjà indiqué, les sous-blocs numériques sont caractérisés à partir des fichiers .lib de la bibliothèque des cellules standard. L'impédance du réseau d'alimentation doit être extraite par ailleurs. Concernant les sous-blocs analogiques, ils sont caractérisés directement grâce aux modèles SPICE des transistors. La prise en compte des impédances du réseau d'alimentation et des impédances parasites se fait en simulant le circuit après la phase du dessin de masques (*layout*). Pour des *IP*, en plus des blocs numériques et analogiques qu'ils contiennent, il faut considérer les paramètres hors circuit tels que les circuits gérant les entrées/sorties, le boitier ou encore la carte de test. Deux moyens sont à disposition pour rendre compte de la consommation électrique : la mesure de la différence de tension aux bornes d'une résistance ou l'utilisation d'une sonde pour capter les émissions électromagnétiques. La difficulté va être d'extraire les informations de consommation qui nous intéressent de l'ensemble du circuit ou du *SoC*. Cela va déprendre notamment du ratio de ces consommations.

TABLEAU 2.1 : Analyse de consommation d'une *IP* dans un *SoC*

	Sous bloc numérique	**Sous bloc analogique**	***IP* dans un *SoC***
Fichiers d'entrée pour évaluer la puissance	Fichiers .lib	Modèle SPICE	Modèle SPICE et Fichiers .lib
Caractérisation de la signature en simulation	Nécessite un modèle prenant en compte l'impédance du réseau d'alimentation	Simulation après *layout* niveau transistor	Nécessite une modèle avec l'impédence de la grille d'alimentation pour : 1. *IP* 2. *IO* 3. Boîtier 4. Carte de test
Moyens de mesure	NA – Doit être isolé du reste du circuit		1. Tension au bornes d'une résistance 2. Sonde électromagnétique
Résultats attendus	Les modèles donnent la puissance consommée avec une précision allant jusqu'au pA		1. A cause de la consommation élevée d'un *SoC*, la résistance de mesure doit être suffisamment petite pour limiter la chute de tension aux bornes du *SoC* 2. Difficulté d'identifier la consommation de l'*IP* si l'alimentation est partagée 3. L'extraction de la consommation dépend du ratio entre celle de l'*IP* et celle de l'ensemble du *SoC*

Dans un premier temps on va voir comment extraire les capacités du réseau d'alimentation nécessaires pour déterminer la signature d'un circuit numérique.

2.2.3 Extraction de la capacité de grille d'alimentation

Les capacités que l'on cherche à extraire sont celles vues depuis l'extérieur du circuit entre le nœud d'alimentation et la masse.

2.2.3.1 Capacités à extraire

Plusieurs éléments contribuent à la capacité de la grille d'alimentation. La première contribution provient des composants élémentaires à savoir les transistors MOS. En

effet, dans la logique CMOS, le drain ainsi que le substrat des transistors de type P sont généralement connectés au nœud d'alimentation du circuit. La capacité totale due aux transistors est donc la somme des capacités de jonctions internes sur le nœud d'alimentation (C_{GS}, C_{GB} et C_{DB}) en plus de la capacité entre le caisson de type N et le substrat qui est de type P (C_{n_w/p_s} sur la figure 2.7).

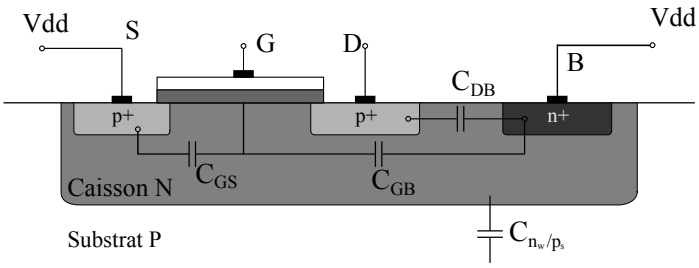


FIGURE 2.7 : Capacités du transistor MOS de type P

La deuxième contribution à la capacité de grille provient de la capacité métallique. Elle est due aux lignes de métaux qui permettent de connecter et d'alimenter l'ensemble des composants du circuit. Il s'agit en fait d'une capacité parasite qui va dépendre de la géométrie des rails d'alimentation.

Les deux types de capacité identifiées ont des méthodologies d'extraction différentes que nous allons détailler ci-dessous.

2.2.3.2 Méthodologie d'extraction

Capacité métallique. L'extraction de la capacité métallique se fait à partir de la liste des interconnexions du circuit après le dessin des masques. La valeur de la capacité dépend uniquement de la géométrie des lignes de métaux, il n'y a donc pas besoin d'une étape de simulation du circuit. On va directement utiliser un outil tel que Synopsys *StarRC*, qui est un outil d'extraction d'impédances parasites sur les interconnexions d'un circuit.

Pour réaliser l'extraction, on part du dessin de masque du circuit étudié (figure 2.8). On va alors procéder à l'étape de génération de la *netlist* au niveau transistor. Cette étape comprend la génération des capacités parasites aussi bien sur les signaux que sur les alimentations. Il existe deux types d'extractions réalisées par l'outil :

- capacité couplée à la masse (C lumped) : la capacité extraite sur un fil donné est référencée à la masse,
- capacité de couplage (Cc) : les capacités de couplage entre les fils sont également extraites en plus de la capacité référencée à la masse.

La méthode d'extraction Cc est plus précise que l'extraction C, néanmoins elle demande plus de temps d'extraction et la taille du fichier contenant les éléments extraits augmente

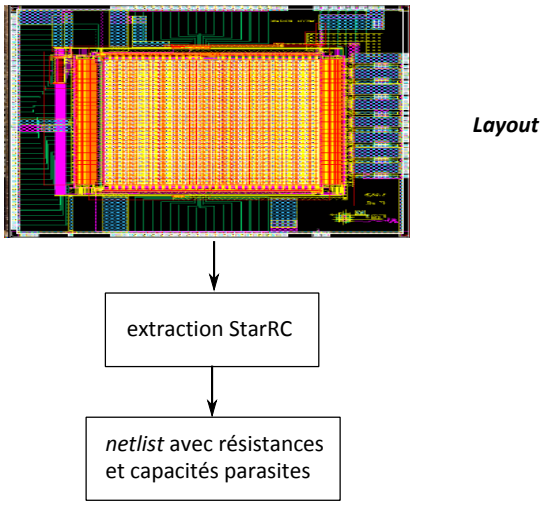


FIGURE 2.8 : Flot de génération des éléments parasites

car on multiplie le nombre de capacités extraites. On choisit ici de faire une extraction C car on s'attend à ce que la capacité des composants CMOS soit plus importante que la capacité métallique, et donc le besoin de précision serait moindre sur cette capacité.

Extraction de la capacité des composants. Pour extraire la capacité des composants, on doit se placer au niveau transistor. Les modèles SPICE des transistors contiennent les paramètres nécessaires pour calculer les différentes capacités. Par ailleurs, la capacité entre le caisson de type N et le substrat de type P (C_{n_w/p_s}) dépend de la topologie du circuit étudié. Par conséquent, on a besoin de la *netlist* niveau transistor après le placement routage.

Analyse petit signal du courant La première méthode d'extraction consiste à réaliser une analyse petit signal du circuit au niveau transistor. Dans cette méthode, on part de la loi d'Ohm pour calculer la capacité à partir du courant et de la tension. Si on prend l'exemple d'un condensateur, la loi d'Ohm nous donne :

$$
\begin{aligned}
U &= Z \cdot I \\
20 \cdot log|U| &= 20 \cdot log(|Z| \cdot |I|) = 20 \cdot log|Z| + 20 \cdot log|I| \\
&= 20 \cdot log\left(\frac{1}{2 \cdot \pi \cdot f \cdot C}\right) + 20 \cdot log|I| \\
&= -20 \cdot log(2 \cdot \pi \cdot f \cdot C) + 20 \cdot log|I| \\
2 \cdot \pi \cdot f \cdot C &= 10^{\frac{20 \cdot log|I| - 20 \cdot log|U|}{20}} \\
C &= \frac{1}{2\pi f} \cdot 10^{\frac{20 \cdot log|I| - 20 \cdot log|U|}{20}}
\end{aligned}
\tag{2.9}
$$

Dans cette équation, U représente la tension aux bornes du condensateur, I le courant le traversant, Z l'impédance complexe, C la capacité du condensateur et f la fréquence. Ainsi, en appliquant une valeur connue de tension aux bornes du condensateur et en mesurant le courant, il est possible de calculer la capacité grâce à l'équation 2.9. En choisissant 1 V comme amplitude de la tension, on obtient alors $log|U| = 0$ dans la précédente équation. On a alors :

$$C = \frac{1}{2\pi f} \cdot |I| \tag{2.10}$$

Pour un condensateur idéal, la courbe qui représente le courant en fonction de la fréquence est une droite dont la pente vaut 20 dB par décade.

Le même principe est utilisé pour extraire la capacité entre le nœud d'alimentation et la masse pour un circuit numérique. Le circuit doit être polarisé dans le même état que lors de son fonctionnement si on veut extraire la capacité. En effet, la valeur de la capacité peut changer en fonction de l'état des transistors dans le circuit. La difficulté est que la simulation au niveau transistor d'un circuit numérique peut s'avérer coûteuse en temps et en ressources de calcul. Afin de contourner cette difficulté, on se place dans un mode statique c'est à dire qu'aucun changement n'a lieu sur les entrées du circuit. De plus, la gamme de fréquences de l'analyse petit signal peut être réduite (1 à 2 décades).

Le résultat de la simulation petit signal d'un circuit numérique est présenté sur la figure 2.9. Le circuit simulé est composé d'environs 10 000 cellules dont 200 bascules. On utilise les modèles SPICE des transistors et le simulateur électrique *Eldo* de Mentor Graphics. La durée de simulation pour obtenir la réponse du circuit sur 4 décades est de 30 minutes (avec une précision de 20 points par décade). On peut distinguer deux zones sur la figure 2.9 :

- entre 1 KHz et 100 KHz, le courant mesuré est quasi constant. Ce comportement est équivalent à celui d'une résistance.
- à partir de 100 KHz environ, la courbe de courant devient proportionnel à la fréquence. Le circuit a un comportement capacitif.

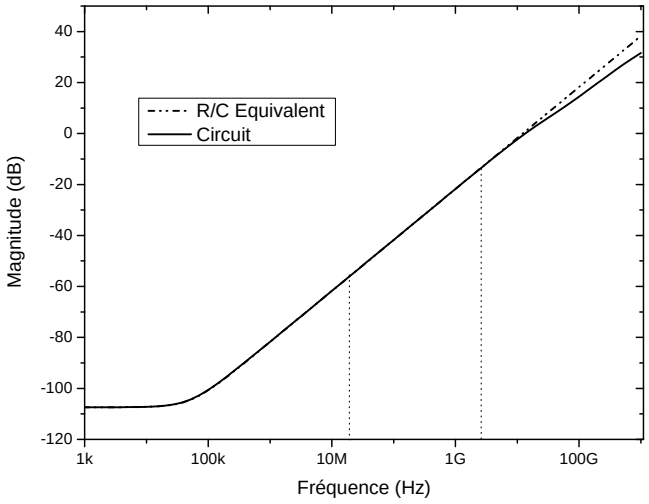


FIGURE 2.9 : Analyse petit signal du courant traversant le circuit étudié et de son circuit équivalent

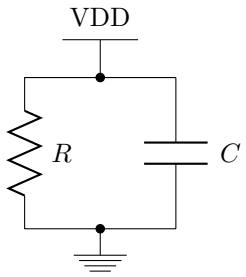


FIGURE 2.10 : Circuit équivalent en analyse petit signal

Le circuit équivalent est donc un condensateur branché en parallèle d'une résistance (figure 2.10). On peut voir sur la figure 2.9 la réponse fréquentielle du courant pour le circuit étudié ainsi que le circuit équivalent (en pointillés). Au-delà de 10 GHz, la pente de la courbe du circuit étudié commence à dériver de celle du circuit équivalent. Pour extraire la capacité recherchée, on se place dans la zone où la pente de la courbe vaut 20 dB par décade. Dans ce cas, seul l'effet de la capacité est considéré.

L'extraction s'est faite dans un cas avec une certaine polarité sur les entrées et donc une certaine polarisation des transistors. Toutefois lorsque le circuit est en fonctionnement, le point de polarisation des transistors change en fonction du temps. On peut alors se demander comment varie la capacité extraite en fonction de la polarisation des signaux d'entrée ou encore en fonction de la tension d'alimentation du circuit. Pour se rendre compte de l'influence de ces paramètres, plusieurs simulations ont été réalisées :

- dans le premier cas le circuit est dans un état de réinitialisation (*reset*)
- dans le deuxième cas, le circuit est dans son état initial (*stand-by*)

– dans le dernier cas, on modifie la tension d'alimentation en restant en état de réinitialisation.

Dans chacune de ces situations, la capacité est extraite en utilisant l'équation 2.10. Les résultats de ces simulations sont donnés dans le tableau 2.2. Il est notable qu'en modifiant la tension d'alimentation, il y a une variation très faible de la capacité extraite. Entre les deux valeurs extrêmes de tension de fonctionnement, à savoir entre 0.8 V et 1.4 V, l'écart de capacité est de 2.4 %. Cela peut s'expliquer par le fait que la diode

TABLEAU 2.2 : Capacité extraite en fonction de la polarisation du circuit étudié

Tension [V]	0.8	1.1	1.1	1.4
Etat du circuit	*reset*	*reset*	*stand-by*	*reset*
Capacité extraite [pF]	8.91	8.85	8.85	8.70

formée par le caisson de type N et le substrat de type P est polarisée en inverse. Ainsi, en augmentant la tension d'alimentation, on augmente la tension de polarisation inverse de la diode, ce qui a pour effet l'élargissement de la zone de charge espace (ZCE). En effet, l'épaisseur de la zone de charge espace est définie par [Mathieu 2009] :

$$W_i = W\sqrt{1 - \frac{V_j}{V_b}} \tag{2.11}$$

W est l'épaisseur de la ZCE de la jonction PN non polarisée, W_i celle de la jonction polarisée en inverse, V_j la différence de potentiel créée par la source extérieure au niveau de la jonction et V_b le potentiel de barrière. En tenant compte de l'évolution de l'épaisseur de la ZCE en fonction de la tension de polarisation dans l'expression de la capacité de transition de la jonction on trouve :

$$C_t(V_j) = \frac{C_t(0)}{\sqrt{1 - V_j/V_b}} \tag{2.12}$$

Et donc plus on augmente en valeur absolue la polarisation inverse, (V_j de plus en plus négative), plus la capacité de la jonction diminue.

L'écart est encore plus petit lorsqu'on change l'état de polarisation des entrées puisqu'on se retrouve ici avec la même valeur de capacité. Finalement, l'état de polarisation a peu d'influence sur la capacité.

Il ressort de cette analyse qu'il est possible de se placer dans l'état de réinitialisation pour extraire la capacité sans pour autant perdre en précision de mesure.

Outil d'analyse d'intégrité du réseau d'alimentation *Totem* *Totem* est un outil d'aide à la conception et la vérification du réseau d'alimentation développé par Apache. Cet outil adresse à la fois l'analyse statique et dynamique de l'intégrité de

l'alimentation. *Totem* fournit un modèle de calcul de la puissance d'un circuit appelé CPM (*Chip Power Model*), qui est un circuit équivalent du réseau d'alimentation en plus de la consommation dynamique en courant. Le modèle CPM permet de considérer à la fois la conception du réseau d'alimentation du circuit intégré, le boîtier ainsi que la carte électronique. Ce modèle peut être utilisé pour vérifier le réseau d'alimentation hors-circuit en évaluant l'impact des impédances parasites provenant de la puce sur le réseau global. L'outil permet de diagnostiquer des résonances potentielles LC dues au boîtier, ou encore optimiser le placement des capacités de découplage hors-circuit [Apa 2013]. Le CPM fournit un modèle simplifié de la puce étudiée qui peut être réduit à une résistance R_{die} connectée en série à un condensateur C_{die} et en parallèle une source de courant (figure 2.11). Cet outil peut être utilisé pour notre besoin car il fournit à la

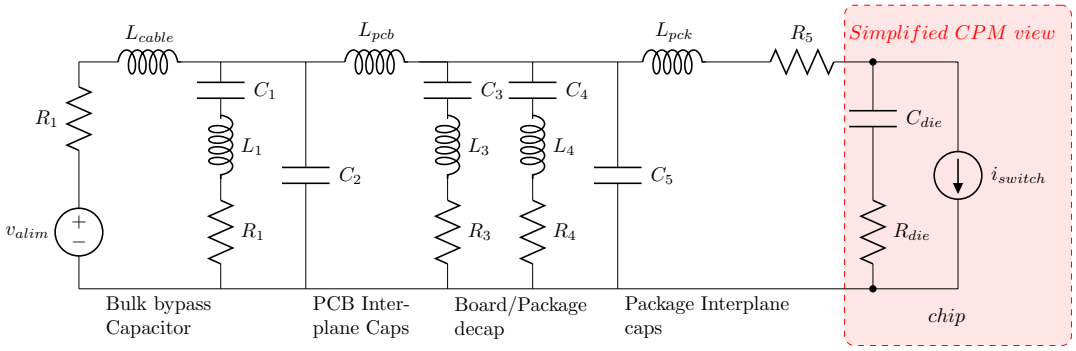


FIGURE 2.11 : Réseau de distribution d'alimentation et modèle simple de puissance d'un circuit [Apa 2013]

fois la capacité entre le nœud d'alimentation et la masse ainsi que le courant en fonction du temps. Toutefois ce courant est obtenu par une simulation SPICE du circuit au niveau transistor. Le problème du temps de simulation et de la capacité à simuler un circuit numérique est toujours présent. Cependant, le CPM peut être utilisé pour extraire la capacité aux bornes du circuit. L'avantage de l'outil est qu'il utilise des algorithmes de réduction qui devraient réduire le temps d'extraction de la capacité en comparaison avec une simulation petit signal du circuit complet.

Le flot de génération du CPM est donné sur la figure 2.12. Le modèle se base sur la topologie du circuit (au format GDSII), la liste des interconnexions du circuit ainsi que le fichier contenant le banc de test (*testbench*). À partir des fichiers d'entrée, une phase de pré-caractérisation a lieu pour déterminer le courant consommé par chaque transistor du circuit. Cette phase fait intervenir un simulateur SPICE externe. Le courant de chaque transistor est transformé en un modèle électrique utilisé par *Totem*. Trois fichiers sont générés par cette étape :

– un fichier .spcurrent contenant un modèle de courant transitoire pour chaque transistor du circuit

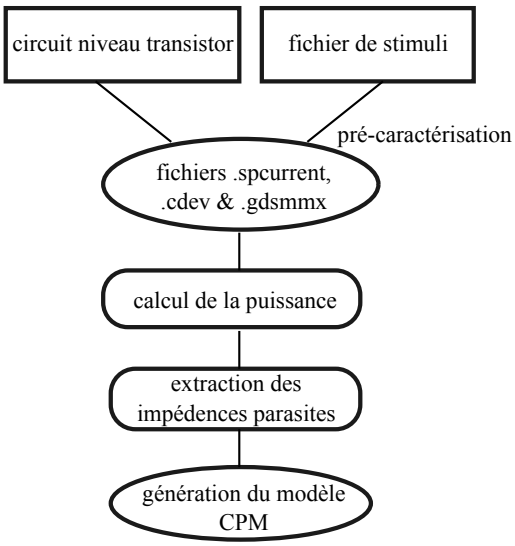


FIGURE 2.12 : flot de génération du modèle CPM

– un fichier .cdev avec les informations sur les capacités des transistors
– le fichier .gdsmmx qui contient notamment la localisation des composants

La génération du modèle CPM utilise un algorithme de réduction des résistances et capacités de grille d'alimentation qui contient à la fois des impédances parasites entre la ligne d'alimentation et la masse, mais aussi des éléments statiques. Les composants qui commutent sont modélisés comme des sources de courant transitoires. À partir de ces informations, la puissance moyenne est calculée puis les éléments parasites sur les lignes de métaux conduisant les alimentations sont extraits. Enfin, le modèle CPM est généré.

Comparaison des méthodes d'extraction. L'utilisation de deux méthodes d'extraction permet de comparer les résultats obtenus. La même *netlist* est utilisée en entrée afin d'évaluer la capacité du système vue depuis l'extérieur du circuit entre les nœuds V_{dd} et Gnd. Les résultats de cette comparaison sont fournis dans le tableau 2.3. Les deux méthodes nécessitent les mêmes fichiers d'entrée pour réaliser l'extraction. À noter que Totem requiert en plus le dessin des masques du circuit au format GDSII. Comme présenté précédemment, Totem fournit en plus de la capacité la résistance série associée de même que le courant en fonction du temps. Cette information supplémentaire augmente le temps de simulation de l'outil car la *netlist* fournie est simulée au niveau transistor pour obtenir le courant transitoire consommé par le circuit. Le courant donné, bien que précis ne peut pas être exploité pratiquement car pour des longues simulations temporelles, il serait difficile de lancer plusieurs simulations avec différents vecteurs d'entrées pour une analyse plus exhaustive de la signature. Étant donné que seule la capacité

TABLEAU 2.3 : Comparaison des deux méthodologies d'extraction de capacité

Méthode d'extraction	Totem	Analyse AC *netlist*
Fichiers d'entrée	GDSII, stimuli de test, *netlist post-layout*, corner	stimuli de test, *netlist post-layout*, corner
Sorties	– courant en fonction du temps – Résistance – Capacité	Capacité
Capacité extraite	7.69 pF	8,85 pF (+15 %)
Limitations	Caractérisation temporelle i(t) augmente le temps de simulation	Simulation sur la *netlist* globale

est recherchée ici, Totem permet de comparer la valeur de capacité extraite. L'analyse AC utilise la *netlist* globale mais polarisée dans un mode statique (aucune variation des entrées). Néanmoins la taille du circuit peut être un élément limitant pour des circuits ayant un nombre important de transistors, par exemple un *SoC*.

Concernant les valeurs de capacités extraites, on note un écart de 15 % supérieur avec la méthode d'analyse petit signal de la *netlist* niveau transistor. Cet écart pourrait s'expliquer par l'utilisation d'algorithmes de réduction au niveau des nœuds d'alimentation utilisés par Totem pour réduire le temps de calcul du courant de la capacité du circuit.

2.2.4 Modèle équivalent de la signature en courant

Un circuit générique peut être composé d'un bloc numérique et/ou d'un bloc analogique. Ces deux parties consomment du courant électrique provenant de la source d'alimentation. À partir des éléments extraits dans le paragraphe précédent, on peut élaborer un modèle électrique de la consommation d'un circuit.

2.2.4.1 Présentation du modèle

L'analyse de la signature de consommation électrique est basée sur la simulation d'un modèle équivalent d'un composant étudié. Le modèle est composé d'une source de courant et de la capacité entre l'entrée d'alimentation et la masse vue depuis l'extérieur du circuit (figure 2.13). Vdd est la tension d'alimentation du système. On retrouve pour la partie numérique les capacités des transistors (C_{MOS}), les capacités parasites dues au lignes de métaux ($C_{metal.}$), et la capacité entre le caisson de type N et le substrat de type P (C_{n_w/p_s}) dont l'extraction a été présentée en section 2.2.3.2. La source $i_{num.}$ correspond au courant consommé par le circuit numérique. Dans la partie analogique,

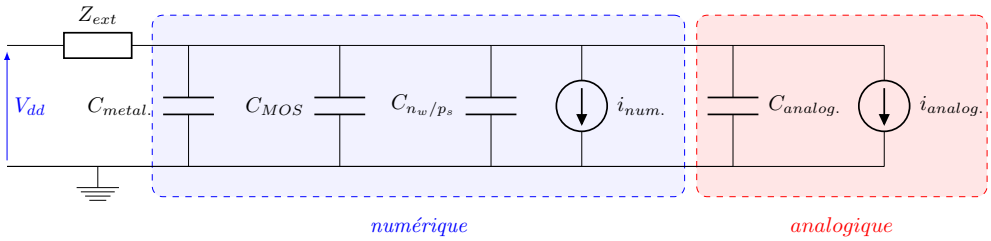


FIGURE 2.13 : Modèle équivalent pour l'analyse de signature d'un circuit générique

on a également la capacité entre V_{dd} et la masse ($C_{analog.}$) ainsi que le courant des transistors (i_{analog}). $Z_{ext.}$ représente l'impédance des éléments hors du circuit considéré. Ce paramètre sera détaillé dans le paragraphe ci-après.

2.2.4.2 Cas particuliers et paramètres additionnels

Pour obtenir la signature électrique d'un bloc analogique, on a besoin de la *netlist* après dessin des masques du circuit. Cette *netlist* est simulée dans les conditions de fonctionnement du circuit (stimuli de l'opération considérée à tension nominale) et on mesure le courant tiré depuis la source d'alimentation. Les modèles SPICE des transistors prennent en compte les capacités MOS, ainsi avec cette *netlist*, on a aussi accès à la capacité de jonction entre le caisson et le substrat. En ce qui concerne les capacités parasites, elles sont générées en même temps comme indiqué dans la partie 2.2.3.2. Ainsi, pour un circuit purement analogique, il n'est pas nécessaire de séparer la phase d'extraction des capacités et l'extraction du courant pour obtenir la signature car les simulations transitoires utilisant les modèles SPICE sont assez précises pour caractériser la puissance consommée. Par conséquent une simulation électrique classique pour un bloc analogique peut être utilisée. La modélisation des circuits analogiques pour l'optimisation des temps de simulation n'est pas abordée dans cette étude.

Certains éléments du circuit qui est testé sur silicium ne sont pas représentés en détails sur la figure 2.13. Parmi ces éléments, on a le boitier, la carte de test ou encore les sondes utilisées pour effectuer les mesures. On peut également ajouter les résistances du réseau de distribution d'alimentation au niveau du circuit ou même la résistance série utilisée pour mesurer le courant. Certains de ces paramètres peuvent être négligés, si ce n'est pas le cas on peut les prendre en compte dans le paramètre Z_{ext}, l'impédance externe au circuit. On a choisi de séparer ces paramètres car ils ne dépendent pas du circuit ou de sa fonctionnalité. Le boîtier ou la carte de test peuvent être différentes d'une expérience à une autre, les valeurs correspondantes sont prises dans la documentation du fabricant. À titre d'exemple, pour des boîtiers de type BGA à 144 billes, l'inductance est évaluée à 2.25 nH en valeur typique et la capacité (*C lumped*) à 60 fF dans [Clark 2003].

La résistance série équivalente du réseau de distribution d'alimentation peut être

estimée à partir de l'extraction des résistances parasites. L'extraction donne la résistance entre le point d'arrivée de l'alimentation et chaque transistor connecté à la grille d'alimentation. Toutes ces résistances en parallèle représentent la résistance série équivalente du réseau d'alimentation. En fonction de la valeur de la résistance de mesure, la résistance du réseau d'alimentation peut être négligée. Un exemple illustrant ce dernier point sera présenté dans la partie expérimentale.

2.2.5 Résultats du modèle obtenu

Dans cette partie on s'intéresse aux résultats donnés par le modèle en simulation et également aux mesures réalisées sur silicium, ce qui nous permettra de comparer et d'évaluer le modèle développé.

2.2.5.1 Paramètres expérimentaux

Description du circuit d'étude. Le circuit de test est une *IP* composée d'un cœur analogique et d'un contrôleur numérique de mémoire non volatile, conçu en technologie silicium massif 40 nm. La tension de fonctionnement est comprise entre 0.90 V et 1.20 V. Pour les expériences réalisées, le circuit fonctionne à tension nominale à savoir 1.1 V. Le contrôleur de mémoire permet de gérer l'adressage des cellules, les opérations de lecture et d'écriture ainsi que les permissions associées. Les mots stockés dans la mémoire ont une longueur de 32 bits. Il est constitué d'environ 10 000 portes logiques parmi lesquelles 200 bascules. Pour les tests, la fréquence de fonctionnement est réglée à 10 MHz. La consommation moyenne du circuit est requise car elle permet de fixer les paramètres de mesures. L'opération que l'on veut observer consomme environ 35 μA. Cette valeur est obtenue lors de la génération des courbes de puissance (cf. section 2.1.2.2). Grâce à ces informations, la signature électrique du circuit va pouvoir être mesurée.

Remarque importante L'*IP* qui est étudiée est intégrée dans un circuit de test qui possède une alimentation partagée entre les différentes parties, parmi lesquelles les entrées/sorties (*IO*), la logique de contrôle du circuit de test et d'autres circuits non utilisés au cours du test. Ces éléments ne sont pas actifs au cours du test de l'*IP*, ils présentent donc uniquement une consommation statique. La consommation correspondante n'a pas été évaluée. Il devrait donc y avoir une signature avec des niveaux de courant mesurés plus élevés lors des tests expérimentaux.

Dispositif expérimental. Les mesures sont effectuées à partir du point d'entrée de la tension d'alimentation de l'*IP*, à savoir la carte de test. Le dispositif expérimental nécessaire pour les mesures est présenté sur la figure 2.14. Pour visualiser les informations temporelles, un oscilloscope numérique est à disposition. Une sonde différentielle permet de mesurer la tension aux bornes de la résistance série. L'avantage de ce type

de sonde est qu'elle permet d'observer à l'oscilloscope la tension entre deux points d'un montage, même si aucun de ces deux points n'est relié à la masse. La génération des stimuli d'entrée du circuit se fait par l'intermédiaire de la carte de test. Et enfin, le générateur de tension continue qui sert à délivrer la tension d'alimentation.

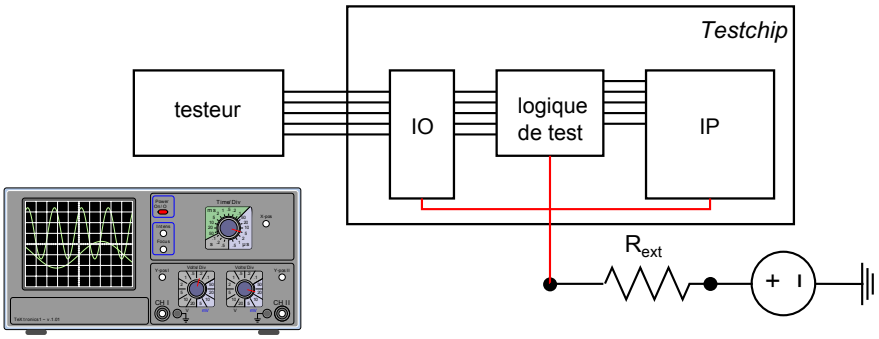


FIGURE 2.14 : Dispositif expérimental

La mesure de courant se fait donc par l'intermédiaire de la tension aux bornes de la résistance R_{ext}. Afin d'obtenir des données exploitables, la valeur de la tension aux bornes de la résistance doit être mesurable à l'oscilloscope. Celui-ci permet de mesurer des variations de tension autour de 10 mV. La tension mesurée doit donc être supérieure à ce seuil. Pour cela, les outils de caractérisation de puissance (en simulation) donnent les informations nécessaires. Il faut que la tension aux bornes de la résistance soit suffisamment faible pour que le circuit continue de fonctionner correctement. Le circuit est conçu pour fonctionner entre 0.9 V et 1.20 V, il reste une gamme de 300 mV pour effectuer les mesures. En notant I_{test} le courant moyen consommé par le circuit, on a :

$$\begin{aligned} 10 < R_{ext} I_{test} < 300 \\ \frac{10}{I_{test}} < R_{ext} < \frac{300}{I_{test}} \end{aligned} \tag{2.13}$$

En prenant en compte les données de consommation moyenne, la résistance choisie pour la mesure est de 1.3 kΩ. Cela donne une chute de tension moyenne aux bornes du circuit de :

$$1300 * 35 \cdot 10^{-6} = 45.5\, mV. \tag{2.14}$$

Il s'agit là de la tension moyenne, en prenant une valeur en dessous de la limite des 300 mV et en vérifiant que les pics de consommation, on s'assure que la limite n'est pas franchie et donc que le circuit continue de fonctionner normalement. Le choix d'une résistance de grande valeur permet de s'affranchir de l'utilisation d'un amplificateur de signal, ce qui alourdit le dispositif de test. Compte tenu de la valeur de résistance externe ici, on peut considérer que la résistance du circuit (éléments parasites et composants CMOS) est négligeable.

La sonde différentielle utilisée pour les mesures peut également introduire des perturbations. En effet, en fonction de la valeur de la résistance aux bornes de la sonde, la résistance équivalente vue par le circuit peut être modifiée. La résistance de la sonde et celle du circuit sont en parallèle, en notant R_s la résistance de la sonde et R_{eq} la résistance équivalente, on a :

$$R_{eq} = \frac{R_s R_{mes}}{R_s + R_{mes}} \tag{2.15}$$

À noter qu'ici R_{mes} correspond à la résistance externe R_{ext} sur la figure 2.14. Pour avoir une résistance équivalente aussi proche que possible de celle qui a été choisie, il faut que R_{mes} soit négligeable devant R_s par conséquent il faut une grande valeur de R_s. La résistance de la sonde ici est de 100 kΩ, ce qui ne modifie que très peu la résistance équivalente.

La fréquence de fonctionnement du circuit est de 10 MHz. On choisit une fréquence d'échantillonnage de 250 MHz qui permet d'avoir un nombre suffisant de points par période (25). Ces réglages seront utilisés pour réaliser les mesures sur le circuit réel.

2.2.5.2 Résultats expérimentaux

Les courbes obtenues à l'oscilloscope sont présentées sur la figure 2.15. Afin de minimiser le bruit de mesure, les résultats de dix acquisitions ont été moyennés pour chacune des traces de courant. L'opération mesurée est un accès en lecture en récupérant

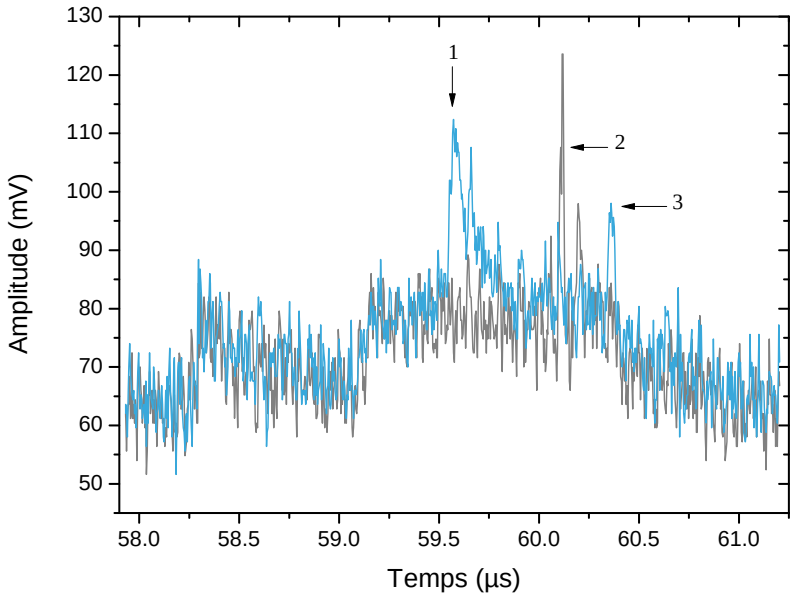


FIGURE 2.15 : Courbes de consommation obtenues à l'oscilloscope

deux données différentes. La première donnée a un poids de Hamming minimal (courbe

en bleu), tandis que la seconde a un poids de Hamming maximal (courbe grise). Notons que le choix de la résistance de mesure à partir des niveaux de tension observés peut être vérifié. Le niveau statique de tension tourne autour de 70 mV sur la durée de cette acquisition et les pics de tension n'excèdent pas les 125 mV. Ces réglages permettent d'avoir une tension d'alimentation suffisante pour le circuit puisqu'on est en dessous des 300 mV spécifiés dans le paragraphe précédent.

En première approche, les courbes ont des niveaux similaires, ce qui montre qu'il s'agit bien de la même opération dans les deux cas. Néanmoins, des différences existent à trois niveaux. Ceux-ci sont indiqués par des flèches sur la figure 2.15. La première différence est un pic de courant qui apparaît étalé dans le temps. Il s'agit en fait d'une transition asynchrone sur le bus de données qui intervient dans ce cas. En fonction de la vitesse de transition de chaque bit du bus, le pic de courant est plus ou moins important et étalé dans le temps. À noter que cette transition n'a lieu que dans un cas ce qui explique cette différence de consommation. La deuxième différence est cette fois-ci un pic de courant beaucoup plus abrupt qui intervient un peu avant la fin de la lecture. Il y a une opération de synchronisation du bus de données, les registres du bus changent donc de valeurs au même moment sur un front d'horloge. Là aussi, le poids de Hamming de la donnée a une grande influence sur la valeur du pic enregistré : plus il y a de bascules qui changent de valeurs, plus le pic de courant est important. Cela se traduit par une augmentation de la tension aux bornes de la résistance. Le dernier pic correspond une fois de plus à une transition asynchrone sur le bus de données, qui peut être assimilé à un retour à l'état initial (*stand-by*).

Tous les points identifiés ci-dessus doivent pouvoir être retrouvés à l'aide du modèle électrique. Nous allons voir les résultats du modèle pour le même circuit étudié.

2.2.5.3 Résultats de simulation

Le modèle électrique simulé est présenté figure 2.16. La partie numérique représente

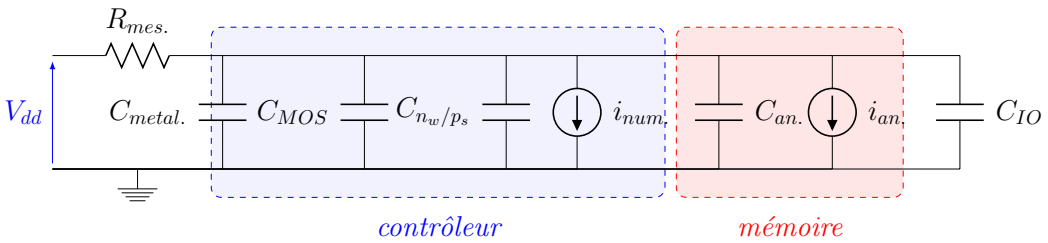


FIGURE 2.16 : Modèle électrique équivalent de l'*IP* étudiée

le contrôleur de mémoire tandis que la partie analogique représente le cœur de mémoire. Plusieurs éléments ont été pris en compte : la résistance de mesure R_{mes}, les courants du contrôleur et de la mémoire ainsi que les capacités des différents blocs.

Le courant de la partie numérique est obtenu en caractérisant à l'aide de l'outil *PrimeTime PX* le circuit au niveau portes logiques après l'étape du placement routage. L'extraction de la capacité de grille est réalisée par une simulation petit signal en utilisant la méthode présentée section 2.2.3.2. La partie analogique est entièrement caractérisée par simulation SPICE au niveau transistor. A partir de ce modèle, on obtient la signature en courant représentée sur la figure 2.17. Les courbes sont obtenues pour des

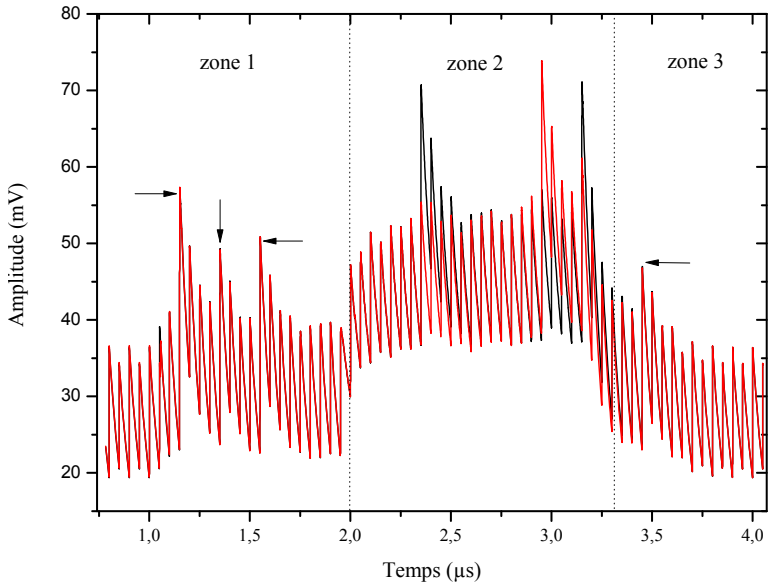


FIGURE 2.17 : Signature de consommation donnée par le modèle électrique : les trois zones représentent les différentes phases de l'opération

données en cours de traitement ayant des poids de Hamming différents (poids de 0 pour la première courbe contre 32 pour la deuxième). Les pics de courant observés (indiqués par des flèches) ont des niveaux identiques, de même que les niveaux statiques. En effet, sur la première phase et la dernière phase on mesure une tension moyenne aux bornes de la résistance de 30 mV tandis que sur la deuxième phase, la consommation moyenne est de 45 mV. La première phase (zone 1) est une phase de préparation de l'opération de lecture. La deuxième (zone 2) correspond à la lecture et la troisième (zone 3) le retour à l'état initial. À noter que le niveau moyen de tension pendant l'opération de lecture (45 mV) correspond bien à une valeur de courant moyen de 35 μA.

Nous allons maintenant tenter de valider ce modèle électrique en comparant les résultats obtenus avec les mesures sur silicium. Cette étape nous permettra aussi d'en évaluer la précision.

2.2.5.4 Comparaison et analyse

On se place dans les mêmes conditions de fonctionnement en simulation et en test (température, tension d'alimentation et stimuli). La figure 2.18 présente les mesures et la simulation du modèle électrique équivalent du circuit d'étude. D'abord, on remarque un offset sur la mesure du courant de consommation par rapport au modèle. L'offset mesuré est d'environ 40 mV. Cette consommation additionnelle peut être imputée aux éléments hors circuit tels que les IOs, la logique de contrôle à l'intérieur du circuit de test ou encore les autres composants présents dans le circuit de test partageant la même alimentation. Ces éléments n'ont pas été caractérisés précisément car on s'intéresse principalement à la consommation dynamique du circuit d'étude. En effet, lors de l'opération de lecture, la configuration du circuit est déjà effectuée donc la logique de contrôle des signaux d'entrée et sortie est inactive. D'autre part, la caractérisation de ces éléments demanderait des étapes supplémentaires d'extraction de courant en fonction du temps. Les trois pics de consommation numérotés de 1 à 3 sur la figure ont le même niveau si on enlève l'offset mentionné. Le dernier pic mesuré est moins important que sur la simulation. Ce pic correspond à la transition du bus de données comme indiqué section 2.2.5.2. Il s'agit d'une transition asynchrone (les bits du bus de données ne changent pas à la même vitesse). Par contre, au niveau de la simulation numérique, la modélisation est telle que tous les bits du bus changent au même moment, le pire cas étant retenu pour le temps de transition. Ainsi le pic de consommation est le cas pessimiste pris en compte en simulation.

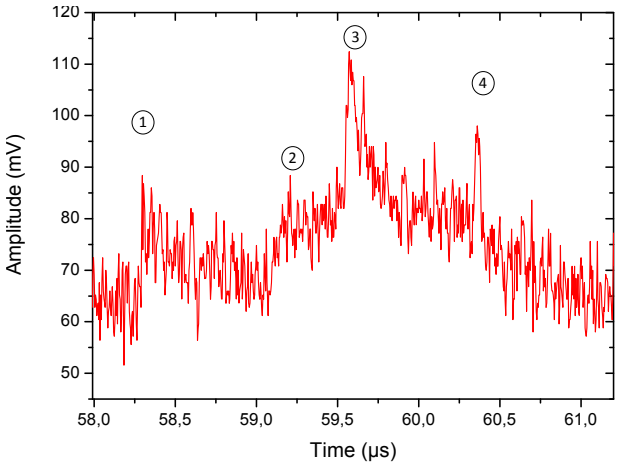
Le tableau 2.4 résume les valeurs de consommation pour les différentes transitions du bus de données. Dans le cas de la transition asynchrone, le modèle donne un pire

TABLEAU 2.4 : Comparaison des pics de consommation entre les mesures et le modèle

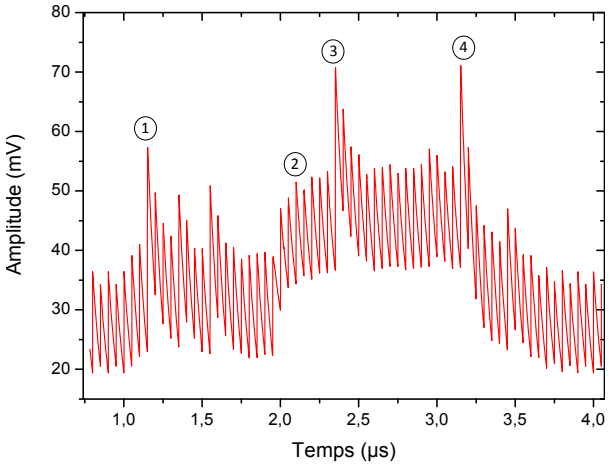
Type de transition	32 bits synchrone	32 bits asynchrone
Simulation	35.54 mV	jusqu'à 35 mV
Test	48 mV	30 mV

cas correspondant au changement simultané de tous les bits du bus. Pour la transition synchrone, les mesures donnent une valeur plus importante que le modèle. Cela peut s'expliquer par la consommation additionnelle des portes logiques de l'arbre d'horloge du circuit de test. En effet, même si les bascules à l'extérieur du circuit d'étude ne changent pas de valeur, les transitions du signal d'horloge font qu'il y a des pics de consommation à chaque front.

Les deux courbes ont été obtenues pour la même opération et sont synchronisées par rapport au même signal de contrôle. La différence observée au niveau de l'échelle temporelle vient du fait que les opérations n'ont pas été lancées au même moment par



(a) Mesure de la tension aux bornes de la résistance série



(b) Tension aux bornes de la résistance obtenue par simulation du modèle

FIGURE 2.18 : Comparaison du modèle électrique et des mesures sur circuit

rapport à l'origine de l'échelle de temps. Toutefois, les pics de consommation observés apparaissent au même moment ce qui est important pour l'exploitation du modèle.

Nous avons pu analyser les résultats de notre modèle tant sur le plan de la précision temporelle que de l'amplitude de la consommation mesurée. Les résultats obtenus sont suffisamment précis pour être exploités dans le cadre d'étude de vulnérabilité d'un circuit contre des attaques par analyse de consommation. L'établissement du modèle ne comporte que peu de phases supplémentaires par rapport à la caractérisation fonctionnelle du flot de conception. La véritable étape additionnelle est l'évaluation de la capacité de la grille d'alimentation. Toutefois on peut relever certaines limitations à l'utilisation de ce modèle. Si la génération de la consommation électrique se fait rapide-

ment, l'extraction de l'ensemble des capacités est plus complexe. L'étape de simulation pour obtenir la capacité des composants entre V_{dd} et la masse peut s'avérer coûteuse en temps et en ressources de calcul pour des circuits comportant plusieurs dizaines de milliers de portes logiques. On peut donc se poser la question de l'utilisation de ce modèle pour des circuits comprenant un nombre élevé de portes logiques, si on prend l'exemple d'un système sur puce. Une solution pourrait être de décomposer le *SoC* en sous-circuits indépendants et de réaliser l'extraction sur chacun d'entre eux. L'idée est d'additionner ces capacités qui sont en parallèle.

D'autre part, pour essayer de modéliser au mieux la signature électrique, il faut prendre en compte tous les éléments implantés dans le circuit c'est-à-dire la logique à l'intérieur du circuit de test, les *IOs* notamment qui ne sont pas nécessairement caractérisés en même temps que le circuit d'étude.

Si le modèle permet de mettre en exergue certaines vulnérabilités, l'intégration de contremesures est une étape supplémentaire à considérer pour la sécurité des circuits. De la même façon, il va être utilisé pour la validation et le développement de contremesures en phase de conception.

2.3 Évaluation de contremesures à l'aide du modèle

Dans cette partie, différentes méthodes visant à limiter les fuites d'information par la voie de la consommation électrique sont présentées. Plusieurs solutions ont été étudiées dans la littérature depuis le développement de ces attaques. Le but ici est de trouver des solutions peu coûteuses aussi bien en termes de consommation, de surface, de performance mais aussi de complexité ce qui va directement influencer le temps de développement et l'intégration de ces solutions dans le flot de conception. Ces notions ont déjà été introduites dans le premier chapitre, et sont primordiales dans le cadre d'une application industrielle. Tous ces critères vont nous servir à évaluer et caractériser les différentes contremesures que nous allons considérer, en mettant l'accent sur le coût surface.

2.3.1 Les catégories de contremesure

Les contremesures aux attaques par canaux auxiliaires peuvent être classées en deux catégories : les techniques de masquage et les techniques de dissimulation (obfuscation).

2.3.1.1 Techniques de masquage

Les contremesures de masquage sont basées sur la dissimulation des données manipulées. Le principe de cette technique est de rompre la corrélation qui existe entre la donnée manipulée et la puissance consommée en masquant les données intermédiaires. Cette corrélation peut être liée par exemple, au poids de Hamming ou à la distance

de Hamming entre deux données manipulées pendant la même opération. L'opération de masquage consiste à modifier les données en cours d'utilisation dans le circuit avec une donnée aléatoire. Le calcul effectué par une porte logique est caché en masquant la donnée courante avec un masque aléatoire ou pseudo-aléatoire. Le bit masqué peut être retrouvé en fin d'opération. La technique de masquage peut être implantée lorsque l'opération f à protéger est linéaire. Si on considère un groupe G au sens mathématique muni d'une loi *, x un bit et m un masque, la fonction f est linéaire si on a la relation :

$$f(x * m) = f(x) * f(m)$$

Il est donc possible de retrouver la valeur du bit masqué après application du masque. Concernant les opérations non linéaires, deux calculs ont lieu en parallèle : un pour la donnée masquée et l'autre pour la donnée elle-même. c'est notamment le cas de plusieurs algorithmes de cryptographie qui utilisent des tables de substitution (*substitution box*) [Akkar 2001, Standaert 2005, Lu 2010, Maghrebi 2009], ou encore des opérations logiques ou arithmétiques non linéaires [Golic 2007]. À titre d'exemple, l'idée dans [Akkar 2001] est de transformer un masquage booléen (i.e. un masquage utilisant l'opération OU exclusif) par un masquage multiplicatif dans $GF(2^8)$ car, étant donné deux variables x et y dans $GF(2^8)$, on a :

$$(x \cdot y)^{-1} = x^{-1} \cdot y^{-1} \tag{2.16}$$

Une autre approche consiste à masquer le résultat de l'opération logique réalisée par une porte. Cette approche a été présentée par Suzuki et al. dans [Suzuki 2004]. La contremesure appelée RSL (*Random Switching Logic*) est basée sur une modification de la probabilité de transition du signal de sortie des portes logiques. Cette probabilité est ramenée à 1/2 ce qui la rend indépendante de la valeur des signaux d'entrée (à l'exception du masque). La logique RSL est composée de trois portes logiques (porte non-OU, non-ET et OU exclusif). Le même bit de masque est utilisé pour le circuit entier mais peut changer d'un cycle d'horloge à l'autre.

Les techniques de masquage nécessitent généralement de modifier l'architecture du système car il faut rajouter le chemin de données correspondant au masque. L'implantation d'une technique de masquage est donc dépendante du circuit à protéger. De plus, il faut prendre en compte la génération des nombres aléatoires pour construire le masque, et son stockage pour l'utilisation tout au long du calcul. La difficulté est de générer le masque et de l'intégrer dans la fonctionnalité du système.

Implantation d'une contremesure de masquage. On va tester l'efficacité d'une contremesure de masquage sur le circuit d'étude précédent. Les données sensibles sont masquées lorsqu'elles sont extraites du cœur analogique avant d'être manipulées dans le contrôleur numérique. Le même masque est appliqué pour récupérer la donnée correcte à la sortie comme indiqué sur la figure 2.19. Ici on utilise le même masque car les

opérations réalisées dans le contrôleur sont linéaires, on n'a donc pas besoin de faire des transformations particulières sur le masque. Les données doivent être protégées lors de

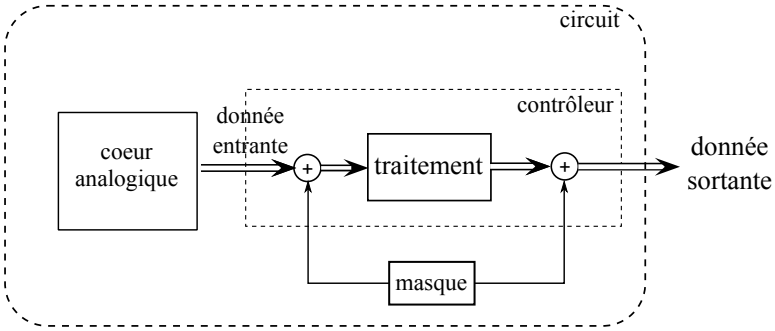


FIGURE 2.19 : Architecture du schéma de masquage implémenté

la transition du bus ainsi que lorsqu'elles sont capturées à l'intérieur du contrôleur numérique. Pour cela, on a besoin d'un masque généré aléatoirement comportant le même nombre de bits que la donnée à protéger. Un nouveau masque est généré à chaque nouvelle opération de lecture, par conséquent le même masque n'est utilisé qu'une seule fois. La valeur du masque aléatoire est disponible quand la donnée sort du cœur analogique. Ce n'est qu'à la fin de l'opération de lecture que la donnée est démasquée. Dans notre implantation, le masquage consiste à faire un OU-exclusif entre la donnée à protéger et le nombre aléatoire avant que la donnée ne soit traitée comme le montre la figure 2.19. Le même masque est appliqué en sortie en fin d'opération. À noter qu'ici la génération du nombre aléatoire est à l'extérieur du circuit numérique, sa consommation ne sera pas comptabilisée dans l'évaluation de la signature du contrôleur. De même, le masque est considéré comme disponible tout au long de l'opération. Son stockage est donc externalisé, par conséquent le coût lié n'est pas évalué de manière quantitative. La deuxième catégorie de contremesures proposées dans la littérature est également abordée.

2.3.1.2 Techniques de dissimulation

Ces techniques peuvent avoir deux objectifs :

- rendre aussi faible que possible la différence de consommation électrique qui peut exister entre deux mêmes opérations utilisant des données différentes.
- rendre la consommation du circuit la plus aléatoire possible.

Cette contrainte peut se décrire formellement comme présenté dans [Mesquita 2005], en utilisant le rapport signal sur bruit (SNR). Si on considère I_c le courant consommé par le circuit attaqué, I_n le courant apporté par la contremesure, et $k = I_c/I_n$ l'atténuation du signal occasionnée par le courant I_n, on a alors :

$$SNR = 20 \cdot log(\frac{I_c}{kN})$$

avec N représentant l'amplitude du bruit. Plus le rapport signal sur bruit est faible, plus il est difficile de corréler le courant mesuré et celui consommé par le circuit. Pour parvenir à réduire ce rapport, différents types de logiques ont été proposées.

La logique différentielle à précharge utilise deux phases, une de précharge et l'autre d'évaluation de telle sorte qu'une seule transition a lieu par cycle et par porte logique. De plus, étant donné qu'on considère le signal de sortie et son inverse, il y a toujours une transition entre la précharge et l'évaluation. La logique de type WDDL (*Wave Dynamic Differential Logic*) a été présentée dans [Tiri 2004]. Les états logiques haut et bas sont représentés par une couple de valeurs complémentaires (1,0) et (0,1) respectivement, avec un séparateur (0,0) au début de chaque cycle d'horloge. Un opérateur de précharge est inséré à l'entrée de la logique combinatoire qui est limitée à trois opérations : ET, OU et NON. Toutefois la logique WDDL a des vulnérabilités : l'évaluation anticipée due à une différence de propagation entre deux signaux d'une porte logique peut modifier l'équilibre de consommation. Il peut aussi y avoir une différence de routage entre un signal et son complémentaire qui déséquilibre la consommation.

Pour éviter ces vulnérabilités, la logique STTL (*Secure Triple Track Logic*) a été introduite [Soares 2008]. Ce type de logique utilise un troisième signal qui permet d'indiquer la validité des signaux d'entrée des portes logiques. Ce signal ne comporte donc aucune information sur la valeur des signaux d'entrée des portes, il s'agit plutôt d'un signal de synchronisation. Il permet de masquer les effets de déséquilibre des charges des signaux dues aux capacités parasites de routage.

Un autre type de logique à précharge a été présenté par Nassar et al. dans [Nassar 2010]. La logique BCDL (*Balanced Cell-based Dual-rail Logic*) utilise un signal de synchronisation commun à plusieurs portes logiques. Cela permet d'une part, d'éviter l'évaluation anticipée et d'autre part d'avoir des temps d'évaluation optimisés par rapport aux autres types de logique différentielle à précharge.

Même si les techniques utilisant les logiques différentielles à précharge peuvent s'avérer efficaces contre les attaques par analyse de consommation, leur implantation reste complexe avec de nombreux surcoûts. Tout d'abord, au niveau de la surface, toutes ces solutions ont un surcoût minimum de 50 % par rapport à un circuit non protégé. Ensuite le temps d'exécution est augmenté de 50 à 75 %. Il peut y avoir aussi des contraintes spécifiques de routage ou de délai de propagation pour assurer la bonne fonctionnalité.

On peut également générer de la consommation parasite en ajoutant du bruit, des instructions "fictives" qui n'ont aucune utilité fonctionnelle mais qui sont présentes uniquement pour rendre difficile l'analyse faite par un attaquant. Les exécutions en parallèle constituent un moyen pour ajouter du bruit. Ces techniques ont été utilisées pour sécuriser une implantation de l'AES dans [Gurkaynak 2005]. L'exécution d'opérations non ordonnées, l'utilisation de périodes d'horloge aléatoires et indépendantes rendent difficile l'analyse de consommation électrique.

D'autres techniques permettent de réguler le courant consommé depuis l'extérieur du circuit à protéger en utilisant un circuit dédié. De telles solutions ont été présentées dans [Mesquita 2005, Ratanpal 2004, Muresan 2008]. Un capteur de courant sert à produire le signal d'entrée d'un amplificateur qui délivre plus ou moins de courant en fonction de l'activité du circuit à protéger. Ces systèmes fonctionnent en boucle de rétroaction. Le niveau de protection dépend de la sensibilité et des caractéristiques des éléments implantés (capteur de courant, bande passante, gain de la boucle etc.) car certains pics de courant peuvent ne pas être atténués. D'autre part, le niveau de régulation du courant est basé sur le pire cas de consommation du circuit, il y a donc un surcoût de consommation qui peut être important. À noter également que ces contremesures sont conçues au niveau transistor et non au niveau portes logiques, ce qui les rend plus difficilement intégrables dans le flot de conception numérique.

Une solution intéressante a été proposée par Shamir [Shamir 2000]. L'idée est d'utiliser deux capacités connectées à l'alimentation qui vont à tour de rôle être chargées à travers la tension d'alimentation et déchargées en étant connectées au circuit à protéger. Le courant tiré depuis la source d'alimentation est donc constant quelles que soient les opérations réalisées et correspond au courant nécessaire pour charger la capacité. Le principal inconvénient de cette méthode est qu'elle nécessite une capacité de valeur importante (100 nF), qui pourrait difficilement être ajoutée dans les circuits intégrés actuels. Néanmoins, on peut se servir de capacités en vue d'atténuer les pics de courant.

Implantation de capacités de découplage intégrées. Dans les circuits synchrones, les transitions du signal d'horloge provoquent des pics de consommation dus au fonctionnement des éléments de mémorisation qui sont, en partie, responsables des fuites d'information. Ces pics ayant une durée très courte, ils peuvent être lissés en utilisant des capacités de découplage. L'effet de lissage dépend d'une part de la valeur initiale du pic de courant et d'autre part de la capacité implantée. Dans le flot de développement numérique, une fois les portes logiques placées et routées, l'espace entre les portes est rempli par des cellules spécifiques (*filler cells*). Pour ajouter de la capacité sur la grille d'alimentation, on remplace les *filler cells* par des capacités de découplage (*decap cells*). Les capacités de découplage usuelles utilisées dans les cellules standard sont constituées de transistors de type N et de type P. Le drain et la source du transistor de type P sont connectés à V_{dd}, la grille étant connectée à la masse. De même, le drain et la source du transistor de type N sont connectés ensemble à la masse, tandis que la grille est connectée à V_{dd} (figure 2.20) [Meng 2006]. L'utilisation de capacités de découplage a déjà été étudiée dans les travaux [Shimazaki 2009, Chen 2008] pour la réduction de bruit d'alimentation. Ici, le but est de dissimuler l'information en réduisant l'intensité des pics de courant reliés aux transitions de données. De cette façon, on peut obtenir une atténuation significative du courant mesuré sans contrepartie en termes de surface. La valeur de capacité ajoutée dépend de la surface libre disponible après l'étape de rou-

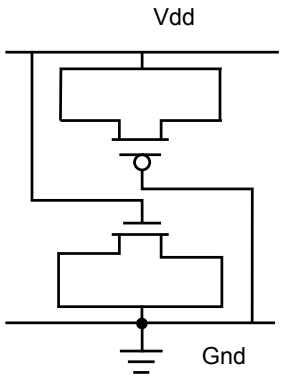


FIGURE 2.20 : Capacité de découplage intégrée dans les cellules standard

tage. Dans le cas où on aurait besoin de plus de capacité, il est possible d'ajouter de la capacité MIM (*metal-insulator-metal*). Ce type de capacité est supporté dans plusieurs procédés de fabrication, qu'il s'agisse d'interconnexions cuivre ou aluminium. Les capacités MIM utilisent généralement le niveau de métal supérieur et son niveau inférieur pour former les électrodes dans le but de minimiser les capacités de couplage parasites entre l'électrode basse et le substrat [Goh 2004]. Toutefois, les niveaux de métaux sont généralement tous utilisés pour réaliser le routage entre les cellules. Cette option n'est donc pas systématiquement disponible.

L'avantage principal de l'utilisation de capacités de découplage est qu'elles ne nécessitent pas de modifier la conception du circuit, car il s'agit de solutions intégrées en phase de placement routage. De plus, elles sont localisées parmi les cellules standard donc il est difficile de les déconnecter en passant par une étape de désassemblage. Toutefois, elles ont pour inconvénients leur dépendance de la technologie utilisée, et d'autre part la capacité totale rajoutée dépend de la surface libre en fin de conception. De ce fait, un circuit déjà dense ne pourra bénéficier que d'une capacité limitée.

Le tableau 2.5 récapitule les avantages et les limitations des différentes contremesures présentées ci-dessus. Elles présentent toutes un surcoût élevé en termes de consommation. Certaines comme la logique différentielle à précharge ou encore les circuits de régulation nécessitent des cellules spécifiques ou de sortir du flot de conception numérique standard.

Afin d'évaluer quantitativement l'efficacité de ces catégories de contremesures, elles ont été implantées dans le circuit qui nous sert de support d'étude. Le modèle développé dans la partie précédente va nous permettre de quantifier leur apport. Le prochain paragraphe présente les résultats obtenus pour la contremesure de masquage et l'utilisation de capacités de découplage intégrées. Ces deux méthodes peuvent convenir dans le cas du circuit étudié car elles présentent un surcoût limité en termes de consommation mais aussi de surface et de performances.

TABLEAU 2.5 : Comparaison des contremesures existantes

Techniques	Masquage	Logique à précharge	Instructions fictives	Régulation
Avantages	Réduction de la corrélation sans modif. directe de la consommation	Diminution rapport signal sur bruit		
Limitations	– Génération et stockage des nb. aléatoires – Surcoût surface et consommation élevé si opérations non-linéaires	– Surcoût surface élevé – Utilisation de portes spécifiques	– Augmentation de la complexité du circuit – Augmentation de la consommation	– Conception *full-custom* – Augmentation de la consommation

2.3.2 Résultats des modèles simulés

Dans cette section on présentera les résultats de simulation du modèle électrique équivalent dans le cas de la contremesure de masquage et de l'implantation de capacités de découplage. Ces deux contremesures sont évaluées sur le contrôleur numérique de mémoire en technologie CMOS 32 nm.

2.3.2.1 Masquage des données

L'intégration d'une architecture de masquage requiert une modification de la conception du circuit. Après la modification de l'architecture au niveau RTL, on passe par l'étape de synthèse logique. Les résultats de l'implantation sont donnés dans le tableau 2.6. On relève des surcoûts surface et consommation relativement faibles par rapport à

TABLEAU 2.6 : Caractéristiques du circuit protégé

Paramètres	**Surcoût du circuit protégé** (sans le générateur de nombres aléatoire)
nombre de portes	28 %
surface	7 %
consommation moyenne	2 %
vitesse	< 1 cycle (quelques centaines de ps sur des chemins non critiques)

l'architecture non protégée. La pénalité au niveau de la vitesse de fonctionnement ne modifie que très légèrement le chemin critique, ce qui permet de fonctionner à la même fréquence que le circuit de référence. Un nouveau masque est généré à chaque fois qu'une nouvelle donnée est disponible à l'entrée du contrôleur, par conséquent, un masque n'est

utilisé qu'une seule fois. Le nombre aléatoire est fixé juste avant l'arrivée des données provenant du cœur analogique. Le démasquage s'effectue en fin d'opération.

Pour évaluer la contremesure on va s'intéresser à la corrélation entre la donnée lue et la consommation électrique du modèle. L'attaque réalisée ici est de type SPA (*Simple Power Analysis*, voir section 1.1.2.1). Le modèle de fuite utilisé dans ce cas est le poids de Hamming c'est-à-dire qu'on considère qu'il y a une relation directe entre le poids de Hamming de la donnée et le courant mesuré. Le procédé de simulation consiste à mesurer à trois instants différents au cours de la simulation l'amplitude du courant. Ces instants ont été minutieusement choisis pour avoir la plus forte corrélation possible sur le circuit non protégé. Il s'agit des instants où des transitions interviennent sur le bus de données. Ceci est possible en connaissant exactement le fonctionnement du système à étudier. Le pire cas en termes de fuite d'informations est donc considéré ici. Pour chacune des données à lire, 30 opérations de lecture sont réalisées, le mot lu étant à chaque fois masqué avec un nombre aléatoire. 30 mesures de courant sont alors obtenues pour chacun des trois instants de mesure. Ces mesures vont ensuite être moyennées et on va calculer le coefficient de corrélation linéaire de Pearson r_p défini par la formule :

$$r_p = \frac{\sum_{i=1}^{N} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2}} \tag{2.17}$$

Dans cette équation, x_i représente le poids de la donnée i, $\bar{x}$ la moyenne des x_i, y_i la valeur du pic de courant enregistrée pour la donnée x_i et $\bar{y}$ la moyenne des y_i. Le coefficient de Pearson mesure la dépendance linéaire de deux variables quelconques. Il est compris entre -1 et 1 respectivement pour des variables inversement proportionnelles et proportionnelles, 0 indique que les deux variables ne sont pas corrélées du tout [Adler 2011]. Les résultats sont présentés sur la figure 2.21. La courbe en bleu clair montre une corrélation quasi maximale entre la consommation et le poids de Hamming pour le circuit de référence. La première implantation du circuit protégé présente une corrélation réduite par rapport au circuit de référence mais la corrélation reste notable. En observant cette implantation de plus près, nous avons remarqué qu'il y avait un ajout de buffers à l'entrée du circuit avant l'opération de masquage. La consommation de ces buffers lors de la transition des données notamment contribuait à augmenter la corrélation. Nous avons donc refait une implantation (implantation 2 sur la figure 2.21) en interdisant l'ajout de cellules avant l'opération de masquage. On relève une nouvelle baisse de la corrélation pour cette implantation. Toutefois, on remarque qu'à l'instant t_1, la corrélation est plus importante qu'aux instants t_2 et t_3. Cela peut s'expliquer par le fait que t_1 est le moment où l'opération de masquage s'effectue. On relève donc ici la consommation des portes logiques OU-exclusif réalisant l'opération de masquage. Concernant la différence de corrélation entre les instants t_2 et t_3 des deux implantations, il peut s'agir de différence d'aléas des masques générés. En effet, pour des nombres aléatoires,

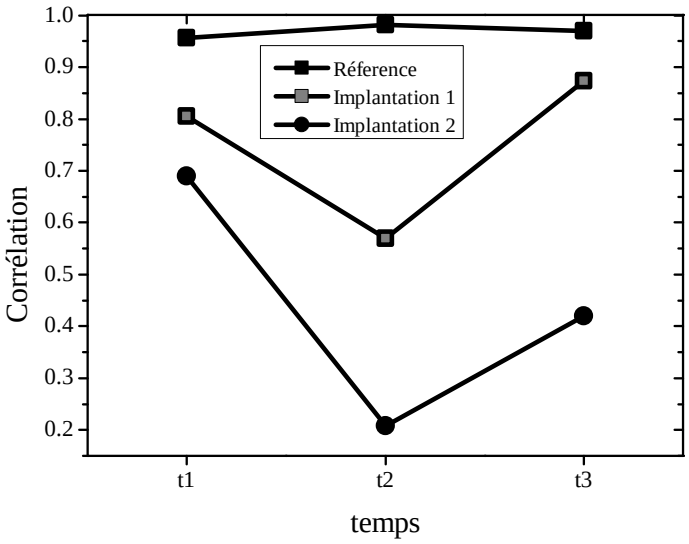


FIGURE 2.21 : Mesure de l'effet de masquage sur la corrélation entre la consommation et le poids de Hamming des données

on peut estimer la consommation moyenne qu'on obtiendra en répétant l'opération de lecture. Si on part de l'hypothèse que les données ayant le même poids de Hamming ont la même consommation, on aura une consommation en fonction du poids de Hamming de la donnée lue. On désigne par (H_i) une donnée dont le poids de Hamming est i, et c_i la valeur du pic de courant mesuré associé. Considérons ici une donnée de poids nul avant masquage. La donnée est composée de 32 bits, donc la donnée est constituée uniquement de 0. La probabilité d'avoir un poids de Hamming nul après masquage est notée $p(H_0)$. Le pic de courant associé à cette probabilité est donné par l'expression :

$$\sum_{k=0}^{32} \binom{32}{k} \frac{1}{2^{32}} c_k \tag{2.18}$$

Pour expliquer ce résultat, on peut voir qu'il y a une seule valeur de masque qui permet de garder un poids de Hamming de 0, si la donnée non masquée a un poids de 0. Cette valeur unitaire est divisée par le nombre total de tirages possibles pour obtenir le coefficient correspondant à c_0, ce qui donne $1/2^{32}$. De la même façon, pour avoir une donnée dont le poids de Hamming est de 1 après masquage, il y a 32 possibilités (chacun des bits de la donnée à 1 et les autres à 0) que l'on divise par le nombre total de combinaisons. Si le masque n'est pas suffisamment aléatoire, chacun des coefficients multipliant les c_k va être modifié, ce qui aura pour conséquence de les rendre non équiprobables. Ainsi, certaines valeurs de données resteront corrélées à la consommation enregistrée.

Nous allons évaluer de la même façon l'approche utilisant des capacités de découplage intégrés dans le circuit. Ce sera l'objet du prochain paragraphe.

2.3.2.2 Capacités de découplage

La méthode de protection consiste à ajouter des capacités de découplage intégrées parmi les cellules standard à la fin de l'étape de routage. L'efficacité de cette solution va dépendre de l'espace libre après le routage d'une part et de l'intensité des pics de courant d'autre part. Afin de mieux évaluer l'impact de cette contremesure, nous avons simulé grâce au modèle développé la signature électrique sur notre circuit d'étude à savoir le contrôleur de mémoire conçu en technologie CMOS 32 nm.

La première étape est d'évaluer la surface disponible en fin de placement routage pour y insérer les capacités. En effet, la valeur des capacités de découplage par unité de surface est connue, cela permet d'estimer la capacité qu'il est possible d'ajouter. Une fois la valeur connue, on peut directement l'intégrer dans le schéma électrique du modèle équivalent, il s'agit d'une capacité parallèle entre le nœud d'alimentation et la masse. Elle vient s'ajouter à la capacité globale incluant les parasites et les composants CMOS. Ainsi, avec la surface disponible dans le contrôleur, il est possible d'ajouter une capacité de 22 pF soit pratiquement autant que la capacité des composants MOS (27 pF). La résistance de mesure est ensuite choisie. Celle-ci est fixée à 1 Ω car même en ayant une différence de potentielle faible aux bornes de la résistance, on est capable de la mesurer en simulation ce qui n'est pas le cas à l'oscilloscope, où il faudrait une valeur minimale de tension. Le tableau 2.7 donne la comparaison entre les implantations avec et sans capacités de découplages intégrées. Les instants t_1, t_2 et t_3 correspondent à des pics de consommation qui ont une forte corrélation avec la donnée (voir section 2.3.2.1).

TABLEAU 2.7 : Évaluation de l'effet des capacités de découplage sur les pics de consommation

Instants	circuit de référence	circuit avec capacité de découplage
t_1	76.8 mV	47.5 mV
t_2	123.6 mV	75.4 mV
t_3	125.9 mV	76.9 mV

On peut voir que l'amplitude maximale des pics de courant est lissée grâce aux capacités. L'amplitude maximale est diminuée d'une valeur allant de 37 à 40 % sans modifications de l'architecture du circuit. L'atténuation de la forme d'onde du courant est présentée sur la figure 2.22. Nous avons ensuite évalué l'apport de la capacité MIM sur le circuit. Le résultat est donné sur la figure 2.23. Cette courbe montre le lissage des pics de courant obtenu en fonction de la capacité ajoutée. Sachant que la capacité MIM

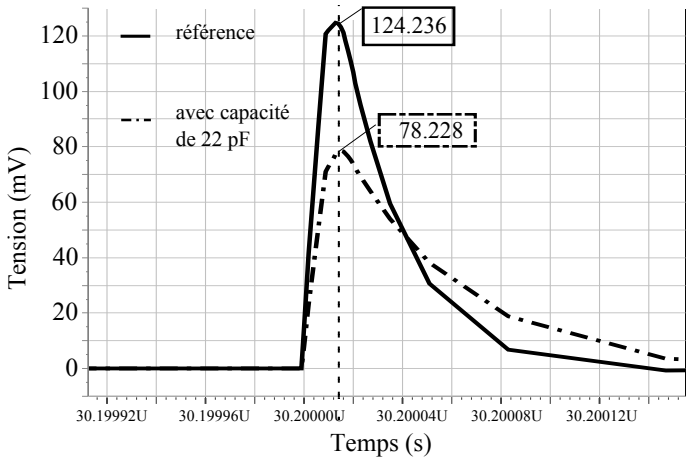


FIGURE 2.22 : Évaluation de l'atténuation de pics de courants avec des capacités de découplages intégrées

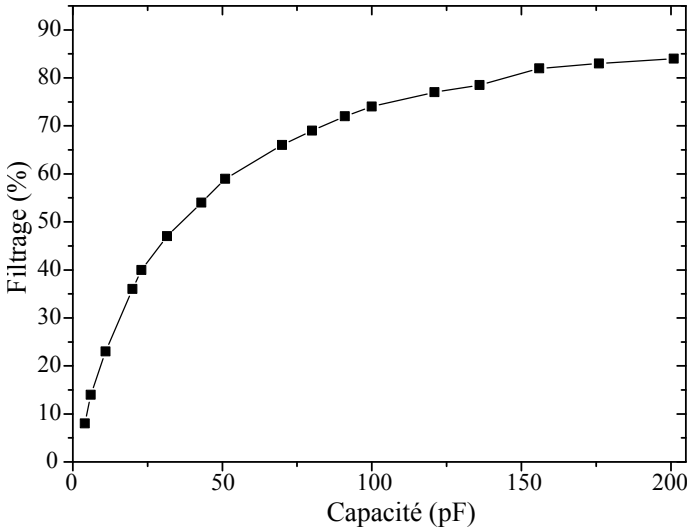


FIGURE 2.23 : Lissage additionnel apporté par la capacité MIM

est placée au-dessus de la surface du circuit, si on recouvre la totalité de sa surface on obtient une capacité d'environ 175 pF, ce qui correspond à une atténuation des pics de courant de 82 %. On remarque que la courbe de lissage en fonction de la capacité a une forme logarithmique avec une asymptote à 100 %. Il est important de rappeler que le lissage obtenu est dépendant de l'intensité du pic de courant consommé par le circuit par conséquent plus le pic est important, moins il sera atténué à capacité équivalente.

Dans cette section on a utilisé un modèle électrique de simulation pour évaluer différentes catégories de contremesures, qu'il s'agisse de technique de masquage ou de dissimulation. Chacune de ces méthodes présente des limitations. Le masquage nécessite

le recours à un générateur de nombres aléatoires ou pseudo-aléatoires. De plus, l'architecture du circuit de départ doit être modifiée de façon plus ou moins complexe en fonction des opérations qui sont effectuées. Son utilisation peut conduire à un surcoût de surface important. En ce qui concerne les capacités de découplage intégrées, leur efficacité dépend fortement de la densité du circuit à protéger, c'est-à-dire la surface occupée par les cellules standard par rapport à la surface totale disponible. Si le circuit est déjà très dense, l'apport des capacités n'aura que peu d'influence sur les pics de courant. Il faudra alors augmenter la surface totale pour bénéficier de plus de capacité ou utiliser une capacité MIM.

En résumé, il est possible de quantifier l'apport de ces contremesures en utilisant le modèle développé. Concernant le masquage, il a permis de diminuer la corrélation d'un facteur 2.5 à 5 avec l'utilisation d'un masquage booléen. À noter que le gain est moins important pour le point d'entrée du masque. Les capacités de découplage intégrées ont permis une atténuation de l'ordre de 40 % des pics de courant. Ce nombre dépend toutefois de la consommation initiale du circuit et de la valeur de capacité qu'il est possible de rajouter. Pour améliorer l'efficacité de cette contremesure, il est important de limiter le courant consommé par le circuit à protéger. Néanmoins, le surcoût en termes de consommation ou de surface est extrêmement limité car seul l'espace initialement inoccupé est utilisé.

Les résultats liés à l'évaluation des contremesures présentés précédemment sont uniquement basés sur le modèle développé. Pour en déterminer l'apport réel, une évaluation sur circuit fabriqué permettrait de valider définitivement l'approche et de procéder à une analyse quantitative. Les résultats obtenus sont néanmoins une première étape vers cette validation.

Les deux techniques implantées ici peuvent être combinées, à la fois pour décorréler les données et la consommation d'une part et lisser les pics de consommation dépendants des données d'autre part. Cela permet dans ce cas d'obtenir une contremesure efficace en limitant les coûts aussi bien en consommation qu'en surface en comparaison avec des méthodes de type logique différentielle à précharge ou des méthodes nécessitant une conception niveau transistor (*full-custom*).

2.4 Conclusion du chapitre

À travers cette étude, nous avons développé un modèle de consommation électrique à partir du besoin de connaître la signature électrique d'un circuit numérique. Nous avons d'abord procédé à l'analyse des paramètres nécessaires à la construction de ce modèle et ensuite, nous sommes partis des outils et des méthodes à notre disposition pour extraire chacun de ces paramètres. La figure 2.24 rappelle les différentes étapes de la construction du modèle. Pour obtenir la signature recherchée, on a besoin de la

consommation électrique du circuit, de la capacité entre le nœud d'alimentation et la masse, et éventuellement de la résistance série sur l'alimentation. Le courant transitoire est obtenu à partir de la *netlist* niveau portes logiques et de la bibliothèque de caractérisation des cellules. Concernant la grille d'alimentation, le dessin de masque au format GDSII est utilisé afin d'extraire la *netlist* niveau transistor ainsi que la capacité due au routage. Ensuite, par une simulation petit signal cette *netlist*, la capacité totale est calculée. Une fois ces paramètres extraits, on obtient un circuit équivalent que l'on peut simuler en SPICE. On peut ainsi combiner la simulation d'un circuit numérique avec un circuit conçu au niveau transistor (circuit analogique). On a ensuite procédé à l'évaluation de ce modèle en le construisant pour un circuit, un contrôleur de mémoire non volatile. On a obtenu des résultats cohérents par rapport aux mesures réalisées sur silicium concernant les pics de courant notamment, qui révélaient le plus d'informations sur le poids de Hamming des données traitées. Dès lors, cette méthode a été utilisée pour évaluer des contremesures face aux attaques se basant sur la signature électrique. Cela a permis de quantifier les avantages et également les limitations des deux solutions étudiées, notamment une complexité d'implantation pour le masquage et une dépendance importante vis-à-vis de la surface pour les capacités de découplage intégrées.

La construction de ce modèle peut s'avérer complexe pour un certain nombre de circuits notamment ceux comportant un grand nombre de transistors. En effet, l'étape limitante est l'extraction de la capacité vue de l'extérieur du circuit entre la masse et l'alimentation. Quelle que soit la méthodologie d'extraction, le temps d'exécution peut être important, voire rendre l'opération infaisable si on prend le cas d'un *SoC* entier. Une solution pourrait être de subdiviser le *SoC* en sous-parties afin d'extraire la capacité de chacune de ces parties de manière indépendante comme indiqué sur la figure 2.24. Sur la droite, le partitionnement permet d'obtenir plusieurs sous-circuits (bloc 1, bloc 2, ..., bloc N). L'analyse petit signal donne la capacité de chacun de ces blocs. Cela peut également servir à répartir les domaines d'alimentation. En effet, les blocs les plus sensibles peuvent être placés dans des domaines d'alimentation à forte capacité, ce qui aura pour effet de réduire les pics de courant, à l'image de l'utilisation de capacités de découplage intégrées. Cela demande néanmoins un travail important de partitionnement et de polariser les différentes parties à leur valeur de fonctionnement.

Les contremesures implantées dans cette étude ont montré des résultats intéressants en termes de protection contre les attaques par analyse de consommation électrique. Une prochaine étape de validation de la démarche serait de tester ces solutions sur silicium, ce qui permettra d'avoir un deuxième niveau de comparaison et de validation du modèle utilisé et ainsi de mener une caractérisation sécuritaire du circuit.

Notre étude s'est concentrée sur les attaques portant sur le nœud d'alimentation, celui-ci étant relativement accessible pour être observé. Il est possible d'en tirer des informations sur l'activité globale du circuit observé, avec une distinction plus ou moins

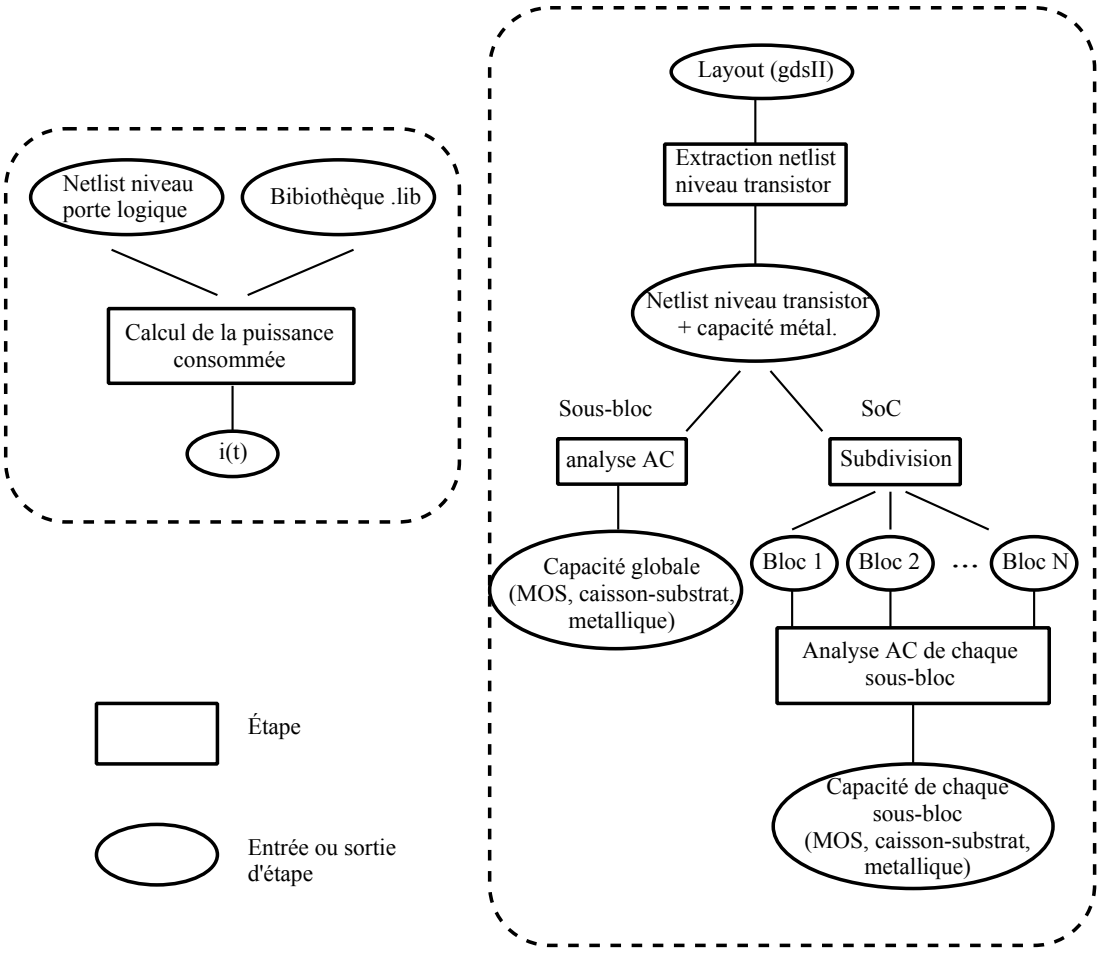


FIGURE 2.24 : Résumé des étapes de construction du modèle électrique

limitée de l'activité individuelle des sous-blocs. Une information supplémentaire peut être apportée par l'analyse électromagnétique qui fait intervenir une composante spatiale en plus de l'intensité de la consommation. Cela pourrait s'avérer utile pour affiner les mesures de consommation et se focaliser sur un sous-système en particulier. On peut noter aussi qu'on a en contrepartie une perte d'information sur la consommation statique, car les mesures électromagnétiques sont basées sur la variation du courant électrique. Toutefois, avec la réduction de la taille des circuits dus aux évolutions technologiques, il serait intéressant de voir à quel point il est possible de faire ressortir une information de consommation locale et la pertinence de cette information.

Au-delà de l'observation des variations de courants électriques lors du fonctionnement du circuit, les rails d'alimentation peuvent également être vecteur de perturbations, qui peuvent se traduire par des fautes injectées dans le circuit. L'étude de ce phénomène est détaillée dans le prochain chapitre.

Chapitre 3

Attaques actives en tension

Sommaire

Parallèlement aux attaques passives (par canaux auxiliaires), plusieurs attaques physiques sont apparues, notamment pour récupérer des informations secrètes d'algorithme de cryptographie [Ali 2011b, Bae 2011, Li 2012, Mirbaha 2013]. Elles sont qualifiées d'attaques actives car elles induisent une perturbation de l'exécution normale des opérations d'un circuit. Lorsqu'une valeur erronée apparait en sortie du circuit, on parle alors d'injection de fautes. Les différentes méthodes d'injections ont été présentées dans le chapitre 1. Parmi ces attaques, il y en a qui ont pour vecteur la tension d'alimentation. En effet, la mise en œuvre de ces attaques ne demande pas d'expertises particulières ou de préparation du circuit. A cela s'ajoute le fait que tous les circuits nécessitent une tension d'alimentation pour fonctionner, ce qui les rend d'autant plus vulnérables. Après avoir étudié les fuites d'information à travers la tension, ce chapitre se concentre sur la détection de fautes injectées par ce biais, plus particulièrement en envoyant des impulsions de durée variable sur la tension d'alimentation (à la fois en sous-tension et en surtension).

Tout d'abord, il est nécessaire de bien cerner les mécanismes mis en jeux lors de ces différentes attaques. Une partie de cette étude sera donc consacrée à l'analyse des fautes injectées par des modifications abruptes de la tension d'alimentation. Le but est de mettre en évidence le comportement des cellules qui composent le circuit, mais aussi de voir l'impact sur les chemins de données. Une fois les mécanismes identifiés, la deuxième partie portera sur la détection des fautes induites par ces attaques en tension. Les contremesures devront à la fois être efficaces dans la détection d'erreurs, et avoir un impact limité sur les performances et le coût du système. De plus, le développement de ces solutions doit s'intégrer dans le flot de conception numérique afin d'être utilisables facilement au niveau *SoC*. Enfin, un bilan sera dressé sur les attaques par impulsion sur la tension puis les perspectives de ce travail seront proposées.

3.1 Attaques par impulsions sur l'alimentation

Dans cette partie, les effets des attaques par modification de la tension d'alimentation sont étudiés sur les circuits numériques. Cela permettra de mieux comprendre les phénomènes intervenant lors d'une attaque. Les attaques par impulsions sur l'alimentation représentent un cas particulier des attaques en tension. Dans ce cas, la tension d'alimentation est modifiée sur une durée inférieure à une période et pouvant aller jusqu'à plusieurs cycles d'horloge. Le choix de la durée et du moment d'injection peut servir à cibler une opération particulière.

3.1.1 Rappels théoriques

Le principe et les contraintes de fonctionnement des circuits numériques synchrones sont rappelés dans les paragraphes suivants.

3.1.1.1 Contraintes temporelles de fonctionnement

Les circuits synchrones sont cadencés par une horloge, c'est-à-dire, la valeur des signaux est mise à jour à la sortie des bascules à chaque nouveau front d'horloge. Généralement, les bascules fonctionnent sur front montant d'horloge pour valider les données qui transitent. Entre deux fronts d'horloge consécutifs, les données passent par un ensemble de portes logiques combinatoires et doivent arriver à l'entrée du registre pendant une certaine durée avant le front d'horloge suivant. Cette durée constitue le temps d'initialisation (*setup time*) du registre. De même, les données doivent rester stables à l'entrée des registres pendant une courte durée après le front d'horloge, appelée temps de maintien ou *hold time* (figure 3.1). Pour fonctionner correctement, chaque chemin de données entre deux registres doit respecter les contraintes définies par (3.1) et (3.2).

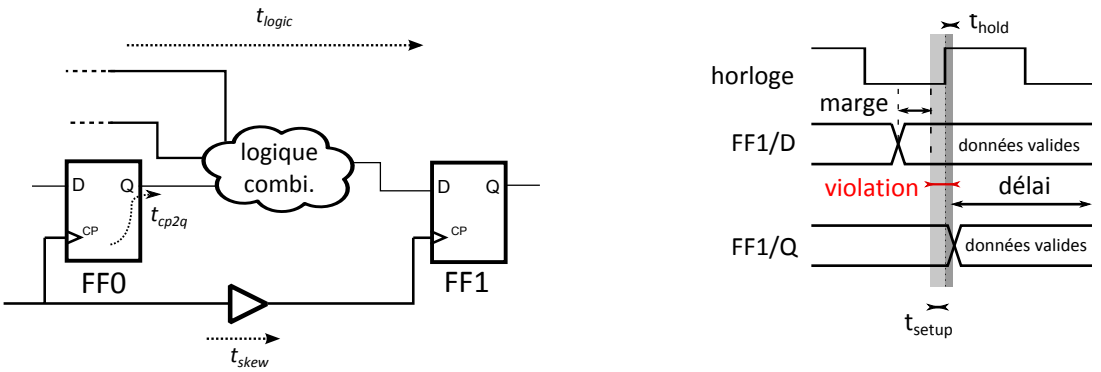
$$T > t_{cp2q} - t_{skew} + t_{logic} + t_{setup} \tag{3.1}$$

$$t_{hold} < t_{cp2q} - t_{skew} + t_{logic} \tag{3.2}$$

Dans ces inéquations :

- T est la période du signal d'horloge,
- t_{cp2q} est le temps de traversée de la donnée à l'intérieur de la bascule,
- t_{logic} est le temps de propagation à travers la logique combinatoire,
- et t_{skew} est la différence entre les temps d'arrivée du signal d'horloge aux entrées des deux bascules.

t_{setup} et t_{hold} sont liés à la bascule d'arrivée FF1 (figure 3.1).



(a) Illustration d'un chemin logique

(b) Contraintes de temps de propagation

FIGURE 3.1 : Temps de propagation d'un chemin en logique synchrone

La figure 3.1(b) montre le diagramme temporel d'un chemin séquentiel. Trois zones peuvent être distinguées :

- avant le front d'horloge (défini comme la marge) : la marge est la différence de temps entre l'arrivée de la donnée à l'entrée de la bascule et le début de la fenêtre

de temps de *setup*. Lorsqu'elle est positive, les contraintes de temps de propagation sont respectées.

- entre le temps de *setup* et le temps de *hold* : la sortie de la bascule FF1 est dans un état métastable, elle va prendre une valeur logique aléatoire entre 0 et 1. A ce stade, la donnée capturée peut donc être correcte ou erronée.
- après le front d'horloge (après le temps de *hold*) : la donnée n'est plus dans la zone d'incertitude, la valeur en sortie de bascule FF1 est donc stable. Toutefois, étant donné que le temps de propagation est plus long que la période, i.e. le signal d'entrée de la bascule ne change pas avant le front d'horloge, la sortie garde la valeur précédente. Ce type d'erreur est qualifié de faute de délai.

Le respect des temps de *setup* et *hold* est donc primordial pour le bon fonctionnement d'un circuit synchrone. Ce sont des caractéristiques de chaque élément de mémorisation (bascules, verrous etc.). Le paragraphe suivant présente la définition de ces temps caractéristiques sur des bascules de type maître-esclave.

3.1.1.2 Définition des temps de *setup* et *hold* sur des bascules

Pour mieux comprendre les temps de *setup* et de *hold*, le fonctionnement d'une bascule D est analysé. La figure 3.2 représente l'architecture d'une bascule maître-esclave de type D (*master-slave*). Lorsque le signal d'horloge (CLK) est à l'état bas, la

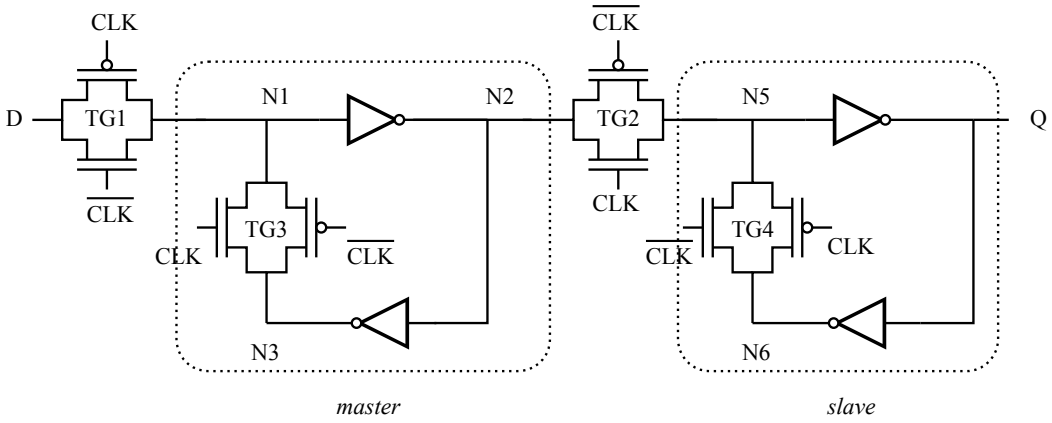


FIGURE 3.2 : Bascule maître-esclave de type D

porte de transmission à l'entrée TG1 est passante et TG3 est bloquée. Le chemin D-N1-N2-N3 est activé, le nœud N2 prend la valeur $\overline{D}$ et N3 prend la valeur D. La porte TG2 est bloquée mais TG4 est passante. Le chemin N5-Q-N6 est actif c'est-à-dire Q maintient la valeur initialement présente sur la sortie. Lorsque le signal d'horloge passe à l'état haut, TG1 et TG4 deviennent bloqués et inversement, TG2 et TG3 deviennent passants. Les chemins N1-N2-N3-N1 et N2-N5-N6 sont maintenant actifs. La valeur du nœud N2 ($\overline{D}$) est transmise au nœud N5 ce qui donne la valeur D à la sortie Q. On peut

noter ici que la sortie Q est mise à jour lors de la transition de l'horloge de l'état bas à l'état haut. Il s'agit donc d'une bascule fonctionnant sur front montant de l'horloge.

Les temps de *setup* et *hold* peuvent maintenant être définis. Comme observé précédemment, lorsque CLK est à l'état bas, le chemin D-N1-N2-N3 est actif. La transition de l'horloge de l'état bas à l'état haut va permettre à l'étage esclave de verrouiller la valeur présente sur le nœud N3. Il faut donc que la donnée soit stable sur ce nœud avant le front montant de CLK. Par conséquent, la donnée doit arriver au nœud N3 une durée t_{setup} avant la transition de l'horloge qui correspond au temps de traversée de TG1 et des deux inverseurs tête-bêche : il s'agit du temps de *setup*.

Il a été vu dans la section précédente que le temps de *hold* était défini par rapport au front d'horloge. De plus, les signaux CLK et $\overline{CLK}$ ne sont pas établis au même moment. Il y a un délai, généralement très court entre ces deux signaux, par conséquent, les portes de transmission ne sont pas activées instantanément. Il est donc nécessaire de maintenir une valeur stable à l'entrée D pendant le blocage de TG1 pour avoir également une valeur stable sur le nœud N1 qui à son tour va se propager vers la sortie Q, d'où l'existence de la contrainte de temps de *hold*.

Contrairement au temps de *setup* qui est toujours positif, le temps de *hold* peut être positif, nul ou même négatif. En effet, en fonction du type de bascule D, il peut y avoir de la logique entre l'entrée D et la première porte de transmission TG1 pour différentes raisons : logique de chaîne de scan, d'activation, de réinitialisation, etc. Notons t_{add} le temps de propagation entre l'entrée D et TG1 (incluant le délai de la logique additionnelle), et t_{TG} le temps nécessaire pour rendre passant ou bloqué la porte de transmission TG1. Trois cas peuvent alors être identifiés :

- $t_{TG} > t_{add}$: dans ce cas le temps de *hold* est positif. La donnée doit rester stable après le front d'horloge.
- $t_{TG} = t_{add}$: le temps de *hold* est nul.
- $t_{TG} < t_{add}$: le temps de *hold* est négatif. La donnée doit donc rester stable avant le front d'horloge. t_{hold} étant généralement court, cette durée est incluse dans le temps de *setup* de la bascule (figure 3.3).

Ainsi, les temps de *setup* et *hold* sont liés à des temps de propagation ou à des différences de temps de propagation de cellules logiques. Cette analyse servira pour étudier les effets de la variation de tension sur les circuits logiques.

3.1.2 Étude de l'impact de la variation de la tension sur la logique

La tension d'alimentation est un paramètre important pour le fonctionnement de tout circuit. Toute variation de ce paramètre implique des modifications plus ou moins critiques du comportement des portes logiques et donc du circuit. Cette partie s'intéresse aux conséquences des modifications de la tension d'alimentation aussi bien en régime quasi-statique (modification lente) qu'en régime dynamique (envoi d'impulsions).

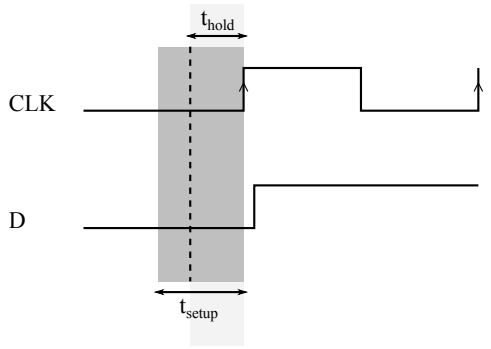


FIGURE 3.3 : Illustration d'un temps de *hold* négatif

3.1.2.1 Étude des temps de propagation

Les temps de propagation ont une influence directe sur la fréquence de fonctionnement du système et donc sur ses performances. Pour comprendre comment sont modifiés ces temps de propagation, l'inverseur CMOS présenté sur la figure 3.4 est étudié. Le

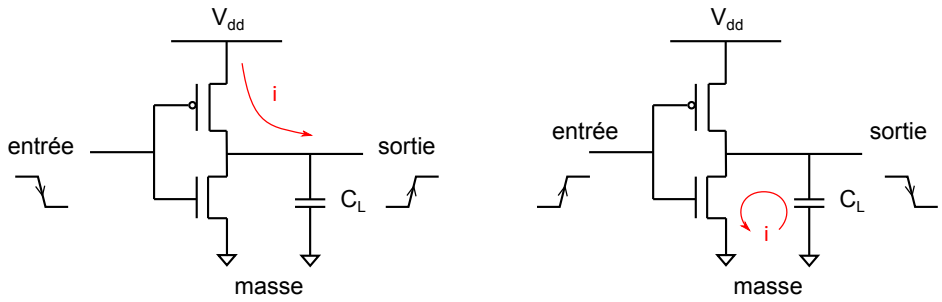


FIGURE 3.4 : Illustration des temps de propagation d'un inverseur CMOS

temps de propagation d'une donnée de l'entrée vers la sortie est donné par l'équation suivante [Sedra 2010] :

$$t_{prop.r} = \frac{C_L \left[\frac{2\left|V_{th,p}\right|}{V_{dd} - \left|V_{th,p}\right|} + ln\left(3 - 4\frac{\left|V_{th,p}\right|}{V_{dd}}\right)\right]}{\mu_p C_{ox} \frac{W_p}{L_p}\left(V_{dd} - \left|V_{th,p}\right|\right)}, \tag{3.3}$$

avec :

- $t_{prop.r}$ le temps de montée de l'inverseur
- C_L la capacité de charge,
- $V_{th,p}$ la tension de seuil du transistor de type P,
- V_{dd} la tension d'alimentation,
- μ_p la mobilité des trous,
- C_{ox} la capacité de l'oxyde de grille

– W_p/L_p le ratio largeur sur longueur du transistor de type P.

Pour le temps de descente, $V_{th,p}$, μ_p et W_p/L_p sont remplacés respectivement par $V_{th,n}$, μ_n et W_n/L_n, qui représentent les mêmes paramètres pour le transistor de type N.

Les temps de propagation des portes logiques en fonction du temps sont similaires à celui de l'inverseur CMOS. L'équation qui définit les temps de propagation d'une porte logique est plus complexe. Cependant, l'évolution en fonction de la tension est la même. La figure 3.5 présente l'inverse des temps de propagation (1/t) de différentes portes logiques (différentes fonctions et tailles) pour une gamme de tensions allant de 0.6 V à 1 V. Notons f_{prop} cette grandeur, qui a la dimension d'une fréquence. Cette

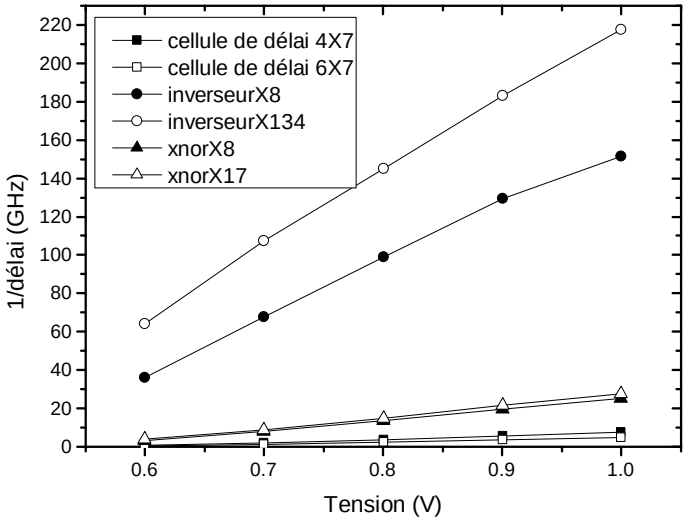


FIGURE 3.5 : Variation des temps de propagation pour différentes portes logiques

caractérisation temporelle a été réalisée en simulation sur une technologie CMOS 28 nm. Les temps de transition ont été relevés à 50 % de la valeur finale aussi bien pour les montées que les descentes de signaux. A noter que la transition du signal d'entrée et la capacité de charge de toutes les portes logiques simulées sont identiques. On peut voir ici que f_{prop} est proportionnel à la tension d'alimentation. En effet, en effectuant comme approximation $|V_{th}| = 0.4V_{dd}$ dans l'équation 3.3, celle-ci donne :

$$\begin{aligned} t_{prop.r} &= \frac{2.78 \cdot C_L}{\mu_p V_{dd} C_{ox} (W_p/L_p)} \\ f_{prop.r} &= 1/t_{prop.r} \end{aligned} \tag{3.4}$$

Le résultat ci-dessus peut être généralisé à l'ensemble des portes logiques, ce qui explique les courbes obtenues. Pour une même charge, chaque porte a une pente différente en fonction de sa capacité à délivrer du courant (*drive*).

Variation du temps de propagation en fonction de la tension. Afin de comprendre l'influence de la taille des portes sur les temps de propagation, les courbes de la figure 3.5 sont analysées. Les inverseurs X8 et X134 réalisent les mêmes fonctions mais ont des capacités différentes à délivrer du courant. Les pentes de leurs courbes sont également différentes : la porte ayant des largeurs de transistors plus grandes (inverseurX134) a une pente plus importante que celle ayant des transistors de largeurs (W) plus petites (inverseurX8). Pour comprendre cette différence, intéressons-nous à l'équation 3.4 avec le cas de l'inverseur CMOS. La pente des courbes représente la variation de f_{prop} en fonction de la tension d'alimentation. Cette variation est notée f'_{prop}, elle est donnée par la dérivée de f_{prop} par rapport à la tension

$$f'_{prop} = \frac{\mathrm{d}(f_{prop})}{\mathrm{d}V_{dd}} = \frac{\mu C_{ox}(W/L)}{2.78 \cdot C_L} = \frac{1}{k} \cdot W/C_L, \tag{3.5}$$

avec $k = 2.78 \cdot L/(\mu \cdot C_{ox})$. La variation de la pente dépend directement du rapport W/C_L : pour une capacité de charge fixée, plus la largeur du transistor est grande, plus la variation de f_{prop} est importante (pour une longueur L donnée).

Dans le cas d'un chemin composé de N inverseurs, notons f_{path} l'inverse du temps de propagation pour parcourir l'ensemble des N inverseurs, f'_{path} sa dérivée par rapport à la tension d'alimentation V_{dd}, C_i et W_i respectivement la capacité de charge et la largeur du transistor de l'inverseur i ($1 \leq i \leq N$). À noter que W_i peut être ici $W_{n,i}$ ou $W_{p,i}$ en fonction de du temps de propagation considéré (temps de montée ou de descente). Ainsi :

$$f_{path} = \frac{1}{\sum_{i=1}^{N} \frac{k}{V_{dd}} \frac{C_i}{W_i}} \tag{3.6}$$

$$\begin{aligned} f'_{path} &= \frac{\mathrm{d}}{\mathrm{d}V_{dd}} \left(\frac{1}{\frac{k}{V_{dd}} (C_1/W_1 + \cdots + C_N/W_N)} \right) \\ &= \frac{1}{k} \frac{1}{\sum_{i=1}^{N} \frac{C_i}{W_i}} \end{aligned} \tag{3.7}$$

Chaque terme du dénominateur de l'équation 3.7 est positif, il en résulte que plus il y a de cellules dans le chemin, plus la valeur de la dérivée est faible. Comme la pente de la courbe qui représente f_{path} en fonction de V_{dd} est la dérivée définie par l'équation 3.7, il en ressort deux résultats :

- la pente (f'_{path}) décroît quand le nombre de cellules du chemin augmente.
- un chemin rapide a une pente plus importante qu'un chemin lent.

En effet, un temps de propagation rapide signifie que le ratio C/W de chaque inverseur est faible, ainsi la pente qui dépend de l'inverse (W/C) est donc élevée.

L'exemple précédent prenant en compte un chemin composé d'inverseurs est un cas particulier car pour un inverseur, l'étage d'entrée est également l'étage de sortie.

C'est également le cas de certaines portes élémentaires inversées telles que Non-OU et Non-ET. Les autres portes logiques sont composées d'au moins deux étages : un étage d'entrée et un étage de sortie (figure 3.6). Le temps de propagation à travers une porte

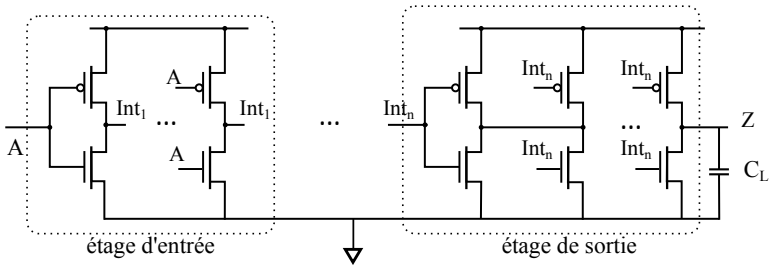


FIGURE 3.6 : Étages d'entrée et de sortie d'une porte logique

logique est la somme des temps de propagation de chacun des étages. Par conséquent, f_{path} ainsi que sa dérivée dépendent du rapport W/C de chaque étage où C est la capacité d'entrée de l'étage suivant et W la largeur du transistor de l'étage courant (W_p ou W_n). Finalement, les résultats obtenus pour un chemin composé d'inverseurs peuvent être étendus à un chemin composé de portes logiques CMOS.

En résumé, la variation de la tension d'alimentation a des effets plus ou moins marqués en fonction des caractéristiques des portes logiques qui composent un chemin. La fréquence maximale atteignable par un chemin long est moins influencée par une variation de tension. De même, à capacité de charge constante, une porte logique ayant une faible capacité à délivrer du courant a une variation de fréquence maximale (f_{prop}) moindre. Ces résultats sont importants pour contrôler des temps de propagation des chemins logiques.

Capacités parasites. Dans les nœuds technologiques avancés, les éléments parasites tiennent une place de plus en plus importante, notamment à cause de la réduction de la taille des composants. La figure 3.7 indique l'évolution de la place des éléments parasites dans les temps de propagation pour différentes technologies. À titre d'exemple, les éléments parasites dus aux interconnexions et aux composants élémentaires représentent plus de 60 % des délais en technologie CMOS 28 nm [Shah 2009]. Les éléments parasites majoritaires sont dus aux interconnexions. Ils dépendent du type de métaux utilisés et de la température. En effet, la température modifie la résistivité des matériaux. Par contre, le changement de tension d'alimentation ne modifie pas les caractéristiques des matériaux. La tension n'a donc pas d'influence directe sur les éléments parasites dus aux interconnexions.

Ce paragraphe a montré la sensibilité des temps de propagation des portes logiques vis-à-vis de la tension d'alimentation. La modification de ces paramètres a des réper-

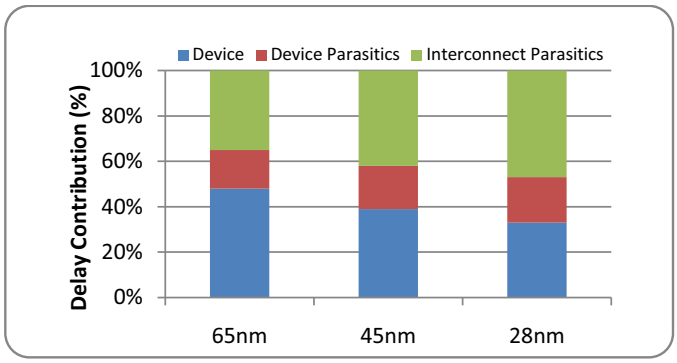


FIGURE 3.7 : Impact des éléments parasites pour des nœuds technologiques avancés [Shah 2009]

cussions sur les contraintes temporelles de fonctionnement qui vont être abordées par la suite.

3.1.2.2 Conséquence sur les paramètres des contraintes temporelles

Le but ici est d'étudier les conséquences des variations de tension sur l'évolution des contraintes temporelles. Le membre de gauche de la contrainte temporelle de *setup* (inéquation 3.1) est la période de l'horloge du circuit. Ce signal est considéré ici comme externe, c'est-à-dire qu'il n'est pas perturbé par une modification de la tension d'alimentation. Une violation de la contrainte temporelle de *setup* intervient si le membre de droite de l'inéquation devient supérieur à la période. Pour cette raison, l'analyse est faite dans le cas où le membre de droite augmente.

La partie droite de l'inéquation 3.1 est composée de t_{logic}, t_{skew}, t_{cp2q} et t_{setup}. t_{logic} représente le temps de propagation d'un ensemble de portes logiques. L'analyse effectuée dans la section précédente montre que t_{logic} augmente lorsque la tension d'alimentation est abaissée. t_{skew} est la différence de propagation du signal d'horloge entre deux bascules d'un chemin. Ce temps peut être dû au routage ou à la propagation à travers les *buffers* de l'arbre d'horloge. Généralement, les circuits sont conçus de façon à équilibrer l'arbre d'horloge, c'est-à-dire que les différences de propagation de l'horloge d'une bascule à l'autre sont minimales. Dans ce cas, une diminution de la tension peut augmenter t_{skew}, mais restera inférieur à t_{logic} par exemple car la logique est souvent composée de plusieurs portes. Enfin, t_{cp2q} est le temps nécessaire pour mettre à jour la sortie Q de la bascule à partir de l'arrivée du front du signal d'horloge. En reprenant la figure 3.2, ce temps correspond au parcours du chemin N3-N1-N2-N5-Q.

Excepté t_{skew} et T, tous les paramètres de l'inéquation 3.1 sont des temps de propagation à travers des portes logiques. Une baisse de la tension d'alimentation entraine une hausse systématique de ces temps de propagation.

Contrairement à la contrainte temporelle de *setup*, l'inéquation 3.2 ne dépend pas de la période. t_{logic}, t_{skew} et t_{cp2q} sont les mêmes paramètres que ceux de l'inéquation 3.1. t_{hold} est défini comme une différence entre deux paramètres, comme montré dans la section 3.1.1.2. Lorsque la tension d'alimentation diminue, t_{skew} et t_{hold} pourraient augmenter ou diminuer. Dans la pratique, ces deux termes augmentent car l'augmentation des temps de propagation fait augmenter également leur différence. Par conséquent, aussi bien t_{hold} d'une part, que la somme $t_{logic} + t_{skew} + t_{cp2q}$ d'autre part augmentent. Toutefois, t_{hold} et t_{cp2q} sont généralement définis par un nombre réduit de portes logiques en comparaison avec t_{logic}. En s'appuyant sur l'analyse faite dans la section 3.1.2.1, à savoir que le nombre de portes d'un chemin fait diminuer sa variation en fréquence, tout en diminuant son temps de propagation, la variation de t_{logic} est plus grande que celle de t_{hold} et de t_{cp2q}. Finalement, une baisse de tension n'est pas préjudiciable à la contrainte temporelle de *hold*.

La contrainte temporelle de *hold* est par contre plus sensible à une hausse de tension. En effet, si t_{skew} est dû essentiellement au routage, la partie gauche de l'inéquation (t_{hold}) peut décroitre plus vite que la partie droite ($t_{logic} + t_{skew} + t_{cp2q}$). À titre d'exemple, la figure 3.8 donne une comparaison des paramètres intervenant dans la contrainte temporelle en *hold*. Les temps sont donnés pour le même chemin (composé d'une seule porte logique ici) et pour plusieurs valeurs de tension. Pour ce chemin, t_{cp2q}

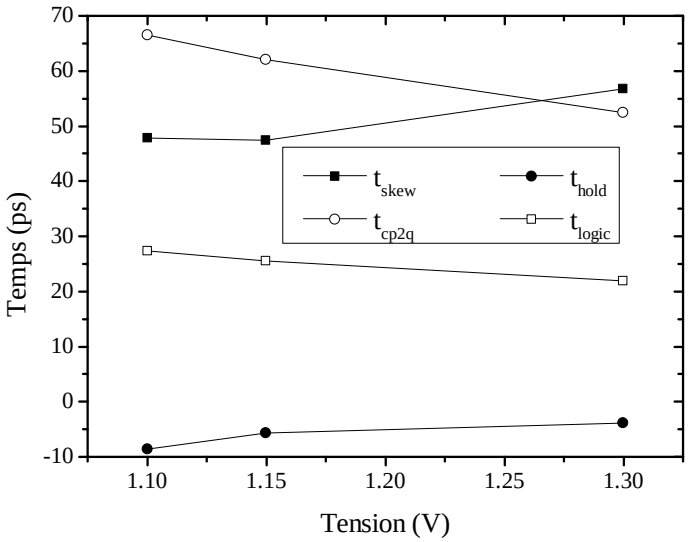


FIGURE 3.8 : Évolution des paramètres de la contrainte en *hold* sur un chemin

a la plus grande valeur, suivi de t_{logic}, t_{skew} et finalement t_{hold}. Cependant, si tous les paramètres diminuent (en valeur absolue) lorsque V_{dd} augmente, ce n'est pas le cas de t_{skew} qui présente une variation non uniforme et devient supérieur à t_{cp2q}. Puisque t_{skew} est une différence de temps de propagation entre deux chemins d'horloge, ce temps peut

augmenter même si le temps de propagation de ces deux chemins diminue simultanément. Ainsi, la somme $t_{logic} + t_{skew} + t_{cp2q}$ peut être réduite et provoquer une violation de l'inéquation 3.2.

En résumé, les paramètres intervenant dans les contraintes temporelles ont des variations en fonction de la tension d'alimentation similaires aux temps de propagation des portes logiques. De plus, la contrainte temporelle de *setup* est plus sensible à la baisse de tension tandis que celle de *hold* est plus sensible à une hausse de tension. Ces résultats sont récapitulés dans le tableau 3.1.

TABLEAU 3.1 : Effet de la variation de tension d'alimentation sur les contraintes de temps de propagation

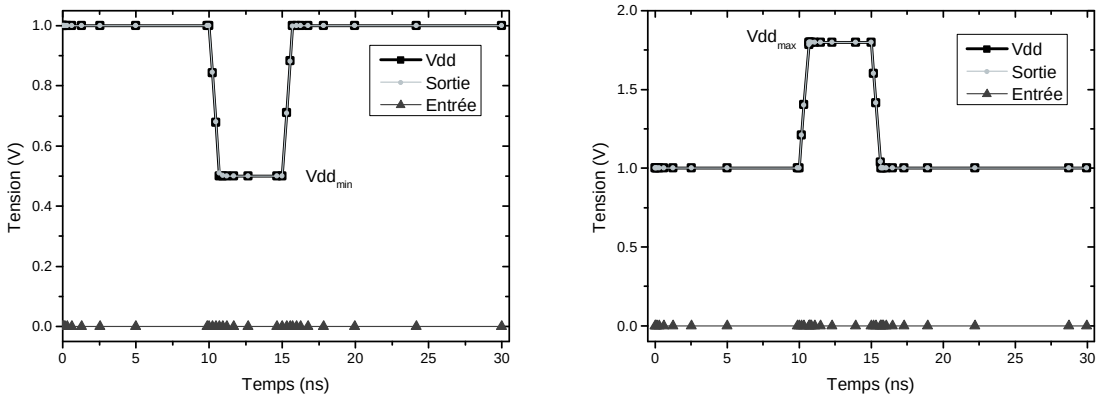
Variation de la tension	Inéquation *setup*	Inéquation *hold*
+	favorable	défavorable
-	défavorable	favorable

Ce paragraphe a été consacré aux modifications des propriétés temporelles de la logique, par la suite, le comportement de la logique en présence d'impulsions sur l'alimentation est analysé.

3.1.2.3 Fonctionnement de la logique

La logique dans les circuits synchrones est composée de portes combinatoires et d'éléments de mémorisation (bascules, verrous etc.). Lorsqu'une impulsion est appliquée sur la tension d'alimentation des portes, celles-ci vont avoir un comportement transitoire qui dépend de la forme de l'impulsion (surtension ou sous-tension), de son amplitude et de sa durée. Tout d'abord, le fonctionnement des cellules combinatoires en présence d'impulsions de tension est étudié, suivi ensuite des éléments de mémorisation.

Étude de la logique combinatoire. Le comportement d'un inverseur CMOS tel que présenté sur la figure 3.4 est analysé. Lorsqu'une impulsion est appliquée sur l'alimentation, la valeur de tension de l'état logique haut suit la variation de la tension d'alimentation, soit vers $V_{dd_{min}}$, soit vers $V_{dd_{max}}$, comme indiqué sur la figure 3.9. En effet, dans ce cas, le transistor de type P est passant par conséquent, la valeur de la sortie est ramenée à celle du nœud V_{dd}. La figure 3.9 montre les résultats de simulation d'impulsions sur l'alimentation d'un inverseur. Dans le cas d'une impulsion positive, la valeur de tension correspondant à l'état haut est modifiée mais la valeur logique n'est pas changée. Ce résultat peut être généralisé à toute porte logique combinatoire car le ou les transistors de type P des étages de sortie vont avoir un comportement identique. En résumé, la fonctionnalité de la porte logique n'est pas modifiée tant que le niveau de



(a) Impulsion négative (b) Impulsion positive

FIGURE 3.9 : Modification de la sortie d'un inverseur en présence d'une impulsion sur l'alimentation

la tension d'alimentation reste supérieur à la tension de basculement de la porte logique de l'étage suivant.

Étude de la logique séquentielle. Le fonctionnement d'un élément de logique séquentielle en présence d'impulsions négatives sur la tension d'alimentation a été présenté dans [Djellid-Ouar 2006]. La réponse d'un verrou (*D-latch*), qui est un élément bistable composé de deux inverseurs tête-bêche, y est analysé face à différentes impulsions. L'analyse petit signal de la réponse en tension de l'élément bistable a permis d'obtenir le rapport entre les signaux de sortie des inverseurs et la tension d'alimentation :

$$\frac{\Delta V_A}{V_{dd}} = \frac{5\beta^2+7\beta}{5\beta^2+11\beta+5} \qquad \frac{\Delta V_B}{V_{dd}} = \frac{5\beta^2+4\beta}{5\beta^2+11\beta+5} \tag{3.8}$$

$$\frac{V_B}{V_A} = \frac{5\beta+4}{5\beta+7} \tag{3.9}$$

avec $\beta = W_p/W_n$. Ces rapports étant inférieurs à 1, le verrou ne peut pas basculer d'un état à un autre. Toutefois en appliquant une impulsion négative d'amplitude supérieure à V_{dd}, le point de fonctionnement du verrou peut être changé [Djellid-Ouar 2006]. Ce cas ne sera pas considéré dans cette étude pour plusieurs raisons. Tout d'abord, les fautes injectées par ce moyen sont difficilement exploitables car l'ensemble des portes logiques y compris les éléments de mémorisations sont perturbés. Cette attaque ressemble à une coupure d'alimentation, le circuit peut difficilement répondre. Ensuite, un moyen simple de détecteur ces attaques est de mettre en place deux bascules témoins conservant des valeurs logiques opposées. La comparaison permanente des valeurs de sorties permet de détecter une anomalie.

La figure 3.10 montre un verrou de type D au niveau transistor. D est le signal

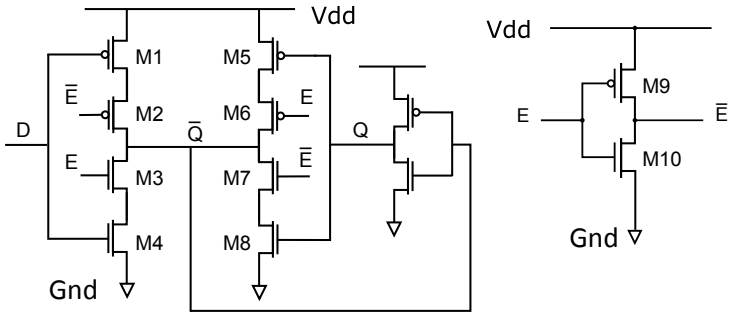


FIGURE 3.10 : Un verrou en technologie CMOS

d'entrée du verrou, E est le signal d'activation. Lorsque E est à l'état haut, le verrou est transparent, sinon le verrou est dans l'état bloqué et la sortie Q maintient son état. Le comportement de ce verrou en présence d'une impulsion positive sur la tension d'alimentation est décrit ci-dessous. Considérons une impulsion de valeur maximale $V_{dd_{max}} = 3V_{dd}$. Lorsque le verrou est dans l'état bloqué, c'est-à-dire $V_E = \boldsymbol{Gnd}$, $V_{\bar{E}} > V_{dd}$ pendant l'impulsion. Les transistors M2 et M3 sont bloqués, par conséquent, ni $\bar{Q}$ ni Q ne peuvent basculer : l'impulsion ne modifie pas le comportement du verrou. Lorsque le verrou est transparent, $V_E > V_{dd}$, $V_{\bar{E}} = \boldsymbol{Gnd}$. Si $V_D = \boldsymbol{Gnd}$, M1 et M2 sont passants, $V_{\bar{Q}}$ monte à $V_{dd_{max}}$, entraînant V_Q à $\boldsymbol{Gnd}$. Si $V_D > V_{dd}$, M3 et M4 sont passants, donc $V_{\bar{Q}}$ descend à $\boldsymbol{Gnd}$ et V_Q monte à $V_{dd_{max}}$. Ainsi, la fonctionnalité du verrou n'est pas modifiée.

Les conséquences de l'application d'impulsions sur l'alimentation des portes logiques viennent d'être analysées. Le comportement des signaux de sortie des portes ne modifie pas directement leur fonctionnalité, c'est-à-dire la fonction logique des portes est respectée. Cependant, la combinaison des modifications transitoires des valeurs de tension et des modifications des caractéristiques temporelles des cellules logiques peut engendrer des mauvais fonctionnements. Ce sera l'objet de la section suivante.

3.1.3 De la modification de tension à l'injection de fautes

Les perturbations provoquées par la modification abrupte de la tension d'alimentation, en fonction de leurs amplitudes et du moment où elles apparaissent, ont des répercussions plus ou moins néfastes sur le fonctionnement de la logique. Le cycle de l'injection permet de cibler une opération particulière, le choix de l'amplitude et de la largeur d'impulsion permet de sélectionner le nombre de chemins qui seront perturbés. L'injection de fautes intervient alors lorsque les perturbations des signaux sont capturées par des éléments de logique séquentielle. Différents mécanismes menant à ces injections de fautes sont examinés.

3.1.3.1 Violations de contraintes temporelles

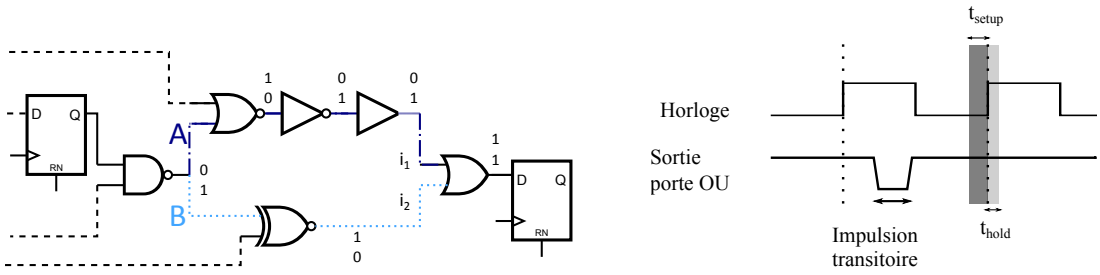
Impulsions négatives. La première conséquence de la modification de tension est le changement des caractéristiques temporelles des cellules logiques. Lorsqu'une impulsion négative est appliquée, l'ensemble des temps de propagation va être rallongé d'une durée plus ou moins longue en fonction de l'amplitude et de la largeur de l'impulsion. En considérant la contrainte donnée par l'inéquation 3.1, l'augmentation des temps de propagation provoque une capture d'une donnée erronée. Deux cas sont alors possibles dans cette situation. Dans le premier, la donnée arrive dans la fenêtre de temps de *setup* ou de *hold* de la bascule d'arrivée : une faute peut alors être injectée, néanmoins le taux de reproductibilité n'est pas de 100 % car le signal de sortie de la bascule est métastable. Dans le deuxième cas, la donnée arrive après cette fenêtre. Il y a donc une injection de faute systématique car la donnée capturée est la valeur précédente et non la nouvelle donnée mise à jour comme indiqué dans la section 3.1.1.1.

Impulsions positives. Lors d'une attaque par impulsion positive, les temps de propagation sont cette fois-ci diminués. Dans l'inéquation 3.2, une diminution excessive des temps de propagation peut avoir pour conséquence une violation de la contrainte. À l'image de la contrainte en *setup*, une donnée arrivant dans la fenêtre de temps de *hold* entraîne une valeur métastable en sortie de bascule, et donc une injection non systématique. Par contre, au-delà de cette fenêtre, la donnée est à nouveau stable mais cette fois-ci, elle est erronée de façon systématique. À noter que la violation des contraintes de temps de *hold* est plus difficile à obtenir que celle des temps de *setup*. En effet, en observant l'inéquation 3.2, l'augmentation de la tension peut provoquer à la fois la diminution de t_{hold} d'une part, et de $t_{cp2q} + t_{logic} + t_{skew}$ d'autre part, ce qui rend difficile la violation de cette inéquation.

Une manière simple d'augmenter la robustesse du circuit vis-à-vis des violations de la contrainte de temps de *hold* est d'augmenter la profondeur logique de t_{logic} lorsque c'est possible.

3.1.3.2 Impulsions transitoires à la sortie des portes combinatoires

Les violations de temps de propagation peuvent intervenir lorsque la valeur de sortie de la bascule d'arrivée change. Le but ici est de voir le comportement d'une bascule lorsque sa valeur de sortie reste la même entre deux fronts d'horloge consécutifs. La figure 3.11 montre un chemin séquentiel entre deux bascules. La porte logique NON-OU pilote deux sous-chemins notés A et B sur la figure. Ces sous-chemins sont ensuite recombinés pour piloter l'entrée de la bascule d'arrivée. Les valeurs logiques en sortie des portes sont indiquées pour deux cycles d'horloge consécutifs par exemple, la sortie de la porte NON-OU vaut 0 pour le premier cycle et 1 pour le cycle suivant. L'entrée i_1 de la porte OU passe de 0 à 1 tandis que l'entrée i_2 passe de 1 à 0. La valeur de



(a) Chemin logique (b) Diagramme temporel associé

FIGURE 3.11 : Exemple d'impulsion transitoire à l'entrée d'une bascule

sortie de la porte est donc 1 pour les deux cycles considérés. Néanmoins, le chemin B est plus rapide que le chemin A car il est composé de moins de portes logiques. Lorsque l'entrée i_2 change, la sortie reste à l'état 0 pendant un court instant avant que l'entrée i_1 ne soit mise à jour. Cette impulsion transitoire est représentée sur la figure 3.11(b). Ce phénomène peut se produire pour toutes les portes logiques du moment que les deux entrées changent et que la sortie garde sa valeur précédente. La modification des temps de propagation due à une impulsion négative sur l'alimentation peut déplacer cette impulsion transitoire dans la fenêtre de temps de *setup* de la bascule d'arrivée et induire une violation de contrainte temporelle.

3.1.3.3 Cas de domaines d'alimentation séparés

Les SoCs actuels sont de plus en plus complexes et possèdent généralement plusieurs domaines d'alimentation. Ces domaines peuvent avoir des tensions identiques ou différentes. La séparation des domaines d'alimentation peut avoir plusieurs utilités : il peut s'agir d'alimenter des blocs sous une tension différente pour des raisons de performances (augmentation de la tension pour plus de performance), ou encore de séparer les circuits qui ont toujours besoin d'être alimentés de ceux qui peuvent être mis en veille. Ce peut être notamment le cas des applications mobiles telles que la téléphonie. La présence de plusieurs domaines peut entraîner des phénomènes particuliers aux interfaces. Ces phénomènes sont étudiés lorsque des impulsions positives ou négatives sont appliquées.

Impulsions positives. La figure 3.12 illustre le cas où des portes logiques appartenant à des domaines d'alimentation séparés sont connectées ensemble. L'attaque est effectuée ici sur le domaine B et donc sur V_{dd_B}. Soient $V_{dd_{max}} = 3 \cdot V_{dd}$ l'amplitude maximale de l'impulsion et $V_{dd}(t)$ la tension d'alimentation pendant l'impulsion. Lorsque $V_{in} = Gnd$, le transistor T2 est bloqué et T1 est passant. Pendant l'impulsion, V_{int} passe de V_{dd_B} à $V_{dd_{max}}$. Cette modification n'a pas d'effet sur la fonction logique de l'inverseur. Toutefois, lorsque $V_{in} = V_{dd_A}$, les changements suivants ont lieu : la tension

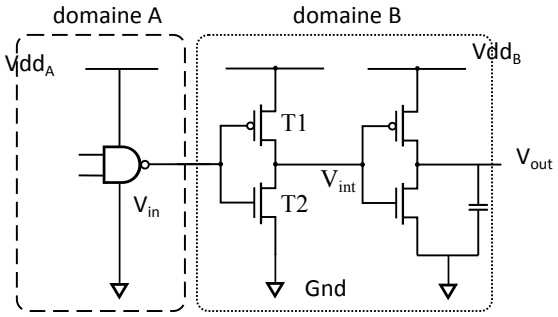


FIGURE 3.12 : Exemple de logique dans deux domaines d'alimentation

entre la grille et la source de T2 ($V_{gs,n}$) est plus grande que la tension de seuil $V_{th,n}$ par conséquent, T2 est passant. De même, la tension entre la grille et la source de T1 $|V_{gs,p}| = V_{dd}(t) - V_{dd_A}$ augmente jusqu'à ce que T1 devienne à son tour passant. En fonction du dimensionnement des transistors, V_{int} va prendre une valeur intermédiaire entre Gnd et $V_{dd}(t)$ et peut alors être lue comme un 0 ou un 1 logique par l'inverseur suivant. La valeur logique des signaux aux interfaces entre domaines d'alimentations est donnée dans le tableau 3.2.

TABLEAU 3.2 : Valeur des signaux logiques aux interfaces en présence d'impulsions positives de tension

Tension domaine A	Tension domaine B	sortie domaine A	entrée domaine B
V_{dd}	$3V_{dd}$	1	1 ou 0
		0	0
$3V_{dd}$	V_{dd}	1	1
		0	0

La logique séquentielle peut également être perturbée face à l'application d'une impulsion en présence de plusieurs domaines d'alimentation. Le verrou tel que présenté sur la figure 3.10 est analysé lorsqu'une impulsion touche son alimentation, les entrées n'étant pas perturbées car alimentés par une autre source.

Lorsque $V_E = Gnd$, $V_{\bar{E}}$ monte à $V_{dd_{max}}$ et le verrou est dans l'état bloqué. M2 et M3 sont bloqués donc $\bar{Q}$ maintient l'état précédent. La sortie Q du verrou ne peut pas basculer. Lorsque le verrou est transparent : $V_E = V_{dd}$ mais $V_{\bar{E}}$ peut prendre n'importe quelle valeur entre V_{dd} et Gnd en fonction du dimensionnement des transistors. Tout d'abord, supposons que $V_D = Gnd$, ce qui rend passant M1. Soient $V_{th,i}$ et $V_{gs,i}$ respectivement la tension de seuil et la tension entre la grille et la source du transistor Mi ($1 \leq i \leq 10$).

- Si $|V_{\bar{E}} - V_{dd}(t)| > |V_{th,2}|$, M2 est passant, $V_{\bar{Q}}$ augmente jusqu'à $V_{dd}(t)$ et V_Q décroît à Gnd.
- Si $|V_{\bar{E}} - V_{dd}(t)| < |V_{th,2}|$ M2 est bloqué, par conséquent, Q et $\bar{Q}$ maintiennent leur état.

Maintenant considérons le cas où $V_D = V_{dd}$.

- Si $|V_{\bar{E}} - V_{dd}(t)| > |V_{th,2}|$, M2 est passant. Lorsque $V_{dd}(t)$ augmente, $|V_D - V_{dd}(t)|$ devient supérieur à $|V_{th,1}|$, donc les quatre transistors de M1 à M4 sont passants : $\bar{Q}$ et Q peuvent être à l'état logique haut ou bas. Toutefois, pour que $\bar{Q}$ soit à l'état haut et Q à l'état bas, il faut que les tensions $V_{gs,1}$ et $V_{gs,2}$ soient supérieures aux tensions de seuil de M1 et M2 respectivement. Cela implique une valeur de $V_{dd_{max}}$ supérieure à $2V_{dd}$. D'autre part, une telle valeur de $V_{dd_{max}}$ entraîne une valeur élevée de $V_{gs,9}$ et par conséquent, une valeur faible de $V_{gs,2}$. Ainsi, si les inverseurs sont équilibrés, Q et $\bar{Q}$ ne peuvent pas se retrouver dans un état erroné.
- Si $|V_{\bar{E}} - V_{dd}(t)| < |V_{th,2}|$, M2 est bloqué mais M3 et M4 sont passants : $V_{\bar{Q}} = Gnd$ et $V_Q = V_{dd}(t)$.

Ces résultats sont résumés dans le tableau 3.3.

TABLEAU 3.3 : Fonctionnement d'un verrou lorsqu'une impulsion positive est appliquée sur l'alimentation

État du verrou	Bloqué	Transparent
Domaine d'alimentation unique	Q ne change pas	comportement attendu : $\bar{Q} = \bar{D}$ & $Q = D$
Plusieurs domaines d'alimentation	Q ne change pas	$D = 0$, $\|V_{\bar{E}} - V_{dd}(t)\| < \|V_{th,2}\|$: état bloqué $D = 1$, $\|V_{\bar{E}} - V_{dd}(t)\| > \|V_{th,2}\|$: possible sorties inversées

Finalement, deux cas de figure peuvent perturber le fonctionnement correct d'un verrou : le premier est quand $V_E = V_D = V_{dd}$ et $|V_{\bar{E}} - V_{dd}(t)| > |V_{th,2}|$, Q et $\bar{Q}$ peuvent valoir 0 ou 1 si les transistors de type N et P ne sont pas équilibrés. Le deuxième cas est lorsque $V_E = V_{dd}$, $V_D = Gnd$ et $|V_{\bar{E}} - V_{dd}(t)| < |V_{th,2}|$: le verrou est bloqué au lieu d'être transparent. Une mauvaise valeur peut être capturée si le verrou passe du premier au second cas, c'est-à-dire, V_D passe à Gnd et $V_{dd}(t)$ revient à la valeur V_{dd}. Cependant, ces deux événements ne peuvent intervenir simultanément : si V_D passe à Gnd en premier, $V_Q = V_{dd}(t)$ and $V_{\bar{Q}} = Gnd$ ce qui correspond au fonctionnement attendu du verrou. De même, si $V_{dd}(t)$ décroît à V_{dd}, M2 se bloque, $V_{\bar{Q}} = Gnd$ and $V_Q = V_{dd}(t)$ ce qui correspond également au comportement attendu du verrou. Par conséquent, une impulsion positive ne peut pas changer la valeur capturée par un verrou quelle que soit l'amplitude de cette impulsion. Les bascules D étant composées de deux verrous, ils sont également insensibles aux impulsions positives.

Impulsions négatives. On se place dans un cas de figure avec deux domaines d'alimentation séparés avec la même valeur de tension, notés V_{dd_1} et V_{dd_2} tel que présenté sur la figure 3.13. Lorsque l'impulsion est appliquée sur le domaine V_{dd_1}, V_{dd_2} n'est pas modifiée. Ainsi, si l'amplitude de l'impulsion est suffisamment grande, un niveau logique haut sur le nœud A peut passer sous le seuil de basculement de l'inverseur, et de cette manière être lu à 0 pour les étages logiques en aval. Si l'étage logique suivant est un

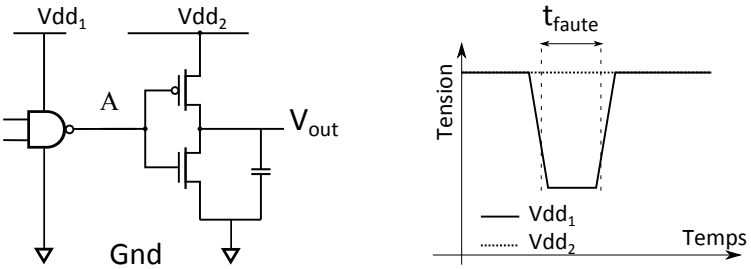


FIGURE 3.13 : Impulsion négative sur la tension en présence de deux domaines d'alimentation

élément de mémorisation (verrou, bascule, etc.), le signal lu à la mauvaise valeur peut être capturé, la faute se propage donc dans le circuit.

Les circuits qui sont constitués de plusieurs domaines d'alimentation, qui communiquent entre eux, font face à des menaces supplémentaires en cas d'attaques par impulsions sur la tension d'alimentation. Les valeurs logiques des signaux peuvent directement être modifiées aux interfaces et ainsi se propager dans le reste du circuit. Ces phénomènes sont à prendre en compte lorsque pour des circuits sécurisés.

3.1.3.4 Autres phénomènes possibles

L'application d'une impulsion d'amplitude suffisamment élevée et de durée courte pourrait avoir des répercussions sur les autres lignes de métaux par un phénomène de diaphonie (*crosstalk*). L'apparition de ce phénomène est dû aux capacités et aux inductances de couplage entre deux lignes de transmission [Wang 2011]. L'application d'une impulsion de forte amplitude sur les rails d'alimentation pourrait créer un pic de tension à l'entrée d'une bascule qui peut à son tour être capturé s'il est synchronisé avec un front d'horloge (figure 3.14).

Un autre phénomène lié à la présence d'éléments parasites sur les rails d'alimentation peut avoir lieu. Dans [Yanci 2008], il est précisé qu'à cause des impulsions, différents sous-circuits peuvent être alimentés à des tensions différentes, ce qui provoque les fautes injectées. Ce mécanisme est illustré sur la figure 3.15 En effet, pendant la durée t_{inj}, une partie du circuit est alimentée sous V_{dd1} tandis que la bascule l'est sous V_{dd2}. De

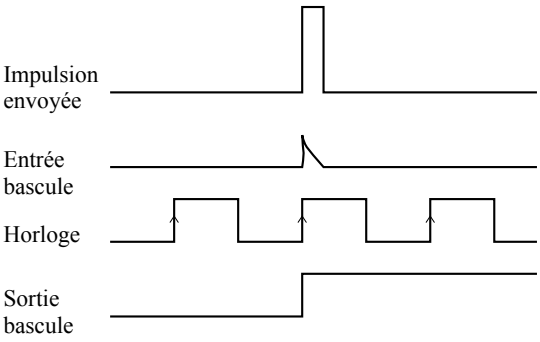
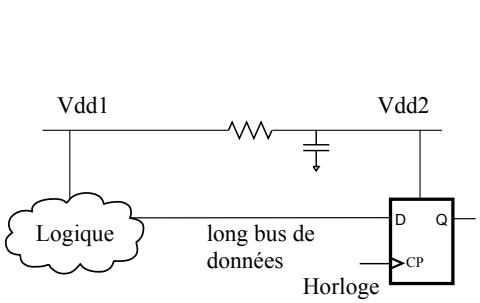
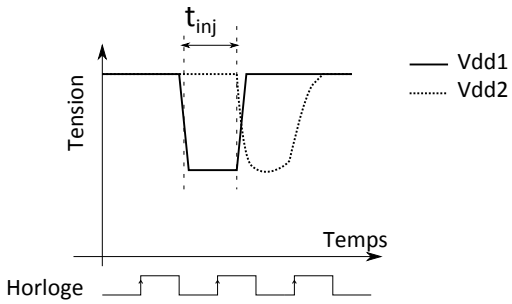


FIGURE 3.14 : Injection d'une faute par diaphonie



(a) Éléments parasites sur les rails d'alimentation



(b) Conséquence de l'application d'une impulsion négative

FIGURE 3.15 : Propagation d'une impulsion sur les rails d'alimentation

ce fait, la valeur 1 logique dans le domaine V_{dd1} peut être lue comme un 0 logique dans le domaine V_{dd2}. L'arrivée d'un front d'horloge va capturer cette valeur erronée qui va se propager dans le reste du circuit. On peut noter que ce mécanisme est le même qu'en présence de deux domaines d'alimentation distincts tels que montré dans le paragraphe précédent.

Les deux phénomènes présentés ici peuvent se produire mais sont difficiles à montrer en pratique. Néanmoins, des essais seront effectués dans la partie expérimentale pour tenter de mettre en évidence ces phénomènes.

Dans cette partie, plusieurs mécanismes d'injection de fautes ont été étudiés. Il apparaît alors que la modification des temps de propagation intervient de façon systématique dès lors que la tension est modifiée. Cela nous pousse à avoir une vue plus globale sur la distribution des temps de propagation dans un circuit et de voir si elle est perturbée par les attaques par impulsion de tension.

3.1.4 Distribution temporelle des chemins dans la logique synchrone

Dans la logique synchrone, les données transitent à travers un certain nombre de portes logiques entre deux fronts d'horloge. Ces données, en fonction du chemin parcouru, présentent des temps de propagation qui diffèrent les uns des autres. Le but est de voir comment sont répartis les délais de propagation dans un circuit et connaître la variation de cette répartition en fonction des paramètres de fonctionnement.

3.1.4.1 Profil de distribution des chemins

La distribution des chemins dans un circuit numérique synchrone est évaluée à travers un exemple. Le circuit utilisé est un circuit de cryptographie exécutant l'algorithme *AES* [Pub 2001]. Il a été conçu en technologie CMOS 28 nm, et est spécifié pour fonctionner à 100 MHz pour des tensions d'alimentation comprises entre 0.90 V et 1.1 V. Les marges sont obtenues à partir d'analyse statique des temps de propagation après le placement et le routage.

Distribution des chemins vis-à-vis des contraintes des temps de *setup*. Les marges de temps de *setup* sont données sur la figure 3.16. Cette distribution permet de visualiser les chemins les plus critiques à une tension donnée. Ici, les 27 premiers chemins sont répartis dans une fenêtre temporelle de 400 ps (entre 5.1 ns et 5.5 ns). Cela donne une résolution du nombre de chemins qui peuvent être attaqués en configurant l'amplitude et la largeur de l'impulsion appliquée sur l'alimentation. En effet, plus l'écart entre les chemins est important, plus il sera facile de créer des violations en ciblant un nombre réduit de chemins.

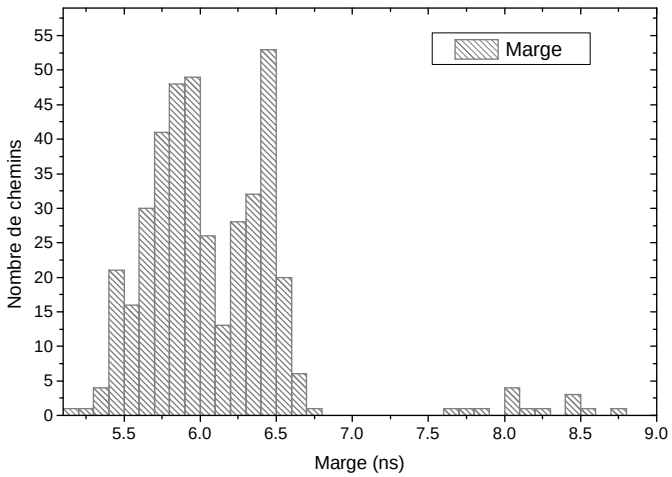


FIGURE 3.16 : Distribution des marges de temps de *setup* à 0.90 V

Distribution des chemins vis-à-vis des contraintes des temps de *hold*. La distribution des chemins est donnée sur la figure 3.17. Les 58 premiers chemins ont une marge comprise entre 0.16 ns et 0.18 ns soit une fenêtre temporelle de 20 ps. Dans ce cas, il est plus difficile de cibler précisément une sélection de chemins qui seront en violation car il faudrait avoir une résolution très précise des réglages des amplitudes et largeurs d'impulsions à appliquer. La violation de la contrainte de temps de *hold*

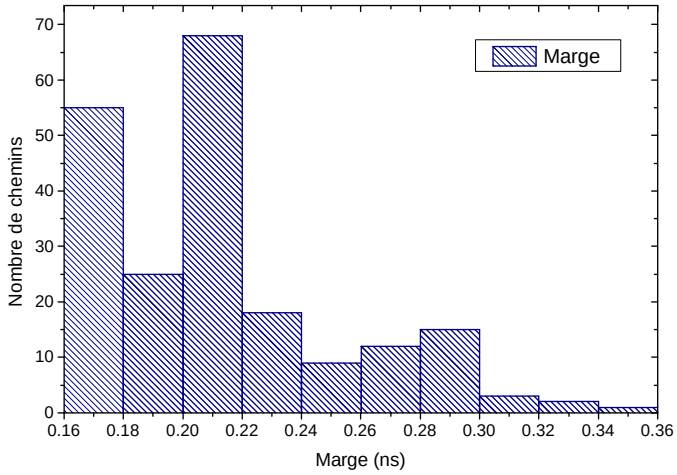


FIGURE 3.17 : Distribution des marges de temps de *hold* à 1.10 V

paraît moins intéressante pour un attaquant car elle permettrait moins de flexibilité en terme de fautes injectées. De plus, on peut voir que la marge minimum est de 0.16 ns sur la figure. En augmentant cette valeur, cela obligerait à augmenter l'amplitude de l'impulsion à appliquer, et donc le risque d'endommager le circuit.

3.1.4.2 Paramètres modifiant la distribution temporelle

La distribution des temps de propagation dans un circuit peut varier de façon plus ou moins importante selon plusieurs paramètres tels que les procédés de fabrication, la température, la tension, etc. Pour étudier ce phénomène, la marge de temps de *setup* du circuit présentée précédemment a été évaluée en simulation pour les dix chemins les plus lents après routage, en prenant donc en compte les délais supplémentaires dus aux éléments parasites. Ces mêmes chemins ont été évalués sans les délais parasites (en prenant en compte uniquement la propagation due aux cellules). Les résultats sont présentés sur la figure 3.18. Les labels au-dessus des histogrammes indiquent le rang de chemin en termes de délai de propagation, le premier étant le plus lent. Le chemin $ch1$ est donc le chemin critique après routage. La première observation est la différence entre les marges avec et sans éléments parasites : la marge diminue de 26 à 29 % à cause du délai supplémentaire ajouté par le routage. Cela confirme les observations de la section 3.1.2.1 sur la part des éléments parasites sur les temps de propagation. D'autre

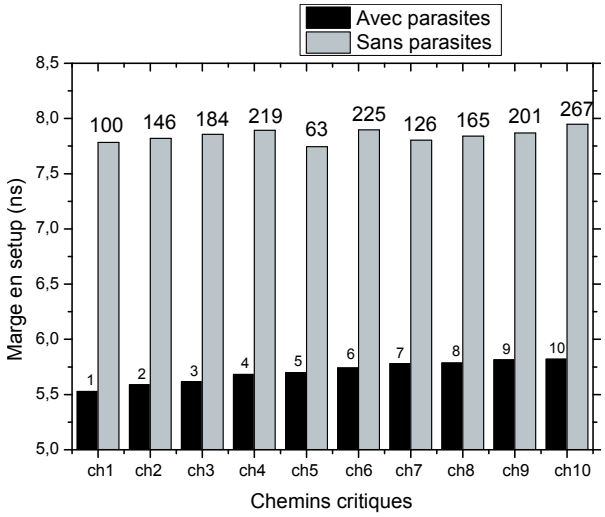


FIGURE 3.18 : Changement de la distribution des temps de propagation du aux éléments parasites de routage

part, l'ordre des chemins en termes de temps de propagation a complètement changé. Les chemins de $ch1$ à $ch10$ sont devenus les plus critiques tandis qu'ils n'étaient pas dans les 60 chemins les plus lents sans les parasites des interconnexions.

Pour connaitre la variation de la distribution des temps de propagation, celle-ci a été tracée pour différentes tensions d'alimentation, en gardant les paramètres de fabrication (*process)* et la température identiques (à -40°C). Seules les marges à 1 V sont données pour une température de 25°C. La figure 3.19 montre les marges obtenues. La marge est

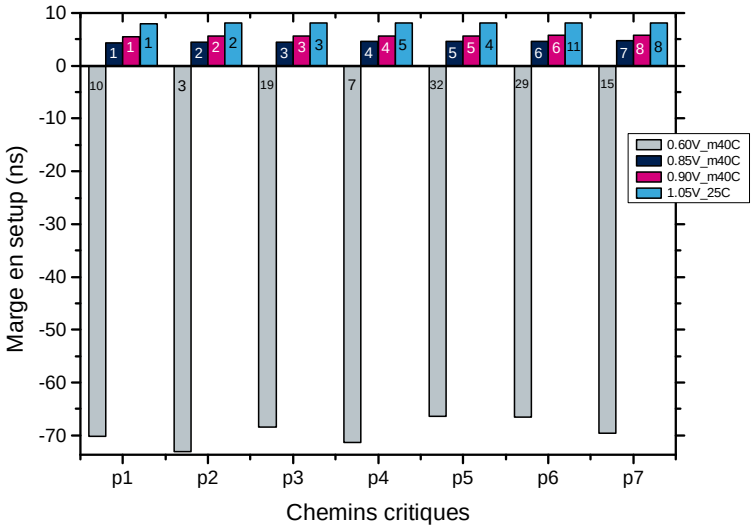
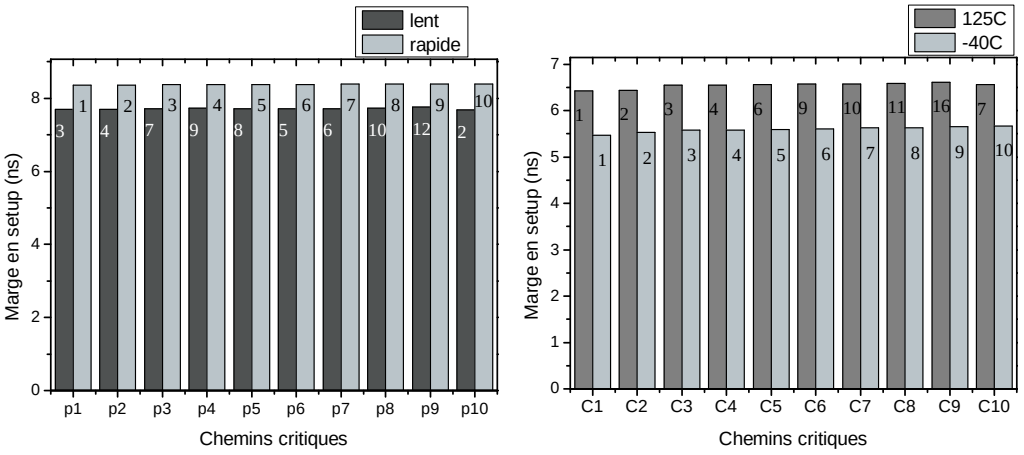


FIGURE 3.19 : Évolution des marges en *setup* pour différentes tensions

indiquée pour le circuit final (après placement et routage). Les chemins $p1$ à $p10$ ont les marges les plus réduites lorsque le circuit est alimenté sous une tension de 0.85 V. Pour des tensions autour de la valeur nominale, l'ordre des chemins critiques n'est que très peu modifié : $p1$ est le chemin critique à 0.85 V, 0.90 V et 1 V. A noter que même en modifiant la température (pour la distribution à 1 V et 25°C), l'ordre des chemins critiques est conservé. Toutefois, cet ordre est modifié à 0.60 V, $p1$ ne se retrouve qu'en dixième position. Les autres chemins sont également modifiés, par exemple le chemin $p3$ se retrouve en 19^e position tandis que $p5$ se retrouve en 32^e position.

De la même manière, l'influence des paramètres de fabrication et de la température est analysée en regardant la distribution temporelle pour deux procédés de fabrication différents. Sur la figure 3.20(a), la distribution temporelle des 10 chemins les plus critiques est représentée pour un procédé de fabrication rapide et un autre lent. La tension d'alimentation est de 1.10 V. Les labels représentent toujours l'ordre temporel des chemins. Cet ordre est légèrement modifié lorsque les deux procédés de fabrication sont comparés. Le même circuit fabriqué d'un procédé à l'autre peut donc avoir des distributions temporelles modifiées. La figure 3.20(b) compare cette fois-ci deux valeurs de



(a) Influence du procédé de fabrication à1.10 V (b) Influence de la température à 0.90 V

FIGURE 3.20 : Influence de la température et du *process* sur les marges en setup

température à savoir 125°C et -40°C, la tension d'alimentation étant de 0.90 V, ce qui explique des marges moins importantes que sur la figure 3.20(a). L'ordre des chemins est également légèrement changé, ce qui peut modifier le chemin critique.

Cet exemple montre que dans certains cas, la température de fonctionnement, les procédés de fabrication et surtout la tension d'alimentation peuvent changer l'ordre des chemins en terme de marge de *setup* notamment. En effet, chacune des portes logiques qui compose un chemin a une variation en fonction de la tension, comme étudié dans la section 3.1.2.1. En modifiant la tension d'alimentation, les temps de

propagation des portes changent et peuvent rendre un chemin plus rapide qu'un autre. La détection de modifications temporelles requiert de prendre en compte cette situation où le chemin critique dépend de la tension d'alimentation, de la température ou du procédé de fabrication.

3.1.5 Bilan

L'étude du fonctionnement de la logique synchrone, ainsi que des attaques par impulsion sur la tension d'alimentation et des phénomènes sous-jacents ont permis de mettre en évidence plusieurs mécanismes. L'effet principal est la modification des propriétés temporelles des portes logiques, qui se traduisent par des injections de fautes, qui inversent la valeur d'un ou plusieurs bits. D'autres effets peuvent se manifester en présence de plusieurs domaines d'alimentation. Des niveaux logiques aux interfaces peuvent alors être mal lus et induire des fautes. Enfin, d'autres phénomènes plus difficiles à mettre en évidence tels que la diaphonie ou l'effet des parasites sur les rails d'alimentations ont également été analysés.

En se basant sur ces analyses, l'observation des temps de propagation peut permettre de détecter des variations anormales de la tension d'alimentation. Pour cela il faut pouvoir définir une méthode à la fois pour surveiller et pour contrôler les temps de propagation dans un circuit. Ces différents points seront abordés dans la suite de cette étude.

3.2 Réalisation de circuits de détection

L'objectif de cette partie est la mise en œuvre de solutions qui permettent de détecter de façon fiable les attaques par impulsions sur la tension d'alimentation. Le développement de ces mesures de protection doit se faire dans un flot de conception numérique, pour faciliter leur intégration aussi bien au niveau d'une IP que d'un SoC. L'une des contraintes de l'utilisation de ce flot est l'impossibilité de réaliser des simulations dynamiques par modification de la tension d'alimentation. En effet, les résultats de simulation se basent sur une caractérisation statique des propriétés temporelles des cellules. Il faudra donc vérifier la validité des modèles statiques ou trouver une méthodologie pour accéder à ses informations dynamiques. Toutefois, les analyses menées dans la section précédente permettent de considérer les temps de propagation pour détecter des anomalies sur la tension. Pour cette raison, les propriétés de temps de propagation des cellules seront utilisées.

Il existe peu de travaux dans la littérature utilisant des circuits numériques pour la détection d'attaques par impulsion de tension. Néanmoins, des circuits permettant de détecter des erreurs de temps de propagation pour diverses applications ont été proposés. Leur principe de fonctionnement est abordé dans le paragraphe suivant.

3.2.1 Travaux relatifs à la détection de violation de temps de *setup*

Plusieurs systèmes ont été proposés dans la littérature pour détecter des erreurs de temps de propagation dues à des variations dites statiques (procédés de fabrication) ou dynamiques (température, tension, vieillissement etc.).

L'idée du circuit razorI [Das 2005] est d'introduire pour chaque chemin jugé critique un verrou secondaire (*shadow latch*). La capture est effectuée par ce verrou au front descendant du signal d'horloge, ce qui permet de capturer la donnée avec un retard additionnel d'une demi-période par rapport à la bascule principale du chemin. Une erreur est détectée lorsque la sortie de la bascule principale diffère de la sortie du verrou secondaire. Cette solution requiert de contrôler le rapport cyclique du signal d'horloge de façon précise car le verrou fonctionne sur le niveau bas de l'horloge. Il faut ajouter aussi un détecteur de métastabilité car la sortie de la bascule principale peut devenir métastable.

RazorII [Das 2009] est une amélioration de la première version (figure 3.21). Ici, la bascule principale est remplacée par un verrou. Les transitions d'un nœud interne du verrou sont observées pour vérifier un changement tardif. Pour cela, un détecteur de transitions basé sur des inverseurs et des portes de transmissions est utilisé. Le circuit razorII ne nécessite pas de détecteur de métastabilité, par contre le rapport cyclique du signal d'horloge doit être convenablement contrôlé.

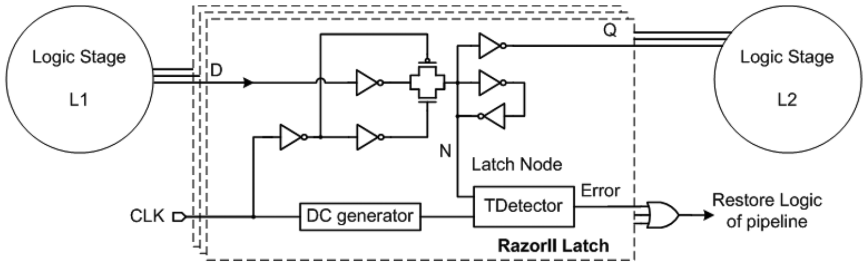


FIGURE 3.21 : Bascule razorII [Das 2009]

Les bascules canary [Kunitake 2011] introduisent également une bascule secondaire, avec un élément de délai additionnel pour créer un chemin plus long que le chemin critique. La comparaison entre les signaux de sortie de la bascule principale et secondaire permet de détecter une erreur. Cette technique a quelques avantages par rapport au circuit razorI : le contrôle du rapport cyclique de l'horloge n'est pas requis, toutefois, il faut inclure un détecteur de métastabilité.

Dans [Bowman 2009, Bowman 2011], les auteurs ont présenté deux circuits *EDS* : détecteur de transition avec partage de temps (*TDTB : Transition Detector with Time Borrowing*) et double échantillonnage avec partage de temps (*DSTB : Double Sampling with Time Borrowing*). Le premier circuit détecte une transition tardive de la donnée

d'entrée pour générer une alarme (figure 3.22). Le deuxième (DSTB), utilise une bascule maître-esclave additionnelle pour capturer les données à un moment différent. La sortie du verrou et de la bascule sont ensuite comparées. Ces deux circuits utilisent un verrou à la place d'une bascule sur le chemin principal. Le verrou est transparent sur niveau haut du signal d'horloge, ce qui autorise un délai supplémentaire pour capturer la donnée valide. Le contrôle du rapport cyclique de l'horloge est donc indispensable.

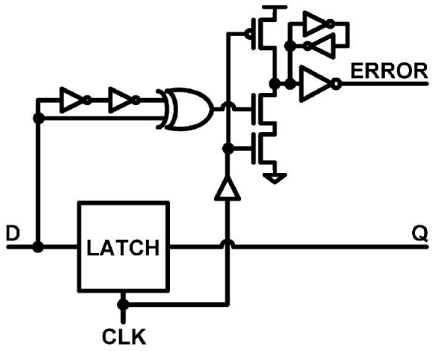


FIGURE 3.22 : Détecteur de transition avec partage de temps [Bowman 2009]

Selmane *et al.* [Selmane 2011] ont proposé une contremesure sur *FPGA* qui consiste à ajouter un chemin composé d'un ensemble de *buffers* et d'un inverseur entre deux bascules. La bascule source délivre un signal qui est inversé à chaque cycle. L'erreur est détectée en comparant la sortie des deux bascules. L'avantage de cette solution est qu'elle est indépendante de l'architecture ou de la fonctionnalité du circuit à protéger, néanmoins il faut caractériser précisément le chemin critique en fonction des différents paramètres de fonctionnement (température, procédé de fabrication etc.) pour configurer le chemin.

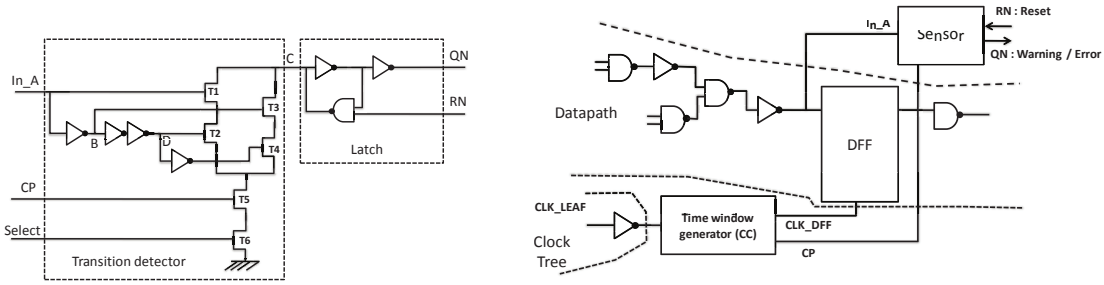
Un circuit répliqué réglable (*TRC : Tunable Replica Circuit*) est présenté dans [Bowman 2011, Raychowdhury 2011]. Un arbre de délai réglable, qui est plus long que le chemin critique est utilisé pour détecter une erreur de temps de propagation. Le chemin combinatoire est constitué d'inverseurs et de multiplexeurs pour sélectionner le chemin et donc le délai correspondant. Ce circuit offre la possibilité de reconfigurer le délai après fabrication. Toutefois, cela augmente le temps de test du circuit.

Un circuit de protection au niveau architectural a été proposé dans [Endo 2012]. L'idée est d'empêcher la capture des données lorsque des temps de propagation incorrects sont détectés. Pour cela, une logique de contrôle permet d'activer ou de désactiver un multiplexeur placé en entrée de chaque bascule. Le signal d'activation est généré par un bloc de délai configurable, qui est calibré après fabrication. Le délai nécessaire, quant à lui, est obtenu par analyse statique de temps de propagation (*STA*) en phase de conception. Cette méthode est utile pour protéger les circuits contre les attaques par analyse de sensibilité aux fautes, qui exploite la dépendance des temps de propagation

vis-à-vis des données. Elle a pour limitation d'être intrusive et exige une caractérisation précise du délai configurable.

Dans [Agarwal 2007], un capteur pour détecter des temps de propagation longs est introduit. Les auteurs présentent deux capteurs de vieillissement, avec un vérificateur de stabilité pour détecter une transition tardive à l'entrée d'une bascule avant le front d'horloge. Une fenêtre temporelle est définie afin de prévenir d'une transition lente inattendue. Ce circuit permet de détecter une déviation progressive des temps de propagation avant qu'une faute ne se produise. Par contre, il est nécessaire de modifier les bascules au niveau transistor pour intégrer les capteurs.

D'autres capteurs ont été proposés [Rebaud 2009, Das 2010] pour détecter des transitions dans une certaine fenêtre temporelle avant la transition du signal d'horloge (figure 3.23. Ces méthodes utilisent un nombre réduit de transistors en comparaison avec des circuits razorI ou razorII. Cependant, ces solutions doivent être conçues au niveau transistor, et ne sont donc pas directement intégrables dans un flot de conception numérique.



(a) Détecteur de transition à une entrée (b) Système complet

FIGURE 3.23 : Système de contrôle de marge de temps [Rebaud 2009]

Les solutions existantes ci-dessus ont été testées pour des variations lentes de la tension d'alimentation et peuvent détecter des violations de temps de propagation. Cependant, elles n'ont pas été validées pour des attaques dynamiques de tension. De plus, elles ne prennent pas en compte les accélérations de temps de propagation. Par ailleurs, plusieurs de ces solutions se basent sur des chemins qui ne sont pas activés à chaque cycle, ce qui est une condition nécessaire pour la détection d'attaques. Le but sera donc de réaliser des circuits de détection qui prennent en compte les variations dynamiques de la tension d'alimentation. Pour cela, la variation de la fréquence de fonctionnement en fonction de la tension sera considérée. Par la suite, les principes de fonctionnement des circuits de détection seront présentés.

TABLEAU 3.4 : Avantages et limitations des solutions rapportées dans la littérature

	Avantages	Limitations
RazorI	La détection d'erreur se fait directement sur un chemin fonctionnel	Contrôle du rapport cyclique de l'horloge
RazorII		Conception analogique
Canary		Faible fenêtre de détection
TDTB		Contrôle du rapport cyclique de l'horloge, conception analogique
DSTB		Schéma intrusif (modifie l'architecture du circuit)
[Endo 2012]		Schéma intrusif
[Rebaud 2009]		Conception analogique
[Agarwal 2007]		Conception analogique
TRC	Chemin de détection toujours actif	Difficulté à régler le délai combinatoire
[Selmane 2011]		

3.2.2 Principes de fonctionnement des circuits de détection

L'idée est de concevoir un circuit capable de détecter des erreurs de temps de propagation dues à des impulsions de tension. Deux types de détection peuvent être considérés :

- les erreurs sont détectées dès qu'elles apparaissent (le même cycle d'horloge),
- la détection peut être périodique (test sur plusieurs cycles), dans ce cas la détection pourrait se faire plusieurs cycles après l'apparition de la faute.

Dans le cadre d'une attaque, la détection doit avoir lieu dès qu'il y a une violation de temps de propagation pour empêcher l'attaquant d'exploiter son injection de fautes.

La détection d'erreurs de temps de propagation peut se baser sur deux types de chemins :

- le premier est un chemin fonctionnel existant du circuit (qui a donc une fonction dans le circuit),
- le deuxième est un chemin additionnel dédié à la détection d'erreurs.

Les circuits utilisant un chemin additionnel seront qualifiés de circuits répliqués, tandis que ceux utilisant un chemin existant sont généralement appelés circuits EDS (*Error-Detection Sequential*) [Bowman 2009].

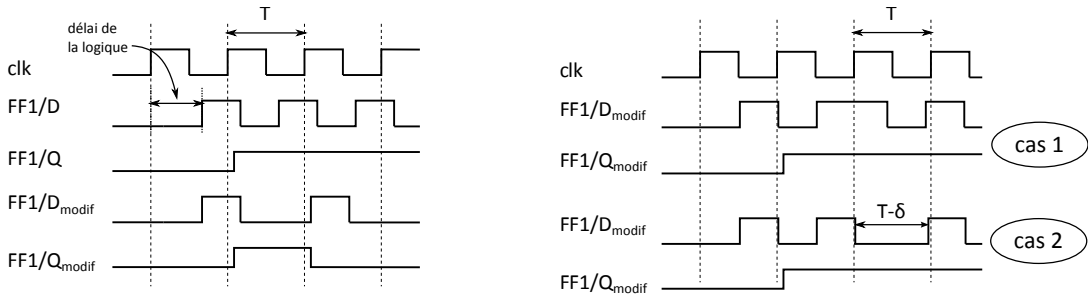
3.2.2.1 Description du fonctionnement

Circuits répliqués. Le principe est d'utiliser un chemin de test, dont les valeurs des signaux sont connues, pour détecter des violations de temps de propagation. Le chemin de test est composé de deux bascules, une de départ et une à l'arrivée (bascule de capture), comme indiqué sur la figure 3.24. Ce chemin ayant une entrée unique, la

FIGURE 3.24 : Principe de fonctionnement d'un circuit répliqué

bascule de départ doit changer de valeur à chaque cycle pour être en mesure de détecter une violation sur la bascule de capture. À noter que le signal d'horloge pourrait être utilisé à la place de la bascule FF0, mais cette méthode présente des limitations.

La figure 3.25(a) montre le diagramme temporel d'un détecteur lorsque la bascule de départ est remplacée par le signal d'horloge. Ainsi, le même signal est utilisé en tant que signal d'entrée et signal d'horloge. En l'absence de perturbations, FF1 maintient sa valeur (état haut). Si les temps de propagations sont ralentis, une des périodes à l'entrée de FF1 dure plus longtemps que la période de référence T. Une valeur erronée peut alors être capturée (0 au lieu de 1). Néanmoins, l'absence de changement de valeur à chaque cycle peut masquer une violation de temps de propagation.



(a) Fonctionnement normal

(b) Cas de masquage

FIGURE 3.25 : Utilisation du signal d'horloge comme entrée des circuits répliqués

La figure 3.25(b) montre deux cas intéressants lorsque les temps de propagations sont plus lents que prévus. Dans le premier cas, l'entrée de FF1 reste à l'état haut pendant une durée supérieure à la demi-période, cependant la sortie est également à l'état haut, la détection est donc masquée. Dans le second cas, le délai entre la montée et la descente du signal d'entrée est plus grand que la demi-période mais plus petit que la période T. La violation n'est donc pas détectée. Ces cas de figure exposent la limite de l'utilisation de l'horloge comme signal d'entrée de la bascule de capture. A noter aussi que dans ce cas, la sensibilité du détecteur dépend du rapport cyclique de l'horloge, qui n'est pas toujours bien défini.

Circuits *EDS*. Les circuits de type *EDS* utilisent des chemins existants (qui ont une utilité fonctionnelle), mais introduisent en plus des éléments de délai et de mémorisation (généralement des verrous mais aussi des capteurs de transition) sur des chemins critiques afin de détecter des violations de temps de propagation (figure 3.26). La détec-

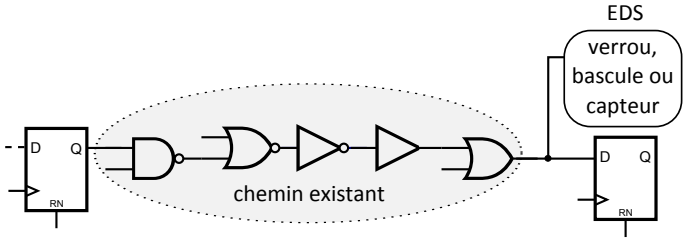


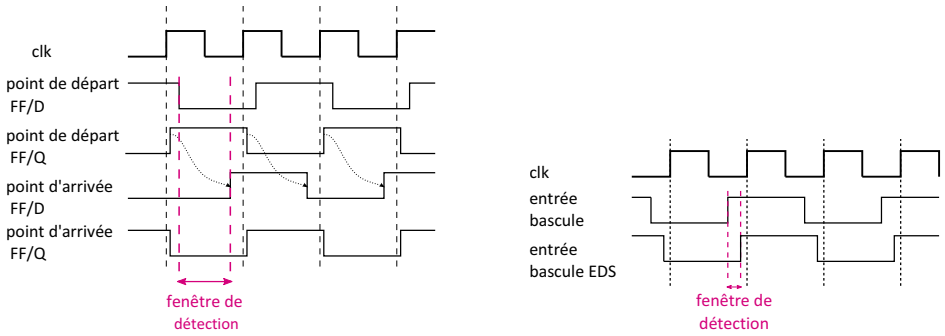
FIGURE 3.26 : Principe de détection avec circuits *EDS*

tion est réalisée en comparant la sortie de la bascule d'arrivée avec la sortie de l'élément additionnel de l'*EDS*, ou en observant l'entrée de cette bascule pour détecter l'arrivée tardive d'une donnée. L'inconvénient de cette méthode est la nécessité que le chemin observé soit actif (changement de valeur à l'entrée de la bascule de capture) pour être en mesure de détecter une violation.

L'avantage principal des circuits répliqués par rapport aux circuits de type *EDS* est qu'ils offrent plus de flexibilité. Le chemin ajouté est indépendant du reste du circuit et peut donc être intégré indépendamment de la fonctionnalité de celui-ci. Toutefois, quel que soit le type de solutions utilisé, celles-ci doivent inclure les marges de fonctionnement qui permettent de s'assurer de la validité de la détection d'erreurs. Cet aspect est abordé dans le paragraphe suivant.

3.2.2.2 Détermination des marges de fonctionnement

Fenêtre de détection. La fenêtre de détection est définie comme la différence temporelle entre l'arrivée de la donnée à l'entrée de la bascule de départ et l'arrivée de la donnée à l'entrée de la bascule de capture comme le montre la figure 3.27. Plus cette fenêtre de détection est grande, plus les modifications de tension de forte amplitude et de largeur importante pourront être détectées. Pour les circuits répliqués, la valeur en sortie de bascule au point d'arrivée est comparée à celle au point de départ. La valeur de référence dans ce cas est la sortie de la bascule du point de départ. En ce qui concerne les circuits *EDS*, la comparaison est faite entre la bascule du point d'arrivée et la sortie de la bascule ou du verrou additionnel. Ainsi, la valeur de référence est celle de la bascule du point d'arrivée. Par conséquent, la fenêtre de détection n'est pas la même pour ces deux méthodes de détection. Dans le cas de circuits *EDS*, la fenêtre de détection est généralement plus petite car le chemin fonctionnel est déjà suffisamment long par rapport à la période d'horloge.



(a) Circuit répliqué (b) Circuit *EDS*

FIGURE 3.27 : Fenêtre de détection des circuits répliqués et *EDS*

Marge de fiabilité. L'une des difficultés liées à la détection de violations de temps de propagation est de configurer les délais combinatoires pour assurer une marge suffisante entre le détecteur et le circuit. La marge de fiabilité est définie ici comme la différence de temps entre le chemin critique et le chemin du détecteur, comme le montre la figure 3.28. En effet, les temps de *setup* et de *hold* sont également dépendants de la tension d'après leur définition. Par conséquent, les marges de fiabilité changent en fonction de la tension d'alimentation. Les temps de *setup* et de *hold* définissent la fenêtre temporelle dans laquelle la sortie des bascules est indéterminée ou encore dans un état de métastabilité [Horstmann 1989]. Lorsqu'une violation de temps de propagation a lieu dans le circuit, la donnée à l'entrée de la bascule du détecteur doit être au-delà de cette fenêtre d'incertitude. À noter que pour les circuits *EDS*, la marge de fiabilité et fenêtre

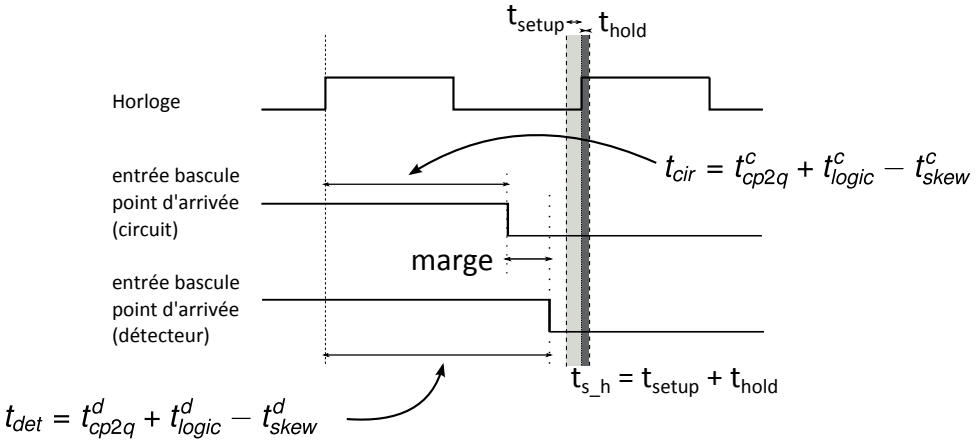


FIGURE 3.28 : Marges de fiabilité

de détection sont confondues.

Considérons un chemin fonctionnel dans le circuit à protéger ainsi que le chemin du

circuit de détection. Les bascules des points d'arrivée sont supposées identiques (mêmes t_{setup}, t_{hold} et t_{cp2q}). La figure 3.28 illustre cette situation : ici le chemin du détecteur aussi bien que celui du circuit respectent les contraintes de temps *setup* et de *hold*. Cependant, le temps de propagation du circuit de détection est plus grand que celui du circuit. Dans le cas d'une attaque par impulsion négative, les temps de propagations seront augmentés. Si l'hypothèse selon laquelle l'augmentation des temps de propagation est uniforme est considérée, le détecteur va être le premier à entrer dans la fenêtre de violation. On suppose ici que qu'il n'y a pas de différence d'arrivée des signaux d'horloge et donc $t_{skew} = 0$. Notons $t_{s\_h} = t_{setup} + t_{hold}$. Ici, la marge est plus grande que t_{s_h} donc au moment où le chemin du circuit entre dans cette fenêtre, le détecteur en sera sorti, il en résulte que la violation sera détectée. Formellement, la contrainte est donnée par l'inéquation 3.10. t^c sont les temps liés au circuit à protéger et t^d sont liés au détecteur.

$$\begin{aligned} t_{det} &= t^d_{cp2q} + t^d_{logic} - t^d_{skew} \\ t_{cir} &= t^c_{cp2q} + t^c_{logic} - t^c_{skew} \\ t_{det} - t_{cir} &> t_{hold} + t_{setup} = t_{s\_h} \end{aligned} \tag{3.10}$$

Généralement, plusieurs types de bascules sont utilisées au sein du même circuit donc les temps de *setup* et de *hold* sont différents pour chaque chemin. Dans ce cas, il faut considérer les temps de *setup* et de *hold* les plus grands pour vérifier les marges. Lorsque t_{hold} est négatif, t_{s_h} vaut $max(t_{setup}, |t_{hold}|)$ et l'inéquation 3.10 est toujours valable. À noter également que cette inéquation doit être respectée quelles que soient les conditions de fonctionnement (température, tension d'alimentation et procédés de fabrication). Dans le cas où t_{skew} n'est pas nul, l'inéquation 3.10 devient :

$$t_{det} - t_{cir} > t_{hold} + t_{setup} - (t^d_{skew} + t^c_{skew}) = t_{s\_h} - t^{dc}_{skew} \tag{3.11}$$

L'étude ci-dessus a été faite pour une augmentation des temps de propagation. Néanmoins, dans le cas d'une réduction des temps de propagations, les contraintes à respecter demeurent identiques.

Il existe peu de travaux dans la littérature utilisant des circuits numériques pour la détection d'attaques par impulsion de tension. Par contre, des circuits permettant de détecter des erreurs de temps de propagation pour diverses applications ont été proposés. Les différents travaux portant sur ces détections d'erreurs font l'objet de la section suivante.

3.2.3 Solutions étudiées

Afin de détecter les fautes provoquées par les attaques par impulsions sur la tension, plusieurs solutions sont étudiées en prenant en compte les critères suivants :

- les attaques par impulsions positives ou négatives doivent être détectées.
- la fenêtre de détection doit être la plus grande possible pour détecter des impulsions de différentes amplitudes et largeurs
- la réalisation des circuits de détection doit se faire en suivant un flot de conception numérique standard, ce qui exclut la conception au niveau transistor
- les solutions proposées ne modifient pas l'architecture du circuit à protéger (non-intrusives)
- la détection d'erreurs doit pouvoir se faire à chaque cycle d'horloge, ce qui exige que le chemin considéré soit toujours actif

Ces différents critères nous orientent vers des solutions de type circuits répliqués. Les sections suivantes présentent les différents circuits étudiés.

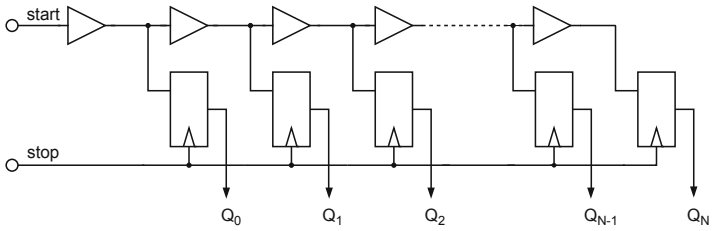
3.2.3.1 Convertisseur temporel vers numérique (*TDC : Time to digital converter*)

Description. Les convertisseurs temporels vers numérique sont des circuits qui permettent d'évaluer un intervalle de temps entre un signal de démarrage et un signal d'arrêt avec une grande précision (quelques picosecondes). Ces circuits sont utilisés dans plusieurs domaines : dans les boucles à verrouillage de phase (*PLL : Phase Lock Loop*), dans les instruments de mesure et d'instrumentation tels que les oscilloscopes numériques, ou encore dans la physique haute énergie [Henzler 2010]. L'idée est de pouvoir mesurer un temps de propagation connu pour détecter une attaque. Un exemple de *TDC* est donné sur la figure 3.29. Le convertisseur est composé d'une ligne de délai formé de *buffers*. La sortie de chaque *buffer* est échantillonnée par une bascule. La propagation du signal *start* est capturée par le signal *stop*. Ainsi, en sortie de chaque bascule, une valeur correspondant à la propagation du signal est obtenue. La valeur du délai mesuré vaut :

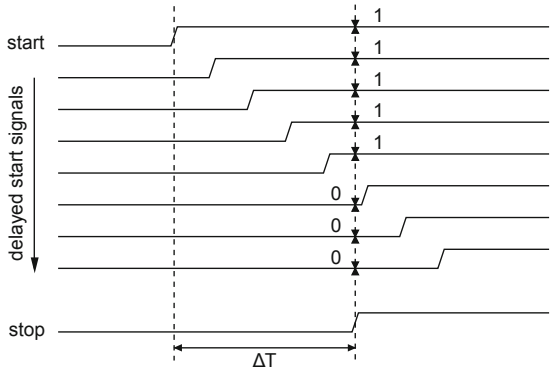
$$\Delta T = N \cdot t_{buf} + \epsilon \tag{3.12}$$

avec ΔT la durée entre le signal *start* et le signal *stop*, t_{buf} le temps de propagation d'un *buffer* et ϵ une valeur comprise entre 0 et t_{buf}.

Besoins pour la détection d'attaques. L'utilisation de *TDC* pour la détection d'attaques est particulière. En effet, dans ce cas de figure, ce sont les délais de traversée des portes logiques qui changent au lieu de la durée des signaux *start* et *stop*. D'autre part, la mesure doit se faire au moins une fois par période d'horloge, afin de détecter une attaque dès qu'elle se produit. De plus, les délais des signaux *start* et *stop* doivent être connus de façon précise car ils constituent la durée du référence pour la détection d'attaques. Les différentes conditions de fonctionnement, à savoir la température, les procédés de fabrication et la gamme de tensions d'alimentation spécifiée (par exemple de 0.85 V à 1.3 V) doivent être pris en compte.



(a) Schéma



(b) Diagramme de fonctionnement

FIGURE 3.29 : Convertisseur temps vers numérique à base d'une ligne de délai [Henzler 2010]

Exemple d'utilisation. Le signal d'horloge et son inverse sont des bons candidats pour les signaux de références : le front montant d'horloge sert de signal de départ et le front montant de son signal inverse sert de signal de capture. Le circuit à protéger est conçu en technologie CMOS 28 nm. Une période $T = 10\,\text{ns}$ est considérée soit une durée ΔT de 5 ns, si le rapport cyclique est de 50 %. La gamme fonctionnelle de tensions varie de 0.85 V à 1.3 V Des cellules de délai ont des temps de propagation de 120 ps à 1.3 V et 570 ps à 0.85 V. Le nombre minimal de cellules requis est :

$$5/0.120 = 42 \qquad 5/0.570 = 9 \tag{3.13}$$

Il faudrait donc au minimum 42 cellules pour un fonctionnement à 1.3 V et 9 cellules pour un fonctionnement à 0.85 V. Ainsi, pour une tension inférieure à 1.3 V, au moins une cellule ne sera pas atteinte (figure 3.30). Les cellules non atteintes (entre la 10^e et la 42^e) doivent être réinitialisées pour éviter la propagation de la période d'horloge précédente, cela doit être fait au même cycle afin de pouvoir détecter une erreur au cycle suivant. Le temps de réinitialisation est défini par la durée nécessaire pour traverser l'ensemble des cellules de délai, soit au pire des cas 33 cellules (de la 10^e à la 42^e). Le délai correspondant vaut : $33 \cdot t_{buf} = 18.81\,\text{ns}$ soit pratiquement deux périodes.

La nécessité de réinitialisation exige de considérer un deuxième *TDC* en parallèle qui fonctionnerait pendant que le premier est réinitialisé et inversement. À noter qu'ici le nombre minimum de cellules a été pris en compte pour la tension maximale de 1.3 V. Si le nombre de cellules est augmenté pour avoir de la marge, le nombre de *TDC* peut augmenter également. Cela complexifierait grandement le système de contrôle des *TDC* et de détection d'attaques.

L'utilisation de *TDC* ne semble donc pas être adaptée à la détection d'attaques par impulsion positives et négatives sur l'alimentation pour une large gamme de tensions. En effet, la gamme fonctionnelle de tensions doit permettre de traverser tous les *buffers* pendant la durée ΔT, y compris à la tension minimale spécifiée. Dans ce cas, seuls des temps de propagation ralentis, donc des attaques par impulsion négatives peuvent être détectées lorsque tous les *buffers* n'ont pas été traversés.

Les *TDC* peuvent donc permettre de détecter des impulsions négatives. Néanmoins, il faut trouver un signal de référence à mesurer suffisamment précis, le choix du signal d'horloge exige de connaitre et de maitriser le rapport cyclique, ce qui en fait une limitation. D'autres solutions sont alors étudiées pour mieux répondre au besoin de détection des attaques.

3.2.3.2 Circuits de détection

En se basant sur les analyses précédentes et l'étude des circuits proposés dans la littérature, plusieurs circuits de détection sont conçus et implantés pour détecter des attaques en tension. Les solutions de détection sont séparées entre impulsions positives et négatives, notamment pour répondre au besoin de fonctionnement en gamme de tensions étendue. Les circuits de détections présentés ici ont fait l'objet de trois publications : [Gomina 2014a, Gomina 2014b, Beringuier-Boher 2014].

Détection d'attaques par impulsion négative. Le principe est d'avoir des chemins qui ont une variation temporelle particulière en fonction de la tension d'alimentation, afin de les mettre en violation de contraintes de temps de propagation plus facilement que le circuit à protéger.

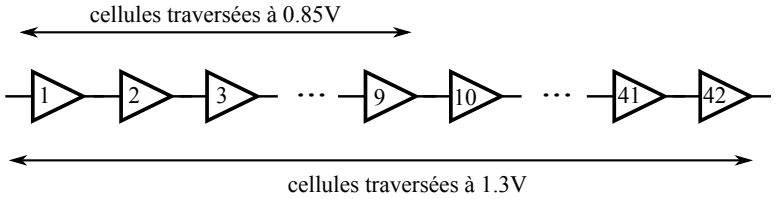
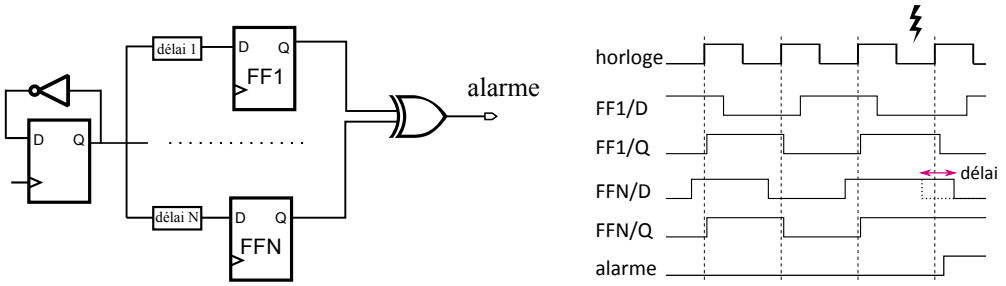


FIGURE 3.30 : Exemple d'implantation d'un TDC pour un fonctionnement entre 0.85 V et 1.3 V

Lignes parallèles de délai L'idée est d'utiliser plusieurs chemins combinatoires pour déterminer l'évolution des temps de propagation dans le circuit. Chaque chemin du détecteur a un temps de propagation différent : le premier chemin est le plus rapide et le dernier est le plus lent, comme présenté sur la figure 3.31. Le chemin le plus lent (délai



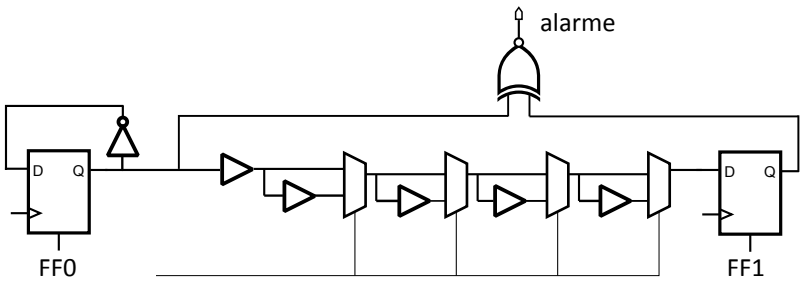
(a) Schéma (b) Diagramme temporel

FIGURE 3.31 : Lignes parallèles de délai

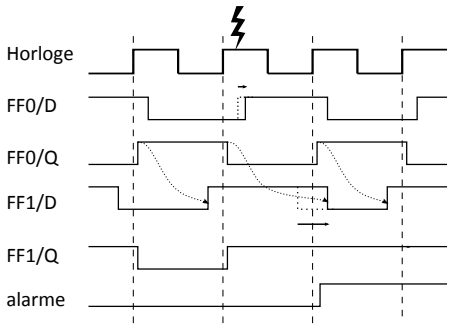
N) est néanmoins plus court que la période d'horloge dans le pire cas de fonctionnement, par exemple, un procédé de fabrication lent, la tension et la température la plus basse. Par conséquent, si la sortie de la bascule FF1 diffère de celle de FFN, une violation de temps de propagation a certainement eu lieu dans FFN, ainsi l'alarme est déclenchée. Cette méthode requiert au moins deux chemins, un pour le plus rapide et l'autre pour le plus lent. Cependant, avec plusieurs chemins, il est possible d'évaluer la variation des délais en observant la sortie de l'ensemble des bascules. Il est important de noter que le délai N doit être plus grand que celui du chemin critique du circuit pour être le premier à être en violation. Pour éviter les problèmes de métastabilité, les marges de fiabilité entre le chemin constitué du délai N et le chemin critique sont réglées en respectant l'inéquation 3.10. Ainsi, la sortie de la bascule FFN est stable lorsqu'un chemin du circuit devient métastable.

Circuit répliqué réglable Le circuit répliqué réglable (*TRC)* utilise un nouveau chemin dans le circuit, dont la logique combinatoire est composée d'un arbre de délai. La logique, faite d'une série de *buffers* et d'inverseurs, est configurée pour être le nouveau chemin critique du circuit à protéger. Un exemple de *TRC* est donné sur la figure 3.32. En fonction de la valeur de sélection, il est possible de reconfigurer le délai après fabrication. Il s'agit du principal avantage de cette méthode. La configuration du délai pourrait servir par exemple à changer les marges de détection pour différentes opérations ou fonctionnalités du circuit.

La figure 3.32 décrit le fonctionnement du circuit en cas d'attaque par impulsion négative de tension. En absence de perturbations, les sorties des bascules FF0 et FF1 sont identiques. Si la donnée arrive tardivement à l'entrée de FF1, la sortie n'est pas



(a) Schéma

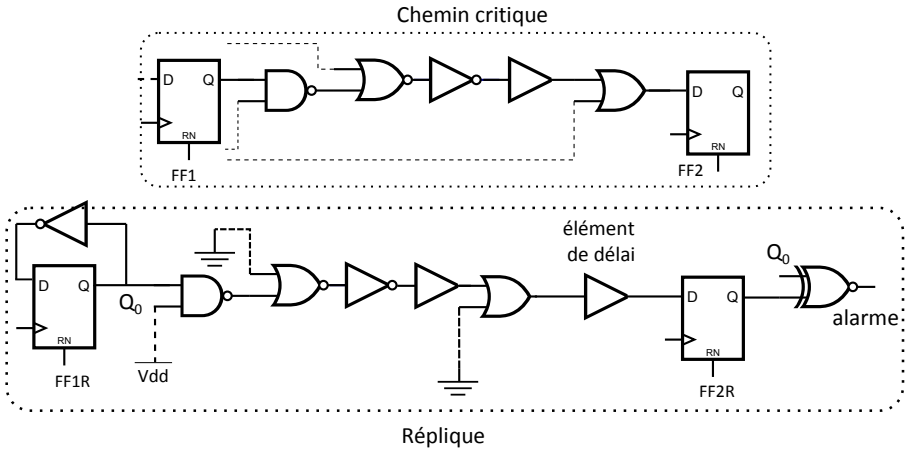


(b) Diagramme temporel

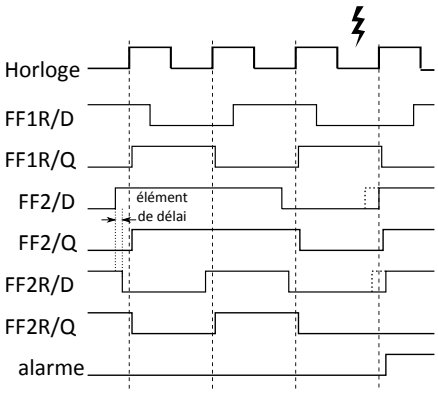
FIGURE 3.32 : Circuit répliqué réglable

mise à jour jusqu'au prochain front d'horloge, par conséquent les sorties des bascules sont dans le même état logique, l'alarme est donc déclenchée. L'alarme reste enclenchée tant que les sorties des bascules sont identiques.

Réplique de chemin critique Cette méthode est basée sur l'introduction d'un nouveau chemin avec des contraintes de placement. Le principe est de dupliquer le chemin critique, y compris la bascule du point de départ et celle du point d'arrivée pour détecter une violation de temps de propagation sur ce chemin dupliqué. Les mêmes cellules sont utilisées pour recréer le chemin en respectant le *fan-out* de chaque porte. Pour rendre le chemin répliqué actif, les entrées des portes sont connectées à V_{dd} ou à la masse de manière à rendre la sortie dépendante de la valeur du signal du point d'entrée. La figure 3.33 montre un exemple de chemin critique et sa réplique. La réplique est placée dans le circuit, aussi proche que possible de la référence dans le but de capter les mêmes variations. À l'image des autres détecteurs, la réplique du chemin critique doit être le premier à violer les contraintes de temps de propagation en cas d'attaque par impulsion négative de tension. Le temps de propagation de la réplique doit donc être supérieur à celui du chemin de référence. Il y a plusieurs façons de ralentir le chemin répliqué. Le *fan-out* d'une ou plusieurs cellules peut être augmenté : la capacité de charge de la porte considérée augmente. Grâce aux résultats de la section 3.1.2.1,



(a) Schéma



(b) Diagramme temporel

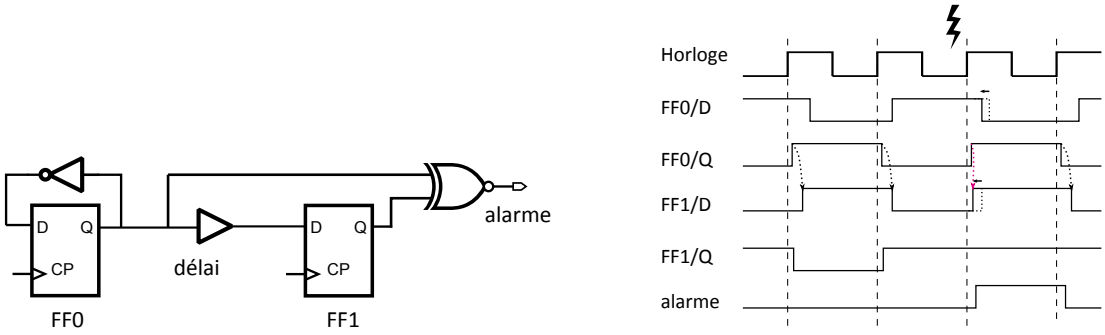
FIGURE 3.33 : Réplique de chemin critique

cette augmentation conduit à une augmentation du temps de propagation. Une autre possibilité consiste à ajouter un élément de délai dans la logique combinatoire. Il peut s'agir d'une série de *buffers* ou d'inverseurs. Ici, la décision d'ajouter un élément de délai a été prise parce qu'il s'agit de la manière la plus simple d'ajouter un délai connu avec précision pour assurer les marges de fiabilité.

On peut observer sur la figure 3.33(b) que le chemin critique n'est pas toujours actif. En effet, la sortie de FF2 ne change pas au deuxième cycle. Cependant, la réplique doit être toujours activée afin de détecter une attaque. Même si le chemin critique est actif, avec l'ajout du délai supplémentaire, la violation a lieu d'abord sur le chemin répliqué comme le montre la figure.

Détection d'attaques par impulsions positives. À l'image des circuits de détection d'attaque par impulsions négatives, l'idée est de pouvoir détecter une violation, notamment de temps de *hold* sur le circuit de détection. Trois circuits sont proposés.

Circuit A Le principe de ce premier circuit de détection est d'utiliser un chemin combinatoire entre deux bascules, ce chemin étant le plus rapide du circuit à protéger (figure 3.34). Par conséquent, le chemin fonctionnel le plus rapide du circuit doit être



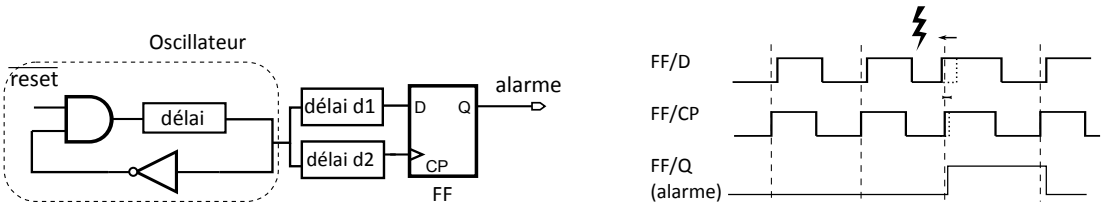
(a) Schéma (b) Diagramme temporel

FIGURE 3.34 : Circuit A

identifié. Le temps de propagation de la logique combinatoire entre doit être suffisamment court pour violer la contrainte de temps de propagation en *hold* de la bascule du point d'arrivée lorsque les délais sont réduits à cause d'une impulsion positive de tension. L'alarme est ainsi déclenchée lorsque la sortie des deux bascules est identique. Il est important de noter que le temps de propagation à travers la boucle de retour de la bascule FF0 doit être plus lent qu'à travers l'élément de délai, ceci afin d'assurer la détection sur FF1.

Circuit B Le deuxième circuit utilise le même principe que la réplique du chemin critique (figure 3.33) : il s'agit de dupliquer cette fois-ci le chemin le plus critique vis-à-vis de la contrainte de temps de *hold*. À noter que l'élément de délai rajouté dans le cas de la détection d'impulsions négatives n'est plus utile ici. Par contre, il faut s'assurer que le chemin répliqué viole la contrainte de temps de *hold* avant tout chemin du circuit à protéger. Cela est fait en choisissant une bascule d'entrée avec un temps de traversée (t_{cp2q}) plus court que celui de la bascule de référence.

Circuit C Le circuit est composé d'un oscillateur en anneau, de deux blocs de délai et d'une bascule, comme illustré sur la figure 3.35. L'oscillateur est utilisé pour cadencer la bascule. Sa fréquence est plus élevée que celle de fonctionnement du circuit. Cela permet de détecter des erreurs à chaque cycle d'horloge du circuit. L'oscillateur



(a) Schéma (b) Diagramme temporel

FIGURE 3.35 : Circuit C

permet d'avoir un signal d'horloge dont le rapport cyclique est connu et maîtrisé. Le signal de sortie traverse les blocs de délai $d1$ et $d2$ avant d'atteindre les entrées de la bascule. Les blocs $d1$ et $d2$ sont constitués de différents types de *buffers*, de telle sorte qu'à tension nominale, $d2$ est plus rapide que $d1$. La sortie de la bascule FF est alors à l'état bas. Lorsqu'un pic de tension est appliqué au-dessus de la tension maximale spécifiée, $d2$ devient plus lent que $d1$. Ainsi, le front montant de l'horloge arrive sur l'entrée D de la bascule avant d'atteindre l'entrée CP : la sortie de la bascule est à l'état haut. Pour définir les blocs de délai $d1$ et $d2$, les résultats de la section 3.1.2.1 sont exploités : $d1$ doit avoir une forte variation de délai en fonction de la tension tandis que $d2$ doit avoir une variation de délai beaucoup plus faible. Cela est possible en choisissant la taille des *buffers* utilisés ainsi que leur charge, en augmentant le *fan-out* par exemple.

Bilan. Trois circuits de détection d'attaques par impulsions positives et par impulsions négatives ont été retenus pour implantation sur silicium. Les solutions proposées ont l'avantage d'avoir une architecture simple, d'être peu coûteuses en surface et de s'intégrer facilement dans le flot de conception numérique. En effet, la conception peut se faire en utilisant les outils commerciaux d'aide à la conception. Par la suite, les différentes étapes et contraintes d'intégration de ces circuits sont présentées : il s'agit de l'analyse des temps de propagation et des contraintes de temps de propagation à appliquer sur le circuit et sur les détecteurs.

3.2.4 Intégration des solutions

La validation des solutions de détection passe par leur intégration dans un circuit réel afin de caractériser leur fonctionnement et de vérifier les seuils de détection. En effet, des seuils de détection trop hauts fragiliseraient le circuit vis-à-vis des attaques tandis que des seuils trop bas auraient des conséquences sur les performances du système en provoquant des fausses détections. La première étape consiste à définir les contraintes de fonctionnement des circuits de détection. L'étape suivante est la vérification de ces contraintes une fois que les détecteurs sont intégrés dans le circuit à protéger.

3.2.4.1 Implantation

L'objectif est d'être capable de détecter des attaques par impulsions de tension à travers la modification des temps de propagation. Pour réaliser cet objectif, une méthodologie est suivie pour introduire les circuits de détection (figure 3.36).

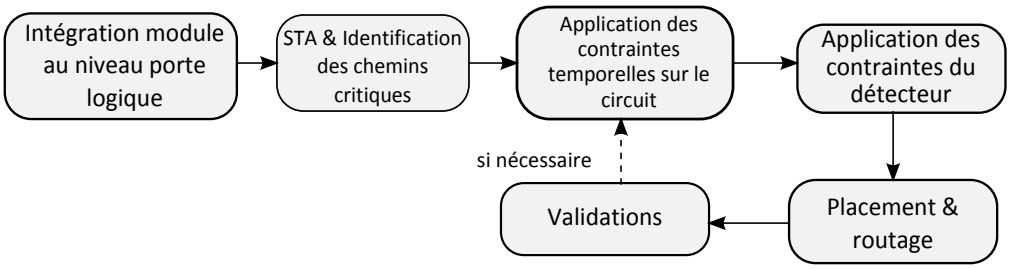


FIGURE 3.36 : Méthodologie d'intégration des circuits de détection

Il est important de noter que les étapes déroulées ici suivent le flot de conception numérique, et n'utilisent que des portes logiques disponibles dans la bibliothèque de cellules standard. Cela rend les solutions développées applicables dans un cadre industriel.

L'insertion du module contenant le détecteur peut se faire au moment de la description *RTL* ou après l'étape de synthèse, en insérant directement les portes logiques. Étant donné que la détection est basée sur la variation des temps de propagation, l'étape suivante est de réaliser une analyse temporelle statique (*STA*) du circuit après synthèse. L'analyse est effectuée pour plusieurs conditions de fonctionnement, notamment aux extrema de tensions spécifiées. Pour cela, un outil d'analyse de temps de propagation est utilisé, tel que PrimeTime de *Synopsys*. On se place dans des conditions de fonctionnement données (procédé de fabrication, température et tension) puis les temps de propagation sont répertoriés. Cette étape fournit la distribution temporelle des chemins logiques. Une attention particulière est portée aux chemins critiques vis-à-vis des contraintes de temps de *setup* et de temps de *hold* car les marges de détections sont définies par rapport à ces chemins. Une fois les chemins critiques identifiés, les contraintes temporelles de fonctionnement sont appliquées sur le circuit, qui vont permettre à la fois de définir les marges de détection et les seuils de détection. Ces contraintes concernent spécifiquement les étapes de placement et de routage. Deux cas de figure peuvent alors se présenter :

– La fréquence de fonctionnement du circuit est connue : les contraintes à appliquer sont définies par rapport à la période et aux temps de *setup* et *hold*. Dans la pratique, les temps de *setup* et de *hold* peuvent être majorés en prenant en compte leur définition tel que présenté dans la section 3.1.1.2. Le temps de *setup* peut être majoré par le temps de traversée de quatre inverseurs tandis que le temps de *hold* par celui d'un inverseur. Par exemple, si la période de fonctionnement est de 10 ns,

la contrainte minimale à appliquer est :

$$délai\_max = 10 - max(t_{setup} + t_{hold}) - max(t_{skew}), \tag{3.14}$$

avec $max(t_{skew})$ l'écart maximum de temps entre l'arrivée du signal d'horloge sur deux bascules du circuit. Ce temps correspond à la marge supplémentaire pour éviter que le détecteur ne se déclenche à tension nominale.

– La fréquence de fonctionnement n'est pas connue à l'avance, elle est déterminée par le chemin critique. Cette fois-ci, la contrainte à appliquer est définie par rapport à la valeur donnée par l'analyse statique de temps de propagation, auquel il faut rajouter un pourcentage de l'ordre de 20 % pour tenir compte du routage. Ainsi, si l'analyse donne un chemin critique à 6.5 ns, le délai de 20 % considéré pour le routage est 1.3 ns. La contrainte à appliquer est :

$$délai\_max = 6.5 + 1.3 + max(t_{setup} + t_{hold}) - max(t_{skew}) \tag{3.15}$$

De la même façon, pour les temps de propagation minimums, il faut définir un écart de temps de propagation entre le circuit et le détecteur d'une valeur correspondante à $t_{setup} + t_{hold} + max(t_{skew})$. D'autre part, si la marge de temps de *hold* t_{slack_h} (*hold slack*) est définie par :

$$t_{slack\_h} = t_{cp2q} - t_{skew} + t_{logic} - t_{hold}, \tag{3.16}$$

une manière simple d'augmenter la robustesse vis-à-vis des violations est d'augmenter t_{logic}. Une contrainte de temps de propagation minimale peut être appliquée, elle doit être supérieure au temps de propagation du chemin le plus rapide du circuit :

$$délai\_min > t_{logic_{min}}. \tag{3.17}$$

Il faut ensuite maîtriser la variation du chemin critique. Pour cela, il peut être nécessaire de lancer une première étape de placement et de routage afin de sélectionner le chemin critique. Ce choix se base sur l'analyse statique de temps de propagation effectuée après ce premier routage. Dès lors, une contrainte supplémentaire est ajoutée sur ce chemin : elle consiste à s'assurer que son temps de propagation minimal soit aussi proche que possible du temps de propagation maximal $délai\_max$. De cette façon, le délai du chemin critique est forcé à une valeur correspondant à $délai\_max$. Cette contrainte augmente le nombre de cellules du chemin critique et donc crée une variation supplémentaire de temps de propagation par rapport aux autres chemins du circuit, tel que montré dans le paragraphe 3.1.2.1.

Par la suite, les contraintes du détecteur sont définies. Celui-ci peut avoir des contraintes de placement, afin d'être localisé à un endroit précis du circuit et également des contraintes de temps de propagation. Pour des blocs entiers ou des modules,

les contraintes de placement sont définis pendant l'étape de définition du plan d'implantation des blocs du circuit (*floorplan*). En ce qui concerne les cellules individuelles, elles peuvent être placées à une location définie grâce aux coordonnées du plan de dessin de masques pendant la phase de placement. Les contraintes de temps de propagation entraînent à choisir la taille des cellules ou encore leur *fan-out*, pour répondre au besoin de variation en fonction de la tension.

Lorsque toutes les contraintes sont définies, le placement et le routage peuvent être lancés, avec des vérifications intermédiaires après le placement. À la fin du routage, une analyse statique de temps de propagation est exécutée, avec en complément des simulations rétro-annotés afin de valider que toutes les contraintes ont été appliquées. Il est possible que pour des raisons de surface ou de routage, certaines contraintes ne soient pas respectées. Les contraintes appliquées sur le circuit ou sur le détecteur peuvent alors être modifiées si elles sont trop agressives. Les étapes de placement et de routage sont réitérées avec les mêmes vérifications comme indiqué sur la figure dans le but de valider toutes les contraintes en fin de conception. Il s'agit de la vérification du respect des contraintes de délai par analyse statique de temps de propagation.

Le circuit répliqué réglable (figure 3.32) et les lignes parallèles de délai (figure 3.31) sont implantés afin de réduire leur surface et d'être en mesure de les comparer de façon équitable. Ainsi, il n'y a que deux lignes de délai dans le circuit à base de lignes de délai et 4 multiplexeurs dans le circuit répliqué réglable.

La méthodologie décrite ci-dessus va être appliquée pour intégrer l'ensemble des détecteurs étudiés dans la partie précédente. Les résultats de l'implantation et de la caractérisation des détecteurs seront présentés.

3.2.4.2 Vérification et caractérisation en phase de conception

Les différents circuits de détection sont caractérisés contre les attaques par impulsion positives et négatives. Pour cela, ils ont été utilisés pour détecter des attaques sur un circuit cryptographique exécutant l'algorithme *AES*. Ce circuit est utilisé à titre d'exemple, son architecture ou son fonctionnement n'a pas d'incidence sur les performances des détecteurs testés.

Le circuit de test a été conçu en technologie CMOS 28 nm. Le flot de conception numérique a été appliqué depuis les spécifications jusqu'aux tests finaux et la mise en boîtier. Les principales caractéristiques du circuit complet (*AES* et détecteurs) sont données dans le tableau 3.5. Le circuit est composé d'environ 15000 portes logiques pour une surface de 0.05 mm^2. La fréquence spécifiée de fonctionnement est de 100 MHz.

La simulation d'attaque dynamique en tension requiert de modifier de manière transitoire la tension d'alimentation, ce qui n'est pas possible en utilisant le flot de simulation rétro-annotée habituel. En effet, les simulations rétro-annotées sont basées sur des temps mesurés à des tensions d'alimentation statiques. C'est pourquoi l'étude ici est

TABLEAU 3.5 : Caractéristiques du circuit testé

Surface	0.05 mm²
Nombre de portes logiques	15000
Fréquence de fonctionnement spécifiée	100 MHz
Technologie	CMOS 28 nm

focalisée sur la variation des temps de propagation en plus des temps de propagation statiques. La caractérisation consiste donc à mesurer les temps de propagation statiques mais également à regarder leur variation en fonction de la tension d'alimentation.

Caractérisation des détecteurs d'impulsions positives. Le chemin critique à protéger est le compteur de la machine d'état de l'*AES*. Pour chaque détecteur, la marge de temps de *hold* est évaluée sur le circuit final après routage. Les résultats de simulation sont présentés sur la figure 3.37. Les marges de temps de *hold* sont données

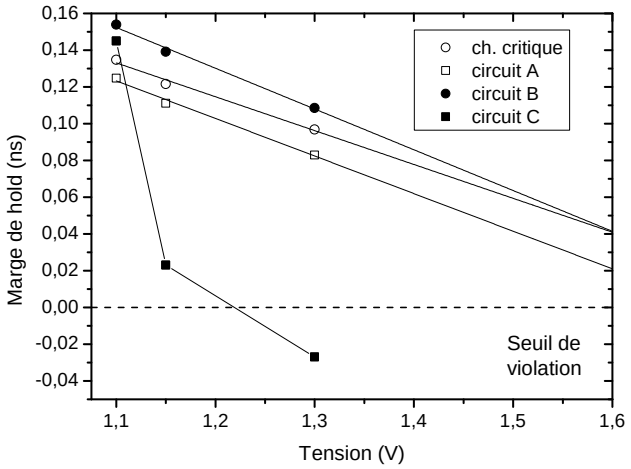


FIGURE 3.37 : Évaluation des détecteurs d'impulsions positives

pour des tensions allant de 1.1 V à 1.6 V. Les performances des différents détecteurs ont été obtenues pour différents procédés de fabrication. Toutefois, les résultats sont donnés pour un procédé de fabrication rapide et une température de fonctionnement de 125°C car c'est dans cette configuration que les marges obtenues sont les plus petites. La variation obtenue pour le circuit C est due à la variation différente des blocs de délai $d1$ et $d2$. La différence de temps de propagation entre ces deux blocs entraîne une variation non-linéaire de la marge de temps de *hold*. À l'exception du circuit C, dans la gamme de tensions regardée, la variation des marges peut être approchée par une droite. Au-delà de cette gamme, les temps de propagations sont proches de 0 et la pente des courbes décroît.

Le circuit C est le premier à détecter une attaque. En effet, pour une tension statique de 1.22 V, l'alarme est déclenchée. À partir de cette tension, le bloc de délai $d1$ devient plus rapide que $d2$, ainsi la bascule capture une valeur logique haute. Les deux autres circuits ont leur marge de temps de *hold* plus proche de celle du chemin critique. Cependant les pentes de ces courbes sont différentes. Les marges du circuit A sont toujours plus petites que celle du chemin critique. De plus, la pente est plus grande (en valeur absolue) que celle de la courbe du chemin critique. Le circuit A viole donc la contrainte de temps de *hold* avant le chemin critique. La marge du circuit C est plus grande que celle du chemin critique jusqu'à 1.6 V. Au-delà de cette tension, les deux marges se croisent et celle du circuit C devient plus petite. À l'image des circuits A et B, le circuit C devrait violer les contraintes de temps de propagation avant le chemin critique, ce qui correspond au comportement attendu.

Les circuits A et B sont réglés en fonction de la marge du chemin critique. Leur marge en est donc proche. Par contre, le circuit B est réglé pour détecter un niveau de tension basé sur les temps de propagations des portes utilisées pour constituer les blocs $d1$ et $d2$. Pour cette raison, le circuit B est plus apte à détecter des attaques par impulsions positives quel que soit l'effet sur le circuit tandis que les deux autres sont destinés à détecter ces attaques lorsqu'une faute est effectivement injectée.

Caractérisation des détecteurs d'impulsions négatives. La figure 3.38 présente les résultats obtenus en simulation, pendant la phase de conception, pour les différents circuits de détection. La fréquence maximale (l'inverse du temps de propagation) du chemin critique, ainsi celle des détecteurs est donnée pour des tensions d'alimentation statiques allant de 0.6 V à 1.0 V. Les caractéristiques sont évaluées pour un procédé de fabrication lent et une basse température (-40°C) ce qui correspond au pire cas de fonctionnement en ce qui concerne les temps de propagation.

Tout d'abord, le temps de propagation du chemin critique du circuit à protéger est plus petit que celui de l'ensemble des détecteurs, quelle que soit la tension considérée. Ensuite, la variation de la fréquence en fonction de la tension peut être comparée. Elle est donnée par la pente de chacune des droites. Les valeurs de ces pentes sont indiquées dans le tableau 3.6. Le chemin critique présente la variation en fréquence la plus grande, suivi du détecteur à base de chemin critique répliqué, des lignes parallèles de délai et enfin le chemin répliqué réglable. En considérant à la fois les temps de propagation et la variation en fréquence, on s'assure que pour toute variation de la tension d'alimentation (amplitude et largeur d'impulsion), le chemin combinatoire des détecteurs est plus lent que celui du chemin critique. En effet, la fréquence maximale donnée par les différents circuits se déplace le long de la droite qui définit la variation de fréquence en fonction de la tension. Si les courbes ne s'entrecroisent pas, l'ordre des chemins n'est pas modifié. Ainsi, les détecteurs sont les premiers à violer les contraintes de temps propagation.

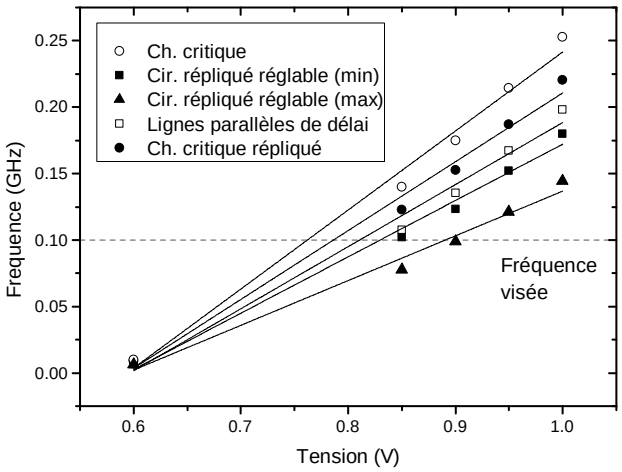


FIGURE 3.38 : Caractérisation des détecteurs d'impulsions négatives

TABLEAU 3.6 : Variation de fréquence en fonction de la tension des détecteurs et du chemin critique

Circuit	Pente ($GHz.V^{-1}$)
Cir répliqué réglable (max)	0.3367
Cir. répliqué réglable (min)	0.4241
Lignes parallèles de délai	0.4672
Ch. critique répliqué	0.5180
Ch. critique	0.5948

Finalement, en considérant à la fois la fréquence et sa variation, toutes les solutions présentées détectent une violation de temps propagation avant qu'elle n'intervienne sur le circuit protégé.

Parmi les détecteurs, le chemin critique répliqué est celui qui a un seuil de détection le plus proche du circuit. Les autres solutions ont des seuils de détections plus hauts, donc sont plus susceptibles de créer des cas de détection faux-positifs. Le tableau 3.7 indique les marges minimums requises et celles relevées pour le chemin critique répliqué à différentes tensions. Les marges requises sont données par l'inéquation 3.11. Ces marges permettent de prendre en compte la marge minimum de fiabilité du détecteur mais aussi un délai supplémentaire pour qu'il n'y ait pas de détection à tension nominale.

Une fois que tous les circuits ont été caractérisés et validés en simulation, ils ont été envoyés en fabrication. Les solutions proposées ici ont donc pu être testées sur un circuit réel. La partie suivante présente les performances de ces circuits de détection obtenues suite aux tests réalisés sur silicium.

TABLEAU 3.7 : Marges de détection requises et mesurées en simulation pour le chemin critique répliqué

Tension (V)	Marge requise (ns)	Marge mesurée (ns)
0.60	3.69	11.36
0.85	0.27	1.03
0.90	0.21	0.83
0.95	0.17	0.68
1.00	0.15	0.58

3.2.5 Résultats des tests silicium

Cette section se consacre à l'évaluation des circuits de détection par des tests électriques. Dans un premier temps, le dispositif expérimental pour la réalisation des tests sera décrit. Par la suite, les mesures effectuées sont présentées puis analysées. Finalement, les résultats des tests sont comparés avec ceux attendus à savoir les résultats de simulation, ce qui permettra de voir les avantages et les limitations des circuits de détections étudiés.

3.2.5.1 Dispositif expérimental

La figure 3.39 illustre le dispositif utilisé. La procédure consiste à appliquer des impulsions de différentes formes puis de regarder la réponse du circuit et également celle des détecteurs. Pour cela, un générateur de vecteurs de test est utilisé. Le générateur permet de programmer de façon précise la valeur de l'amplitude de l'impulsion mais aussi sa largeur. Il peut délivrer une tension comprise entre -3.8 V et 3.8 V, ce qui est suffisant pour notre étude sachant que la tension nominale du circuit est de 1 V. La largeur d'une impulsion peut être réglée par pas de 5 ns. On dispose également d'un oscilloscope qui permet d'observer l'impulsion qui est envoyée sur l'entrée d'alimentation du circuit de test et de se synchroniser vis-à-vis du fonctionnement du circuit. La génération des stimuli d'entrée du circuit se fait à partir du testeur. Enfin, un générateur de tension continue délivre la tension d'alimentation de la logique de test à l'intérieur du circuit.

Pour réaliser des impulsions positives, le générateur de vecteurs de test utilisé ne permet pas de délivrer un courant suffisamment élevé compte tenu des valeurs de tension appliquées. Le générateur de tension continue est donc utilisé pour fournir le courant nécessaire et le générateur de vecteurs permet de générer l'impulsion. L'association des deux générateurs se fait à l'aide d'un *bias tee* tel que présenté sur la figure 3.40. Il s'agit d'un composant à trois ports permettant de polariser un circuit sous une tension ou un courant continu sans affecter la composante fréquentielle. La capacité autorise la composante fréquentielle du signal d'entrée (le générateur de vecteurs de tests), mais

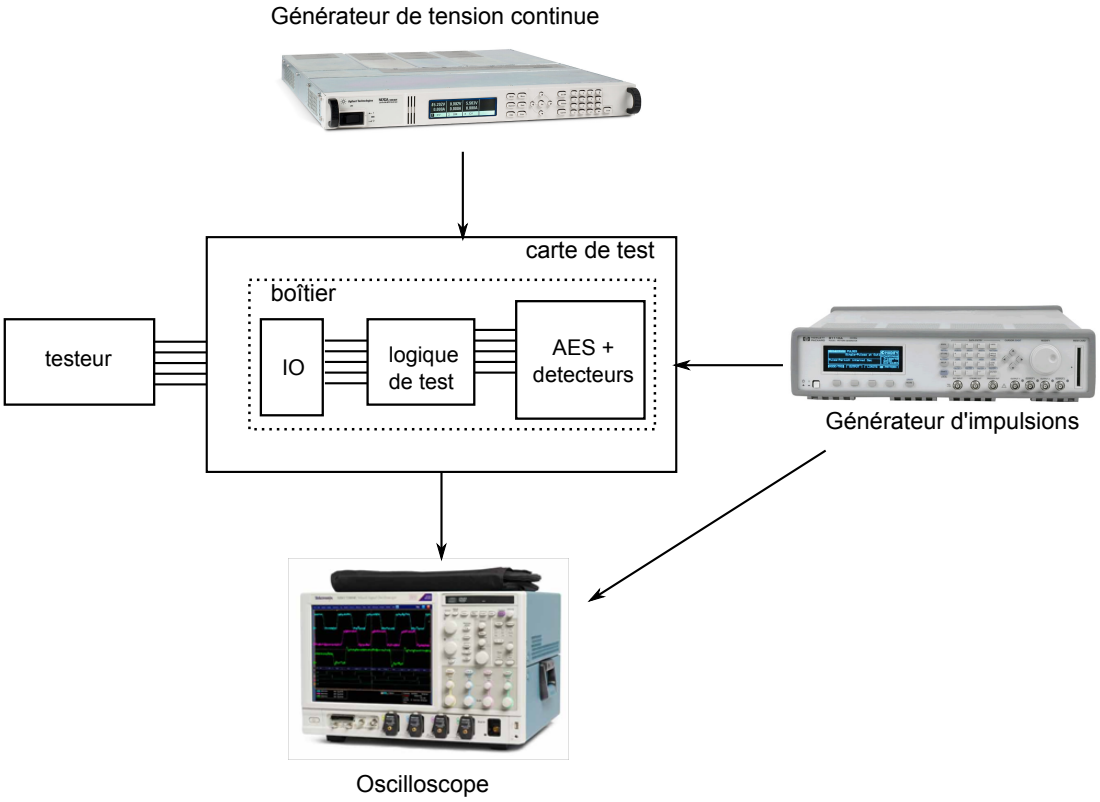


FIGURE 3.39 : Dispositif de test des circuits de détection

filtre la composante continue, et inversement pour l'inductance. Ainsi, à la sortie sont transmises à la fois les composantes continues et fréquentielles.

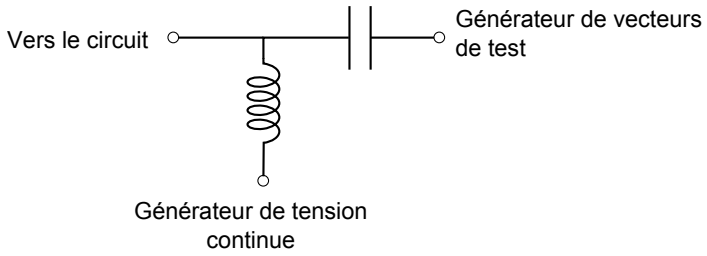


FIGURE 3.40 : *Bias tee*

Le testeur utilisé fonctionne sous une plateforme *FPGA* virtex-5 [Xilinx 2009]. Il permet d'envoyer les stimuli d'entrée au circuit de test (les textes à chiffrer), mais également de récupérer et d'observer les signaux de sortie (les textes chiffrés et les signaux d'alarme des détecteurs).

Les fautes injectées dans le circuit de test ainsi que le fonctionnement des circuits de détection d'impulsions négatives dépendent de la fréquence du signal d'horloge. Compte

tenu du testeur utilisé, la fréquence pour les mesures est limitée à 30 MHz. Les tests sont réalisés à cette fréquence pour la détection d'impulsions négatives. Pour la caractérisation des détecteurs d'impulsions positives, la fréquence est ramenée à 25 MHz. Afin d'effectuer des tests à 100 MHz, il aurait fallu utiliser un générateur externe d'horloge, ainsi qu'un testeur capable d'échantillonner les sorties du circuit de test à une fréquence 2 à 3 fois supérieure (200 à 300 MHz) pour une meilleure observabilité des signaux de sortie.

3.2.5.2 Procédure de test

La réalisation d'une campagne de test se fait en suivant la procédure définie ci-dessous.

Algorithme 1: Protocole de test

Régler la fréquence de fonctionnement du circuit
for $i = 0$ **to** $nombre\_impulsions$ **do**
  Choisir le moment d'injection de l'impulsion (*trigger*)
  Régler de l'amplitude (positive ou négative) de l'impulsion de tension
  Régler la largeur de l'impulsion de tension
  Choisir le vecteur d'entrée (texte à chiffrer)
  Choisir le vecteur d'entrée (clé de chiffrement)
  Lancer le cryptage
  Observer les alarmes
  Récupérer le texte chiffré
  Comparer avec le résultat attendu
end

Les premières étapes consistent à régler les paramètres de fonctionnement, à savoir la fréquence de fonctionnement, les vecteurs d'entrée et la configuration de l'impulsion de tension. Une fois ces paramétrages effectués, le chiffrement est réalisé. Les 128 bits du texte chiffré sont ensuite récupérés en série, puis comparés au résultat attendu à l'aide d'un script. La valeur des alarmes est également récupérée, ce qui permet de voir si l'attaque a été détectée par chacune d'entre-elles. Pour l'envoi d'impulsions, les tests ont été réalisés à partir de la $6^{ème}$ ronde, et peuvent s'étendre jusqu'à la $10^{ème}$ en fonction de la largeur de l'impulsion. Le chiffrement s'effectuant en 11 cycles d'horloge, ce moment d'injection permet de s'assurer que la largeur des impulsions couvre la durée du chiffrement. À noter également, que les réglages de l'impulsion (moment d'injection, amplitude et largeur) ne sont pas automatisés, il faut donc changer manuellement ces paramètres pour chaque test effectué.

3.2.5.3 Mesures et résultats

Cette section présente les résultats de caractérisation des différents circuits de détection. L'analyse de ces résultats sera faite dans la partie suivante. Lorsque le dispositif expérimental est mis en place, la première étape consiste à vérifier le fonctionnement du circuit. Pour cela, le circuit de test est alimenté à une tension nominale puis plusieurs vecteurs de tests sont envoyés pour vérifier que le chiffrement s'effectue correctement. D'autre part, la valeur des signaux d'alarme de chacun des détecteurs est également vérifiée, les alarmes doivent être à 0 lorsque le circuit fonctionne sans attaques. Par la suite, les seuils de déclenchement des alarmes sont caractérisés pour des valeurs statiques de tensions d'alimentation . La tension est abaissée ou augmentée progressivement à partir de la valeur nominale, jusqu'à l'apparition des premières alarmes. Les résultats sont reportés dans les tableaux 3.8 et 3.9 sous forme de shmoos. Les cases vertes indiquent l'absence de fautes ou des alarmes non déclenchées pour les détecteurs. Les cases rouges indiquent des fautes ou des alarmes enclenchées.

Mesures statiques. La tension d'alimentation est abaissée progressivement par pas de 100 mV. Les premières fautes de chiffrement pour une diminution statique de la tension apparaissent pour une tension statique de 0.55 V. A cette tension, les détecteurs ont tous signalé une violation. Le circuit répliqué réglable est le premier à être déclenché, à une tension de 0.58 V, suivi des lignes parallèles de délai à 0.57 V, et enfin du chemin critique répliqué qui est activé à 0.56 V. Cependant, lorsque la tension est abaissée davantage, il y a une gamme de tensions entre 0.53 V et 0.52 V où le chemin répliqué réglable n'est plus activé. De la même façon ce phénomène existe pour les lignes parallèles de délai pour une tension de 0.50 V.

TABLEAU 3.8 : Caractérisation des circuits de détection pour des tension statiques inférieures à la tension nominale

Tension (V)	1.00	0.90	0.80	0.70	0.65	0.60	0.58	0.57	0.56	0.55
AES										
Ch. répliqué réglable (max)										
L. parallèles de délai										
Ch. crit. répliqué										

Tension (V)	0.54	0.53	0.52	0.51	0.50	0.49	0.48	0.47	0.45	0.40
AES										
Ch. répliqué réglable (max)										
L. parallèles de délai										
Ch. crit. répliqué										

TABLEAU 3.9 : Caractérisation des circuits de détection pour des tensions statiques supérieures à la tension nominale

Tension (V)	1.00	1.05	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1
AES													
Circuit A													
Circuit B													
Circuit C													

En ce qui concerne les tensions supérieures à la tension nominale, les résultats sont présentés dans le tableau 3.9. Cette fois-ci, aucune erreur n'a été observée sur le circuit de test en faisant monter la tension jusqu'à 2.1 V. Seul le circuit C détecte les surtensions à partir de 1.05 V. Les autres circuits de détection ne déclenchent pas d'alarmes sur la gamme de tensions testée. La valeur maximale de tension a été limitée à 2.1 V pour ne pas endommager le circuit et risquer de le détruire.

Mesures dynamiques. La vérification des seuils de détection statiques a permis de valider le fonctionnement des circuits de détection. Dès lors, les différents circuits sont testés face à des attaques par impulsions. La procédure consiste à appliquer des impulsions de différentes largeurs et amplitudes et de vérifier le seuil de déclenchement des alarmes. Une campagne d'attaque par impulsions négatives est d'abord réalisée, puis une autre concernant les impulsions positives. Compte tenu du dispositif utilisé, peu de données d'entrée (textes à chiffrer) ont pu être testées sur l'*AES*. Néanmoins, les impulsions sont réalisées à des cycles différents en fonction de la largeur d'impulsion, ce qui permet d'attaquer des données différentes.

Impulsions négatives Les résultats des tests réalisés sont présentés dans les tableaux 3.10, 3.11, 3.12 et 3.13. Les impulsions envoyées sur le circuit ont des amplitudes allant de 350 mV à 650 mV, et des largeurs allant de 30 ns à 150 ns. Cette largeur d'impulsion permet de couvrir des durées allant d'une à plusieurs périodes.

Les premières fautes apparaissent sur l'*AES* pour des impulsions d'amplitude supérieure à 450 mV. Pour une largeur d'impulsion de 30 ns, il faut augmenter l'amplitude de l'impulsion à 650 mV. Les impulsions inférieures à 450 mV d'amplitude n'ont pas provoqué de fautes dans l'exécution de l'algorithme. À noter que pour deux cas de figure, à savoir une amplitude de 650 mV et des largeurs de 125 ns et 150 ns, aucune réponse du circuit n'a été enregistrée (les sorties sont restés à 0). Cela correspond aux cases grisées dans le tableau 3.10.

Le chemin répliqué réglable détecte les impulsions à partir de 400 mV d'amplitude et de largeur supérieure à 50 ns. Pour des impulsions plus courtes, il faut monter l'amplitude à 550 mV pour déclencher une alarme. Il est important de noter qu'il existe

deux zones où l'alarme n'est pas enclenchée : pour une impulsion de largeur de 100 ns et d'amplitude variant de 550 à 600 mV, et pour une impulsion de 100 ns et 650 mV d'amplitude.

Les premières détections des lignes parallèles de délai interviennent lorsqu'une impulsion d'amplitude supérieure à 400 mV est appliquée. Les largeurs d'impulsion détectées pour cette amplitude sont de 100 ns et 125 ns. Il faut augmenter l'amplitude à 425 mV pour les autres largeurs d'impulsion. Par contre, pour 30 ns de largeur d'impulsion, une amplitude de 550 mV est nécessaire pour détecter l'attaque. A noter également qu'il existe des impulsions non détectées pour une impulsion de largeur de 100 ns et d'amplitude allant de 550 mV à 600 mV et un couple largeur-amplitude de (150 ns, 650 mV). Le chemin critique répliqué détecte des attaques lorsque l'amplitude de l'impulsion atteint 425 mV, pour une largeur comprise entre 75 ns et 100 ns. À partir de 450 mV d'amplitude, les impulsions de largeur supérieure à 50 ns sont toutes détectées. Enfin, pour une largeur de 30 ns, les attaques sont détectées à partir de 550 mV d'amplitude. Pour ce circuit, il n'y a pas eu d'impulsions de forte amplitude non détectées, comme c'est le cas avec les deux autres circuits de détection.

Impulsions positives Les tableaux 3.14, 3.17, 3.15 et 3.16 présentent les résultats obtenus pour les différents lors de la campagne de mesures concernant les attaques par impulsions positives. Les largeurs d'impulsions appliquées varient de 30 ns à 150 ns, et les amplitudes, de 100 mV à 1.1 V au-dessus de la tension nominale. Lors des différents tests réalisés, aucune faute n'a été injectée sur le circuit de test pour la gamme de tensions et d'amplitudes indiquée dans le tableau 3.14. De la même manière, les circuits A et C n'ont détecté aucune faute pendant les différents tests réalisés. Seul le circuit B a détecté les attaques par impulsions : toutes les impulsions d'amplitude réalisées à partir de 1.1 V ont déclenché l'alarme du détecteur.

Les mesures effectuées sur les différents détecteurs permettent d'évaluer leur performance et leur efficacité, mais aussi de considérer la complexité de réglage et de définition des seuils de détection en phase de conception. L'analyse des résultats obtenus donne les avantages et les limitations des solutions étudiées.

TABLEAU 3.10 : Impulsions négatives sur l'*AES*

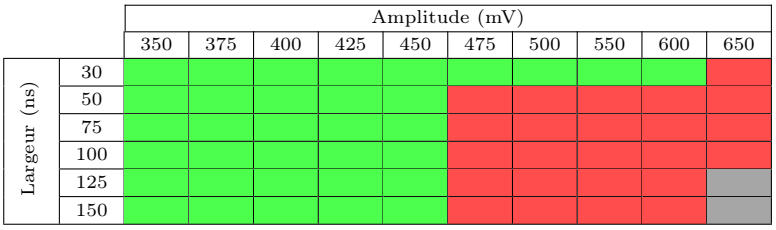


TABLEAU 3.11 : Impulsions négatives sur le circuit répliqué réglable

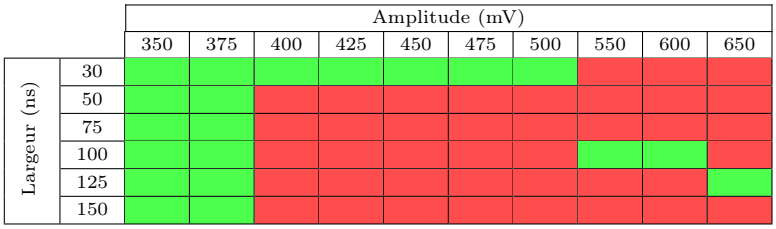


TABLEAU 3.12 : Impulsions négatives sur les lignes parallèles de délai

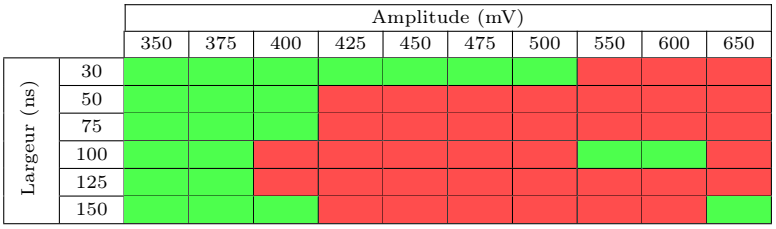


TABLEAU 3.13 : Impulsions négatives sur le chemin critique répliqué

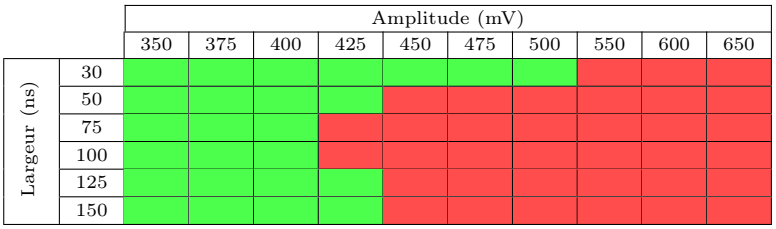


TABLEAU 3.14 : Impulsions positives sur l'*AES*

		Amplitude (mV)									
		100	300	400	500	600	700	800	900	1000	1100
Largeur (ns)	30										
	50										
	75										
	100										
	125										
	150										

TABLEAU 3.15 : Impulsions positives sur le circuit A

		Amplitude (mV)									
		100	300	400	500	600	700	800	900	1000	1100
Largeur (ns)	30										
	50										
	75										
	100										
	125										
	150										

TABLEAU 3.16 : Impulsions positives sur le circuit B

		Amplitude (mV)									
		100	300	400	500	600	700	800	900	1000	1100
Largeur (ns)	30										
	50										
	75										
	100										
	125										
	150										

TABLEAU 3.17 : Impulsions positives sur le circuit C

		Amplitude (mV)									
		100	300	400	500	600	700	800	900	1000	1100
Largeur (ns)	30										
	50										
	75										
	100										
	125										
	150										

3.2.6 Analyse et comparaison des mesures et des simulations

Cette partie vise à définir dans quelles mesures les attaques par impulsions sur l'alimentation peuvent être anticipées à travers les caractérisations faites pendant la phase de conception et les contraintes utilisées pour définir les seuils de détection. Dans un premier temps, les résultats des tests électriques sur circuit sont comparés à ceux attendus et définis par les contraintes de conception. Par la suite, les mesures sont analysées plus en détails afin de déterminer les performances et l'efficacité des solutions de détection.

3.2.6.1 Comparaison avec les résultats théoriques

Comme expliqué précédemment, le flot de développement numérique ne permet pas de faire des simulations dynamiques en ce qui concerne la tension d'alimentation. Les différentes analyses sont donc effectuées pour des tensions d'alimentation statiques. La configuration des solutions de détection étudiées ici s'est appuyée sur les caractéristiques statiques des portes logiques. Une première étape de comparaison consiste à vérifier l'ordre de déclenchement des alarmes pour une tension d'alimentation statique, aussi bien pour des surtensions que des sous-tensions. Ensuite, le positionnement absolu des seuils de détection par rapport à ceux définies pendant la phase de conception sera analysé.

Comparaison des sous-tensions. L'étude de la variation des temps de propagation et de leur variation a permis d'obtenir les seuils de détection en simulation tels que présentés sur la figure 3.38. L'ordre d'apparition des alarmes est le même quelle que soit la fréquence considérée. La contrainte d'effectuer les tests à une fréquence de 30 MHz ne modifie donc pas les résultats. Le chemin répliqué réglable était le premier à détecter une attaque compte tenu de la fréquence maximale de fonctionnement observée. Il y avait ensuite les lignes parallèles de délai et enfin le chemin critique répliqué. Ces résultats peuvent être comparés à ceux du tableau 3.8. En effet, l'ordre de déclenchement des alarmes relevé est le même que sur la figure. Cela valide la caractérisation faite par analyse statique de temps de propagation.

Les seuils de détection mesurés peuvent également être comparés. Les valeurs de ces seuils sont indiqués dans le tableau 3.18. Les seuils mesurés sur silicium sont beaucoup plus bas que ceux relevés en conception. Ce résultat était attendu car les circuits de détections ont été conçus pour fonctionner pour différents procédés de fabrication et une large gamme de températures. Les valeurs données dans le tableau ci-dessous correspondent à un scénario où les temps de propagation sont les plus longs (procédés de fabrication lent et une température de 125°C). Les mesures sur circuit ont été réalisées à température ambiante, ce qui explique ces différences importantes observées. Le fait de prendre en compte différents procédés de fabrication et la gamme de spécification

TABLEAU 3.18 : Seuils de détections des circuits et de violation de l'*AES* pour des sous-tensions

Circuit \ Seuil (V)	En conception	Mesures électriques
Ch. répliqué réglable (max)	0.68	0.58
Lignes parallèles	0.66	0.57
Ch. critique répliqué	0.65	0.56
Ch. critique (*AES*)	0.64	0.55

en température du circuit à protéger rend difficile le positionnement précis d'un seuil à température ambiante par exemple.

Comparaison des surtensions. Comme pour les sous-tensions, les seuils de détections statiques sont indiqués dans le tableau 3.19. Les mesures réalisées n'ont pas permis

TABLEAU 3.19 : Seuils de détections des circuits et de violation de l'*AES* pour des surtensions

Circuit \ Seuil (V)	En conception	Mesures électriques
Circuit A	> 1.6	> 2.1
Circuit B	> 1.6	> 2.1
Circuit C	1.22	1.05
Ch. critique (*AES*)	> 1.6	> 2.1

de mettre en évidence le seuil à partir duquel les premières fautes apparaissent sur le circuit de cryptographie. En effet, jusqu'à la tension appliquée, aucune faute n'a été relevée. Les circuits A et B n'ont pas été déclenchés non plus. En simulation, les bibliothèques de caractérisation des cellules logiques ne contiennent pas d'informations pour des tensions supérieures à 1.3 V, c'est pour cette raison que le seuil n'a pas pu être évalué précisément. En réalisant une extrapolation linéaire sur une gamme de tensions réduite (de 1.30 V à 1.6 V), le seuil de détection se trouve au-delà de 1.6 V. Seul le circuit C détecte les surtensions à partir de 1.05 V. La différence mesurée par rapport à la caractérisation en simulation peut s'expliquer par des variations de temps de propagation entre les cellules. En effet, les mêmes cellules sont utilisées en quantités importantes pour créer les blocs de délai. Il faudrait donc prévoir une marge supplémentaire pour les simulations rétro-annotées qui ont permis d'établir ces seuils. Les variations entre cellules sont habituellement prises en compte pour les analyses statiques de temps de propagation. L'architecture particulière du circuit B ne comprenant pas l'horloge globale du système, cette marge n'a pas été ajoutée par les outils de caractérisation. En effet, l'outil d'analyse de temps de propagation considère les chemins à travers les délais

$d1$ et $d2$ comme des chemins de données. Les contraintes appliquées pour calculer les marges de temps de propagation n'ont pas été prises en compte pour ce circuit. Ces contraintes prennent en compte les variations de procédés de fabrication qu'il peut y avoir à l'intérieur d'un même circuit.

Les mesures statiques de seuils de détection ont permis de valider le fonctionnement du circuit de test et des seuils de détection. La deuxième étape consiste donc à évaluer les performances et l'efficacité des solutions face aux attaques par impulsions.

3.2.6.2 Analyse des mesures

Le but ici est de déterminer l'efficacité des solutions proposées faces aux attaques par impulsions sur la tension d'alimentation. L'analyse se fera en deux parties, d'abord les impulsions négatives, suivies des impulsions positives.

Impulsions négatives. Les résultats de la caractérisation sur circuit des détecteurs ont été présentés dans les tableaux 3.10, 3.11, 3.12 et 3.13. L'application d'impulsions d'amplitude autour de 450 mV est toujours détectée par les circuits de protection avant qu'une faute n'apparaissent à la sortie de l'*AES*. À l'image de la détection des tensions statiques, le circuit répliqué réglable est le premier à détecter une attaque puis viennent les lignes parallèles de délai et le chemin répliqué réglable. En jouant sur l'amplitude ou la largeur de l'impulsion, les premières fautes, notamment celles dues à la métastabilité des données dans l'*AES* sont détectées. Toutefois, lorsque la largeur et l'amplitude des impulsions sont augmentées, le circuit répliqué réglable et les lignes parallèles de délai font apparaître deux zones de non-couverture de fautes. Une gamme d'impulsions en particulier (entre 550 mV et 600 mV d'amplitude et 100 ns de largeur) n'est pas détectée par ces deux solutions. Ces cas particuliers de non couverture apparaissent aussi lorsque ces deux circuits de détections sont testés pour des tension statiques (tableau 3.8). Pour les impulsions de 650 mV d'amplitude et 150 ns de largeur, toutes les sorties du circuit sont restées à l'état bas. Les opérations ne se sont donc pas exécutées. La tension d'alimentation se trouvant autour de la tension de seuil des transistors, l'ensemble du circuit peut donc être considéré comme non fonctionnel. En ce qui concerne les cas de non-détection, une hypothèse serait la métastabilité de la bascule qui délivre le signal de comparaison. L'augmentation des temps de propagation ferait que ce signal est capturé à chaque cycle dans la fenêtre $t_{setup} + t_{hold}$ de la bascule. Cela empêcherait le signal de basculer à chaque cycle, d'où une alarme qui n'est pas déclenchée. D'autre part, à très basse tension (proche de la tension de seuil des transistors), il peut y avoir un déséquilibre important entre les temps de montée et les temps de descente des signaux lorsque ceux-ci traversent un grand nombre de portes logiques comme illustré sur la figure 3.41. Ainsi, le signal a l'entrée D de la bascule est déformé, ce qui peut entraîner une valeur de sortie constante égale à 0 dans ce cas.

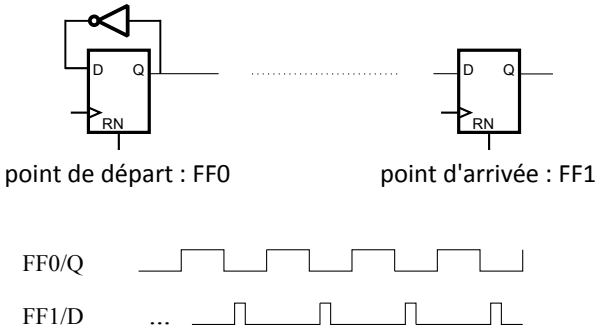


FIGURE 3.41 : Déformation des signaux envoyés due au déséquilibre entre les temps de montée et de descente des portes logiques

La combinaison de ces deux phénomènes provoquerait l'absence d'alarmes. Compte tenu du dispositif de test et du circuit utilisé, la valeur de sortie de la bascule de départ n'a pas pu être directement vérifiée expérimentalement pour confirmer ces hypothèses. Néanmoins, le phénomène de déformation des signaux dû aux différences de temps de montée et de descente des portes a pu être confirmé par des simulations des portes logiques au niveau transistor. Pour améliorer la détection de fautes, il faut donc essayer d'équilibrer au maximum les temps de montée et de descente des signaux. Pour cela, l'ajout d'inverseurs peut permettre de rééquilibrer les temps de propagation en montée et en descente comme on le montre la figure 3.42.

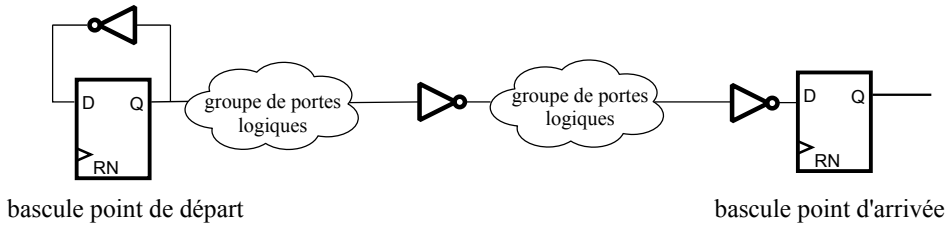


FIGURE 3.42 : Exemple de rééquilibrage des temps de propagation avec l'insertion d'inverseurs

Impulsions positives. Les résultats de ces attaques ont été donnés dans les tableaux 3.14, 3.17, 3.15 et 3.16. Seul le circuit B détecte les différentes impulsions envoyées, et ce, à partir de 100 mV d'amplitude. De même, les impulsions de courtes largeurs sont détectées. En ce qui concerne les largeurs minimums d'impulsions détectées, celles-ci sont directement dépendantes de la fréquence de l'oscillateur. En effet la comparaison des deux blocs de délai se fait à chaque front montant du signal d'oscillateur, ainsi il est possible de détecter des largeurs d'impulsions minimums correspondant à la durée d'une période d'horloge.

Avec les différents tests réalisés, les phénomènes autres que les violations de temps de propagation (éléments parasites sur les rails d'alimentation ou diaphonie) ont voulu être mis en évidence. C'est notamment pour cela que des impulsions d'amplitude deux fois supérieure à la tension nominale ont été appliquées. Ces expériences n'ont pas permis d'observer ces phénomènes. En effet, les impulsions positives n'ont pas généré de fautes dans le circuit, et les impulsions négatives ont toutes été détectées, notamment par le chemin critique répliqué.

Plusieurs formes d'impulsions ont pu être testées dans cette partie expérimentale. En fonction des perturbations qu'elles provoquent, elles ont pu être détectées ou non par les circuits de protection. Les impulsions sont envoyées dans le circuit par l'entrée de la tension d'alimentation de la carte de test, sur laquelle est connecté le boîtier du circuit. Ces impulsions traversent un ensemble d'éléments parasites avant d'arriver sur le circuit. On peut alors imaginer que l'impulsion envoyée est perturbée et peut potentiellement être différente de celle qui arrive effectivement sur le circuit. Le paragraphe suivant apporte une discussion sur ce phénomène.

Impulsion de tension à l'intérieur du boîtier. Pour déterminer la forme d'onde des impulsions effectives dans le circuit, il faut essayer de reconstruire les impédances des différents éléments du dispositif expérimental. Cela peut être représenté par un réseau à base de résistances, de capacités et d'inductances. Ainsi, en fonction de la valeur de ces différents éléments, l'impulsion envoyée peut être amortie ou provoquer des oscillations à l'intérieur du circuit. Chaque circuit ayant une valeur de capacité différente, la forme de l'impulsion effective dépend du circuit attaqué. Pour les tests réalisés, il n'a pas été prévu de dispositif permettant de connaître la perturbation exacte qui se déroulait à l'intérieur du boîtier. Cependant, la sensibilité du circuit B, permet de détecter des impulsions d'amplitude relativement faible (100 mV) pour une largeur inférieure à 30 ns. Lors des attaques par impulsion négatives, cette alarme n'a pas été déclenchée. Pour les impulsions positives, l'alarme restait à l'état haut pendant une durée proche de celle de l'impulsion. Ces premiers résultats montrent a priori qu'il n'y a pas d'oscillations de période supérieure à 30 ns et d'amplitude supérieure à 100 mV. Une étude plus approfondie sera nécessaire pour évaluer les perturbations effectives perçues par le circuit.

Les détecteurs proposés permettent de couvrir une certaine gamme d'attaques par impulsions compte tenu des résultats observés. Un comparatif des performances et des limitations de ces contremesures sera maintenant dressé.

3.2.6.3 Comparaison des détecteurs

Les circuits étudiés sont comparés selon plusieurs critères tels que présentés dans les tableaux 3.20 et 3.21. Les seuils de détection des circuits A et B sont au-delà 2.1 V.

Les largeurs d'impulsion détectées sont déterminées par la fréquence de l'horloge qui cadence chacun des circuits. Ainsi, les circuits A et B peuvent détecter des largeurs supérieures à la période de l'horloge du circuit à protéger (10 ns pour une amplitude supérieure à 2.1 V) tandis que le circuit C détecte des largeurs supérieures à la période de l'oscillateur (< 6.25 ns pour une amplitude de 1.1 V). En ce qui concerne la surface, les circuits A et B utilisent un nombre réduit de portes en comparaison avec le circuit C. En effet, celui-ci utilise deux blocs de délai pour fonctionner. Ces éléments de délai peuvent requérir un grand nombre de portes logiques pour créer la variation nécessaire à la détection. Ainsi le réglage du délai de ce circuit peut s'avérer difficile compte tenu du choix des portes pour créer la variation adéquate. Le circuit A a un délai spécifié qui est proche du chemin critique (l'écart est déterminé par la marge de détection). Le réglage du délai peut se faire par des simples contraintes de temps de propagation à appliquer pour le placement et le routage. Le délai du circuit B se base sur celui du chemin critique étant donné qu'il utilise les mêmes cellules. Il faut néanmoins prendre en compte le développement d'une solution permettant de reproduire le chemin critique de manière identique pendant la phase de placement et de routage, ce qui rend cette solution complexe. En effet, il faut être en mesure d'identifier toutes les cellules du chemin critiques et d'appliquer des contraintes de placement pour les positionner aussi proche que possible des cellules de référence. Il faut également reproduire le *fan-out* de chaque cellule en y mettant des contraintes pour éviter les optimisations de l'outil de placement. Pour ces mêmes raisons, la conception de ce circuit est complexe. La

TABLEAU 3.20 : Comparaison des circuits de détection d'attaques par impulsions positives

Circuit	A	B	C
Seuil de détection (V)	> 2.1	> 2.1	1.1
Largeur minimum détectée (amplitude supérieure au seuil)	Période de l'horloge du système		Période de l'oscillateur
Nombre de portes logiques utilisées	6	10	226
Réglage du délai	modéré	difficile	difficile
Complexité	faible	très forte	forte
Testabilité	faible	faible	forte

complexité du circuit C réside dans le choix des cellules permettant de régler le seuil, tandis que le circuit A est peu complexe car ne nécessite pas de cellules particulières. La testabilité est également un paramètre important à prendre en compte. De par leur seuil de détection élevé, les circuits A et B sont plus difficiles à tester. Le circuit C a une testabilité plus grande car le choix des blocs de délai permet de définir à l'avance les seuils qui correspondent à la tension où les deux blocs de délai ont les mêmes temps de propagation.

Les solutions détections d'impulsions négatives ont des seuils de détection compa-

rables. Cependant, l'architecture du chemin critique répliqué permet de s'approcher plus facilement du seuil de violation du circuit, et donc entraine une marge de détection plus faible. Ceci autorise de mieux se rapprocher du comportement du circuit à protéger. Comme pour les détecteurs d'impulsions positives, la largeur minimum d'impulsions détectée est définie par la période de l'horloge du système, car les comparaisons des sorties des bascules sont font à chaque front. Les lignes parallèles de délai sont les plus coûteuses en surface car il y a au moins deux chemins combinatoires là où les autres schémas n'en utilisent qu'un. La taille du chemin critique répliqué est fortement dépendante du circuit car les cellules identiques au chemin de référence sont utilisées. Les délais des chemins combinatoires pour le chemin répliqué réglable et les lignes parallèles de délai peuvent être définis par des contraintes de temps de propagation avant la phase de placement et de routage. En ce sens, les temps de propagation de ces solutions sont plus faciles à mettre en place que ceux du chemin critique répliqué qui nécessite une phase d'automatisation afin de reproduire le chemin critique et d'y ajouter une marge de détection. C'est également pour cette raison que cette solution est plus complexe que les deux autres. Les trois contremesures ont des niveaux de testabilité similaires, la détection des seuils en phase de conception se fait en étudiant la variation des temps de propagation en fonction de la tension. Le chemin répliqué réglable présente l'avantage de pouvoir être reconfiguré une fois que le circuit est fabriqué. Cela peut être utile pour changer les seuils de détection dans les cas où la tension serait perturbée par le bruit d'alimentation qui serait mal maîtrisé. Les trois solutions présentent aussi une limitation au niveau des performances du système protégé. En effet, la marge de détection insérée pour chaque détecteur crée une diminution de la fréquence maximale de fonctionnement du circuit protégé. À titre d'exemple le chemin critique répliqué est le détecteur présentant une marge la plus faible. La limitation en fréquence est mesurée à 500 ps à la tension nominale soit une limitation de 5 %, la période spécifiée étant de 10 ns.

TABLEAU 3.21 : Comparaison des circuits de détection d'attaques par impulsions négatives

Circuit	Cir. répliqué réglable	Lignes parallèles de délai	Ch. critique répliqué
Seuil de détection (V)	0.58	0.57	0.56
Largeur minimum détectée (amplitude supérieure au seuil)	Période de l'horloge du système		
Surface (μm^2)	103	145	121
Réglage du délai	modéré	modéré	difficile
Configuration du délai après fabrication	possible	non	non
Complexité	faible	faible	très forte
Testabilité	forte	forte	forte

En prenant en compte les différents critères de comparaison et les résultats des tests réalisés, le chemin critique répliqué paraît la meilleure solution pour détecter les attaques par impulsions négatives et le circuit C offre les meilleures performances pour détecter les impulsions positives. La complexité de réalisation et les temps de développement sont supérieurs à ceux des autres solutions, cependant une fois que la méthodologie de conception est mise en place, elle peut être réutilisée notamment pour la réplique du chemin critique. L'association de ces deux contremesures permet d'être robuste pour la détection des attaques par impulsions de tension d'alimentation.

3.3 Conclusions

Ce chapitre a permis d'étudier différents aspects concernant les attaques par impulsions de tension d'alimentation sur les circuits numériques synchrones. Tout d'abord, l'objectif était de bien comprendre les phénomènes qui étaient mis en jeu lors de ces attaques. Pour cela, le fonctionnement des circuits numériques a été analysé aussi bien à l'échelle d'une porte logique que pour des chemins logiques complets. Ensuite, la réponse d'un circuit numérique face aux attaques par impulsions a été étudiée à travers l'effet des variations de la tension d'alimentation sur la logique. Cette étape a mis en avant les vulnérabilités des circuits. La principale vulnérabilité identifiée est liée à la violation des contraintes de temps de propagation inhérent au fonctionnement de la logique synchrone. D'autres phénomènes ont également été mis en évidence, notamment des changements de valeurs de tension aux interfaces entre différents domaines d'alimentation. Les changements de propriétés temporelles des cellules logiques intervenant de façon systématique lors d'attaques, une solution efficace consiste à utiliser les temps de propagation pour détecter les attaques modifiant la valeur de tension. Les difficultés résident dans le fait que ces temps de propagation et leur variation dépendent de plusieurs paramètres comme les procédés de fabrication, la température, et la tension. Il a donc fallu trouver un moyen de gérer ces paramètres. La méthode utilisée a été de contraindre les chemins critiques et d'étudier leur variation en fonction de la tension.

Des solutions de détection d'attaques ont été proposées et étudiées à travers leur conception et leur implantation dans un circuit. Le principe de fonctionnement consiste à rajouter un chemin dans le circuit à protéger, et de détecter des violations de contraintes de temps de propagation sur ce chemin. Les avantages de ces solutions sont qu'elles sont peu coûteuses en surface et en consommation et surtout elles ne nécessitent pas de modifier l'architecture du circuit à protéger. Elles peuvent donc s'intégrer facilement quelle que soit la fonctionnalité du circuit.

La méthodologie d'intégration de ces circuits a été développée. Elle consiste à appliquer les contraintes de temps de propagation d'une part pour gérer les marges de détection, et d'autre part pour créer une variation des délais permettant de déclencher la détection avant de provoquer une faute dans le circuit. Les circuits ont ensuite été

évalués sur silicium en technologie CMOS 28 nm. Les résultats ont montré que tous les détecteurs d'impulsions négatives sont capables de protéger le circuit. Toutefois, le détecteur basé sur un chemin critique répliqué s'est montré plus efficace en détectant toutes les attaques provoquant des fautes dans le circuit, et en permettant de prendre moins de marge par rapport à la période de fonctionnement. Pour la détection d'impulsions positives, c'est le circuit basé sur la variation des temps de propagation qui a été en mesure de détecter toutes les attaques. Les expérimentations ont permis de mettre en évidence la nécessité d'équilibrer les temps de montée et de descente pour une meilleure couverture des attaques, notamment lorsque l'amplitude des impulsions dépasse les 600 mV (tension de fonctionnement proche de la tension de seuil des transistors). Il faudrait toutefois vérifier le comportement de la bascule fournissant le signal de référence pour valider que celle-ci change bien de valeur à tous les cycles lorsque la tension d'alimentation avoisine la tension de seuil des transistors.

Lors de cette étude, la variation des temps de propagation sur les cellules logiques du circuit attaqué a été considérée plus ou moins uniforme. Les résultats expérimentaux n'ont pas permis de mettre en évidence une variation locale de la tension d'alimentation lorsqu'une impulsion est appliquée. Une prochaine étape serait de mesurer l'impact de la taille du circuit sur les impulsions appliquées, et sur la création de modifications locales de la tension d'alimentation. Cela nécessite de caractériser la forme d'onde de l'impulsion, telle qu'elle est perçue à l'intérieur du boîtier du circuit.

Conclusion générale et perspectives

Le travail réalisé dans le cadre de cette thèse avait pour objectif d'évaluer la vulnérabilité des circuits numériques face aux attaques par analyse et par injection de fautes autour de la tension d'alimentation, et de concevoir des solutions de protection dans un flot de conception numérique standard.

La première partie s'est intéressée aux menaces existantes sur les circuits, qui obligent les concepteurs à trouver des protections pour les systèmes. De par le grand nombre d'attaques qui existent, l'analyse montre qu'il est difficile de les couvrir dans leur intégralité. Toutefois, il est apparu que les attaques non invasives présentent une menace réelle pour la sécurité des circuits, car leur mise en œuvre et leur exploitation ne présentent pas de difficultés particulières. Les attaques en tension, qu'il s'agisse de l'analyse de la consommation ou de l'injection de fautes par ce biais se sont avérées être des attaques non invasives privilégiées pour compromettre la sécurité d'un circuit. C'est pour cette raison qu'elles ont fait l'objet de notre étude. Il existe des contremesures mais elles sont bien souvent coûteuses en surface ou en consommation, ou nécessitent de sortir du flot de conception numérique. Cette dernière contrainte est particulièrement importante dans le cadre d'un développement de produits, pour une application industrielle.

L'étude s'est ensuite portée sur les attaques par analyse de consommation électrique. Le but était de pouvoir déterminer la signature électrique d'un circuit numérique en phase de conception, telle qu'elle est mesurée après fabrication afin de pouvoir y apporter des contremesures. Pour cela, un modèle électrique équivalent du circuit cible a été développé. La méthodologie complète d'obtention des paramètres qui composent le modèle a été établie, elle comprend l'extraction de la capacité globale du circuit entre l'alimentation et la masse, ainsi que le courant transitoire consommé par le circuit. La construction de ce modèle utilise les outils commerciaux d'aide à la conception, et ne rajoute que peu d'étapes supplémentaires par rapport aux caractérisations fonctionnelles déjà effectuées. Afin d'évaluer cette méthodologie, le modèle d'un circuit numérique en technologie CMOS 40 nm a été réalisé puis ensuite comparé aux mesures effectuées sur ce même circuit. Les valeurs de courant comparées ont permis de valider l'approche de construction et la méthodologie d'extraction des paramètres électriques. Deux contreme-

sures, l'une utilisant le masquage des données, et l'autre une implantation de capacités de découplage intégrées ont été évaluées grâce au modèle développé. Le masquage a fait disparaître la corrélation entre la signature et les données sauf aux points d'entrée et de sortie du masque, tandis que les capacités de découplage ont réduit les pics de courant de plus de 30 % sans ajout de surface. Cette dernière solution est intéressante pour réduire la signature électrique à faible coût et pourrait être appliquée systématiquement pour le développement de produits. La principale contribution de cette partie est le développement du modèle d'évaluation de signature électrique et son utilisation en phase de conception pour prévenir les attaques par analyse de consommation électrique.

Dans la troisième partie de cette thèse, une étude complète des attaques par impulsion de tension d'alimentation sur les circuits numériques a été réalisée. Le point de départ a été de comprendre les différents mécanismes d'injection de fautes dans la logique, en fonction du type d'impulsions engendrées. La logique synchrone a été étudiée à travers les conséquences de la variation de la tension d'alimentation sur son fonctionnement. Les principaux phénomènes liés à l'injection de fautes ont pu être déterminés et correspondent à la violation de contraintes de temps de propagation entre les cycles d'horloge.

Dès lors, une méthodologie de conception de circuits de détection a été proposée. Elle se base sur l'application de contraintes de temps de propagation permettant de contrôler la variation des délais dus au procédé de fabrication, à la température et à la gamme de tensions de fonctionnement spécifiée. Trois circuits de détection d'impulsions positives et négatives ont été conçus en technologie CMOS 28 nm puis évalués à la fois en simulation puis sur circuit. Deux circuits en particulier ont montré des performances satisfaisantes en termes de détection mais présentent une complexité d'implantation supérieure. Cependant, la compatibilité du développement de ces circuits avec le flot de conception numérique les rend utilisables pour fournir une solution peu onéreuse et efficace contre les attaques par impulsions de tension dans des circuits industriels sécurisés. Les contributions de cette partie portent d'une part, sur la méthodologie de conception des détecteurs d'attaques par impulsions dans une gamme de fonctionnement prenant en compte les procédés de fabrication, la température et la tension. D'autre part, les mesures de leur performance et de leur efficacité sur silicium ont été validées.

Pour poursuivre les travaux réalisés sur les attaques passives en tension, deux points pourront être privilégiés :

- l'évaluation de contremesures à base de capacité de découplage à la fois en simulation et après implantation sur silicium afin de comparer les résultats et quantifier l'apport définitif du modèle.
- effectuer des mesures électromagnétiques afin de vérifier si le modèle peut être utilisé également pour cette attaque par observation.

En ce qui concerne les attaques par impulsions de tension, il serait intéressant de

mettre en place une méthodologie de simulation par modification dynamique de la tension dans un environnement de conception numérique. Cela permettrait de voir directement en fonctionnement la conséquence des fautes injectées. Un deuxième axe d'approfondissement serait d'évaluer la forme d'onde des impulsions perçues par le circuit à l'intérieur du boîtier. Cela pourrait se faire d'une part en reconstruisant le réseau d'impédances en simulation et d'autre part en effectuant des mesures à l'intérieur de la puce. Enfin, on pourrait comparer les mécanismes d'injection par attaque électromagnétique et voir dans quelles mesures les contremesures développées ici peuvent être réutilisées contre ce type d'attaque.

Glossaire

Glossaire

AES Acronyme de *Advanced Encryption Standard*. 6, 11, 12, 66, 97, 120, 121

BCDL Acronyme de *Balanced Cell-based Dual-rail Logic*. 66

BGA Acronyme de *Bold Grid Array*, matrice de billes. 15, 55

BIST Acronyme de *Built-In Self Test*, test intégré in-situ. 29

CC Acronyme de Critères Communs. 20, 21

CESTI Acronyme de Centres d'Évaluation de la Sécurité des Technologies de l'Information. 1

CMOS Acronyme de *Complementary Metal Oxide Semiconducteur*. 7, 11, 15, 36, 38, 39, 47, 48, 57, 69, 72, 82–85

CPM Acronyme de *Chip Power Model*, modèle de puissance d'une puce. 52, 53

CPU Acronyme de *Computer Processing Unit*, unité centrale de traitement. 9

DES Acronyme de *Data Encryption Standard*. 8

DFA Acronyme de *Differential Fault Analysis*. 13

DRC Acronyme de *Design Rule Checking*, Vérification des règles de dessin des masques. 30

DRM Acronyme de *Design Rule Manual*, manuel de règles de dessin des masques. 30

DSTB Acronyme de *Double Sampling with Time Borrowing*. 102

EAL Acronyme de *Evaluation Assurance Level*, niveau d'évaluation d'assurance. 21

EDS Acronyme de *Error Detection Sequential*. 102, 107

EPROM Acronyme de *Erasable Programmable Read Only Memory*. 13

E²PROM Acronyme de *Electrically Erasable Programmable Read-Only Memory*, mémoire non-volatile programmable électriquement. 10, 24

fan-out Le fan-out d'une porte ou d'un bloc logique désigne le nombre de composants connectés à la sortie de cette porte ou de cet élément. 114, 117, 119, 137

FPGA Acronyme de *Field-Programmable Gate Arrays*, réseau de portes logiques programmables in situ. 10–12, 24, 25, 103, 125

IO Acronyme de *Input Output*, circuit d'interface entre la puce et l'extérieur du circuit. 56, 61, 63

IP Acronyme de *Intellectual Property* (propriété intellectuelle). 33, 45, 56, 101

LVS Acronyme de *Layout Versus Schematic*, Comparaison du schéma électrique et du dessin des masques. 30

MIM Acronyme de *Metal-Insulator-Metal*. 68, 72, 74

MOS Acronyme de *Metal Oxide Semiconductor*. 46, 55, 72

netlist description d'un circuit en termes de composants élémentaires et des connexions entre ceux-ci. 53–55, 74, 75

poids de Hamming Le poids de Hamming est le nombre de 1 contenus dans un mot binaire. 58, 59, 70, 71, 75

PP Acronyme de *Protection Profile*, profil de protection. 20

RAM Acronyme de *Random Access Memory*. 24

RFID Acronyme de *Radio Frequency IDentification*, radio-identification. 11

RSA Acronyme de *Rivest Shamir Adleman*, noms des inventeurs de l'algorithme. 8, 9, 12

RSL Acronyme de *Random Switching Logic*. 64

RTL Acronyme de *Register Transfer Level*. 28, 29, 31, 69, 117

SAIF Acronyme de *Switching Activity Interchange Format*. 41

SoC Acronyme de *System On Chip* (système sur puce). 1, 19, 33, 45, 54, 63, 75, 92, 101

SPA Acronyme de *Simple Power Analysis* (Analyse simple de puissance). 70

SPICE Acronyme de *Simulation Program with Integrated Circuit Emphasis*. 45, 52, 55, 60, 75

SRAM Acronyme de *Static Random Access Memory*, mémoire vive statique. 13, 14

ST Acronyme de *Security Target*, cible de sécurité. 20, 21

STA Acronyme de *Static Timing Analysis*, Analyse statique de temps de propagation. 29, 103

STTL Acronyme de *Secure Triple Track Logic*. 66

TDC Acronyme de *Time to Digital Converter*. 110–112

TDTB Acronyme de *Transition Detector with Time Borrowing*. 102

TOE Acronyme de *Target Of Evaluation*, cible d'évaluation. 20–22

TRC Acronyme de *Tunable Replica Circuit*. 103, 113

UV Acronyme de UltraViolet. 13

VCD Acronyme de *Value Change Dump*. 41, 43

VHDL Acronyme de *Very High speed integrated circuit Hardware Description Language*. 28

WDDL Acronyme de *Wave Dynamic Differential Logic*. 66

ZCE Acronyme de Zone de Charge Espace. 51

Bibliographie

[Abdollahi 2004] A. Abdollahi, F. Fallah and M. Pedram. *Leakage current reduction in CMOS VLSI circuits by input vector control.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 2, pages 140–154, February 2004. Cité page 37.

[Abraham 1991] D. G. Abraham, G. M. Dolan, G. P. Double and J. V. Stevens. *Transaction Security System.* IBM Syst. J., vol. 30, no. 2, pages 206–229, March 1991. Cité page 4.

[Adler 2011] Joseph Adler. R, l'essentiel. Pearson, Paris, 2011. Cité page 70.

[Agarwal 2007] M. Agarwal, B.C. Paul, Ming Zhang and S Mitra. *Circuit Failure Prediction and Its Application to Transistor Aging.* In 25th IEEE VLSI Test Symposium, 2007, pages 277–286, 2007. Cité pages 104 et 105.

[Agoyan 2010] Michel Agoyan, Jean-Max Dutertre, David Naccache, Bruno Robisson and Assia Tria. *When clocks fail : On critical paths and clock faults.* In Smart Card Research and Advanced Application, pages 182–193. Springer, 2010. Cité page 10.

[Agrawal 2003] Dakshi Agrawal, Bruce Archambeault, Josyula R Rao and Pankaj Rohatgi. *The EM side—channel (s).* In Cryptographic Hardware and Embedded Systems-CHES 2002, pages 29–45. Springer, 2003. Cité page 9.

[Akkar 2001] Mehdi-Laurent Akkar and Christophe Giraud. *An implementation of DES and AES, secure against some attacks.* In Cryptographic Hardware and Embedded Systems—CHES 2001, page 309–318, 2001. Cité page 64.

[Ali 2011a] Sk. Subidh Ali, Rajat Subhra Chakraborty, Debdeep Mukhopadhyay and Swarup Bhunia. *Multi-level attacks : An emerging security concern for cryptographic hardware.* In Design, Automation and Test in Europe Conference Exhibition (DATE), 2011. Cité page 10.

[Ali 2011b] S.S. Ali and D. Mukhopadhyay. *A Differential Fault Analysis on AES Key Schedule Using Single Fault.* In 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pages 35–42, 2011. Cité page 78.

[Amiel 2006] Frederic Amiel, Christophe Clavier and Michael Tunstall. *Fault analysis of DPA-resistant algorithms*. In Fault Diagnosis and Tolerance in Cryptography, page 223–236. Springer, 2006. Cité page 10.

[Apa 2013] Apache DA. *Totem User Manual*, 2013. Cité pages v et 52.

[Atmel 2014] Atmel. *Atmel product Website*. http://www.atmel.com/products/microcontrollers/arm/default.aspx, 2014. Cité page 24.

[Bae 2011] KiSeok Bae, SangJae Moon, Dooho Choi, YongJe Choi, Doo-sik Choi and JaeCheol Ha. *Differential fault analysis on AES by round reduction*. In 2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), pages 607–612, 2011. Cité page 78.

[Bar-El 2006] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall and Claire Whelan. *The sorcerer's apprentice guide to fault attacks*. Proceedings of the IEEE, vol. 94, no. 2, page 370–382, 2006. Cité pages 11 et 16.

[Barenghi 2012] A. Barenghi, L. Breveglieri, I. Koren and D. Naccache. *Fault Injection Attacks on Cryptographic Devices : Theory, Practice, and Countermeasures*. Proceedings of the IEEE, vol. 100, no. 11, pages 3056–3076, 2012. Cité page 11.

[Beringuier-Boher 2014] N. Beringuier-Boher, K. Gomina, D. Hely, V. Beroulle, J-B. Rigaud, A. Tria, J. Damiens, P. Gendrier and P. Candelier. *Voltage Glitch Attacks on Mixed-Signal Systems*. In 17th Euromicro Conference on Digital Systems Design, à paraître, 2014. Cité page 112.

[Bowman 2009] K.A. Bowman, J.W. Tschanz, Nam Sung Kim, J.C. Lee, C.B. Wilkerson, S.L. Lu, T. Karnik and V.K. De. *Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance*. IEEE Journal of Solid-State Circuits, vol. 44, no. 1, pages 49–63, 2009. Cité pages vii, 102, 103 et 105.

[Bowman 2011] K.A. Bowman, J.W. Tschanz, S.L. Lu, P.A. Aseron, M.M. Khellah, A. Raychowdhury, B.M. Geuskens, C. Tokunaga, C.B. Wilkerson, T. Karnik and V.K. De. *A 45 nm Resilient Microprocessor Core for Dynamic Variation Tolerance*. IEEE Journal of Solid-State Circuits, vol. 46, no. 1, pages 194–208, 2011. Cité pages 102 et 103.

[Chandrakasan 1995] Anantha P. Chandrakasan and Robert W. Brodersen. *Minimizing power consumption in digital CMOS circuits*. Proceedings of the IEEE, vol. 83, no. 4, page 498–523, 1995. Cité page 38.

[Chen 2008] Po-Yuan Chen, Che-Yu Liu and TingTing Hwang. *Transition-aware decoupling-capacitor allocation in power noise reduction*. In IEEE/ACM International Conference on Computer-Aided Design, 2008. ICCAD 2008, November 2008. Cité page 67.

[Choukri 2005] Hamid Choukri and Michael Tunstall. *Round reduction using faults.* FDTC, vol. 5, page 13–24, 2005. Cité page 11.

[Clark 2003] Sean Clark. IC package design's effects on signal integrity. 2003. Cité page 55.

[Criteria 2012a] Common Criteria. *Part 2 : Security functional components*, september 2012. Cité page 20.

[Criteria 2012b] Common Criteria. *Part 3 : Security assurance requirements*, september 2012. Cité page 21.

[Criteria 2013] Common Criteria. *Application of Attack Potential to Smartcards*, may 2013. Cité pages ix, 21, 22 et 23.

[Danger 2014] Jean-Luc Danger, Nicolas Debande, Sylvain Guilley and Youssef Souissi. *High-order Timing Attacks.* In Proceedings of the First Workshop on Cryptography and Security in Computing Systems, CS2 '14, pages 7–12, New York, NY, USA, 2014. ACM. Cité page 7.

[Das 2005] Shidhartha Das, Sanjay Pant, David Roberts, Seokwoo Lee, David Blaauw, Todd Austin, Trevor Mudge and Krisztian Flautner. *A self-tuning DVS processor using delay-error detection and correction.* In VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on, pages 258–261, 2005. Cité page 102.

[Das 2009] S. Das, C. Tokunaga, S. Pant, Wei-Hsiang Ma, S. Kalaiselvan, K. Lai, D.M. Bull and D.T. Blaauw. *RazorII : In Situ Error Detection and Correction for PVT and SER Tolerance.* IEEE Journal of Solid-State Circuits, vol. 44, no. 1, pages 32–48, 2009. Cité pages vi et 102.

[Das 2010] B.P. Das and Hidetoshi Onodera. *Warning Prediction Sequential for Transient Error Prevention.* In 2010 IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), pages 382–390, 2010. Cité page 104.

[Dehbaoui 2012a] A. Dehbaoui, J. M. Dutertre, B. Robisson, P. Orsatelli, P. Maurine and A. Tria. *Injection of transient faults using electromagnetic pulses Practical results on a cryptographic system.* Rapport technique, Cryptology ePrint Archive, Report 2012/123, 2012. Cité page 12.

[Dehbaoui 2012b] Amine Dehbaoui, Jean-Max Dutertre, Bruno Robisson and Assia Tria. *Electromagnetic Transient Faults Injection on a Hardware and a Software Implementations of AES.* In 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pages 7 –15, September 2012. Cité page 12.

[Dhem 2000] Jean-Francois Dhem, Francois Koeune, Philippe-Alexandre Leroux, Patrick Mestré, Jean-Jacques Quisquater and Jean-Louis Willems. *A practical*

implementation of the timing attack. In Smart Card Research and Applications, pages 167–182. Springer, 2000. Cité pages v, 6 et 7.

[Djellid-Ouar 2006] A. Djellid-Ouar, G. Cathebras and F. Bancel. *Supply voltage glitches effects on CMOS circuits*. In International Conference on Design and Test of Integrated Systems in Nanoscale Technology, 2006. DTIS 2006, pages 257 –261, September 2006. Cité page 89.

[Dutertre 2010] Jean-Max Dutertre, Amir Pasha Mirbaha, Assia Tria, Bruno Robisson and Michel Agoyan. *Revue expérimentale des techniques d'injection de fautes*. In Colloque nationale Groupement De Recherche SOC-SIP, PARIS, France, March 2010. Cité pages 10 et 11.

[Endo 2012] Sho Endo, Yang Li, Naofumi Homma, Kazuo Sakiyama, Kazuo Ohta and Takafumi Aoki. *An efficient countermeasure against fault sensitivity analysis using configurable delay blocks*. In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2012 Workshop on, page 95–102, 2012. Cité pages 103 et 105.

[ENISA 2005] ENISA. *ISO/IEC Standard 15408 — ENISA*. https://www.enisa.europa.eu/activities/risk-management/current-risk/laws-regulation/rm-ra-standards/iso-iec-standard-15408, 2005. Cité page 20.

[Fournier 2003] Jacques JA Fournier, Simon Moore, Huiyun Li, Robert Mullins and George Taylor. *Security evaluation of asynchronous circuits*. In Cryptographic Hardware and Embedded Systems-CHES 2003, page 137–151. Springer, 2003. Cité page 11.

[Genkin 2013] Daniel Genkin, Adi Shamir and Eran Tromer. *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*. Rapport technique, Cryptology ePrint Archive, Report 2013/857, 2013. http ://eprint. iacr. org, 2013. Cité page 9.

[Goh 2004] M.W.C. Goh, Q. Lim, R.A. Keating, A.V. Kordesch and Y. Bin Mohd Yusof. *Design of radio frequency metal-insulator-metal (MIM) capacitors*. In 7th International Conference on Solid-State and Integrated Circuits Technology, 2004. Proceedings, volume 1, pages 209–212 vol.1, October 2004. Cité page 68.

[Golic 2007] J.D. Golic. *Techniques for Random Masking in Hardware*. IEEE Transactions on Circuits and Systems I : Regular Papers, vol. 54, no. 2, pages 291 –300, February 2007. Cité page 64.

[Gomina 2014a] Kamil Gomina, Philippe Gendrier, Philippe Candelier, Jean-Baptiste Rigaud and Assia Tria. *Detecting Positive Voltage Attacks on CMOS Circuits*. In Proceedings of the First Workshop on Cryptography and Security in Computing Systems, CS2 '14, page 1–6, New York, NY, USA, 2014. ACM. Cité page 112.

[Gomina 2014b] Kamil Gomina, Jean-Baptiste Rigaud, Philippe Candelier, and Assia Tria. *Power supply glitch attacks : design and evaluation of detection circuits.* In IEEE Int. Symposium on Hardware-Oriented Security and Trust, à paraître, 2014. Cité page 112.

[Gurkaynak 2005] F. Gurkaynak, S. Oetiker, H. Kaeslin, N. Felber and W. Fichtner. *Improving DPA security by using globally-asynchronous locally-synchronous systems.* In Solid-State Circuits Conference, 2005. ESSCIRC 2005. Proceedings of the 31st European, page 407–410, 2005. Cité page 66.

[Haensch 2006] Wilfried Haensch, Edward J. Nowak, Robert H. Dennard, Paul M. Solomon, Andres Bryant, Omer H. Dokumaci, Arvind Kumar, Xinlin Wang, Jeffrey B. Johnson and Massimo V. Fischetti. *Silicon CMOS devices beyond scaling.* IBM Journal of Research and Development, vol. 50, no. 4.5, page 339–361, 2006. Cité pages v et 39.

[Hely 2006] D. Hely, F. Bancel, M.-L. Flottes and B. Rouzeyre. *Secure scan techniques : a comparison.* In On-Line Testing Symposium, 2006. IOLTS 2006. 12th IEEE International, pages 6 pp.–, 2006. Cité page 29.

[Henzler 2010] Stephan Henzler. Time-to-digital converters, volume 29. Springer, 2010. Cité pages vii, 110 et 111.

[Horstmann 1989] J.U. Horstmann, H.W. Eichel and R.L. Coates. *Metastability behavior of CMOS ASIC flip-flops in theory and test.* IEEE Journal of Solid-State Circuits, vol. 24, no. 1, pages 146–157, 1989. Cité page 108.

[Hutter 2009] M. Hutter, J. Schmidt and T. Plos. *Contact-based fault injections and power analysis on RFID tags.* In European Conference on Circuit Theory and Design, 2009. ECCTD 2009, pages 409–412, 2009. Cité page 11.

[Idate 2013] Post Published : 18 September 2013 Author : M2M World News Idate. *IDATE forecasts 80 Billion things connected in 2020*, 2013. Cité page 1.

[IEEE 1996] IEEE. *IEEE Standard Hardware Description Language Based on the Verilog(R) Hardware Description Language.* IEEE Std 1364-1995, 1996. Cité page 41.

[Joaquim Da Rolt 2012] Jean Joaquim Da Rolt. *Testabilité versus Sécurité : Nouvelles attaques par chaîne de scan et contremesures.* PhD thesis, Université Montpellier II, 2012. Cité page 29.

[Kim 2007] Chong Hee Kim and J.-J. Quisquater. *Faults, injection methods, and fault attacks.* Design & Test of Computers, IEEE, vol. 24, no. 6, page 544–545, 2007. Cité page 6.

[Kömmerling 1999] Oliver Kömmerling and Markus G. Kuhn. *Design principles for tamper-resistant smartcard processors.* In Proceedings of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology, page 2–2. USENIX Association, 1999. Cité pages v, 4 et 5.

[Kocher 1996] Paul C Kocher. *Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems.* In Advances in Cryptology—CRYPTO'96, pages 104–113. Springer, 1996. Cité page 6.

[Kocher 1999] Paul Kocher, Joshua Jaffe and Benjamin Jun. *Differential power analysis.* In Advances in Cryptology—CRYPTO'99, page 388–397, 1999. Cité pages 7 et 8.

[Kunitake 2011] Y. Kunitake, T. Sato, H. Yasuura and T. Hayashida. *Possibilities to miss predicting timing errors in canary flip-flops.* In 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), pages 1–4, 2011. Cité page 102.

[Li 2012] Yang Li, K. Ohta and K. Sakiyama. *New Fault-Based Side-Channel Attack Using Fault Sensitivity.* IEEE Transactions on Information Forensics and Security, vol. 7, no. 1, pages 88–97, February 2012. Cité page 78.

[Lin 2001] Shen Lin and Norman Chang. *Challenges in power-ground integrity.* In Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on, page 651–654, 2001. Cité page 39.

[Loughry 2002] Joe Loughry and David A Umphress. *Information leakage from optical emanations.* ACM Transactions on Information and System Security (TISSEC), vol. 5, no. 3, pages 262–289, 2002. Cité page 9.

[Lu 2010] Y. Lu, K. Boey, P. Hodgers and M. O'Neill. *SEED masking implementations against power analysis attacks.* In Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on, pages 1199 –1202, December 2010. Cité page 64.

[Maghrebi 2009] H. Maghrebi, J.-L. Danger, F. Flament, S. Guilley and L. Sauvage. *Evaluation of countermeasure implementations based on Boolean masking to thwart side-channel attacks.* In Signals, Circuits and Systems (SCS), 2009 3rd International Conference on, pages 1 –6, November 2009. Cité page 64.

[Maingot 2006] V. Maingot and R. Leveugle. *Error Detection Code Efficiency for Secure Chips.* In 13th IEEE International Conference on Electronics, Circuits and Systems, 2006. ICECS '06, pages 561–564, December 2006. Cité page 18.

[Maingot 2009] Vincent Maingot. *Conception sécurisée contre les attaques par fautes et par canaux cachés.* PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2009. Cité pages v, 15, 16 et 18.

[Mangard 2010] Stefan Mangard, Elisabeth Oswald and Thomas Popp. Power analysis attacks : Revealing the secrets of smart cards. Springer Publishing Company, Incorporated, 1st édition, 2010. Cité pages 7, 8, 9 et 19.

[Mathieu 2009] Henry Mathieu and Hervé Fanet. Physique des semiconducteurs et des composants électroniques-6ème édition : Cours et exercices corrigés. Dunod, 2009. Cité page 51.

[Meng 2006] Xiongfei Meng. *Decoupling capacitor design issues in 90nm CMOS*. PhD thesis, The University of British Columbia, 2006. Cité page 67.

[Mesquita 2005] Daniel Mesquita, J.-D. Techer, Lionel Torres, Gilles Sassatelli, Gaston Cambon, Michel Robert and Fernando Moraes. *Current mask generation : a transistor level security against DPA attacks*. In Integrated Circuits and Systems Design, 18th Symposium on, page 115–120, 2005. Cité pages 65 et 67.

[Microchip 2014] Microchip. *PIC microcontroller Website*. http://www.microchip.com/pagehandler/en-us/products/picmicrocontrollers, 2014. Cité page 24.

[Mirbaha 2011] Amir-Pasha Mirbaha. *Etude de la vulnérabilité des circuits cryptographiques l'injection de fautes par laser*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2011. Cité page 13.

[Mirbaha 2013] A.-P. Mirbaha, J.-M. Dutertre and A. Tria. *Differential analysis of Round-Reduced AES faulty ciphertexts*. In 2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), pages 204–211, October 2013. Cité page 78.

[Monnet 2006] Y. Monnet, M. Renaudin and R. Leveugle. *Designing Resistant Circuits against Malicious Faults Injection Using Asynchronous Logic*. IEEE Transactions on Computers, vol. 55, no. 9, pages 1104–1115, September 2006. Cité page 11.

[Muresan 2008] Radu Muresan and S. Gregori. *Protection Circuit against Differential Power Analysis Attacks for Smart Cards*. IEEE Transactions on Computers, vol. 57, no. 11, pages 1540–1549, November 2008. Cité page 67.

[Nassar 2010] M. Nassar, S. Bhasin, J. L Danger, G. Duc and S. Guilley. *BCDL : a high speed balanced DPL for FPGA with global precharge and no early evaluation*. In Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010, page 849–854, 2010. Cité page 66.

[Ope 2013] Open Source Liberty. *Liberty User Guide and Reference Manual Suite*, 2013. Cité pages v et 42.

[ope 2014] *http ://opensourceliberty.org/opensourceliberty.html*, 2014. Cité page 42.

[Pelkins 2007] Gérard Pelkins. *Certification - critères communs.* Rapport technique, © Forum ATENA, 2007. Cité page 20.

[Pub 2001] NIST FIPS Pub. *197 : Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/-NIST. Available from the NIST website*, 2001. Cité page 97.

[Quisquater 2000] Jean-Jacques Quisquater and David Samyde. *A new tool for non-intrusive analysis of smart cards based on electro-magnetic emissions : the SEMA and DEMA methods.* In Eurocrypt rump session, 2000. Cité page 9.

[Quisquater 2002] J. J Quisquater and D. Samyde. *Eddy current for magnetic analysis with active sensor.* 2002. Cité page 12.

[Ratanpal 2004] G.B. Ratanpal, R.D. williams and T.N. Blalock. *An on-chip signal suppression countermeasure to power analysis attacks.* IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 3, pages 179–189, July 2004. Cité page 67.

[Raychowdhury 2011] A. Raychowdhury, J. Tschanz, K. Bowman, Shih-Lien Lu, P. Aseron, M. Khellah, B. Geuskens, C. Tokunaga, C. Wilkerson, T. Karnik and V. De. *Error Detection and Correction in Microprocessor Core and Memory Due to Fast Dynamic Voltage Droops.* IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 1, no. 3, pages 208–217, 2011. Cité page 103.

[Rebaud 2009] B. Rebaud, M. Belleville, E. Beigne, M. Robert, P. Maurine and N. Azemard. *On-chip timing slack monitoring.* In 2009 17th IFIP International Conference on Very Large Scale Integration (VLSI-SoC), pages 89–94, 2009. Cité pages vii, 104 et 105.

[Roscian 2012] Cyril Roscian, Florian Praden, Jean-Max Dutertre, Jacques Fournier and Assia Tria. *Security characterisation of a hardened AES cryptosystem using a laser.* Technical Sciences/University of Warmia and Mazury in Olsztyn, pages 139–154, 2012. Cité pages v, 13 et 14.

[Roscian 2013] Cyril Roscian. *Cryptanalyse physique de circuits cryptographiques à l'aide de sources LASER.* These, Ecole Nationale Supérieure des Mines de Saint-Etienne, October 2013. Cité page 14.

[Roy 2003] K. Roy, S. Mukhopadhyay and H. Mahmoodi-Meimand. *Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits.* Proceedings of the IEEE, vol. 91, no. 2, pages 305–327, February 2003. Cité page 37.

[Sarafianos 2013] Alexandre Sarafianos. *Injection de fautes par impulsion laser dans des circuits sécurisés.* PhD thesis, Ecole Nationale Supérieure des Mines de

Saint-Etienne, 2013. Thèse de doctorat dirigée par Tria, Assia Microélectronique Saint-Etienne, EMSE 2013. Cité page 14.

[Saxena 2003] Jayashree Saxena, Subhendu Kundu, N. V. Arvind, Kenneth M. Butler, Vinay B. Jayaram, Pravin Sreeprakash and Manfred Hachinger. *A case study of IR-drop in structured at-speed testing.* In 2003 IEEE International Test Conference (ITC), page 1098–1098, 2003. Cité page 39.

[Schmidt 2007] Jörn-Marc Schmidt and Michael Hutter. *Optical and EM fault-attacks on crt-based rsa : Concrete results.* In Proceedings of the Austrochip, page 61–67, 2007. Cité page 12.

[Schmidt 2009] J Schmidt, Michael Hutter and Thomas Plos. *Optical fault attacks on AES : A threat in violet.* In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009 Workshop on, pages 13–22. IEEE, 2009. Cité page 13.

[Sedra 2010] Adel S Sedra and Kenneth Carless Smith. Microelectronic circuits. Oxford University Press, New York, 2010. Cité pages 40 et 82.

[Selmane 2008] N. Selmane, S. Guilley and J. L Danger. *Practical Setup Time Violation Attacks on AES.* In Dependable Computing Conference, EDCC Seventh European, May 2008. Cité page 11.

[Selmane 2010] Nidhal Selmane. *Attaques en fautes globales et locales sur les cryptoprocesseurs AES : mise en oeuvre et contremesures.* PhD thesis, Télécom ParisTech, 2010. Cité pages v et 8.

[Selmane 2011] N. Selmane, S. Bhasin, S. Guilley and J.-L. Danger. *Security evaluation of application-specific integrated circuits and field programmable gate arrays against setup time violation attacks.* IET Information Security, vol. 5, no. 4, pages 181–190, 2011. Cité pages 103 et 105.

[Shah 2009] Omar Shah and Shekhar Kapoor. *Extraction Techniques for High-performance, High-capacity Simulation.* Rapport technique, Synopsys Inc., September 2009. Cité pages vi, 85 et 86.

[Shamir 2000] Adi Shamir. *Protecting smart cards from passive power analysis with detached power supplies.* In Cryptographic Hardware and Embedded Systems—CHES 2000, page 71–77, 2000. Cité page 67.

[Shimazaki 2009] K. Shimazaki and T. Okumura. *A minimum decap allocation technique based on simultaneous switching for nanoscale SoC.* In IEEE Custom Integrated Circuits Conference, 2009. CICC '09, pages 21–24, September 2009. Cité page 67.

[Skorobogatov 2003] Sergei P Skorobogatov and Ross J Anderson. *Optical fault induction attacks*. In Cryptographic Hardware and Embedded Systems-CHES 2002, pages 2–12. Springer, 2003. Cité page 13.

[Skorobogatov 2005] Sergei P. Skorobogatov. *Semi-invasive attacks-a new approach to hardware security analysis*. Technical report, University of Cambridge, Computer Laboratory, 2005. Cité pages v, 5, 14 et 15.

[Skorobogatov 2009] Sergei Skorobogatov. *Local heating attacks on Flash memory devices*. In Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on, page 1–6, 2009. Cité page 10.

[Soares 2008] R. Soares, N. Calazans, V. Lomné, P. Maurine, L. Torres and M. Robert. *Evaluating the robustness of secure triple track logic through prototyping*. In Proceedings of the 21st annual symposium on Integrated circuits and system design, page 193–198, 2008. Cité page 66.

[Standaert 2005] F. X Standaert, E. Peeters and J. J Quisquater. *On the masking countermeasure and higher-order power analysis attacks*. In Information Technology : Coding and Computing, 2005. ITCC 2005. International Conference on, volume 1, page 562–567, 2005. Cité page 64.

[STMicroelectronics 2014] STMicroelectronics. *STM32 Website*. http://www.st.com/web/en/catalog/mmc/FM141/SC1169, 2014. Cité page 24.

[Suzuki 2004] D. Suzuki, M. Saeki and T. Ichikawa. *Random switching logic : a countermeasure against DPA based on transition probability*. IACR ePrint, rep, vol. 346, page 2004, 2004. Cité page 64.

[Tiri 2004] K. Tiri and I. Verbauwhede. *A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation*. In Proceedings of the conference on Design, automation and test in Europe-Volume 1, page 10246, 2004. Cité page 66.

[Wang 2011] Jiangwei Wang and Muyan Ma. *Crosstalk analysis in Signal Integrity*. In 2011 International Conference on Multimedia Technology (ICMT), pages 261–264, July 2011. Cité page 95.

[Weste 2010] Neil Weste and David Harris. CMOS VLSI design : a circuits and systems perspective. Addison-Wesley Publishing Company, 2010. Cité page 25.

[White 2013] Michael White. *It's Time To Change To The OASIS Data Format*. http://electronicdesign.com/eda/it-s-time-change-oasis-data-format, 2013. Cité page 31.

[Xil 2009] Xilinx Inc. *ISim User Guide*, 2009. Cité page 41.

[Xilinx 2009] Xilinx. *Virtex-5 Family Overview*. http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf, 2009. Cité page 125.

[Xiu 2007] Liming Xiu. VLSI circuit design methodology demystified : A conceptual taxonomy. John Wiley & Sons, 2007. Cité pages 26 et 30.

[Yanci 2008] A.G. Yanci, S. Pickles and T. Arslan. *Detecting Voltage Glitch Attacks on Secure Devices*. In ECSIS Symposium on Bio-inspired Learning and Intelligent Systems for Security, 2008. BLISS '08, pages 75–80, 2008. Cité pages v, 12 et 95.

[Zhou 2005] YongBin Zhou and DengGuo Feng. *Side-Channel Attacks : Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing*. IACR Cryptology ePrint Archive, vol. 2005, page 388, 2005. Cité page 6.

[Zussa 2012] Loïc Zussa, Jean-Max Dutertre, Jessy Clédière, Bruno Robisson and Assia Tria. *Investigation of timing constraints violation as a fault injection means*. In 27th Conference on Design of Circuits and Integrated Systems (DCIS), Avignon, France, 2012. Cité page 11.

Liste des publications

Conférences internationales avec actes

K. Gomina, J-B. Rigaud, P. Gendrier, P. Candelier, A. Tria, *Power analysis methodology for secure circuits*, IEEE 16^{th} International Symposium on Design and Diagnostics of Electronic Circuits Systems, pp. 102-107, 2013

K. Gomina, J-B. Rigaud, P. Gendrier, P. Candelier, A. Tria, *Detecting Positive Voltage Attacks on CMOS Circuits*, Proceedings of the First Workshop on Cryptography and Security in Computing Systems, pp. 1-6, 2014

K. Gomina, J-B. Rigaud, P. Gendrier, P. Candelier, A. Tria, *Power supply glitch attacks : design and evaluation of detection circuits*, IEEE Int. Symposium on Hardware-Oriented Security and Trust, pp. 136-141, 2014

N. Beringuier-Boher, **K. Gomina**, D. Hely, V. Beroulle, J-B. Rigaud, A. Tria, J. Damiens, P. Gendrier, P. Candelier, *Voltage Glitch Attacks on Mixed-Signal Systems*, 17^{th} Euromicro Conference on Digital Systems Design, pp. 379-386, 2014

Colloques sans actes

K. Gomina, J-B. Rigaud, P. Gendrier, P. Candelier, A. Tria, *A methodology to investigate power analysis attack : application to secure a digital memory controller in advanced CMOS technologies*, Journée de la recherche du centre Microélectronique de Provence, Session Poster, 2012

K. Gomina, J-B. Rigaud, P. Gendrier, P. Candelier, A. Tria, *Detection of voltage attacks in digital circuits*, Workshop on Practical Hardware Innovations in Security Implementation and Characterization, Session Poster, 2013.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2014 EMSE 0751

Kamil GOMINA

METHODOLOGIE ET CONCEPTION DE SOLUTIONS POUR LA SECURISATION DES CIRCUITS NUMERIQUES FACE AUX ATTAQUES EN TENSION

Spécialité: Microélectronique

Mots clefs : Circuits numériques, Méthodologie de conception, Signature électrique, Contremesures, Attaques passives, Attaques par impulsion de tension, Analyse de consommation, Technologies avancées, Injection de fautes, Conception sécurisée

Résumé :

Les applications grand public comme la téléphonie mobile ou les cartes bancaires manipulent des données confidentielles. A ce titre, les circuits qui les composent font de plus en plus l'objet d'attaques qui présentent des menaces pour la sécurité des données. Les concepteurs de systèmes sur puce (SoC) doivent donc proposer des solutions sécurisées, tout en limitant le coût et la complexité globale des applications. L'analyse des attaques existantes sur les circuits numériques nous a orienté vers celles se basant sur la tension d'alimentation, dans des nœuds technologiques avancés.

Dans un premier temps, nous avons déterminé la signature électrique d'un circuit en phase de conception. Pour cela, un modèle électrique a été proposé, prenant en compte la consommation en courant et la capacité de la grille d'alimentation. L'extraction de ces paramètres ainsi que l'évaluation du modèle sont présentées. L'utilisation de ce modèle a permis de mesurer la vulnérabilité d'un circuit mais aussi d'évaluer quantitativement des contremesures, notamment celle utilisant des capacités de découplage.

Ensuite, l'étude se consacre à l'injection de fautes par impulsions de tension d'alimentation. Les mécanismes d'injection de fautes sur des circuits numériques ont été étudiés. Dès lors, des solutions de détection d'attaques ont été proposées et évaluées à la fois en simulation et par des tests électriques sur circuit. Les résultats ont permis de confirmer les analyses théoriques et la méthodologie utilisée.

Ce travail a ainsi montré la faisabilité de solutions à bas coût contre les attaques actives et passives en tension, utilisables dans le cadre d'un développement industriel de produits.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2014 EMSE 0751

Kamil GOMINA

METHODOLOGY AND DESIGN OF SOLUTIONS TO SECURE DIGITAL CIRCUITS AGAINST POWER ATTACKS

Speciality: Microelectronics

Keywords: Digital circuits, Design methodology, Power signature, Countermeasures, Passive attacks, Power glitch attacks, Power analysis, Advanced technologies, Fault injection, Secure design

Abstract:

General use products as mobile phones or smartcards manipulate confidential data. As such, the circuits composing them are more and more prone to physical attacks, which involve a threat for their security. As a result, SoC designers have to develop efficient countermeasures without increasing overall cost and complexity of the final application. The analysis of existing attacks on digital circuits leads to consider power attacks, in advanced technology nodes.

First of all, the power signature of a circuit was determined at design time. To do so, an electrical model was suggested based on the current consumption and the overall power grid capacitance. The methodology to extract these parameters, as well as the evaluation of the model are presented. This model allows designers to anticipate information leakage at design time and to quantify the protection of countermeasures, as the use of integrated decoupling capacitors.

Then, the study was dedicated to power glitch attacks. The different fault injection mechanisms were analyzed in details. From then on, a set of detection circuits were suggested and evaluated at design time and on silicon by electrical tests. Both the theoretical analysis and the given methodology were confirmed by the test campaigns.

This work demonstrated that the design of low-cost solutions against passive and active power attacks can be achieved, and used in a large scale product development.