HAL
open science

Analyse et amélioration de la logique double rail pour la conception de circuits sécurisés

Alin Razafindraibe

To cite this version:

Alin Razafindraibe. Analyse et amélioration de la logique double rail pour la conception de circuits sécurisés. Micro et nanotechnologies/Microélectronique. Université Montpellier II - Sciences et Techniques du Languedoc, 2006. Français. NNT : . tel-00282762

HAL Id: tel-00282762

https://theses.hal.science/tel-00282762v1

Submitted on 28 May 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE MONTPELLIER II
SCIENCES ET TECHNIQUES DU LANGUEDOC

T H E S E

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE MONTPELLIER II

Discipline : Electronique, Optronique et Systèmes
Formation Doctorale : Systèmes Automatiques et Microélectroniques
Ecole Doctorale : Information, Structure et Systèmes

présentée et soutenue publiquement

par

Alin Razafindraibe

Le 27 Novembre 2006

Titre :

Analyse et amélioration de la logique double rail pour la conception de circuits sécurisés

JURY

- M. Gaston Cambon	, Président
- M. Michel Robert	, Directeur de thèse
- M. Marc Renaudin	, Codirecteur de thèse
- M. Christian Piguet	, Rapporteur
- M. Etienne Sicard	, Rapporteur
- M. Philippe Maurine	, Examinateur
- M. Jean Baptiste Rigaud	, Examinateur
- M. Jacques Sonzogni	, Examinateur

A ma défunte mère
A ma soeur
A mon épouse
A ma fille

REMERCIEMENTS

Je souhaite remercier en premier lieu Monsieur Michel ROBERT, mon directeur de thèse, professeur à l'université de Montpellier II et directeur du Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, d'avoir accepté de me prendre en thèse en 2003 et de m'avoir accueilli au sein de son laboratoire.

Je souhaite remercier également Madame Marie-Lise FLOTTES, chef du département de Microélectronique du LIRMM de m'avoir accueilli au sein du département.

Je remercie également Monsieur Marc RENAUDIN, mon co-directeur de thèse, professeur à l'Institut National Polytechnique de Grenoble, de m'avoir permis d'effectuer cette thèse.

Une pensée particulière ainsi que toute ma gratitude à Monsieur Philippe MAURINE, mon encadrant, maître de conférence à l'université Montpellier II, pour son expertise et ses conseils. Sa grande disponibilité et son aide ne m'ont jamais fait défaut tout au long de mes trois années de thèse.

Je remercie sincèrement Monsieur Christian PIGUET, directeur de recherche au Centre Suisse d'Electronique et de Microélectronique à Neuchâtel (CSEM) et Monsieur Etienne SICARD, Professeur à l'Institut National des Sciences Appliquées de Toulouse, qui m'ont fait l'honneur d'accepter d'être rapporteurs et membres du jury de cette thèse.

Tous mes remerciements également à Messieurs Jean-Baptiste RIGAUD, maître de conférence à l'Ecole Nationale Supérieure des Mines Saint-Etienne et Jacques SONZOGNI, design manager de la société STMicroelectronics (Rousset), d'avoir bien voulu participer à mon jury de thèse.

Je remercie également Monsieur Gaston CAMBON d'avoir accepté de présider ce jury de thèse.

Une pensée amicale pour tous les thésards croisés durant ces quelques années passées au LIRMM.

Toutes les personnes m'ayant permis de mener à bien ce travail sont assurées de ma gratitude.

Mes remerciements vont droit à ma chère défunte mère. Sans elle, je ne serai jamais arrivé à ce niveau d'étude. Je remercie également ma soeur de m'avoir soutenu et encouragé tout au long de cette thèse.

Enfin, je remercie affectueusement Mia, mon épouse, pour son soutien indéfectible, sa présence attentionnée, sa gentillesse, sa compréhension et pour son sourire angélique qui m'a apporté un bonheur de tous les instants.

Table des matières

Table des matières

Introduction générale : contexte et motivation 1

Chapitre I : **Les attaques par canaux cachés: état de l'art 5**

I-1-Introduction 5
I-2-La cryptologie 6
I-3-Attaques logiques 11
I-4-Attaques matérielles 12
I-5-Attaques matérielles: contre-mesures 21
I-6-Classification des attaquants et niveaux de sécurité 24
I-7-Conclusion 26

Chapitre II : **Attaque DPA: étude et analyse 27**

II-1-Introduction 27
II-2-Fondement des attaques en puissance 27
II-3-Attaque par analyse simple de consommation (SPA) 33
II-4-Attaque par analyse différentielle de consommation (DPA) 34
II-5-Contre-mesures 40
II-6-Conclusion 43

Chapitre III : **La logique double rail, une contre-mesure à la DPA ? 45**

III-1-Introduction 45
III-2-Pourquoi la logique double rail? 45
III-3-Circuits asynchrones et circuits double rail 51
III-4-Les types d'implantation de cellules double rail 58
III-5-Définition et caractérisation d'une bibliothèque double rail 62
III-6-Analyse de robustesse 75
III-7-Conclusion 79

Chapitre IV : **La logique double rail: impact de la synthèse physique et étude formelle 81**

IV-1-Introduction 81
IV-2-Placement routage et choix de librairie 82
IV-3-Méthodes de placement routage spécifique 89
IV-4-Etude formelle de la robustesse de la logique double rail à la DPA 91
IV-5-Conclusion 111

Chapitre V : **La logique "pseudo" triple rail sécurisée 113**

V-1-Introduction 113
V-2-Fondamentaux de la logique STTL 113
V-3-Analyse des performances 120
V-4-Validations expérimentales 125
V-5-Conclusion 132

Chapitre VI : **Conclusion**..... **133**
Bibliographie..... **137**
Bibliographie personnelle..... **143**

Liste des symboles et acronymes

Liste des symboles et acronymes

AES	Advanced Encryption Standard
CMOS	Complementary Metal-Oxide Semiconductor
DCVS	Differential Cascode Voltage Switch
DCVSL	Differential Cascode Voltage Switch Logic
DEMA	Differential Electro Magnetic Analysis
DES	Data Encryption Standard
DFA	Differential Fault Analysis
DIMS	Delay-Insensitive Min-Term Synthesis
DPA	Differential Power Analysis
DR	Dual-Rail
ECC	Elliptic Curve Cryptosystem
FIPS	Federal Information Processing Standards
FPGA	Field-Programmable Gate Array
LEF	Library Exchange Format
LISAL	Lirmm Synchronous Asynchronous Library
RSA	Rivest Shami Adleman
RTZ	Return To Zero
SABL	Sense Amplifier Based Logic
SEMA	Simple Electro Magnetic Analysis
SPA	Simple Power Analysis
SRAM	Static Random Access Memory
STTL	Secure Triple Track Logic
TAL	Tima Asynchronous Library

Liste des figures

Liste des figures

Fig. 1. Principes de base de la cryptologie 6

Fig. 2. Schéma de Feistel et la fonction f dans le cas de l'algorithme DES 8

Fig. 3. **a)** Extraction de la puce, **b)** Traitement de la couche de passivation 13

Fig. 4. **a)** Microscope électronique à balayage JEOL JSM-6340F, **b)** Une porte AND photographiée par un microscope 14

Fig. 5. Plateforme d'attaque par sondage 15

Fig. 6. Les canaux cachés 15

Fig. 7. Capture des émissions électromagnétiques indirectes issues d'un circuit sécurisé 20

Fig. 9. Inverseur CMOS dimensionné dans une technologie 130 nm 29

Fig. 10. Evolution du courant avec la valeur de la capacité de charge C_L 29

Fig. 11. Evolution du courant avec la valeur de la rampe d'entrée τ_{in} 30

Fig. 12. Schématique d'une porte NAND2 31

Fig. 13. Evolution du courant selon les vecteurs d'entrée 31

Fig. 14. a) Groupe de portes NAND2, b) Evolution du courant en fonction du taux d'activité α 32

Fig. 15. Profils en courant correspondant à un traitement DES 33

Fig. 16. La dernière ronde de l'algorithme DES et la fonction F 35

Fig. 17. Plateforme d'acquisition automatique des traces de courant et des cryptogrammes 37

Fig. 18. Sous circuit de la fonction F 37

Fig. 19. **a)** Courbes DPA obtenues selon $D_i[2]$, **b)** Courbes DPA obtenues selon $D_i[3]$ 39

Fig. 20. Technique de découplage de l'alimentation 42

Fig. 21. Caractéristique de consommation d'un inverseur CMOS 47

Fig. 22. Codage avec retour à l'état invalide (RTZ) 48

Fig. 23. Codage double rail NTRS dit quatre états 49

Fig. 24. Cellule double rail 50

Fig. 25. Communication de type requête/acquittement entre opérateurs asynchrones pour garantir une synchronisation indépendante du temps. 52

Fig. 26. Un protocole requête acquittement deux phases 52

Fig. 27. Un protocole requête acquittement quatre phases 53

Fig. 28. Caractéristiques de consommation d'un DES synchrone 56

Fig. 29. Porte OR2 double rail réalisée à partir de cellules AO222 standard 58

Fig. 30. Porte ‘OR2’ double rail réalisée avec des cellules simple rail CO222 et C-element.....59

Fig. 31. a) Une porte NAND2/AND2 en logique SABL, b) Simulation temporelle des évènements de charge et décharge d'une porte NAND2/AND2 en logique SABL 60

Fig. 32. a) C-element b) une porte NOR2 c) une porte OR3 d) une cellule AND2 double rail sécurisée 61

Fig. 33. Topologie d'une cellule double rail 63

Fig. 34. Schéma partiel d’une cellule OR3 double rail.....67

Fig. 35. Implantation pseudo statique d’une "OR3" double rail.....68

Fig. 36. Implantation statique d’une "OR3" double rail 68

Fig. 37. Autre topologie d'une cellule double rail.....69

Fig. 38. **(a)** Implantation pseudo-statique d’une porte OR3 double rail (28 transistors) **(b)** Half buffer complexe (OR3) (32 transistors) 70

Fig. 39. Nombre de transistors nécessaires à la réalisation de la structure de la figure 44 72

Fig. 40. Protocole de simulation.....73

Fig. 41. Délais de propagation d'une porte 'OR2' double rail sous différents styles d'implémentation 74

Fig. 42. Energie d'une porte 'OR2' double rail sous différents styles d'implémentation.....74

Fig. 43. La fonction F d'un algorithme DES.....75

Fig. 44. Bloc expérimental pour le test de robustesse.....76

Fig. 45. Flot de l'analyse DPA.....77

Fig. 47. Signatures DPA des différents styles d'implantation.....78

Fig. 48. Réalisation d’une porte OR2 double rail à partir de cellules simple rail 84

Fig. 49. Placement routage symbolique de cellules simple rail afin d’obtenir une cellule OR2 double rail et vue du layout 84

Fig. 50. Conception d'une cellule double rail à partir de cellules simple rail 85

Fig. 51. Placement routage symbolique de cellules simple rail complexes afin d’obtenir une cellule OR2 double rail.....86

Fig. 52. Placement routage de cellules OR2 double rail.....87

Fig. 53. Structure considérée pour la comparaison des approches eclatées, semi éclatées et non éclatées de placement routage. 88

Fig. 54. Pourcentage cumulé de nœuds différentiels dont le déséquilibre de charge est inférieur ou égal à $|C_T-C_F|$. 89

Fig. 55. Signatures DPA de la structure figure 57 avant et après placement routage ‘éclaté’ 89

Fig. 56. Routage différentiel proposé par Kris Tiri 90

Fig. 57. Micro-circuit de chiffrement considéré 92

Fig. 58. Profils différentiels en courant des portes NOR2, XOR2 et OR2 et signature DPA complète du micro-circuit de chiffrement pour l'ensemble des valeurs de C considérées 93

Fig. 59. Topologie générique d'une cellule double rail 96

Fig. 60. **(a)** modèle équivalent réduit d'une porte double rail, **(b)** Allures typiques du courant de commutation et des tension d'entrée et de sortie d'une structure CMOS....97

Fig. 61. Profils en courant calculés et simulés d'un inverseur CMOS 99

Fig. 62. Impact d'un déséquilibre de charge sur $\Delta i(t)$....100

Fig. 63. Impact d'un déséquilibre de temps de transition sur $\Delta i(t)$ 101

Fig. 64. Impact d'un déséquilibre de temps de transition sur $\Delta i(t)$ 102

Fig. 65. I_S^{MAX} simulé et calculé en fonction C_T/C_F 104

Fig. 66. I_S^{MAX} simulé et calculé en fonction τ_T/τ_F 105

Fig. 67. I_S^{MAX} simulé et calculé en fonction Δ/τ 106

Fig. 68. Evolutions simulées et calculées de $(C_F/C_T)_{Crit}$ en fonction de I_{MAX} 108

Fig. 69. Evolutions simulées et calculées de $(\tau_F/\tau_T)_{Crit}$ en fonction de I_{MAX} 109

Fig. 70. Evolutions simulées et calculées de $|\Delta/\tau|_{Crit}$ en fonction I_{MAX} 110

Fig. 71. Impact du décalage temporel sur le courant différentiel....114

Fig. 72. **a)** Structures insensibles au décalage temporel, **b)** Mode de fonctionnement des structures insensibles au décalage temporel....114

Fig. 73. Codage des données utilisé par la STTL 115

Fig. 74. Topologie d'une cellule STTL....116

Fig. 75. **a)** Les portes complexes d'une porte XOR2 STTL, **b)** Schéma complet d'une porte XOR2 STTL pseudo statique 117

Fig. 76. Schéma complet d'une porte XOR2 TTL statique....118

Fig. 77. Structure composée de cellules STTL....118

Fig. 78. Chronogramme associé au fonctionnement de la figure 77....119

Fig. 79. Structure d'évaluation....121

Fig. 80. Résultats de placement routage 122

Fig. 81. Protocole de simulation....123

Fig. 82. Délais de propagation d'une porte 'OR2' sous différents styles d'implémentation 124

Fig. 83. Consommation d'une porte 'OR2' sous différents styles d'implémentation 125

Fig. 84. Résultats des attaques DPA menées sur le circuit STTL 126

Fig. 85. Résultats des attaques DPA menées sur un circuit à base des cellules proposées dans [Gui04] 127

Fig. 86. Variation des délais de propagation entrée/sortie 129

Fig. 87. Résultats des attaques DPA menées sur les différents circuits 131

Liste des tableaux

Liste des tableaux

Tableau 1. Algorithme RSA 9

Tableau 2. Table de vérité d'une OR3 64

Tableau 3. Table de correspondance simple rail - double rail 65

Tableau 4. Table de vérité de $a_1 \oplus a_0$ 66

Tableau 5. Quelques fonctionnalités de la bibliothèque LISAL 71

Tableau 6. Tableau de comparaison des coûts d'intégration en nombre de transistors (surface) 71

Tableau 7. Valeurs moyennes et maximales des capacités des équipotentielles 88

Tableau 8. Comparaison des coûts de réalisation en nombre de transistors 120

Tableau 9. Estimation de surface physique des cellules 122

Tableau 10. Tableau reportant les délais de propagation des différents circuits (STTL, [Gui04, Tir02, Mau03, Raz05]) 128

Introduction générale

Introduction

Avec le développement du marché des nouvelles technologies de l'information et des télécommunications, la notion de confidentialité des données occupe une place croissante parmi les différentes préoccupations des concepteurs de circuits intégrés. Pour satisfaire un niveau de confidentialité satisfaisant, ces derniers se basent généralement sur la cryptographie. L'objectif est de rendre les données inintelligibles sauf par celui qui en possède la clef de déchiffrement. Incluant en son sein un ou plusieurs blocs cryptographiques, la carte à puce apparaît comme le support incontournable des applications sécurisées. Les domaines d'applications en sont nombreux : paiement, télévision à la demande, téléphonie mobile, médical, transport public, contrôle d'accès, etc.

En terme de sécurité, notamment dans le domaine plus particulier des transactions bancaires, le code PIN d'une carte à puce, la clef privée qu'elle peut contenir ou toute autre information confidentielle sont considérés comme inviolables. En effet, durant les trois dernières décennies, de gros efforts ont été consentis pour développer des algorithmes de chiffrement. Si ces efforts ont aboutis à la définition d'algorithmes de chiffrement standard particulièrement robustes aux attaques logiques, peu ou pas d'efforts ont été dévolus à la sécurisation des plateformes matérielles exécutant ces algorithmes. Par voie de conséquence, il existe aujourd'hui des attaques matérielles capables de retrouver les informations secrètes contenues dans la puce. On parle de cryptanalyse matérielle.

La cryptanalyse matérielle désigne l'ensemble des techniques qui consistent à exploiter les failles de la puce elle-même pour en extraire les données secrètes. Il en existe trois grandes catégories: les attaques invasives, les attaques semi-invasives et les attaques non-invasives. Une attaque invasive est une attaque qui va agir physiquement sur la puce et peut conduire jusqu'à sa destruction. De la même manière qu'une attaque invasive, une attaque semi-invasive agit physiquement sur la puce sans pour autant la détériorer et comme une attaque non-invasive, elle exploite les corrélations entre les données manipulées et les signaux compromettants. Une attaque non-invasive englobe l'ensemble des techniques qui consistent à exploiter uniquement les signaux compromettants.

Un signal compromettant est un signal qui peut rendre compte des activités ou de l'état interne du circuit, l'idée de base des attaques non-invasives est d'établir statistiquement la corrélation entre les données manipulées et ces signaux compromettants. Ces signaux peuvent être : le courant consommé, le temps de traitement des données, la température, les émissions électromagnétiques, des comportements fautifs, le son, etc..

Parmi ces attaques non-invasives, "l'attaque différentielle en puissance" ou "Differential Power Analysis" (DPA) est reconnue comme étant très efficace. Introduite par Paul Kocher en 1998, cette attaque permet d´extraire toute information secrète contenue dans une puce en mesurant son activité électrique durant quelques milliers de cycles de fonctionnement. Le principe de base sur lequel repose cette attaque est l'exploitation, par des moyens statistiques, des éventuelles corrélations existant entre les données manipulées et les profils en courant de consommation. Cette attaque constitue aujourd'hui une des attaques les plus dangereuses dans la mesure où elle peut être mise en œuvre avec succès avec un faible niveau de compétence en électronique et du matériel à des prix relativement abordables.

Pour les raisons précédemment citées, l'attaque DPA représente aujourd'hui un danger significatif et fait l'objet d'un challenge permanent pour les concepteurs de circuits sécurisés. De ce fait, beaucoup de contre-mesures ont été proposées dans la littérature. Au niveau algorithmique, on distingue quelques méthodes : "Time randomization", "Permutation de l'exécution", "Masquage", etc. Au niveau matériel, on note également quelques techniques de contre-mesure, à savoir: le filtrage de l'alimentation ou encore l'utilisation de logique spécifique, etc. Parmi les logiques spécifiques, l'approche double rail apparaît comme une alternative intéressante à la logique simple rail dans la mesure où elle offre la possibilité d'équilibrer la consommation. Cependant, elle se heurte à un manque d'outil de conception industriel et aboutit généralement à un surcoût en surface excessif. Par ailleurs, sans précaution particulière durant l'étape de placement-routage, les capacités de routage peuvent réduire localement, mais de manière significative la robustesse de cette logique aux attaques DPA.

Dans ce contexte, ce travail de thèse s'est focalisé sur l'analyse des atouts et faiblesses de la logique double rail et à l'amélioration de celle-ci. Cette thèse a débuté par l'acquisition des connaissances indispensables pour aborder ce sujet de thèse et plus particulièrement à l'acquisition de notions relatives à la cryptographie et à la cryptanalyse matérielles. En effet,

une étude bibliographique des attaques par canaux cachés a été effectuée. Le chapitre I dresse un rapide état de l'art sur les différentes attaques répertoriées dans la littérature. Le chapitre II, quant à lui, est dédié à l'étude des attaques exploitant le canal consommation et plus particulièrement à l'étude des fondements de ces attaques.

Les bases de mes travaux de thèse posées, le chapitre III propose une étude comparative de différentes logiques double rail ayant été identifiées ou non comme robustes aux attaques DPA. Cette étude est exclusivement réalisée au niveau schématique de portes, i.e. sans tenir compte des capacités de routage introduites lors de la synthèse physique des circuits. Ce chapitre III introduit également une méthode de construction de portes logiques double rail, méthode qui constitue une première contribution de cette thèse.

Le chapitre IV, est quant à lui, entièrement dédié à l'impact de la synthèse physique sur la robustesse des circuits double rail. Plus précisément, après une formalisation mathématique pertinente de ce qu'est le syndrome réellement capturé par l'attaque DPA, une analyse formelle de la robustesse de la logique double rail aux attaques DPA est proposée. Cette analyse, qui prend notamment en compte l'introduction éventuelle de déséquilibres de charges, de temps de transition et de temps d'arrivée des signaux, nous a finalement permis d'identifier l'espace de conception dans lequel la logique double rail peut être considérée comme robuste aux attaques DPA. Ces deux études formelles constituent une seconde contribution de cette thèse.

Le chapitre V est lui entièrement dévolu à la prise en compte des faiblesses de la logique double rail afin de développer une logique "double rail" améliorée présentant un niveau de robustesse à la DPA nettement plus important que les logiques précédemment considérées. Outre sa robustesse, cette logique semble être plus compacte et donc offrir un très bon compromis "sécurité/surface/vitesse/consommation". Cette logique a été appelée STTL pour Secure Triple Track Logic. Ceci constitue une troisième contribution de cette thèse.

Chapitre I:

Les attaques par canaux cachés: état de l'art

Pour garantir la sécurité des données qui transitent dans les canaux de communication, les concepteurs de composants sécurisés (cartes à puce) font appel à la cryptographie. Après une introduction à la cryptographie et une présentation de quelques algorithmes cryptographiques, nous verrons que si les composants sécurisés sont considérés comme inviolables, la réalité est qu'il existe un certain nombre d'attaques capables d'extraire les informations secrètes contenues en leur sein. On distingue notamment les attaques logiques et les attaques matérielles. Pour rendre ces attaques inopérantes, des contre-mesures ont été proposées. Cependant, la quasi-totalité d'entre elles ne font que rendre les attaques beaucoup plus complexes sans pour autant les éradiquer.

Chapitre I : Les attaques par canaux cachés: état de l'art

I-1-Introduction

Avec le développement des réseaux de télécommunication, les transactions financières n'empruntent plus seulement des réseaux fermés ou privés mais un réseau mondial et accessible par tous. A partir du moment où ces transactions sont susceptibles d'être interceptées ou "écoutées", la sécurité des données doit être garantie. Aujourd'hui, les techniques issue de la cryptographie (cryptologie) sont les seules solutions permettant d'assurer la confidentialité et l'intégrité des transactions sur un réseau ouvert (internet, réseau des cartes bancaires, etc.) [Abr91].

Si les cartes à puce de première génération ne permettaient que de conserver et de gérer des informations sensibles telles que les numéros de compte, les mots de passe ou encore les informations médicales, elles offrent aujourd'hui la possibilité d'effectuer des opérations cryptographiques particulièrement complexes. En effet, les cartes à puce les plus perfectionnées contiennent des microprocesseurs relativement puissants (jusqu'à 32bits), des quantités de mémoire significatives (jusqu'à 100Mb), mais aussi et surtout des cryptoprocesseurs tels que des DES, des triples DES, ou encore des AES.

Du fait de l'importance des enjeux stratégiques et économiques relatifs aux transactions effectuées avec les cartes à puce, ces dernières doivent être capables de résister à toutes sortes d'agressions et attaques connues. Si les algorithmes de chiffrement implantés dans ces cartes garantissent un très haut niveau de résistance aux attaques exploitant les signaux booléens échangés par la carte avec son environnement, les cartes à puce ne présentent pas le même degré de résistance aux attaques matérielles ou physiques.

Les attaques matérielles font partie d'une vaste famille de techniques cryptanalytiques. Celles-ci exploitent des comportements, représentatifs de certaines étapes des algorithmes de chiffrement, pouvant être aisément observés au travers de syndromes physiques qu'exhibe leur implémentation matérielle. Dans le domaine des attaques matérielles, les attaques sont

nombreuses et portent sur différents paramètres ou syndromes. On en distingue trois grandes catégories d'agression : les attaques invasives, les attaques semi-invasives et les attaques non-invasives.

Les attaques non-invasives, communément appelées attaques par canaux cachés, se révèlent être les plus dangereuses dans la mesure où leur mise en œuvre ne nécessite relativement peu de moyen financier et de compétences techniques pour être menées avec succès. Du côté des concepteurs de circuits sécurisés, des contre-mesures ont été proposées pour rendre ces attaques plus difficiles à mettre en œuvre voire inopérantes.

Dans ce chapitre, après quelques généralités sur la cryptologie, nous étudierons ces différentes attaques. Du fait de leur dangerosité, nous nous attarderons un peu plus sur les attaques matérielles. Avant de conclure ce chapitre, nous dresserons un bref état de l'art sur les différentes contre-mesures.

I-2-La cryptologie

Etymologiquement, la cryptologie est la science du secret. Elle englobe la cryptographie et la cryptanalyse.

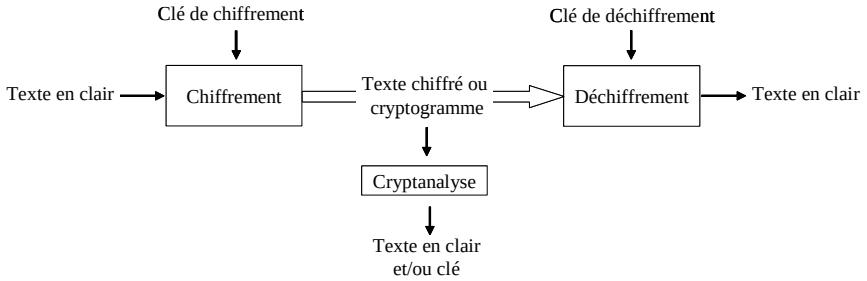


Fig. 1. Principes de base de la cryptologie

I-2-a-Cryptographie

La cryptographie est l'ensemble des techniques permettant de chiffrer des données afin de pouvoir garantir leur confidentialité, leur authenticité et leur intégrité lors de leur transfert sur des canaux non sécurisés. En effet, les algorithmes cryptographiques vont appliquer des transformations bijectives aux données permettant de les rendre inintelligibles à toutes personnes ne possédant pas la clef de chiffrement. C'est ce qu'on appelle le chiffrement qui, à partir d'un texte en clair, donne un texte chiffré ou encore un cryptogramme. Inversement, le déchiffrement est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré et d'une clef secrète.

Les algorithmes cryptographiques sont généralement classés selon deux grandes catégories: les algorithmes cryptographiques à clef secrète et les algorithmes cryptographiques à clef publique.

Les algorithmes à clef privé, ou encore algorithmes symétriques, se caractérisent par l'utilisation de la même clef pour effectuer le chiffrement et le déchiffrement. Ceci implique donc le transfert sécurisé de la clef de chiffrement aux différentes personnes souhaitant communiquer de manière sécurisée par la suite. Parmi les algorithmes de chiffrement à clef privée, on distingue les algorithmes DES, AES [FIPS197], etc..

Les algorithmes à clef publiques, ou encore algorithmes asymétriques, se caractérisent par l'utilisation de deux clefs : une clef privée et une clef publique. Ce type d'algorithme a été proposé par W. Diffie et M Hellman en 1976 [Dif76] afin de résoudre le problème lié aux transferts de clefs secrètes. En effet, dans ce type de chiffrement, tout le monde peut utiliser la clef publique pour expédier des messages chiffrés que seul le détenteur de la clef privée peut déchiffrer. Parmi les algorithmes à clefs asymétriques on trouve notamment le RSA, le DSA-DH, le ElGamal, les courbes elliptiques.

I-2-a-1-L'algorithme DES: Data Encryption Standard

La norme de chiffrement Data Encryption Standard (DES) [FIPS46a] est une proposition du National Bureau of Standards (NSB) américain datant du milieu des années 70. Il s'agissait alors de définir un algorithme robuste, gratuit et destiné au grand public. Comme la majorité des algorithmes de chiffrement à clef secrète, le DES a été conçu selon le principe des schémas de Feistel [Sch01].

Schémas de Feistel

Le premier objectif de tout algorithme à clef secrète est de rendre le message chiffré d'apparence aléatoire afin de limiter les risques d'attaques, par analyse linéaire par exemple. Le problème du chiffrement est donc bien la définition de fonctions aléatoires. Si l'on sait depuis longtemps construire des fonctions pseudo-aléatoires, il a fallu attendre la fin des années 70 et plus particulièrement les travaux de Feistel pour être capables de construire des fonctions bijectives aléatoires.

La solution proposée par Feistel est très élégante. Elle s'appuie sur l'utilisation d'une fonction f pseudo aléatoire prenant en entrées de mots de n bits et délivrant en sortie des mots de n bits. Cette fonction f est en effet introduite dans le schéma de Feistel représenté sur la figure 2.

Comme on peut le constater, l'algorithme de chiffrement va procéder en chiffrant des blocs de 2n bits (G+D), qu'on partage en deux parties : gauche G et droite D. L'image du bloc (G, D) par le schéma de Feistel est le bloc (L, R) de 2n bits tel que L=D et R= G 'xor' f(D) où f est la fonction pseudo aléatoire considérée. Feistel a démontré que la fonction associée à ce schéma est une fonction bijective. Toutefois comme on peut le constater seul les n bits de G sont chiffrés. Pour chiffrer l'ensemble du texte il faut donc appliquer de manière itérative ce schéma. On parle alors de ronde.

Dans le cas plus spécifique du DES, 16 itérations du schéma Feistel sont effectuées pour traiter des mots de 64 bits. La clef de chiffrement utilisée est de 64 bits dont 8 bits de parité. Pour chaque ronde, le DES utilise une sous-clef de 48 bits calculée en fonction de la sous clef précédente grâce à un réseau de permutations et de décalages. Comme l'illustre la figure 2, la fonction f du DES est constituée de 8 boîtes de substitution dont le rôle principal est de rajouter un peu plus de non-linéarité à l'algorithme.

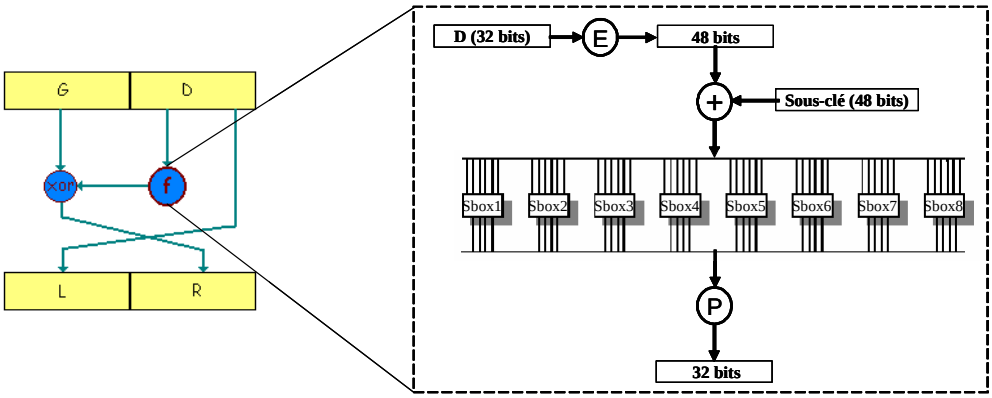


Fig. 2. Schéma de Feistel et la fonction f dans le cas de l'algorithme DES

En 1999, afin d'améliorer la robustesse de l'algorithme DES dont la longueur de clef est jugée trop faible, le Triple-DES a été adoptée comme nouvelle norme [FIPS46b]. Le Triple-DES est une variante du DES qui consiste par exemple à appliquer trois l'algorithme DES avec deux ou trois clefs de chiffrement secrètes distinctes selon le schéma : chiffrement / déchiffrement chiffrement. Dans tous les cas, la longueur efficace de la clef du triple DES ne dépasse pas 112 bits.

Finalement en 2001, le standard DES a été remplacé par un nouveau standard : l'AES pour Advanced Encryption Standard. Ce nouveau standard de chiffrement utilise l'algorithme Rijndael [FIPS197].

I-2-a-2-L'algorithme RSA

Introduit par Rivest, Shamir et Adleman en 1978, l'algorithme à clef publique RSA permet le chiffrement et la signature [FIPS186] de texte. Il est aujourd'hui encore très largement utilisé. Cet algorithme repose sur la difficulté de factoriser de grands nombres.

Voici comment se fait la génération des paires de clefs :

1. On commence par choisir deux grands nombres premiers, **p** et **q**, et on calcule **n = p*q**. **n** est rendu public; **p** et **q** doivent rester secrets et sont donc détruits une fois les clefs générées.
2. On choisit ensuite aléatoirement une clef publique **e** telle que **e** et **(p-1)*(q-1)** soient premiers entre eux.
3. La clef privée **d** est obtenue grâce à l'algorithme d'Euclide : **e*d = 1 mod (p-1)*(q-1).**

Soit **m** le message en clair et **c** le cryptogramme. La fonction de chiffrement est, de façon simplifiée, $\mathbf{c = m^e \bmod n}$ (si **m** est plus grand que **n**, il est séparé en morceaux de valeur inférieure à **n** et chaque morceau est chiffré séparément suivant cette formule). Du fait de la relation entre **e** et **d**, la fonction de déchiffrement correspondante est $\mathbf{m = c^d \bmod n}$. La signature se fait de manière similaire, en inversant **e** et **d**, c'est-à-dire en chiffrant avec une clef privée et en déchiffrant avec la clef publique correspondante : $\mathbf{s = m^d \bmod n}$ et $\mathbf{m = s^e \bmod n}$.

Clefs			
Clef publique	**n = p*q**, où **p** et **q** sont deux grands nombres premiers tenus secrets **e** telle que **e** et **(p-1)*(q-1)** soient premiers entre eux		
Clef privée	$\mathbf{d = e^{-1} \bmod (p-1)*(q-1)}$		
Algorithmes			
Chiffrement	$\mathbf{c = m^e \bmod n}$	Déchiffrement	$\mathbf{m = c^d \bmod n}$
Signature	$\mathbf{s = m^d \bmod n}$	Vérification	$\mathbf{m = s^e \bmod n}$

Tableau 1. Algorithme RSA

Pour un cryptanalyste, retrouver la clef privée à partir de la clef publique nécessite de connaître **(p-1)*(q-1) = (p*q)-p-q+1 = n+1-p-q**, donc de connaître **p** et **q**. Pour cela, il doit

factoriser **n**. Donc **n** doit être suffisamment grand pour que cela ne soit pas possible dans un temps raisonnable par rapport au niveau de sécurité requis. Actuellement, la longueur du module **n** varie généralement de 512 à 2048 bits suivant les utilisations. Compte tenu de l'augmentation des vitesses de calcul des ordinateurs et des avancées mathématiques en matière de factorisation des grands nombres, la longueur minimale des clefs doit augmenter au cours du temps.

I-2-b-Cryptanalyse

A l'inverse de la cryptographie, la cryptanalyse est l'étude des procédés cryptographiques dans le but de trouver leurs faiblesses et de déchiffrer illégitimement les textes chiffrés. Le décryptage est l'action consistant à retrouver le texte en clair sans connaître la clef de déchiffrement. Une tentative de cryptanalyse est communément appelée attaque. Lorsqu'une attaque permet de déchiffrer un message chiffré à l'aide d'un cryptosystème, on dit alors que l'algorithme de chiffrement a été "cassé".

D'une manière générale, on suppose toujours que le cryptanalyste connaît le détail des algorithmes, fonctions mathématiques ou protocoles employés. Même si ce n'est pas toujours le cas en pratique. Il serait en effet particulièrement risqué de se baser sur le secret des mécanismes utilisés pour assurer la sécurité d'un système.

On distingue habituellement quatre méthodes de cryptanalyse:

- Une **attaque sur texte chiffré seulement** consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés;
- Une **attaque sur texte clair connu** consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés, connaissant le texte en clair correspondant;
- Une **attaque sur texte clair choisi** consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair;
- Une **attaque sur texte chiffré choisi** consiste à retrouver la clef de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair.

Il existe deux grandes familles d'attaques contre les cartes à puce ou plus précisément contre les systèmes cryptographiques: les attaques logiques et les attaques matérielles.

I-3-Attaques logiques

Les attaques logiques exploitent les faiblesses mathématiques des algorithmes de chiffrement et les erreurs de conception ou de codage qui sont passées inaperçues lors des tests de sécurité. Elles visent donc le logiciel embarqué dans le cas des cartes à puce. Typiquement, ces attaques permettent de contourner certains mécanismes de protection ou encore de détourner l'utilisation de certaines fonctionnalités de la carte à puce.

Parmi les attaques qui exploitent les faiblesses mathématiques, on distingue les attaques linéaires et les attaques différentielles. Parmi les attaques qui se basent sur les erreurs de codage, on note par exemple le cas du cheval de Troie [Sum97].

I-3-a-Attaque linéaire

L'attaque linéaire, développée par Mitsuru Matsui en 1993 [Mat93], a été spécialement conçue pour casser l'algorithme de chiffrement symétrique DES. Avec une analyse pertinente des couples [texte clair – texte chiffré] obtenus avec la même clef, l'attaque linéaire consiste à faire une approximation linéaire de l'algorithme de chiffrement. En augmentant le nombre de couples disponibles, on améliore la précision de l'approximation et on peut en extraire la clef.

Tous les nouveaux algorithmes de chiffrement doivent veiller à être résistants à ce type d'attaque. Le DES n'était pas conçu pour empêcher ce genre de méthode dans la mesure où les boîtes de substitution présentent des propriétés linéaires alors qu'elles étaient justement prévues pour ajouter une non-linéarité à l'algorithme.

I-3-b-Attaque différentielle

Introduite par Eli Biham et Adi Shamir dans [Bih90], l'attaque différentielle est une analyse statistique des changements dans la structure de la méthode de chiffrement après avoir légèrement modifié les entrées. Avec un très grand nombre de couples [texte clair – texte chiffré], il est possible d'extraire la clef. A titre d'exemple, dans [Bih90] les auteurs ont montré comment un DES à 8 tours peut être cassé en seulement quelques minutes. Toutefois, on sait maintenant que des algorithmes récents (AES, IDEA, etc.) sont conçus pour résister à ce type d'attaque.

I-3-c-Le cheval de Troie

Par définition, un cheval de Troie est un programme malveillant dissimulé dans un programme autorisé.

Les avancées technologiques dans le domaine des cartes à puce font qu'il coexiste plusieurs applications au sein des cartes mais aussi qu'il est possible de charger de nouvelles applications. Ainsi les cartes multi-applicatives se rapprochent des ordinateurs traditionnels. A ce titre, elles sont sujettes aux mêmes failles de sécurité, en particulier, les chevaux de Troie. Par analogie à ce qui passe avec les ordinateurs liés au réseau Internet, un cheval de Troie introduit dans une carte à puce a pour but de sortir des mots de passe, des copies des données sensibles ou encore d'exécuter des tâches visant à affaiblir la sécurité du système [Sum97, Ros01, Bid01].

I-4-Attaques matérielles

Dans ce paragraphe, nous présentons les différentes attaques pouvant être réalisées sur les implémentations matérielles de systèmes sécurisés. Nous commencerons par les attaques invasives dont la mise en œuvre peut détériorer, voire détruire le circuit (cryptoprocesseur) à analyser, ensuite les attaques non-invasives qui se contentent de procéder à une observation extérieure du système et enfin les attaques semi-invasives, une classe intermédiaire entre des deux premières.

I-4-a-Attaques invasives

Les attaques invasives sont des attaques menées en général par des experts et requièrent du matériel spécifique. Typiquement, une attaque invasive se déroule en deux étapes: la préparation des échantillons et l'attaque à proprement parler.

I-4-a-1-La préparation des échantillons

La préparation des échantillons consiste à isoler le composant électronique, pour cela l'attaquant doit mettre à nu la puce. Pour ce faire, l'attaquant a le choix entre utiliser des produits chimiques ou du matériel sophistiqué. Dans le cas d'une carte à puce, cela revient à extraire la puce de son support en plastique et à enlever la couche de passivation [Lee93] [Bec98].

D'un point de vue pratique, l'extraction du module s'effectue aisément en chauffant le plastique. Pour enlever la couche de passivation qui protège la surface de la puce, cette dernière est plongée dans de l'acide nitrique chauffé à 60°. Pour faciliter la mise en œuvre des attaques et plus particulièrement celle des attaques par sondage, l'échantillon ainsi préparée est insérée dans un boîtier de test de type DIP dont les broches sont connectées aux différents points de contact de la puce.

a) **b)**

Fig. 3.a) Extraction de la puce, **b)** Traitement de la couche de passivation

I-4-a-2-Les attaques par reconstruction de layout

La reconstruction de layout (Reverse engineering) est l'activité qui consiste à étudier un composant électronique pour en déterminer de manière fine la structure interne et par la suite en déduire le fonctionnement. Dans le domaine plus particulier des cartes à puce, la reconstruction de layout permet d'une part :

- d'analyser l'architecture de la puce afin d'en comprendre les mécanismes de sécurité et donc de pouvoir les contourner,
- et d’autre part par de lire le contenu d'une mémoire et récupérer les informations secrètes (si non cryptées).

Concrètement, la reconstruction de layout est une démarche qui consiste à extraire des informations sur l'emplacement exact de tous les transistors et de toutes les interconnexions composant la puce. Pour ce faire, les différentes couches technologiques sont successivement "enlevées" et "cartographiées" à l’aide d’un microscope électronique à balayage (SEM, Figure 4). Toutes les informations collectées sont alors assemblées pour permettre la reconstruction complète du layout du circuit ou encore de reconstituer des netlists de simulation.

Compte tenu des moyens nécessaire à sa mise en oeuvre, ce type d'attaque ne peut être réalisée que par des spécialistes en conception de circuits intégrés et par de grands groupes ayant des moyens financiers très importants. En effet, cette attaque exige du matériel très perfectionné et coûteux.

a) b)

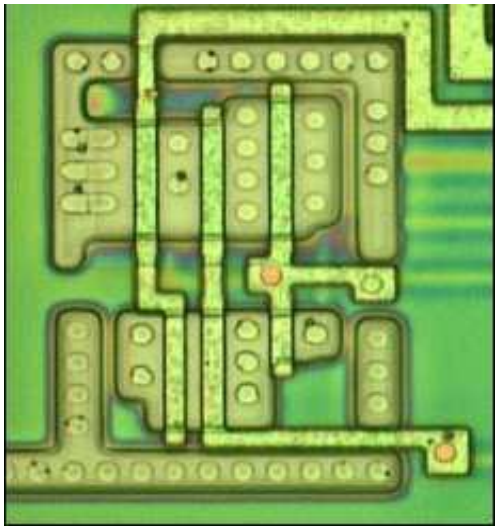


Fig. 4.a) Microscope électronique à balayage JEOL JSM-6340F, **b)** Une porte AND photographiée par un microscope

I-4-a-3-Les attaques par sondage

Le principe d'une attaque par sondage ou "probing attack" est d'espionner l'activité électrique d'un composant électronique du circuit (cryptoprocesseur) en positionnant une sonde suffisamment proche dudit composant. La mise en œuvre de ce type d'attaque exige l'utilisation d'une plateforme sophistiquée représentée par la figure 5. Ce matériel permet non seulement de récupérer les données transitant sur un bus de données mais aussi d'imposer des valeurs logiques sur certains nœuds du circuit. Avec un tel contrôle de l'environnement, l'attaquant peut être en mesure de déduire tout ou une partie du secret du circuit cryptographique.

Pour l'instant ce type d'attaque a un intérêt essentiellement théorique, car supposer que l'attaquant dispose d'un tel contrôle de l'environnement est une hypothèse extrêmement forte (possibilité de manipuler le matériel comme il l'entend, accès à du matériel de mesure très coûteux, etc.).

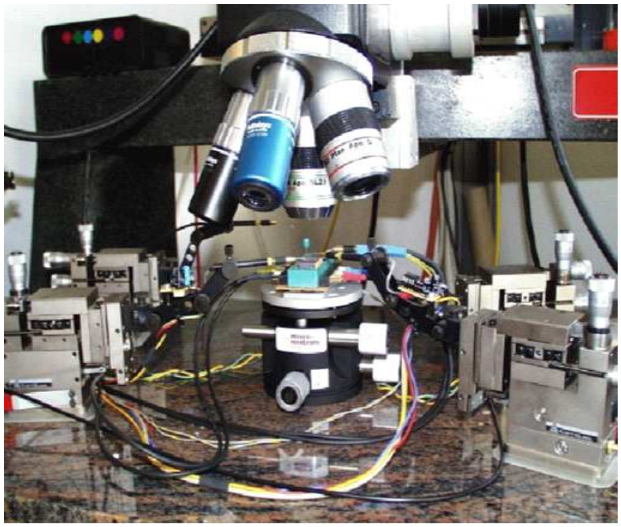


Fig. 5. Plateforme d'attaque par sondage

I-4-b-Attaques par canaux cachés

Contrairement aux attaques invasives, la mise en œuvre des attaques par canaux cachés ne requière pas d'accès direct à la puce. En effet, elles consistent à exploiter les canaux cachés ou les signaux physiques émis par les implémentations matérielles des algorithmes de chiffrement durant les calculs cryptographiques. Comme l'illustre la figure 6, ces canaux cachés peuvent être le temps de calcul ou la consommation électrique du composant, ses rayonnements (Electromagnétique, calorifique, sonores), ou encore les résultats de calculs comportant des erreurs car obtenus par injection de fautes pendant la mise en œuvre du procédé cryptographique. Etroitement corrélés avec l'état interne du circuit, ces canaux cachés peuvent permettre d'extraire des composants les secrets (ex: clef de chiffrement) sur lesquels se base la sécurité des algorithmes mis en œuvre.

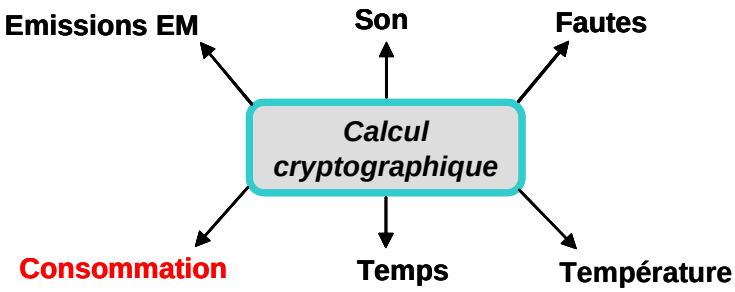


Fig. 6. Les canaux cachés

I-4-b-1-Attaque par analyse de temps de calcul – Timing attacks

Cette attaque exploite la corrélation entre les données manipulées et les temps de calcul (en terme de cycle d'horloge) lors des opérations cryptographiques pour extraire les informations secrètes contenues dans la puce [Koc96, Koe99]. Dans ce cas précis, on dit que l'attaquant exploite le canal temporel.

En effet, qu'ils soient implémentés au niveau logiciel ou matériel, les algorithmes cryptographiques ont des temps de calcul très dépendants des données d'entrée et de la clef secrète. De ce fait, des mesures et analyses sophistiquées de ces temps de calcul peuvent permettre de retrouver la valeur de la clef secrète sur laquelle repose la sécurité du cryptosystème.

La possibilité d'utiliser le canal temps a été proposée par P. Kocher en 1996 dans [Koc96]. Quelques années plus tard, J. Dhem proposa une version simplifiée de cette attaque. Ce dernier démontra en outre dans [Dhe98] comment il était possible de casser une implémentation naïve de l'algorithme RSA [FIP186]. Cet algorithme est d'ailleurs maintenant connu pour avoir des implémentations naïves particulièrement vulnérables aux attaques temporelles. Cette faiblesse du RSA tient au fait que selon la valeur du bit de clef traité, seules deux possibilités de branchement sont possibles, branchements dont les temps de calculs sont très nettement différents et nécessitant des ressources distinctes.

Aujourd'hui, avec l'évolution des techniques des contre-mesures, la majorité des cryptosystèmes sont robustes aux attaques temporelles. Toutefois, combinées avec d'autres attaques, les attaques temporelles restent une menace contre les systèmes sécurisés.

I-4-b-2-Attaque par analyse de consommation

Les attaques par analyse de consommation consistent à étudier les courants et les données manipulées d'un circuit dans le but d'en extraire des informations secrètes ou bien la clef de chiffrement.

Aujourd'hui, la majorité des cryptosystèmes sont à base de la technologie CMOS. Les caractéristiques de consommation de cette technologie font que l'amplitude du courant de consommation correspondante à un évènement {0-1} est beaucoup plus importante que celle correspondante à un évènement {1-0}. De ce fait, l'amplitude du courant de consommation d'un circuit CMOS est proportionnelle au nombre de transition {0-1} et au poids de Hamming dans une certaine mesure. Ainsi, une analyse fine de la consommation peut permettre de

retrouver les données manipulées lors des calculs cryptographiques notamment la clef de chiffrement.

En fonction de la méthode d'analyse de la consommation, on distingue deux types d'attaque: l'attaque par analyse simple de la consommation (SPA) et l'attaque par analyse différentielle de la consommation (DPA).

L'attaque par analyse simple de la consommation [Bih99] consiste à effectuer une analyse directe des profils en courant durant les opérations cryptographiques. Cette attaque permet de récupérer des informations sur le type d'instruction en cours d'exécution et sur la valeur d'une partie de la clef secrète dans le cas d'un algorithme de chiffrement symétrique. Toutefois, la réussite de la SPA exige une connaissance de l'algorithme cryptographique considéré et surtout de la manière dont il est implémenté.

L'attaque par analyse différentielle de la consommation [Koc99] est une version plus puissante de la SPA. En effet, d'une part, la réussite de l'attaque DPA ne suppose pas une connaissance des détails de l'implémentation de l'algorithme de chiffrement et d'autre part, elle utilise des analyses statistiques intelligentes pour établir une corrélation entre les données traitées et la consommation. Par ailleurs, ces méthodes statistiques sont capables d'identifier les plus petites variations de la consommation qui peuvent être par la suite exploitées pour extraire des informations sur la valeur de la clef secrète.

Aujourd'hui encore, ces attaques par analyse de consommation et plus particulièrement les attaques DPA sont prises très au sérieux par les concepteurs de circuits sécurisés. En effet, elles sont jugées très dangereuses dans la mesure où elles sont assez simples à mettre en œuvre et singulièrement efficaces. Plus de détails sur ces attaques seront donnés dans le chapitre II.

I-4-c-Attaques semi-invasives

C'est une nouvelle classe d'attaque publiée pour la première fois en 2002 par Sergei Skorobogatov et Ross Anderson dans [Sko02]. Comme les attaques invasives, elles supposent une décapsulation (depackaging) du circuit pour se rapprocher le plus possible de la surface de la puce sans toutefois détériorer le fonctionnement de cette dernière. Pour leur mise en œuvre, les attaques semi-invasives ne nécessitent pas de contact physique avec la puce. En d'autres termes, les attaques semi-invasives sont dans une certaine mesure, une classe intermédiaire entre les attaques invasives et non-invasives.

Avec la diminution des nœuds technologiques et la complexité toujours grandissante des circuits intégrés, les attaques invasives deviennent de plus en plus contraignantes et coûtent de plus en plus cher. Les attaques semi-invasives profitent de cette situation et deviennent attrayantes dans la mesure où d'une part, elles ne supposent pas l'utilisation d'outils très coûteux et d'autre part, elles permettent d'obtenir des résultats assez rapidement.

Parmi les attaques semi-invasives, on distingue les attaques par injection de fautes, les attaques par analyse des émissions électromagnétiques, etc. Notez cependant que sans décapsulation et sans extraction de la couche de passivation de la puce, les attaques par analyse des émanations électromagnétiques sont classées parmi les attaques non-invasives.

I-4-c-1-Attaque par injection de fautes

Introduites en 1997 par Boneh, Demillo et Lipton dans [Bon97], les attaques par injection de fautes consistent à générer intentionnellement des fautes dans un cryptosystème en cours de fonctionnement afin d'obtenir des comportements anormaux exploitables. Ces injections de fautes peuvent altérer les données manipulées ou corrompre les opérations cryptographiques de telle façon à ce que les informations secrètes soient dévoilées [Bih97].

D'une manière générale, les fautes sont générées par des modifications anormales des paramètres externes du circuit comme par exemple: des variations intempestives des signaux d'horloge et d'alimentation, des variations de température, une exposition du circuit à des faisceaux de lumière de différentes longueurs d'ondes, etc.

Les fautes générées peuvent être permanentes ou bien transitoires. Bien évidement, une faute permanente induit le cryptosystème en erreur et ce de manière permanente. Générer une faute permanente peut par exemple consister à figer le contenu d'une cellule mémoire à une valeur constante [Sko02]. A l'inverse, une faute transitoire induit le cryptosystème en erreur durant une fraction temporelle bien délimitée correspondant à opération bien spécifique du cryptosystème que l’on cherche à corrompre. Injecter une variation intempestive des signaux d'horloge et ou d'alimentation constitue un moyen efficace de générer des erreurs logiques transitoires par violation des contraintes de setup et de hold time.

L’injection de faute à des fins de cryptanalyse nécessite la définition de modèles de fautes tout comme des modèles de fautes sont générés pour effectuer le diagnostic des circuits fautifs. Contrairement au diagnostic des circuits, ces modèles de fautes ne sont pas utilisés pour retrouver le site des pannes. En effet, ces modèles sont utilisés pour retrouver des informations secrètes et plus particulièrement des portions de la clef secrète.

La formalisation d'une attaque par injection de faute exploitant un modèle de faute spécifique a été introduite pour la première fois en 1997 par Boneh, Demillo et Lipton [Bon97]. Les cibles principales de ces attaques introduites dans cette publication étaient dans un premier temps les cryptosystèmes à clef publique et notamment ceux implémentant un RSA [FIP186] utilisant le théorème des restes chinois [Sti96]. D'autres travaux de recherche relatifs aux attaques par injection et analyse de fautes sur le RSA ont également été publiés dans [Yen02, Yen03]. D'autres travaux ont plus spécifiquement portés sur les algorithmes à clef secrète comme par exemple les travaux de Biham et Shamir [Bih97]. En effet, ces derniers ont proposé une attaque par injection de faute appelée DFA (Differential Fault Analysis). L'attaque DFA consiste à analyser les différences entre deux ensembles de résultats de chiffrement dont le premier est correct et le deuxième incorrect et ce en utilisant le même message d'entrée et la même clef. Avec ce procédé, ils ont démontré qu'il est possible de retrouver la clef secrète (dans sa totalité) d'un algorithme DES rien qu'en analysant entre 50 et 200 textes chiffrés. Dans [Bie00], l'attaque DFA a été généralisée à des cryptosystèmes à clef publique notamment les cryptosystèmes à base de courbes elliptiques ou ECC (Elliptic Curve cryptosystem).

D'une efficacité remarquable, les attaques par injection de fautes représentent aujourd'hui une forte menace pour la sécurité des cryptosystèmes (les cartes à puce). Toutefois, la réussite de ces attaques exige des compétences en microélectronique et une connaissance détaillée de la structure interne du circuit (disposition des mémoires, des blocs sensibles, etc.).

I-4-c-2-Attaque par analyse des émissions électromagnétiques

Outre le temps de calcul, la consommation, les fautes, les émissions électromagnétiques des cryptosystèmes sont aussi des signaux compromettants qui peuvent permettre à un tiers malintentionné de retrouver des informations secrètes.

La majorité des cryptosystèmes sont aujourd'hui cadencés par un signal d'horloge qui impose une synchronisation des traitements dans les différentes parties du circuit. Durant chaque cycle d'horloge, de nouvelles données sont stockées dans les registres et de nouveaux calculs sont réalisés. L'existence de ce signal global de synchronisation explique celle de forts appels en courant. Tous ces mouvements de charges électriques sont accompagnés par des rayonnements électromagnétiques. Comme les appels en courant sont étroitement dépendants des données manipulées, une analyse pertinente des émissions électromagnétiques émises par

la puce peut permettre de retrouver des informations secrètes contenues dans la puce. Généralement, on distingue deux types d'émanations électromagnétiques [Arc03]: les émanations électromagnétiques directes et indirectes.

Les émanations électromagnétiques directes résultent des mouvements de charges électriques nécessaires au fonctionnement du circuit. Dans le cas d'un circuit à forte densité d'intégration, la mesure de ces émanations directes peut être très difficile et peut exiger l'utilisation d'une antenne microscopique placée à proximité du composant et de filtres spéciaux permettant de minimiser les interférences. Pour une meilleure qualité de mesures, il est préférable que la couche de passivation soit enlevée.

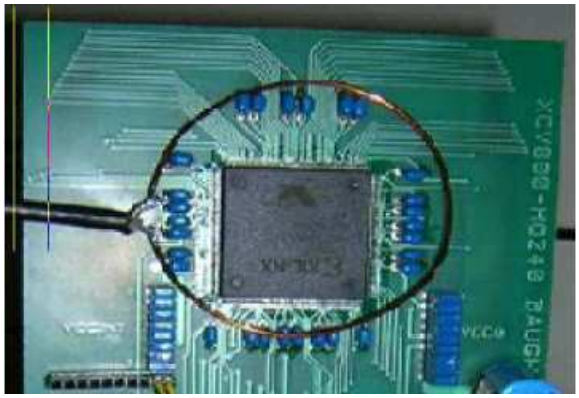


Fig. 7. Capture des émissions électromagnétiques indirectes issues d'un circuit sécurisé

Les émanations électromagnétiques indirectes, quant à elles, sont les résultats de différents couplages électriques et électromagnétiques dus à la proximité des composants. Ces émanations se manifestent alors à travers de phénomènes de modulation d'amplitude, ou de phase ou encore de fréquence de signaux porteurs. Très riche en harmonique, le signal d'horloge est une porteuse de choix lors de l'analyse des émissions. Grâce à une meilleure propagation, la mesure des émanations électromagnétiques indirectes est beaucoup plus aisée que celle des émanations directes. Comme illustré par la figure 7, la mesure des émanations ne nécessite qu'une sonde avec une seule spire placée à distance raisonnable (jusqu'à 30 cm) de la puce.

Dans [Qui00], les émanations électromagnétiques récupérées sont analysées de la même façon que la consommation lors des attaques par analyse de consommation. L'auteur parle alors d'attaque par analyse simple des émissions électromagnétiques (SEMA: Simple ElectroMagnetic Attack) et différentielle des émissions électromagnétiques (DEMA: Differential ElectroMagnetic Attack). Dans [Gan01], Gandolfi, Mourtel et Olivier ont

expérimenté les attaques SEMA et DEMA sur différents algorithmes: DES, RSA, etc. Ils ont notamment démontré la faisabilité des attaques et effectué des comparaisons entre d'une part, les attaques SEMA et SPA et d'autre part, les attaques DEMA et DPA. Malgré l'importance des interférences lors de la phase d'acquisitions des émanations électromagnétiques, les attaques EMA sont aussi efficaces, voire plus efficaces que les attaques SPA et DPA.

Capables d'exploiter les informations locales au niveau d'un circuit, les attaques EMA représentent aujourd'hui une menace réelle contre les crytposystèmes. Cependant, la récupération des émissions électromagnétiques de manière très locale, requièrent des moyens importants (capteur de haute résolution, cage de faraday, table XY, etc.) et une bonne connaissance du circuit cible.

I-5-Attaques matérielles: Contre-mesures

Pour prémunir les cryptosystèmes contre les différentes catégories d'attaques matérielles, les concepteurs de circuits sécurisés ont développé de nombreuses techniques de contre-mesures.

I-5-a-Les contre-mesures contre les attaques invasives

D'une manière générale, les techniques de contre-mesures destinées à prémunir les cryptosystèmes contres les attaques invasives sont basées sur l'utilisation de capteurs, de couches de protection et de procédés de placement routage.

- **Réseaux de capteurs** [Köm99]: Lors de la phase de fabrication de la puce, des couches de métallisation supplémentaires sont déposées sur le module cryptographique pour former un réseau de capteurs. Par ailleurs, toute une circuiterie accompagne généralement le réseau de capteurs pour déclencher la destruction des données sensibles en cas de détection d'une tentative d'intrusion.
- **Couches de protection (Shielding)** [Köm99]: Lors de la réalisation physique des puces, une grille de protection est mise en place sur le dessus du module cryptographique pour empêcher toute attaque par sondage.
- **Placement routage aléatoire** [Sam02]: Appelée aussi brouillage de conception, cette technique de contre-mesure consiste à répartir aléatoirement les éléments de mémorisation dans les blocs combinatoires. L'idée est d'augmenter les difficultés d'identification des registres et des bus de données lors d'attaques par sondage.

D'autre part, cette manière de disposer les mémoires, les bus et d'autres éléments peut complexifier de manière significative les opérations de reconstruction du layout.

I-5-b-Les contre-mesures contre les attaques semi-invasives

I-5-b-1-Contre-mesures contre les attaques par injection de fautes

Les techniques de contre-mesures contre les attaques par injection de fautes se déclinent généralement en deux catégories: les contre-mesures logicielles et les contre-mesures matérielles.

Dans la gamme des contre-mesures logicielles, une technique très simple consiste à vérifier les résultats de calcul [Qui02]. Par exemple, on peut répéter chaque opération et comparer les résultats. Soulignons cependant que cette technique de contre-mesure est très coûteuse en temps de calcul (répétition des calculs) et / ou en surface silicium (doublement du matériel pour effectuer les calculs en parallèle). D'un autre côté, le fait de refaire le calcul peut s'avérer inefficace. En effet, dans le cas d'une faute permanente induite, bien qu'incorrects, les deux résultats seront toujours identiques.

Dans la catégorie des contre-mesures matérielles, Karri [Kar01] propose de rajouter une circuiterie supplémentaire afin de pouvoir effectuer parallèlement le chiffrement et le déchiffrement et comparer les résultats (textes en clair) pour s'assurer qu'aucune erreur ne s'est produite. Toutefois, bien qu'efficace, cette technique est handicapée par un coût en surface exorbitant.

D'autre part, des capteurs sur les signaux d'alimentation, le signal d'horloge ou des capteurs de température peuvent être utilisés pour détecter des modifications anormales des paramètres de fonctionnement du circuit et bloquer les opérations en cours le cas échéant.

Plus récemment, Moore [Moo02] propose l'utilisation de la technologie asynchrone et l'adoption de la logique double rail pour lutter contre les attaques par injection de fautes. L'absence d'horloge globale élimine d'emblée les attaques par variations intempestives du signal d'horloge et le codage double rail offre la possibilité de faire propager "un signal d'alarme" si une erreur est détectée. Un autre gros avantage des circuits asynchrones quasi insensibles aux délais (double rail) est la tolérance aux conditions d'environnement ce qui rendrait les attaques par variations intempestives de l'alimentation et de la température inefficace.

I-5-b-2-Contre-mesures contre les attaques électromagnétiques

Les contre-mesures proposées contre les attaques électromagnétiques se divisent en deux catégories: celles qui consistent à réduire les émanations électromagnétiques et celles qui consistent à les rendre inexploitables.

Dans la première catégorie de contre-mesures, les "fondeurs" de circuits sécurisés rajoutent des couches de métallisation (shielding) supplémentaires au dessus des modules cryptographiques pour limiter le niveau de rayonnement [Gan01]. D'autre part, on envisage même de placer les modules cryptographiques dans une cage de Faraday, mais il n'est pas toujours simple et possible d'en utiliser une [Qui01].

La deuxième catégorie de contre-mesures est celle la plus utilisée par les concepteurs de systèmes sécurisés car elle permet, non seulement de lutter contre les attaques électromagnétiques mais aussi contre les attaques en puissance. En effet, les émanations électromagnétiques ne sont qu'une image des activités électriques du composant. Une contre-mesure largement proposée dans la littérature est l'utilisation de la technologie asynchrone avec le codage double rail [Qui01, Gan01]. L'utilisation de protocole de communication de type poignée de mains (Handshake) permet aux circuits asynchrones de mieux répartir la consommation en courant dans le temps et de réduire considérablement les appels de courant (amplitudes moins importantes) [Bou05].

I-5-c-Les contre-mesures contre les attaques non-invasives

I-5-c-1-Les contre-mesures contre les attaques temporelles

La stratégie globale dans la lutte contre les attaques temporelles est de rendre les temps de calcul des systèmes sécurisés indépendants des données manipulées.

Dans [Koc96], Kocher propose de concevoir les cryptosystèmes de telle sorte qu'ils aient un temps de calcul constant. Bien qu'efficace contre les attaques temporelles, cette contre-mesure dégrade fortement les performances en vitesse des modules cryptographiques, ce qui peut les rendre vulnérable à d'autres attaques. En effet, pour s'assurer un temps de calcul constant, les modules cryptographiques fonctionnent généralement en pire cas. Une autre possibilité pour se prémunir des attaques temporelles est de modifier aléatoirement les variables intermédiaires sans pour autant altérer le résultat du chiffrement. De ce fait, les temps de calcul varieront de manière aléatoire et seront indépendants des données manipulées.

D'autre part, il existe des règles de bon sens qui peuvent aider à se prémunir contre les attaques temporelles comme par exemple d'éviter d'utiliser des instructions ou des branchements conditionnels qui dépendent directement des paramètres de sécurité [Dhe98].

I-5-c-2-Les contre-mesures contre les attaques en puissance

Dans cette partie, nous présentons les contre-mesures contre les attaques en puissance selon deux grandes catégories: contre-mesures logicielles et contre-mesures matérielles.

Dans la catégorie des contre-mesures logicielles, on distingue l'introduction de délais aléatoires (Time randomization) [Dae99], la permutation aléatoire des chemins de données [Gou99] et le masquage des valeurs intermédiaires avec des données aléatoires [Mes00, Cor00, Gou01]. Notez que la majorité de ces méthodes a pour but de réduire le rapport signal sur bruit et rendre ainsi la consommation inexploitable pour une analyse de corrélation. Particulièrement efficace, ces contre-mesures logicielles sont très difficiles à mettre en place et spécifiques à chaque algorithme.

Dans la gamme des contre-mesures matérielles, on distingue l'introduction de bruit dans les mesures de courant à l'aide d'un générateur de nombres aléatoires (RNG) [Dae99], le filtrage du signal d'alimentation [Sha00] et bien d'autres nouveaux styles de conception [Tir03, Mac04, Gui04]. En particulier, ces nouveaux styles de conception utilisent des portes logiques spécifiques (double rail, etc.) dont la particularité est d'avoir une consommation indépendante des données. On notera par ailleurs l'intérêt des architectures reconfigurables (FPGA-SRAM), par exemple pour modifier dynamiquement l'architecture d'un circuit.

L'avantage de ces contre-mesures matérielles est que la susceptibilité des cartes à puce aux attaques en puissance est moins dépendante aux changements d'algorithme/logiciel. Cependant, soulignons que ces techniques de contre-mesure n'éliminent pas la possibilité de mettre en œuvre ces attaques mais les rendent beaucoup plus complexes.

I-6-Classification des attaquants et niveaux de sécurité

Généralement, la mise en place de mécanismes de sécurité doit être précédée par une évaluation des risques de sécurité. Plus particulièrement, il est essentiel de savoir contre qui et contre quoi il faut se protéger.

Dans cette partie, nous nous intéressons aux risques de sécurité des cartes à puce. Nous commençons par présenter une classification des attaquants, puis nous présentons une

classification des niveaux de sécurité

I-6-a-Classification des attaquants

Les risques de sécurité peuvent être évalués selon le niveau de préparation de l'attaque, le temps et les moyens nécessaires. Ainsi, une classification des attaquants a été définie par IBM [Abr91]:

- **Classe I** (Clever outsiders): Cette classe regroupe les attaquants très astucieux, utilisant du matériel obsolète et ayant une connaissance très limitée du système. Ils exploitent uniquement les failles existantes.
- **Classe II** (Knowledgeable insiders): Les attaquants de cette classe sont généralement des spécialistes expérimentés ayant une connaissance parfaite du système visé. Très souvent, ils ont accès à des équipements sophistiqués.
- **Classe III** (Funded organizations): Dans cette classe, les attaquants sont des organisations (Multinationales, pays, etc.) capables de mettre sur pied des équipes de spécialistes dans des domaines très variés et complémentaires. Ces groupes de spécialistes sont capables d'analyser en profondeur le système visé et de concevoir de nouvelles attaques. D'une manière générale, ils ont accès à des équipements ultrasophistiqués et ont des moyens illimités.

I-6-b-Niveaux de sécurité

La norme FIPS140-2 [FIPS140-2] précise les exigences en matière de sécurité qui devront être respectées par le module cryptographique utilisé dans un système de sécurité servant à protéger des informations. La norme fournit quatre niveaux de sécurité:

- **Niveau 1**: C'est le niveau de sécurité le plus bas défini par le standard. Il définit des exigences de sécurité très basiques comme l'utilisation d'un algorithme standard (AES, etc.).
- **Niveau 2**: Outre les exigences du niveau 1, le niveau 2 suppose des mécanismes physiques de sécurité spécifiques. En particulier, une couche de protection doit être placée sur le dessus de la puce afin d'empêcher tout accès physiques non-autorisé.
- **Niveau 3**: Ce niveau de sécurité suppose des mécanismes physiques de sécurité capables de détecter et d'empêcher tout accès aux paramètres de sécurité contenus

dans le module cryptographique. En plus des fonctions de sécurité du niveau 2, un réseau de capteurs peut être rajouté pour déclencher la destruction des paramètres de sécurité en cas d'intrusion.

- **Niveau 4**: C'est le niveau de sécurité le plus élevé défini par la norme FIPS140-2. A ce niveau, le module cryptographique doit être enveloppé par un bouclier de protection dont le rôle est de détecter et de réagir à toute tentative d'accès physique. D'autre part, un module cryptographique de niveau 4 doit être robuste aux modifications anormales des paramètres externes du circuit à savoir: les signaux d'alimentation, le signal d'horloge, la température, etc.

I-7-Conclusion

Dans ce chapitre, nous avons vu qu'une bonne assurance de la sécurité des données passe par l'utilisation d'algorithmes de chiffrement développés par des cryptologues. Nous avons également présenté des techniques cryptanalytiques qui menacent la sécurité des systèmes sécurisés: les attaques logiques et les attaques matérielles. Un bref état de l'art sur les contre-mesures contre les attaques matérielles a également été dressé.

Parmi les attaques matérielles, les attaques en puissance sont considérées comme étant les plus dangereuses. En effet, elles sont particulièrement efficaces et faciles à mettre en œuvre. Du côté des contre-mesures, aucune n'est reconnue comme totalement efficace contre les attaques DPA. C'est pourquoi, dans nos travaux de recherche, nous allons nous focaliser uniquement sur ces attaques DPA. Afin de pouvoir proposer des contre-mesures efficaces, nous nous proposons de les étudier dans le chapitre suivant.

Chapitre II:

Attaque DPA: étude et analyse

A partir d'une analyse différentielle de la consommation, l'attaque DPA est capable d'extraire les informations secrètes contenues dans les composants sécurisés. Pour ce faire, elle exploite les caractéristiques de consommation des circuits CMOS classiques à savoir, la corrélation entre les données manipulées et la consommation. Des contre-mesures ont été proposées pour prémunir les cryptosystèmes contre l'attaque DPA, cependant aucune n'est jusqu'à aujourd'hui reconnue comme étant très efficace.

Chapitre II : Attaques DPA: Etude et Analyse

II-1-Introduction

Les attaques en puissance exploitent les mesures de consommation des circuits sécurisés pour en extraire des informations secrètes comme les clefs cryptographiques. Pour prémunir efficacement les cryptosystèmes contre les attaques en puissance et plus particulièrement contre les attaques DPA, nous allons étudier étudierons ces dernières dans le détail. Dans un premier temps, nous étudierons le(s) fondement(s) des attaques en puissance. Par la suite, un paragraphe sera dédié aux attaques par simple analyse de la consommation. Avant de conclure ce chapitre, nous nous intéresserons à l'analyse différentielle de consommation.

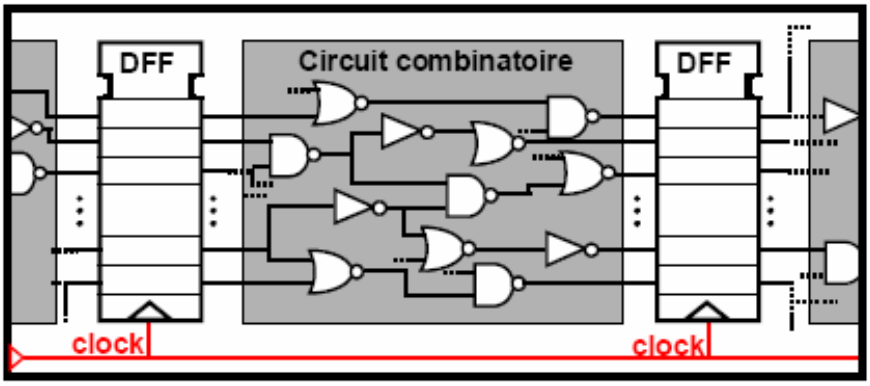


Fig. 8. Schématisation d'un circuit CMOS synchrone

II-2-Fondement des attaques en puissance

Pour comprendre le fondement des attaques par analyse de consommation et plus particulièrement des attaques DPA, il faut se référer aux caractéristiques de consommation de la technologie CMOS.

II-2-a-Sources de dissipation dans les circuits CMOS

Comme l'illustre la figure 8, un circuit CMOS synchrone est typiquement composé de portes logiques et de registres mémoires qui eux-mêmes sont composés de transistors agissant comme des interrupteurs. La puissance consommée par un tel circuit est très dépendante de

l'activité courante de celui-ci. En effet, elle dépend des changements d'état des composants et non des états eux-mêmes [Sze02].

Les sources de dissipation de puissance dans les circuits CMOS sont répertoriées dans deux grandes classes de contribution: les composantes statiques et les composantes dynamiques. Dans le cas d'une porte CMOS, les composantes statiques correspondent à la puissance dissipée lorsqu'elle se trouve dans état d'équilibre. Idéalement, de par sa topologie, la logique statique CMOS devrait avoir une puissance dissipée nulle en l'absence d'activité. Toutefois, les transistors ne sont pas des interrupteurs idéaux. En effet, même bloqués, les transistors ne sont pas complètement fermés laissant ainsi passer des courants de faibles amplitudes.

La puissance dynamique est, quant à elle, la principale source de dissipation dans les circuits CMOS. En effet, cette puissance dynamique est dissipée pendant le changement d'état logique c'est-à-dire lors de la charge et décharge de capacités. Généralement, on admet que la puissance statique est négligeable devant la puissance dynamique lorsque le circuit est en pleine activité comme par exemple lors de l'exécution d'un chiffrement. Par conséquent, il est possible d'exprimer la puissance consommée sur période par une porte CMOS:

$$P = \alpha \cdot C_L \cdot Vdd^2 \cdot F \qquad (1)$$

où α représente le taux d'activité de la porte, C_L la charge, F la fréquence et Vdd la tension d'alimentation. Cette expression (1) peut être extrapolée au niveau d'un circuit complexe. En d'autres termes, nous nous limiterons ici à considérer les charges/ décharges des capacités (Interconnexions, capacités d'entrée des portes, etc.) comme la principale source de consommation.

II-2-b-Analyse des profils de consommation en courant

Les attaques par analyse de consommation exploitent les variations instantanées du courant d'un circuit, nous proposons d'étudier ici les origines des variations fines du profil de consommation en courant des portes logiques.

On sait que les profils de consommation en courant d'une porte CMOS sont très dépendantes des paramètres environnementaux et technologiques [Nik99, Auv00, Mau01] tels que: la rampe d'entrée, la charge (fanout), la topologie, le dimensionnement des transistors, le dessin des masques, etc. Afin d'identifier dans quelle mesure certains de ces paramètres influent sur les profils en courant, nous avons effectué une campagne de simulation électrique sur des circuits CMOS de complexité différente et dimensionnés selon une technologie 130 nm.

- Influence de la charge et de la rampe d'entrée

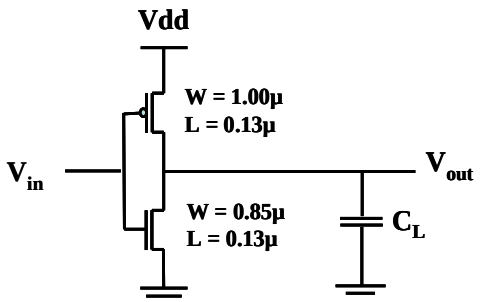


Fig. 9. Inverseur CMOS dimensionné dans une technologie 130 nm

Afin de vérifier l'impact de la charge sur les profils en courant, nous avons simulé l'inverseur CMOS de la figure 9 soumis à une conditions de contrôle constante (τ_{in} = 100 ps) mais à des conditions de charges différentes [4 fF – 160 fF]. Dans un deuxième temps, et ce afin d'observer l'effet de la rampe d'entrée sur les profils en courant, nous avons soumis l'inverseur à une charge constante (40 fF) et à différentes valeurs de rampe d'entrée [10 ps – 600 ps].

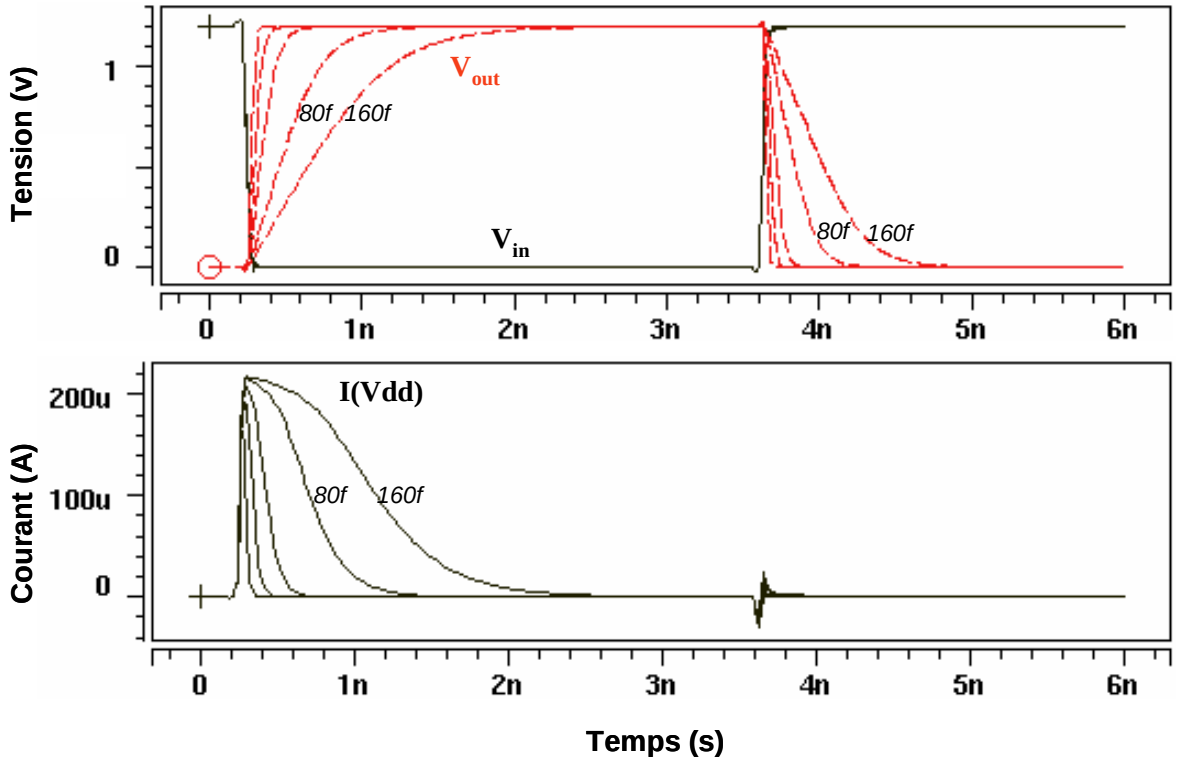


Fig. 10. Evolution du courant avec la valeur de la capacité de charge C_L

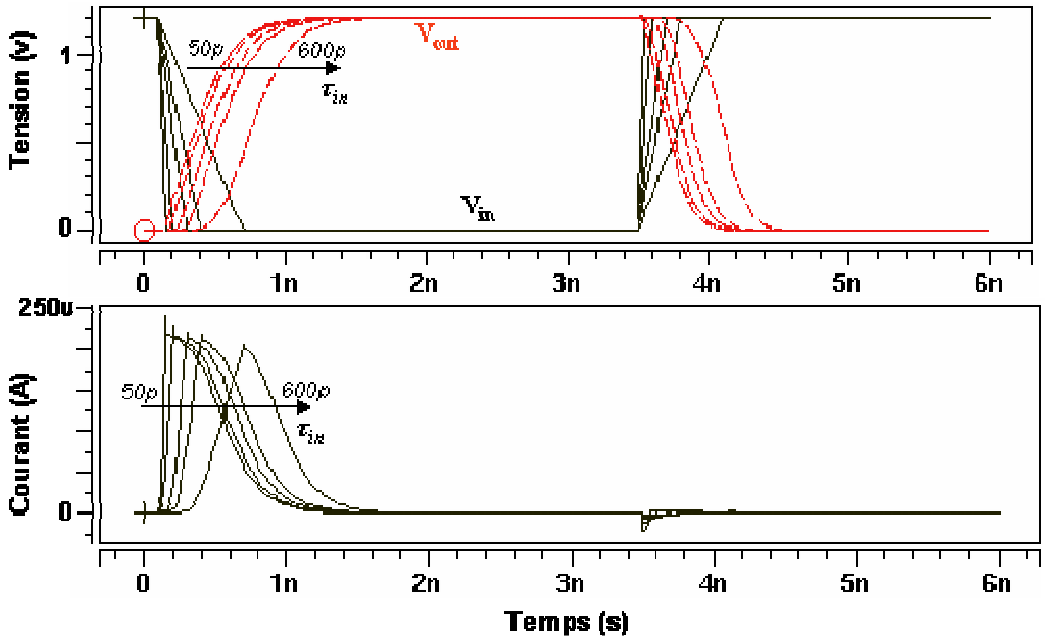


Fig. 11. Evolution du courant avec la valeur de la rampe d'entrée τ_{in}

Les résultats de simulation sont présentés dans les figures 10 et 11. L'analyse de ces résultats nous a permis de dégager les conclusions suivantes:

- Les transitions [0-0] et [1-1] en sortie de l'inverseur ont une consommation nulle et ne génèrent aucun appel de courant sur les rails d'alimentation. Les transitions [0-1] et [1-0] induisent quant à elle la circulation de courants d'amplitudes plus ou moins importantes. Toutefois, vu depuis le rail d'alimentation V_{DD}, les transitions [0-1] induisent des appels en courant nettement supérieur aux transitions [1-0].

- Le profil en courant d'un inverseur dépend fortement de la valeur de C_L. La figure 10 représente l'évolution temporelle du courant de charge et de décharge pour différentes valeurs de C_L. Pour des valeurs plus faibles de C_L, l'amplitude maximale du courant est proportionnelle à la racine carrée la charge. Toutefois, à partir d'une certaine valeur de C_L, l'amplitude maximale du courant ne dépend plus de la valeur de C_L. A ce stade, la valeur maximale du courant est limitée par les possibilités en courant du transistor en conduction. Par conséquence, les profils en courant ont tendance à s'étaler dans le temps et ce de manière proportionnelle à la valeur de la charge.

- Les profils en courant sont très sensibles aux variations de la rampe d'entrée.

Comme le montre la figure 11, l'amplitude maximale du courant est inversement proportionnelle à la valeur de la rampe d'entrée. D'autre part, une augmentation de la valeur de la rampe d'entrée se traduit par un décalage temporel de la réponse de l'inverseur et par conséquent, des profils en courant.

- Influence des réseaux série/parallèle de transistors

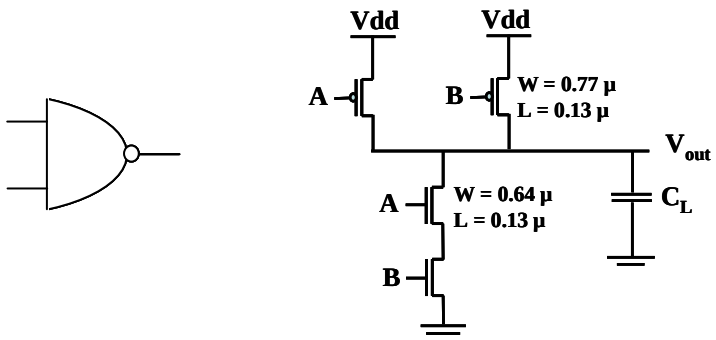


Fig. 12. Schématique d'une porte NAND2

Afin d'évaluer l'impact des réseaux série/parallèle de transistors sur les profils en courant, nous allons considérer la porte NAND2 de la figure 12. Durant les simulations, les conditions de contrôle (τ_{in} = 100ps) et de charges sont constantes (40fF). Les paramètres qui varient sont l'ordre d'arrivée des signaux et le vecteur d'entrée.

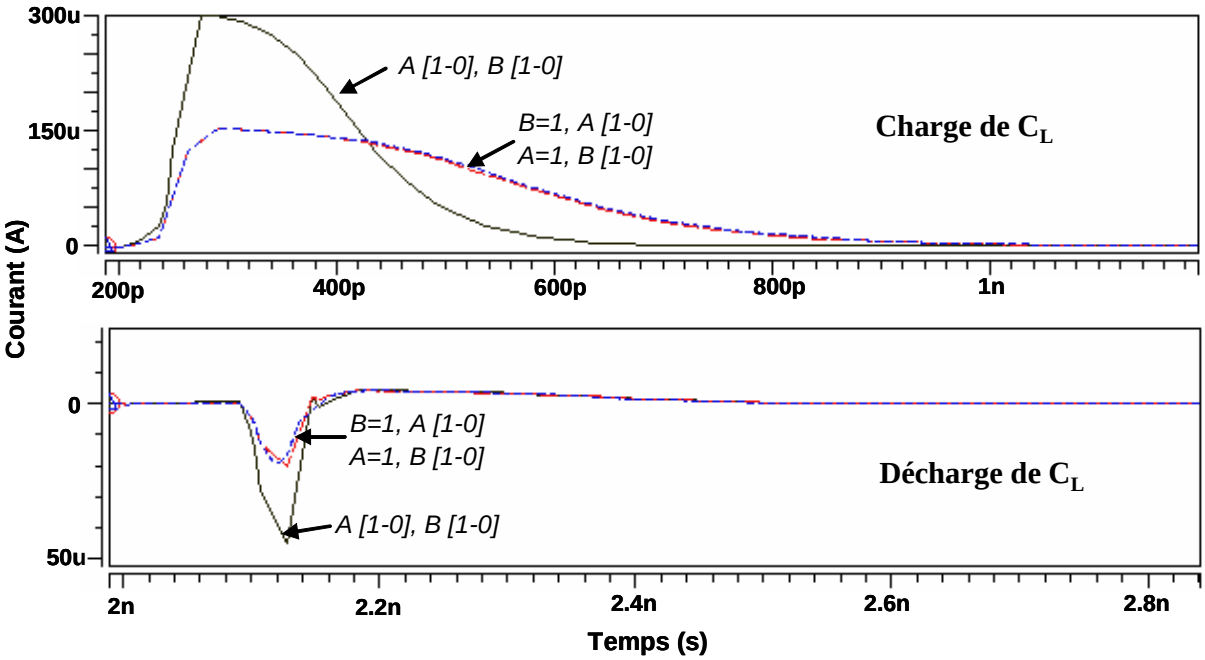


Fig. 13. Evolution du courant selon les vecteurs d'entrée

La figure 13 présente l'évolution du courant d'une porte NAND2 en fonction des vecteurs

d'entrée et l'ordre d'arrivée des signaux. L'amplitude maximale du courant est proportionnelle au nombre de transistors en commutation. D'un autre côté, comme les délais de propagation de la porte, la durée dans le temps des profils en courant est très sensible aux vecteurs d'entrée.

- Influence de l'activité du circuit – Poids de Hamming

Afin d'observer l'impact de l'activité d'un circuit sur les profils en courant, nous avons simulé le groupe de portes NAND2 de la figure 14a. Les conditions de contrôle (τ_{in} = 100ps) et de charges (C_L=20f) sont restées inchangées. Seul, le taux d'activité du circuit a varié.

Par définition, le taux d'activité d'un circuit représente la probabilité de transition des portes à chaque période de l'horloge. Dans notre cas d'étude, nous avons supposé que les simulations se déroulent durant une période d'horloge et qu'à t = 0, toutes les sorties sont à '0'. Dans ces conditions, le taux d'activité n'est autre que le nombre de portes en commutation sur le nombre total de portes et le nombre de portes en commutation représente le poids de hamming de la donnée calculée.

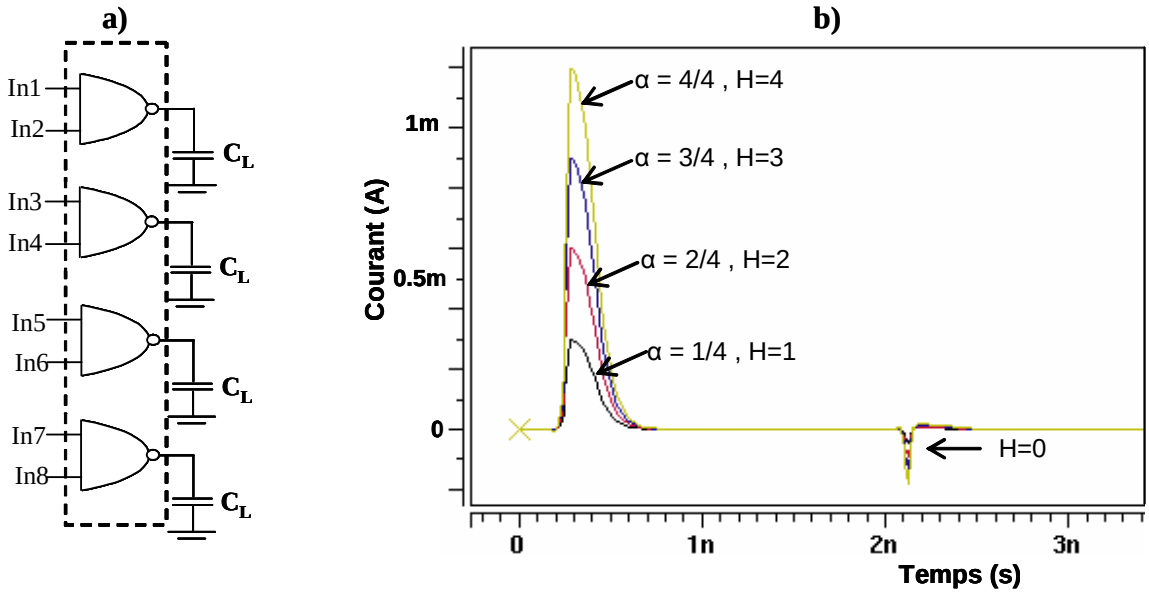


Fig. 14.a) Groupe de portes NAND2, b) Evolution du courant en fonction du taux d'activité α

La figure 14b présente l'évolution du courant en fonction de l'activité du circuit. Comme attendu, plus l'activité du circuit est importante, plus l'amplitude maximale du courant est importante. En d'autres termes, l'amplitude du courant d'un circuit est proportionnelle au poids de hamming de la donnée calculée.

II-2-c-Conclusion

Dans ce paragraphe, nous avons montré par simulation que les profils en courant d'un circuit CMOS sont très dépendants de la valeur de la charge, de la valeur de la rampe d'entrée, du type de transition effectuée, de l'activité du circuit ou encore du poids de hamming de la donnée traitée. Par voie de conséquence, une analyse intelligente des profils en courant peut permettre de retrouver les données manipulées. Prenons le cas de la figure 14b, une simple observation des formes d'onde en courant permet de conclure sur les données calculées. Dans le domaine de la cryptanalyse, on parle alors d'attaque par analyse simple de consommation (SPA).

Du point de vu de la sécurité, il est maintenant clairement établi que le fondement des attaques par analyse de consommation est la dépendance entre les données manipulées et les profils en courant.

II-3-Attaque par analyse simple de la consommation (SPA)

Introduite par Paul Kocher [Koc99], l'attaque SPA est une interprétation directe des traces de courant mesurés lors des calculs cryptographiques. Plus précisément, elle permet d'établir des corrélations entre les profils en courant et les données manipulées. Ainsi, par une simple observation des profils en courant, l'attaque SPA peut permettre d'identifier les instructions en cours d'exécution ou encore de retrouver des clefs cryptographiques.

La figure 15 présente les profils en courant récupérés d'une carte à puce lors d'un traitement DES. Les profils en courant correspondants : à la permutation initiale, aux 16 rondes et à la permutation finale sont clairement identifiables. Dans ce cas précis, l'attaque SPA a permis d'identifier les différentes instructions et leurs instants d'occurrence.

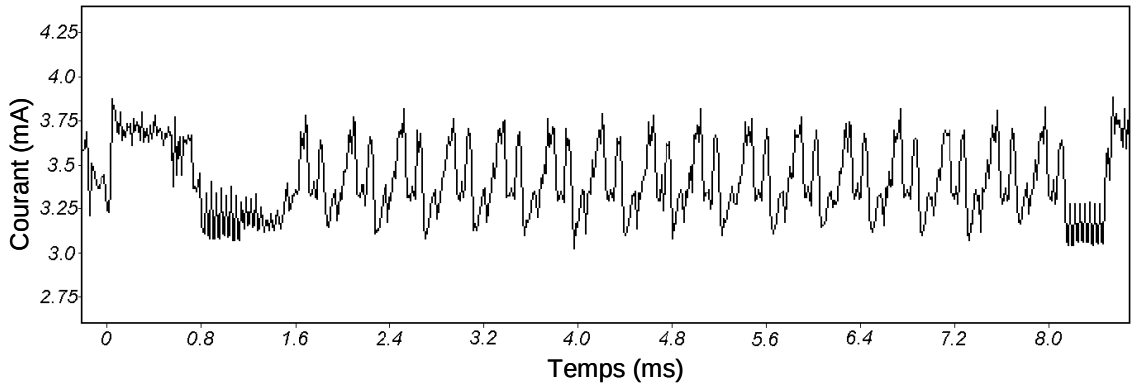


Fig. 15. Profils en courant correspondant à un traitement DES

D'autre part, les profils en courant d'une instruction sont très dépendants des données manipulées. Ceci s'explique par le fait que le chemin des données est très dépendant des données traitées. Dans [Koc99a], un zoom sur les rounds 2 et 3 a permis d'identifier les profils en courant des blocs de génération de sous-clefs du cryptosystème DES et de conclure sur les différentes opérations de modification (permutations et décalages à gauche) auxquelles la clef a été soumise. Par ailleurs, à chaque opération de décalage, un test sur la valeur d'un bit de la clef est effectué. Selon que le bit de test est un '1' ou un '0', les profils en courant du branchement conditionnel diffèrent légèrement. Par conséquent, l'attaquant est en mesure de conclure sur la valeur d'un certain nombre de bits de la clef secrète.

Notez que toute implémentation naïve d'algorithme de chiffrement (TDES, AES, RSA, etc.) [Bih99, Gem01] est vulnérable aux attaques SPA. Par ailleurs, la réussite de ces attaques exige une connaissance détaillée de l'algorithme de chiffrement et de la manière dont il est implémenté.

Aujourd'hui, avec l'apparition de nouvelles techniques de contre-mesure [Mes00, Dae99], l'attaque SPA ne représente plus à elle seule une menace contre les cryptosystèmes. Par contre, nous verrons un peu plus tard qu'elle peut servir d'étape préliminaire à l'attaque DPA.

II-4-Attaque par analyse différentielle de consommation (DPA)

Contrairement à l'attaque SPA, l'attaque par analyse différentielle de consommation ou DPA est beaucoup plus efficace et difficile à contrecarrer [Koc99, Mes99]. D'une part, elle ne nécessite pas de connaître dans les détails l'implémentation du cryptosystème et d'autre part, par des analyses statistiques, elle est capable d'exploiter les plus petites variations du courant pour retrouver des informations secrètes contenues dans les cartes à puce comme la clef privée RSA ou la clef de chiffrement de l'algorithme DES.

Globalement, la stratégie de l'attaque DPA est d'éliminer toute information inutile (bruit de mesure, consommation des parties non sensibles du cryptosystème, etc.) et d'amplifier celle corrélée avec les données secrètes.

L'attaque DPA se déroule en deux étapes: la collection des données et l'analyse des données. L'étape de collection des données consiste à récupérer les couples [cryptogramme – consommation]. L'analyse des données consiste en une analyse statistique de la consommation mesurée durant les calculs cryptographiques. Pour illustrer ces différentes étapes, nous considèrerons le cas d'une attaque DPA sur un cryptosystème DES.

- Étude préliminaire

Pour rappel, le DES est un algorithme de chiffrement par bloc qui repose sur des principes simples dont des permutations, des substitutions, des échanges de blocs de données et une fonction prenant en entrée une clef intermédiaire à chaque round. Chaque clef intermédiaire est calculée à partir de la clef de chiffrement de 64 bits. Dans le cas d'une attaque sur texte chiffré seulement, il est important de noter que pour retrouver la totalité de la clef de chiffrement du DES, il faut au moins retrouver les clefs intermédiaires des deux dernières rondes [Sti01, Sch01].

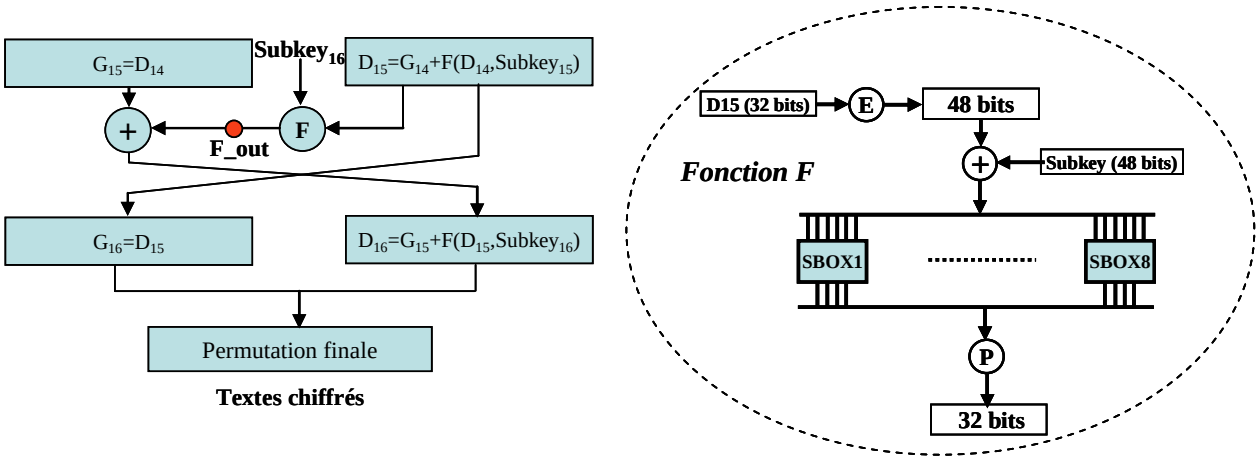


Fig. 16. La dernière ronde de l'algorithme DES et la fonction F

La figure 9 présente la dernière ronde et la fonction F de l'algorithme DES. Si on regarde de près cette dernière ronde et si on a accès aux textes chiffrés, finalement la seule inconnue est la clef intermédiaire $Subkey_{16}$. En ce qui concerne le message D16 (32 bits), les propriétés de la fonction « ou-exclusif » font qu'il peut être fonction de la clef intermédiaire ($subkey_{16}$) et du message chiffré (D_{15} et D_{16}):

$$\mathbf{G_{15} = D_{16} \text{ xor } F_out = D_{16} \text{ xor } F(D_{15}, Subkey_{16})} \qquad (2)$$

Notion de fonction de sélection

Comme définie dans [Koc99], une fonction de sélection est une fonction qui calcule une valeur intermédiaire de l'algorithme DES en fonction d'une partie de la clef intermédiaire et d'une partie du message chiffré. L'expression (2) est une illustration parfaite de ce que c'est une fonction de sélection.

Toutefois, pour des raisons de simplification et d'efficacité des analyses, il est préférable de définir une fonction de sélection ciblant un bloc moins important mais tout aussi sensible de

l'algorithme DES. Dans notre cas d'étude par exemple, nous considèrerons une fonction de sélection D mettant en jeu la première table de substitution (SBOX1), 6 bits de la clef intermédiaire (k) et 6 bits du message chiffré (m):

$$\mathbf{D(m,k) = SBOX1(m \text{ xor } k)} \qquad (3)$$

Soulignons qu'une fonction de sélection est utilisée lors de l'analyse des données pour effectuer les hypothèses de clef et pour vérifier si une hypothèse de clef est la bonne. Avec la bonne hypothèse de clef, la fonction de sélection calculera la valeur intermédiaire correcte de D (4 bits) et ce, pour chaque message chiffré m. En d'autres termes, la fonction de sélection sera corrélée avec chaque bit de D manipulé au niveau du 16ème round du DES. Etant donné la dépendance entre la consommation et les données manipulées, des analyses de corrélation entre chaque bit de D et la consommation du circuit (mesurée durant le 16ème round et avec l'utilisation de la vraie clef k) permet alors de vérifier si une hypothèse de clef est la bonne.

- La collection des données

Pour cette phase, il faut impérativement avoir accès au cryptosystème et mesurer de manière précise sa consommation en fonction du temps. Dans le même temps, il faut récupérer les cryptogrammes disponibles en sortie du circuit. Pour les analyses de corrélation, il faut, en effet, un nombre arbitrairement élevé de couples [cryptogramme – consommation]. La figure 17 présente une plateforme utilisée pour automatiser le processus d'acquisition de ces informations. Généralement, la consommation en courant est mesurée aux bornes d'une petite valeur de résistance [1Ω-5Ω] insérée entre la masse de la puce et celle du dispositif d'acquisition des données. De même, il est possible d'effectuer des mesures de courant à travers l'alimentation (V_{DD}) de la puce.

Pour améliorer la qualité des mesures, des filtres de bruit et des amplificateurs d'instrumentation peuvent être rajoutés dans la chaîne d'acquisition.

Pour faciliter l'analyse des données, les cryptogrammes et les traces de courant doivent être mémorisés dans l'ordinateur plus généralement sous forme matricielle. D'autre part, pour économiser les ressources mémoire, il est nécessaire de délimiter la zone d'acquisition des traces de courant. Dans le cas d'une attaque DPA sur texte chiffré seulement, seules les traces de courant correspondantes au 16ème round sont nécessaires. Cette zone peut être aisément identifiée par une analyse SPA.

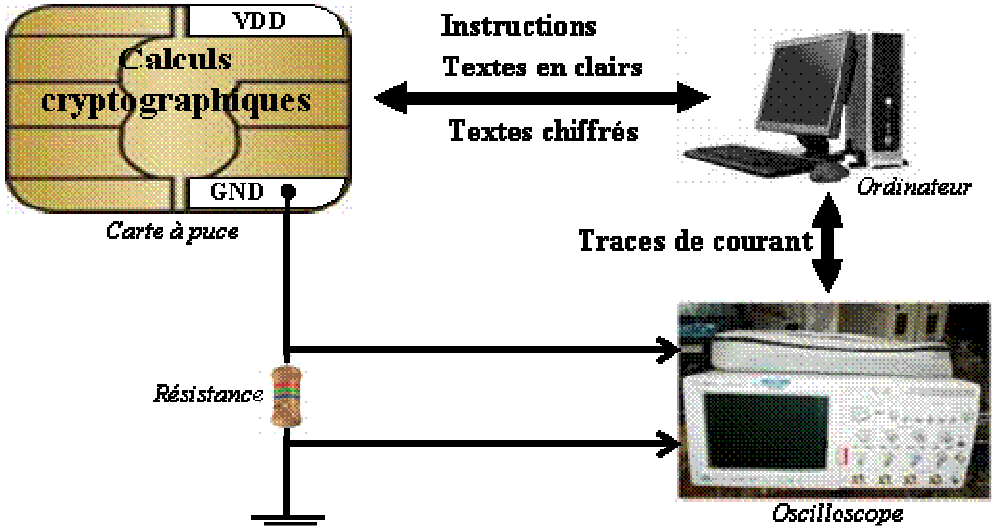


Fig. 17. Plateforme d'acquisition automatique des traces de courant et des cryptogrammes

En l'absence de circuit test, nous avons procédé à des attaques DPA par simulation sur un cryptosystème DES. Pour ce faire, nous avons utilisé une technologie CMOS 130nm et le simulateur électrique "Eldo" (Mentor). Toutefois pour des raisons de temps de calcul, il n'est pas raisonnable de simuler l'intégralité du cryptosystème. Nous nous sommes donc contentés de simuler le bloc cible de notre fonction de sélection (3) présenté sur la figure 18.

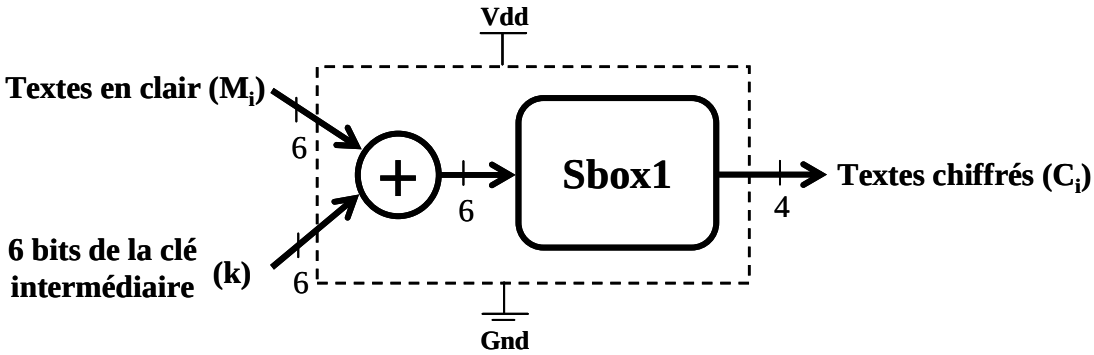


Fig. 18. Sous circuit de la fonction F

Dans un premier temps, nous avons développé la description VHDL de ce sous circuit. Après une vérification fonctionnelle, nous avons procédé à la synthèse logique en utilisant "Ambit", le synthétiseur de Cadence. La bibliothèque cible est une bibliothèque (Technologie 130 nm) de cellules logiques dont on connaît la description Eldo.

Le résultat obtenu est du Verilog structurel qui correspond à l'instanciation de toutes les cellules utilisées et des fils permettant de les relier. Comme un fichier Verilog n'est pas utilisable pour une simulation électrique, nous avons développé un script "Perl" pour convertir du Verilog structurel en netlist Eldo.

La netlist Eldo étant prête, nous avons lancé le chiffrement de n messages aléatoires (M_i) avec la même clef k. Pour chaque message M_i, nous avons mémorisé les traces de courant dans le temps T_{ij} (j indique le numéro de l'échantillon) et le cryptogramme obtenu C_i (4 bits). Comme prévu, les données ainsi collectées ont été mémorisées sous forme matricielle:

$$\underbrace{\begin{pmatrix} M_1 & C_1 \\ \cdot & \cdot \\ M_i & C_i \\ \cdot & \cdot \\ M_n & C_n \end{pmatrix}}_{\textbf{Messages}} \underbrace{\begin{pmatrix} T_{11} & \cdot & \cdot & T_{1j} & \cdot & \cdot & T_{1k} \\ T_{i1} & \cdot & \cdot & T_{ij} & \cdot & \cdot & T_{ik} \\ T_{n1} & \cdot & \cdot & T_{nj} & \cdot & \cdot & T_{nk} \end{pmatrix}}_{\textbf{Traces de courant}}$$

- Analyse des données

Concrètement, cette étape de la DPA consiste à effectuer des hypothèses de clef et à déterminer la bonne hypothèse de clef. Pour arriver à déterminer la bonne hypothèse de clef, on se base sur la dépendance entre les données traitées et les traces de courant.

Considérons la fonction de sélection représentée par l'expression (3). Pour chaque hypothèse de clef k_s (s = [1-64]), elle donnera une valeur théorique notée D_i (4 bits) du cryptogramme obtenu et ce pour chaque message en clair M_i. Pour commencer l'analyse des données, on répartit les traces de courant T_{ij} en deux ensembles en fonction de la valeur du bit b de D_i appelé communément le bit cible. On constituera alors deux ensembles T_0 et T_1:

$$T_{ij} \in T_0 \left| D_i(M_i, k_s)[b] = 0 \right. \quad (4)$$

$$T_{ij} \in T_1 \left| D_i(M_i, k_s)[b] = 1 \right. \quad (5)$$

Pour chaque hypothèse de clef, on calcul M_0 et M_1, la moyenne respective de l'ensemble T_0 et de l'ensemble T_1.

$$M_{0\,j\in[1-k]} = \frac{\sum_{i=1}^{n}(1 - D_i(M_i, k_s)[b]) \cdot T_{ij\in[1-k]}}{n - \sum_{i=1}^{n}(D_i(M_i, k_s)[b])} \quad (6)$$

$$M_{1\,j\in[1-k]} = \frac{\sum_{i=1}^{n}(D_i(M_i, k_s)[b]) \cdot T_{ij\in[1-k]}}{\sum_{i=1}^{n}(D_i(M_i, k_s)[b])} \quad (7)$$

Par la suite, on détermine la différence de ces deux moyennes qui représente la signature DPA notée $S_{K_s}^{DPA}$.

$$S_{K_s}^{DPA} = M_1 - M_0 \qquad (8)$$

Pour la suite de notre analyse, référons-nous à la forme matricielle des données mémorisées. Si l'hypothèse de clef K_s n'est pas bonne, le bit b de D_i calculé par la fonction de sélection sera égal au bit b de C_i (donnée réellement manipulée par le circuit) avec une probabilité de 1/2 pour chaque message M_i. Vu le caractère aléatoire de la fonction de sélection, les traces de courant occasionnées par des transitions de différent type ([0-1] et [1-0]) peuvent se retrouver dans un même ensemble (T_0 ou T_1) et ce, dans une même proportion. Par voie de conséquence, sur un grand nombre d'échantillons n, il est fort probable que les deux moyennes M_0 et M_1 soient égales et que la courbe DPA soit plate et proche de zéro.

D'un autre côté, si l'hypothèse de clef K_s est bonne, le bit b de D_i calculé par la fonction de sélection sera égal au bit b de C_i avec une probabilité de 1. Dans ce cas, la fonction de sélection sera corrélée à la valeur du bit b de C_i manipulé par le circuit. Par conséquent, les traces de courant occasionnées par les transitions d'un même type ([0-1] ou [1-0]) se retrouveront dans un même ensemble (T_0 ou T_1). Sachant que les transitions [0-1] consomment, vu depuis V_{DD}, plus que les transitions [1-0], la courbe DPA sera plate et proche de zéro aux instants où la fonction de sélection ne sera pas corrélée avec le bit cible et affichera des pics de forte amplitude dans le cas contraire.

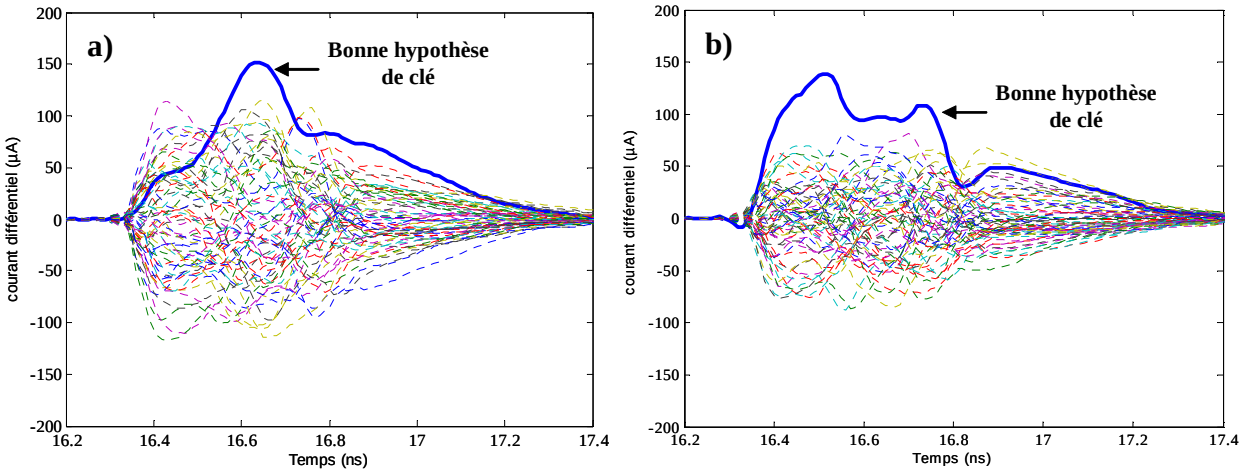


Fig. 19.a) Courbes DPA obtenues selon D_i[2], **b)** Courbes DPA obtenues selon D_i[3]

Dans notre cas d'étude, nous aurons 64 courbes DPA correspondantes aux 64 hypothèses de

clef (6 bits). La figure 19 présente les courbes DPA obtenues selon le bit cible $D_i[2]$ et le bit $D_i[3]$. Pour chacun de ces deux bits cibles, la courbe DPA qui s'écarte le plus de la moyenne et ce de manière notable est bien celle correspondante à l'utilisation de la bonne hypothèse de clef. Avec deux bits cibles donnant le même résultat juste, on peut conclure que l'attaque DPA a été menée avec succès.

Soulignons cependant qu'il peut arriver qu'une mauvaise hypothèse de clef génère des pics de courant bien plus importants que ceux générés par la bonne hypothèse de clef. On parle alors de pics fantômes généralement occasionnés par le bruit. Ceci peut s'expliquer par le fait qu'il n'y ait pas assez d'échantillons pour pouvoir éliminer le bruit par moyennage. Une solution est de prendre un nombre suffisamment important d'échantillons.

D'autre part, il existe une autre version plus efficace de l'attaque DPA qui est "l'attaque DPA de second ordre" ou le "High-Order DPA" [Koc99a]. Si précédemment les analyses de la consommation sont effectuées selon la valeur d'un bit manipulé par le composant, avec la l'attaque DPA de second ordre, elles peuvent être effectuées selon la valeur de plusieurs bits manipulés par le composant. Par exemple, la répartition de la consommation peut se faire selon la valeur de quelques (2, 3) bits de la fonction de sélection. Par ailleurs, lors d'une attaque DPA de second ordre, des signaux de sources différentes (Emissions Electromagnétiques, consommation en courant) et des signaux obtenus selon différentes techniques de mesure peuvent être combinés lors des analyses des données.

II-5-Contre-mesures

Il existe trois types d'approche dans la définition de contre-mesures contre les attaques en puissance. Une première approche consiste à introduire du bruit dans les mesures de courant [Dae99, Gou99]. L'idée est de réduire le rapport signal sur bruit pour rendre les mesures de courant inexploitables. Une deuxième approche consiste à "masquer" les valeurs intermédiaires [Mes00, Cor00] pour faire en sorte que les analyses de corrélation soient impossibles. Enfin, pour que l'attaque DPA soit impossible, une troisième approche consiste à avoir une consommation équilibrée/constante [Sha00, Tir03, Mac04, Gui04].

II-5-a-Méthodes par introduction de bruit

L'introduction de bruit a pour but de rendre la consommation complètement inexploitable par les attaques en puissance [Dae99]. Le bruit peut être introduit soit à l'aide d'un jitter sur les signaux d'alimentation ou le signal d'horloge, soit en utilisant un générateur de nombre aléatoire (RNG). L'idée est de rendre l'exécution de certaines instructions, l'écriture et la lecture des registres complètement aléatoire [Mul01] sans pour autant perturber les résultats des opérations cryptographiques.

Ces méthodes par injection de bruit agissent sur le profil en courant des cryptosystèmes et plus particulièrement sur les amplitudes. Par exemple, en rendant l'occurrence des données aléatoires, on diminue le nombre de portes en commutation à chaque instant. Tout naturellement, ceci a pour conséquence de réduire l'amplitude du courant à chaque instant. En terme de sécurité, l'introduction de bruit permet de réduire le rapport signal sur bruit ce qui peut davantage la complexité des analyses de corrélation et donc des attaques DPA.

D'autre part, ce type de contre-mesure rend très difficile la synchronisation pour les acquisitions et le traitement des courbes.

II-5-b-Méthodes de masquage

Les méthodes de masquage consistent à dissimuler les données manipulées de telle sorte qu'il ne soit plus possible d'établir des corrélations avec le courant consommé. Ces techniques n'agissent pas directement sur le profil en courant mais sur la nature des données manipulées lors des calculs cryptographiques.

Soit la fonction de sélection **D** dépendant de la donnée **m** et d'une partie de la clef **k**: **D(m,k)**. Le principe de la protection est de remplacer la valeur de **m** par une valeur **m'** dépendante de **m** et d'une valeur aléatoire **r**. Ainsi, lors des opérations cryptographiques, au lieu de calculer **D(m,k)** on calcule **D(m',k)** sur laquelle il est impossible de faire des hypothèses de répartition car on ne connaît pas la valeur de **m'**. Par ailleurs, le masque peut aussi être appliqué sur la valeur de **k** telle que **D** soit fonction de **m'** et **k'**: **D(m',k').**

La nouvelle valeur de **m'** est calculée en fonction de l'algorithme par des opérations dites de masquage. Elle doit pouvoir d'une part conserver les propriétés de l'algorithme, et d'autre part elle doit permettre de retrouver le résultat initial. L'opération de masquage est alors très dépendante de l'algorithme considéré [Mes00, Cor00].

En résumé, l'objectif de cette approche est donc de masquer aléatoirement les données d'entrées et les clefs dans les opérations de chiffrement pour empêcher toute analyse de corrélation.

Cependant, les protections par masquage restent vulnérables faces à aux attaques DPA de second ordre [Mes00a]. En effet, il est toujours possible d'établir une corrélation entre la donnée masquée **m'** et la valeur du masque utilisé **r** par analyse du courant lors des opérations de masquages. Pour plus d'efficacité, ce type de contre-mesure doit être associé aux méthodes par introduction de bruit [Akk01].

II-5-c-Méthodes par lissage et uniformisation du profil de courant

Les techniques qui agissent sur le courant du composant sont basées sur l'utilisation de filtres de courant sur l'alimentation et sur des procédés de découplage des alimentations. L'usage du filtre inséré entre les plots d'alimentation et le cœur de la puce permet de lisser le profil de courant [Rak01]. D'autre part, les techniques de découplages de l'alimentation externe et interne d'un composant à l'aide des capacités de découplages [Sha00] permettent d'uniformiser le profil en courant externe (Figure 20).

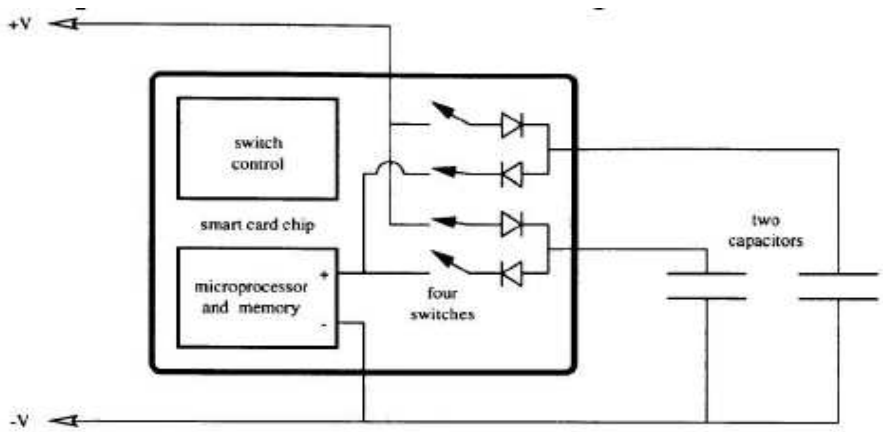


Fig. 20. Technique de découplage de l'alimentation

Au niveau porte logique, les approches sont focalisées sur la suppression des dissymétries observées lors de la charge et décharge d'une porte. Dans [Tir02], Tiri propose l'utilisation de portes logiques spécifiques de type SABL (Sense Amplifier Based Logic) pour l'implémentation de circuits sécurisés. Par ailleurs, il présente, dans [Tir04], un flot de conception sécurisé basé sur les propriétés des portes SABL. Typiquement, ces portes ont été conçues de telle manière à ce qu'elles consomment toujours la même quantité d'énergie durant

un cycle de calcul (Précharge + Evaluation). Toutefois, la suppression des dissymétries observées lors de la charge et décharge d'une porte est conditionnée par le fait qu'aucune dissymétrie ne doit être introduite au niveau des interconnexions différentielles lors de la phase de placement routage.

II-6-Conclusion

Dans ce chapitre, nous avons mené une étude détaillée sur les attaques en puissance et plus particulièrement sur l'attaque DPA. Nous avons clairement identifié le fondement des attaques en puissance qui est la dépendance de la consommation des cryptosystèmes aux données traitées. Nous avons également identifié les paramètres pouvant influer sur les formes d'onde en courant.

Par ailleurs, nous avons présenté l'attaque SPA qui, par une simple observation des profils en courant, permet de déduire les instructions en cours de traitement et les données manipulées. Cependant, la réussite de l'attaque SPA est conditionnée par la connaissance de l'algorithme considéré et de la manière dont il est implémenté. D'autre part, nous avons étudié dans les détails l'attaque DPA. Bien plus efficace que l'attaque SPA, elle exploite les plus petites variations de consommation par des moyens statistiques pour retrouver les données secrètes. Les différentes étapes de l'attaque ont été présentées et illustrées par des attaques par simulation sur une implémentation de l'algorithme de chiffrement DES. Les résultats obtenus confortent clairement l'efficacité remarquable de l'attaque DPA.

Nous avons également dressé un bref état de l'art sur les contre-mesures. Excepté les contre-mesures centrées sur la conception de portes logiques spécifiques qui permettent d'agir sur les dissymétries au niveau électrique, l'ensemble des contre-mesures proposées ne supprime pas l'origine des fuites d'informations mais permet de les rendre quasi-inexploitables en minimisant la corrélation entre les données et la consommation.

Chapitre III:

La logique double rail: une contre-mesure à la DPA?

Il est maintenant clairement établi que l'attaque DPA est rendue possible par le fait qu'il y a une dépendance étroite entre les données manipulées et la consommation. Dans ce contexte, la logique double rail apparaît comme une solution intéressante dans la mesure où le codage associé offre la possibilité d'équilibrer la consommation et d'éviter ainsi toute corrélation entre les données traitées et la consommation. Au niveau physique, des implantations de cellules double rail ont été proposées cependant chacune d'entre elles présente des handicaps majeurs tels qu'une consommation très élevée, un coût excessif en surface, etc. Dans ce contexte, nous proposons dans ce chapitre, une méthode de conception de cellules double rail dont la principale caractéristique attendue est un bon compromis entre le niveau de robustesse à l'attaque DPA et la surface.

Chapitre III : La logique double rail, une contre mesure à la DPA?

III-1-Introduction

Dans le chapitre précédent, nous avons longuement étudié l'attaque différentielle en puissance et son application en vue d'obtenir la clef de chiffrement des circuits sécurisés. La facilité à mettre en œuvre cette attaque met en évidence que le talon d'Achille des systèmes de chiffrement réside dans leur implantation matérielle dont la consommation, ainsi que d'autres syndromes physiques, sont très dépendants des données manipulées.

Outre l'étude de l'attaque DPA, dans le chapitre II, nous avons vu que des contre-mesures ont été proposées dans la littérature tant au niveau algorithmique (méthode de duplication, de masquage, etc.) qu'au niveau implémentation physique (injection de bruit, réduction des pics en courant, etc.). Cependant aucune n'est jusqu'à aujourd'hui reconnue comme étant très efficace.

Partant de ce constat, des efforts significatifs de recherche ont été consentis afin de rendre les attaques DPA, et plus généralement les attaques par canaux cachés, inopérantes. Parmi les contre-mesures, on peut distinguer les attaques visant à masquer ou à noyer dans du bruit les syndromes exploités par ces attaques de celles visant à réduire autant que possible ces derniers. La logique double rail a récemment été identifiée [Tir02, Bou05a, Gui04, Kul05] comme une contre-mesure permettant de réduire très significativement le syndrome exploitée par l'attaque DPA. L'étude de cette contre-mesure constitue l'objet de ce chapitre.

III-2- Pourquoi la logique double rail ?

Nous allons dans un premier temps préciser quel est le syndrome physique qu'exploite l'attaque DPA et puis apporter des éléments de réponse aux deux questions suivantes: pourquoi la consommation des cryptosystèmes est dépendante des données traitées et comment réduire cette dépendance entre les profils en courant et les données manipulées.

III-2-a- Le syndrome DPA

Dans le chapitre II, nous avons étudié l'attaque DPA et démontré que le syndrome $S_{DPA}(Z)$ que révèle cette analyse différentielle s'exprime au premier ordre selon l'expression suivante :

$$S_{DPA}(Z) = \frac{1}{T} \cdot \sum_{v=1}^{T} I_V(t) - \frac{1}{F} \cdot \sum_{w=1}^{F} I_W(t) \qquad (9)$$

$I_V(t)$ et $I_W(t)$ sont respectivement les profils en courant du bloc combinatoire, constitué de p portes logique, lorsque les vecteurs v et w appartenant respectivement aux ensembles de vecteurs T et F qui forcent la valeur du bit attaqué à '1' ou à'0' respectivement. Si cette dernière expression permet d'appréhender intuitivement l'attaque DPA, elle ne permet pas d'identifier formellement la ou les sources d'informations qu'exploite la DPA. Nous cherchons donc à reformuler cette expression (9) de telle sorte que ces sources d'informations apparaissent clairement.

Considérons qu'une attaque DPA est effectuée sur le bit Z d'un bloc combinatoire, constitué de P portes, avec l'ensemble de vecteurs V. Parmi ces V vecteurs, considérons que T d'entre eux forcent Z à rester ou à prendre la valeur logique '1' alors que V-T=F d'entre eux forcent la sortie à rester ou à prendre la valeur '0'.

Si l'on considère l'ensemble T (F) des vecteurs qui forcent Z à prendre la valeur '1' ('0'), trois types de vecteurs peuvent être définis selon l'action qu'ils peuvent avoir sur la porte k du bloc combinatoire. En effet, parmi ces T '(F) vecteurs on peut dénombrer :

- $T^k_{t,1}$ ($F^k_{t,1}$) vecteurs qui induisent une transition de '0' vers '1' à la sortie de la porte k
- $T^k_{t,0}$ ($F^k_{t,0}$) vecteurs qui induisent une transition de '1' vers '0' à la sortie de la porte k
- T^k_s (F^k_s) qui laissent la tension de sortie de la porte k inchangée

En s'appuyant sur ce résultat de dénombrement, il est possible d'exprimer la signature DPA du bloc combinatoire considéré selon :

$$\begin{aligned} S_{DPA}(Z) = & \frac{1}{T} \cdot \sum_{k=1}^{P} \left\{ T^k_{t,1} \cdot i_{k,1}(t) + T^k_{t,0} \cdot i_{k,0}(t) + T^k_S \cdot i_{k,-}(t) \right\} \\ & - \frac{1}{F} \cdot \sum_{k=1}^{P} \left\{ F^k_{t,1} \cdot i_{k,1}(t) + F^k_{t,0} \cdot i_{k,0}(t) + F^k_S \cdot i_{k,-}(t) \right\} \end{aligned} \qquad (10)$$

où $i_{k,1}(t)$, $i_{k,0}(t)$, et $i_{k,-}(t)$ sont les profils du courant consommés par la porte p lorsque la sortie de cette dernière effectue respectivement une transition croissante, descendante ou bien reste stable. En émettant l'hypothèse que $i_{k,-}(t)=0$, i.e. qu'en l'absence d'activité sur la sortie de la

porte k sa consommation est nulle, l'expression devient après quelques simplifications :

$$S_{DPA}(Z) = \frac{1}{T} \cdot \sum_{k=1}^{P-1} \left\{ \varepsilon^k \cdot \Delta i_k(t) \right\} + \varepsilon^P \cdot \Delta i_p(t) \qquad (11)$$

où $\Delta i_p(t)$ est le profil différentiel des courants consommés par la porte p :

$$\Delta i_k(t) = i_{k,1}(t) - i_{k,0}(t) \qquad (12)$$

et ε^k est, pour une porte k et une séquence de vecteur V donnée une constante définie par :

$$\varepsilon^k = \frac{T_{t,1}^k}{T} - \frac{T_{t,0}^k}{T} + \frac{F_{t,0}^k}{F} - \frac{F_{t,1}^k}{F} \qquad (13)$$

L'expression (11) du syndrome DPA est très intéressante. En effet, celle-ci met en évidence que l'analyse différentielle de la consommation exploite deux sources distinctes d'information. Plus précisément, le dernier terme de l'expression (11) démontre que l'attaque DPA exploite la dissymétrie entre les profils en courant d'une porte selon qu'elle calcule un '1' ou un '0' alors que le premier terme montre que la DPA exploite également les corrélations introduites par la projection technologique de la fonction que réalise le bloc combinatoire. Toutefois, pour un ensemble de vecteurs V grand, il est légitime de penser que ε^k est faible et que la principale source d'information qu'exploite la DPA est le profil différentiel de la porte contrôlant le bit Z.

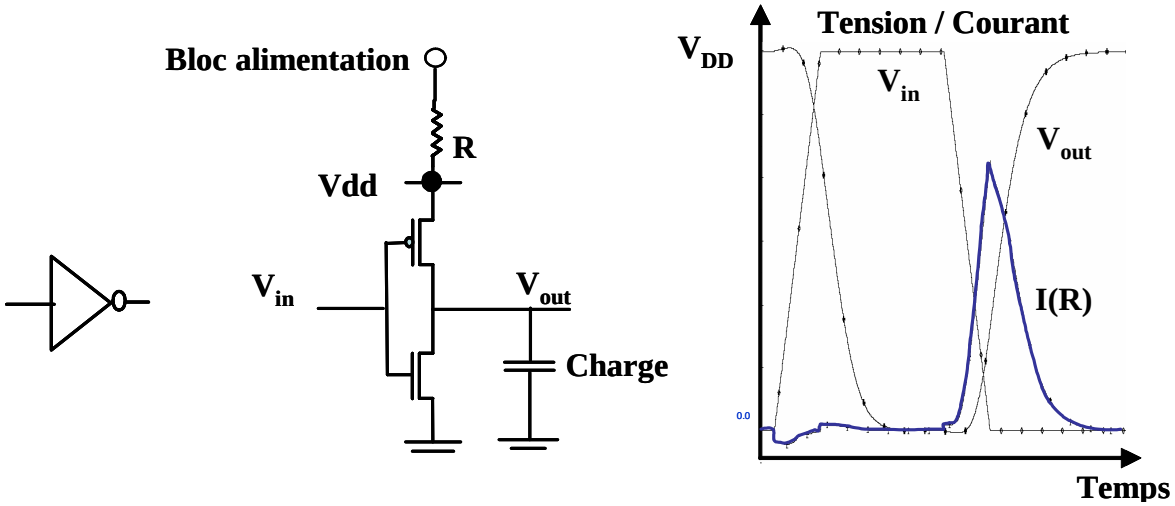


Fig. 21. Caractéristique de consommation d'un inverseur CMOS

En s'appuyant sur le constat que les implantations matérielles des algorithmes de chiffrement sont généralement réalisées avec des portes statiques CMOS, dont un profil de consommation en courant caractéristique est représenté sur la figure 21, on comprend bien dès lors pourquoi il existe une corrélation entre la consommation d'un bloc et les données qu'il manipule et plus

précisément une corrélation entre la consommation de la porte contrôlant Z et les données traitées. C'est d'ailleurs cette corrélation qu'exploite la DPA pour retrouver la clef de chiffrement. Toutefois cette corrélation peut être significativement réduite, voire éliminée, en développant des portes dont le profil en courant est identique quel que soit la transition effectuée par le signal de sortie, i.e. indépendant des données appliquées sur ses entrées. Dans Dans ce contexte, la logique double rail apparaît comme une alternative intéressante à la logique CMOS statique simple rail. Le codage des données associé à cette logique double rail offre en effet la possibilité d'équilibrer la consommation.

Le reste de ce chapitre est organisé comme suit: dans un premier temps, nous allons nous intéresser brièvement à l'historique de la conception de circuit double rail avant d'en définir les avantages et inconvénients non seulement en terme de conception de CIs mais surtout en terme de sécurité et plus particulièrement de résistance aux attaques différentielles en courant. Dans un deuxième temps, nous allons passer en revue différents styles d'implantation qui ont été proposés dans la littérature. Dans un troisième temps, nous introduirons notre propre méthode de conception de cellules double rail. Enfin, avant de conclure, nous mènerons une analyse comparative des performances et de la robustesse à la DPA des différents styles d'implantation considérés dans ce chapitre.

III-2-b-Le codage double rail

Concrètement, le codage double rail se caractérise par l'utilisation de deux fils pour coder une valeur binaire. L'utilisation de deux fils pour coder deux valeurs binaires distinctes offre différents choix de codage. Les deux codages les plus couramment utilisés sont les codages avec retour à l'état invalide (RTZ) et sans retour à l'état invalide (NRTZ). Les figure 22 et 23 reportent respectivement un exemple de codage RTZ et NRTZ.

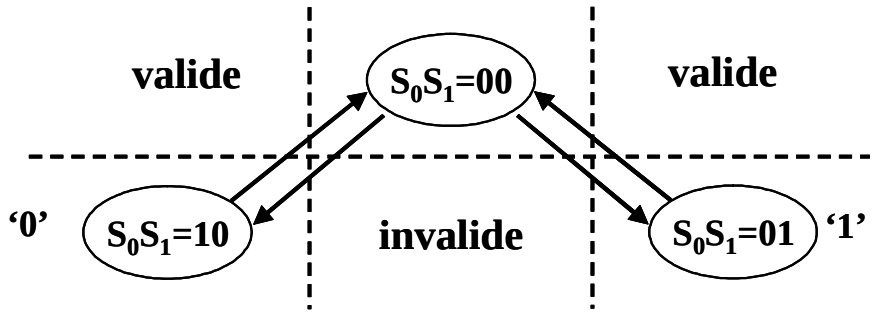


Fig. 22. Codage avec retour à l'état invalide (RTZ)

Dans le cas d'un codage avec retour à l'état invalide, comme par exemple celui représenté sur

la figure 22, un des deux fils prend la valeur 1 pour coder une donnée à 1 et l'autre fil prend la valeur 1 pour coder la donnée 0. L'état "11" est interdit alors que l'état "00" représente l'invalidité d'une donnée. Comme on peut le constater sur la figure 22, le passage d'un état valide à un autre état valide s'effectue nécessairement par le passage par l'état invalide que l'on appelle également 'sparser'. L'existence de cette séquence immuable données valides / données invalides / données valides est précieuse dans le cadre de la conception de circuits asynchrones. En effet, elle permet de détecter la fin des calculs et donc d'assurer le cadencement des circuits en l'absence de signal d'horloge. Toutefois, ce n'est pas la seule manière d'assurer le cadencement des circuits, et d'autres types de codage de données offrent des avantages similaires. Le codage NRTZ dit quatre états est l'un d'entre eux [Mca92].

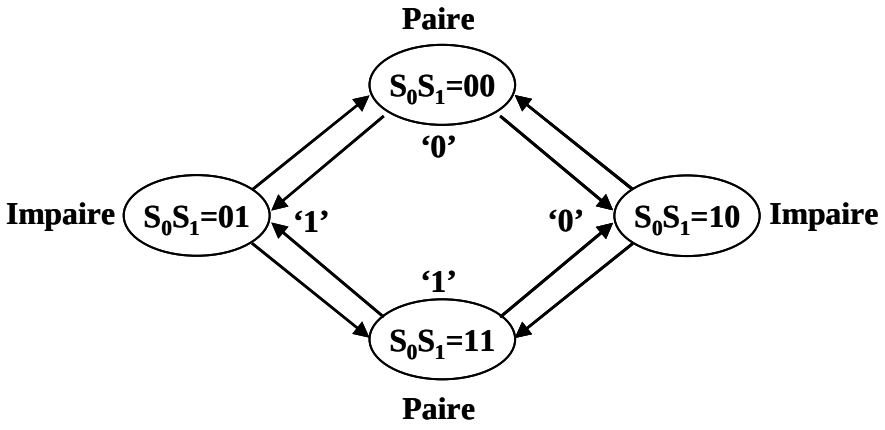


Fig. 23. Codage double rail NTRS dit quatre états

Dans le cas de ce codage, chaque valeur binaire '0' ou '1' est codée avec deux combinaisons. L'une des combinaisons est considérée comme étant de parité impaire et l'autre de parité paire comme l'illustre la figure 23. En l'absence de sparser, l'émission de nouvelles données s'accompagne nécessairement d'un changement de parité qui permet de détecter la fin des calculs sans ambiguïté. Ceci présente potentiellement de nombreux avantages en terme de performance temporelle dans la mesure où il n'est plus nécessaire de passer par un état invalide pour émettre de nouvelles données. Toutefois bien qu'élégante, cette méthode reste très coûteuse en terme d'implantation dans la mesure où l'équivalent d'un half buffer (20 transistors) [Ren00a] n'est autre qu'une bascule maître esclave "clockée" (34 transistors).

Outre les caractéristiques spécifiques de ces deux codages bifilaires, ces derniers présentent une caractéristique commune. En effet, comme on peut le constater sur les figures 22 et 23, tout établissement d'une donnée à la sortie d'une cellule s'accompagne de l'établissement

d'un rail et un seul à V_{DD}. En d'autres termes, la distance de Hamming entre deux états successifs d'un circuit double rail est constante.

Ceci confère à priori aux circuits double rail des propriétés intéressantes. Plus particulièrement, l'énergie consommée par un circuit double rail pour changer d'état doit être théoriquement une constante caractéristique de celui. Cette propriété théorique a un corollaire intéressant dans le contexte de l'accroissement de la robustesse des circuits aux attaques DPA : "**la consommation des circuits double rail est indépendante des données traitées**". C'est ce corollaire qui récemment a suscité un vif intérêt pour la logique double rail dans la communauté des concepteurs de circuits sécurisés.

III-2-b- Du codage double rail aux cellules double rail

Si l'on considère les chemins de conduction permettant de charger ou décharger les sorties d'une cellule double, il apparaît qu'une cellule double rail peut être considérée comme la juxtaposition de deux cellules complexes contrôlant chacune une des sorties comme cela est schématisé sur la figure 24.

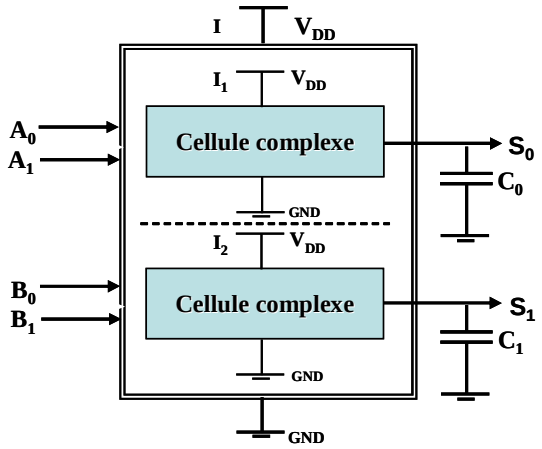


Fig. 24. Cellule double rail

Si l'on se place dans le contexte d'un codage RTZ ou codage trois états, ces deux cellules complexes sont structurellement différentes ou bien, si ce n'est pas le cas, reçoivent sur leurs entrées des signaux différents dans la mesure où l'état '11' est interdit. Ces différences topologiques (elles existent dans la très grande majorité des cas) impliquent que les consommations des deux cellules ne sont pas strictement identiques et n'absorbent donc pas les mêmes courants ($I_1 \neq I_2$) pour commuter. Par conséquent, les propriétés relatives à la consommation que nous avons déduites des spécificités du codage double rail ne sont pas automatiquement vérifiées après implantation. Une attention toute particulièrement doit donc

être apportée à la conception des cellules afin de conserver d'assurer un très faible niveau de corrélation entre les données et la consommation. Nous reviendrons largement sur ce dernier point dans les paragraphes suivants.

III-3- Circuits asynchrones et circuits double rail

III-3-a-Historique

Historiquement, la projection technologique d'un circuit sur une bibliothèque de cellules double rail a été effectuée dans le contexte de la conception de circuits asynchrones [Ren00]. En effet, l'absence de signal global de synchronisation des calculs, le codage double rail, et plus généralement le codage 1 parmi n, a largement été adopté par la communauté 'asynchrone' afin de pouvoir détecter sans ambiguïté la fin des calculs ou bien la présence de nouvelles données. Nous allons donc dans les paragraphes ci-après brièvement rappeler quels sont les grands principes de ce type de logique et ses principaux avantages et inconvénients potentiels.

III-3-a-1-Circuit asynchrone : principe de base

En l'absence de signal global de synchronisation, les échanges d'information au sein d'un circuit asynchrone sont gérés localement par la mise en place d'une signalétique adéquate. Cette dernière, généralement appelée protocole de communication entre canaux, doit assurer la causalité des évènements au niveau local et donc la correction fonctionnelle du système dans son ensemble. Pour ce faire, le contrôle local doit assurer les tâches suivantes :

- être à l'écoute des communications entrantes,
- déclencher le traitement localement si toutes les informations sont disponibles (rendez-vous)
- mais également produire un signal d'acquittement informant les opérateurs amont que les données ont bien été consommées.

En effet, sans acquittement comment savoir que le récepteur est prêt à traiter de nouvelles données ? Dans ce contexte, deux opérateurs asynchrones connectés entre eux, communiquent l'un avec l'autre en échangeant entre eux des informations selon un protocole pré-établi et ce indépendamment des autres opérateurs constituant le circuit. Compte tenu des taches que doit assurer le contrôle local, les échanges d'informations se font de manière bidirectionnelle et

tout échange d'information doit être acquitté afin que l'émetteur puisse émettre à nouveau. Ces contraintes ont donné lieu à la définition de divers protocoles de communication de type requête-acquitement.

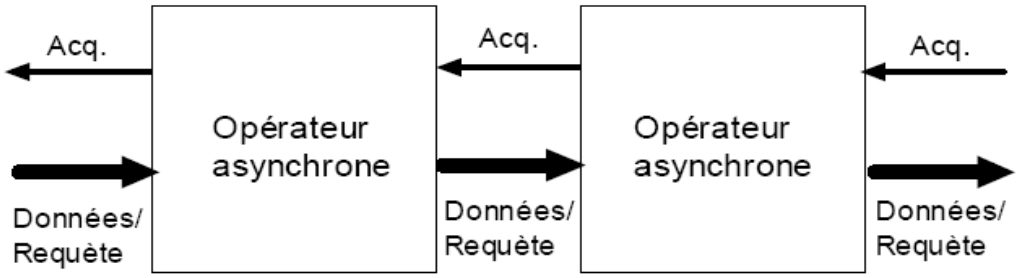


Fig. 25. Communication de type requête/acquittement entre opérateurs asynchrones pour garantir une synchronisation indépendante du temps.

III-3-a-2-Protocoles de communication

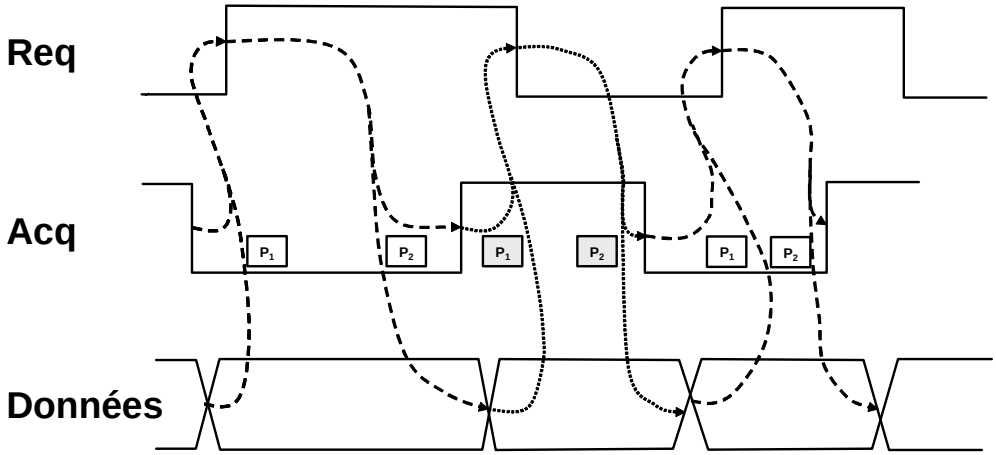


Fig. 26. Un protocole requête acquittement deux phases

De nombreux protocoles de communication de type requête - acquittement (Figure 25) satisfaisant les contraintes établies auparavant ont été rappelés dans [Rig02]. L'ensemble de ces protocoles peut être organisé en deux ensembles distincts:

- les protocoles 2 phases (ou NRZ pour Non Retour à Zéro ou encore "Half-handshake"), qui nécessite l'implantation d'une logique de contrôle sensible aux fronts des signaux (Figure 26),
- et les protocoles 4 phases (ou RZ pour Retour à Zéro ou encore "Full-handshake") qui nécessite l'implantation matérielle d'une logique à niveaux traditionnelle (Figure 27).

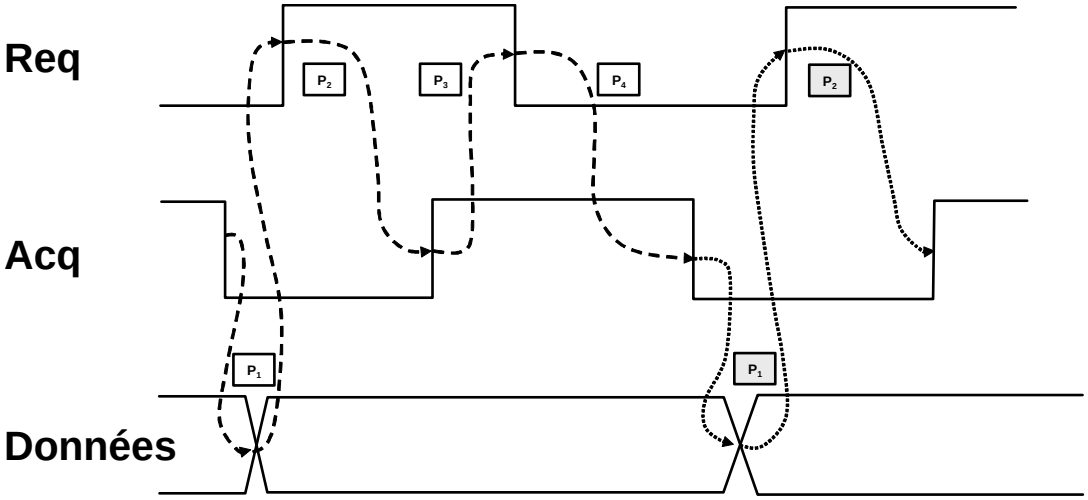


Fig. 27. Un protocole requête acquittement quatre phases

Ces deux types de protocole peuvent aussi bien se décliner en logique simple rail qu'en logique multi rail. Dans ce dernier cas, la requête pouvant être véhiculée par les données elles mêmes.

Toutefois, dans la pratique, pour des raisons de coûts d'implantation et de performance, les protocoles quatre phases supplantent généralement les protocoles deux phases [Ren00]. Ce surcoût des protocoles deux phases est essentiellement dû à l'utilisation de logique sensible aux fronts. La figure 26 décrit dans les grandes lignes le fonctionnement de ce type de protocole, alors que la figure 27 décrit elle le principe sur lequel repose le protocole quatre phases. Comme cela est représenté, ce dernier s'organise comme suit :

- Phase 1 : c'est la première phase active de l'émetteur qui, après avoir détecté que le récepteur est susceptible d'accepter de nouvelles données, positionne celles-ci sur les entrées du récepteur et émet une requête.
- Phase 2 : c'est la première phase active du récepteur qui détecte le signal de requête, et inverse le signal d'acquittement après traitement de l'ensemble des données,
- Phase 3 : c'est la deuxième phase active de l'émetteur qui après avoir détecté la fin de calcul du récepteur positionne l'ensemble des données à l'état invalide et invalide la requête.
- Phase 4 : c'est la deuxième phase active du récepteur, qui détecte le retour à zéro de la requête et abaisse son signal d'acquittement après que l'ensemble de ses données internes aient été invalidées

Si la séquence des quatre phases constituant le codage quatre états est intuitive, la définition

des phases actives du récepteur (2 et 4) soulève une question capitale : comment le récepteur détecte t-il que suffisamment de temps s'est écoulé pour considérer que les données valides ou invalides appliquées sur son entrée ont été complètement traitées ?

III-3-a-3- Détection de la fin des calculs et codage des données

Deux solutions sont possibles pour détecter la fin des calculs selon le modèle temporel que l'on adopte pour les éléments constitutifs des circuits intégrés et plus particulièrement les portes logiques et les interconnexions.

Dans une première approche, on peut supposer que les éléments constitutifs d'un circuit intégré ont des délais bornés dont la valeur est connue. Sous cette hypothèse, couramment utilisée dans le cadre de la conception de circuits synchrones, la détection de calcul s'effectue grâce à l'adjonction de portes logiques une telle sorte que celle-ci se propage plus lentement que les données. On parle de circuits micropipelinés ou 'bundled data' [Sut89]. Toutefois ceci revient à considérer un modèle temporel pire cas tout comme dans le cadre de la conception de circuits synchrones ou encore à remplacer le signal d'horloge global par une série de signaux locaux de synchronisation très similaires à des horloges locales.

Une autre solution, nous l'avons déjà évoquée, consiste à adopter un codage spécifique permettant de détecter sans ambiguïté la fin des calculs. Les codages m parmi n, avec m<n et plus spécifiquement le codage double rail est l'un de ces codages. Dans le cas où l'on adopte ce type de codage, la détection de la fin des calculs d'un module double rail s'effectue en vérifiant tout simplement :

- qu'au moins un des fils de l'ensemble des paires de rail codant les bits de sortie est '1' si l'on se trouve dans la phase 2
- que tous les fils de sortie sont à '0'.

Matériellement cela est réalisé, notamment dans les circuits Quasi-Delay Insensitive [Mar90], en effectuant le 'ou' de toutes le paires de rail et en effectuant un rendez-vous de l'ensemble des signaux délivrés par ces portes 'ou'.

III-3-b- Circuits asynchrones double rail et DPA

Compte tenu qu'historiquement la projection technologique d'un circuit sur une bibliothèque double rail a été effectuée dans le cadre de la conception de circuits asynchrones, c'est tout naturellement que nous nous sommes posés la question suivante : les circuits asynchrones sont-ils intrinsèquement mieux adaptés à la conception de circuits sécurisés que les circuits synchrones ?

Afin d'apporter des éléments de réponses à ces questions, nous passons en revue les principaux avantages potentiels et les inconvénients d'ors et déjà connus.

III-3-b-1-Absence de référence temporelle

Dans le chapitre II, nous avons longuement étudié la mise en œuvre d'une attaque DPA sur un circuit sécurisé. Nous avons notamment vu que l'attaque DPA nécessite la collection d'une quantité importante de traces de consommation. Cette collection est généralement réalisée en prenant le signal d'horloge externe comme référence temporelle stable permettant de synchroniser les courbes collectionnées entre elles.

En l'absence de signal d'horloge l'acquisition et la synchronisation des traces de consommation s'avèrent plus délicate mais pas impossible. En effet, des techniques de traitement ad-hoc [Fur03] doivent être mise en œuvre pour re-synchroniser les courbes entre elles et les ramener sur une même base temporelle (période de calcul) et pouvoir effectuer l'attaque à proprement parlé.

Pour résumer l'absence de signal global et de périodicité dans le calcul n'est pas un obstacle incontournable. Toutefois, ne pas avoir de référence temporelle requière l'utilisation de techniques spécifiques qui compliquent la mise en œuvre de la DPA et suppose un degré de compétence supplémentaire de la part de l'attaquant.

III-3-b-2-Profils en courant non répétitifs et de faibles amplitudes

Comme l'illustre la figure 28 [Koc99], d'une manière générale les circuits synchrones comme la plupart des circuits sécurisés ont des caractéristiques de consommation répétitives. En terme d'attaque DPA cette répétitivité des formes d'onde en consommation est très intéressante dans la mesure où elle permet d'identifier clairement l'instruction mise en jeu. Par exemple dans la figure 28, chaque forme répétitive correspond au traitement d'un round de

l'algorithme DES. La lecture directe de cette donnée par simple SPA constitue une information précieuse pour l'attaquant dans la mesure où celui-ci sait exactement où débuter son attaque (premier ou dernier round) pour récupérer les premières sous-clefs.

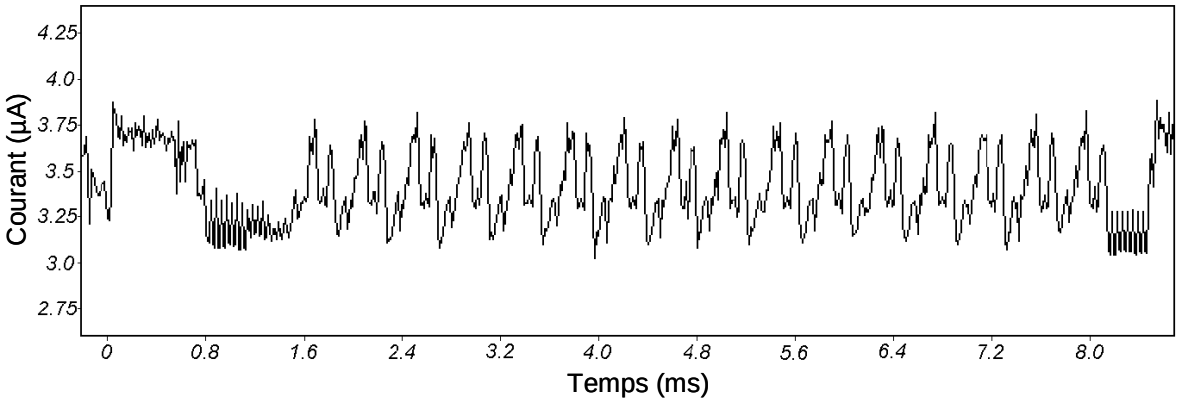


Fig. 28. Caractéristiques de consommation d'un DES synchrone

Dans le cas des circuits asynchrones, cette identification de l'étape de calcul n'est pas aussi évidente. Avec un fonctionnement de type flot de données, les profils en courant des circuits asynchrones peuvent être différents d'un cycle à un autre. Plus précisément, le temps de traitement d'un cycle est étroitement dépendant des données manipulées. Par ailleurs, l'utilisation de protocole de communication de type poignée de mains permet aux circuits asynchrones de mieux répartir la consommation du courant dans le temps et donc de réduire considérablement les appels en courant. Typiquement, le profil en courant d'un circuit asynchrone est assez plat. Ainsi en terme de résistance à l'attaque DPA, deux autres gros avantages des circuits asynchrones sont: la variation du temps de cycle qui peut être vu comme un 'jitter' temporel et la faible amplitude des pics de consommation.

Pour conclure sur les avantages potentiels des circuits asynchrones on peut donc dire que quelques propriétés intrinsèques de l'absence de signal d'horloge rende la mise en œuvre des attaques différentielles en courant plus difficile, sans la rendre impossible. Toutefois, ces avantages dus à l'absence de signal global de synchronisation s'additionnent à l'utilisation de la logique double rail dans les circuits asynchrones dont le fonctionnement est insensible au délai des éléments constitutifs (circuits Quasi Delay Insensitive dans le jargon [Mar90]. Ce type de circuits semble donc être nettement plus résistants que les circuits synchrones ou les circuits asynchrones bundled data [Sut89, Fur03].

III-3-c- Limitations des circuits asynchrones double rail

Bien que les circuits asynchrones double rail et plus spécifiquement les circuits QDI semblent être les circuits les plus robustes aux attaques différentielles en courant, la très grande majorité des cryptosystèmes sont aujourd'hui des circuits synchrones réalisés avec de la logique CMOS statique.

Plusieurs raisons peuvent expliquées cet état de fait. Parmi celles-ci, le manque d'outils (si l'on exclut Tangram [Kes01]) de conception de circuits asynchrones semble être la plus importante. Certes, des outils de conception universitaires ont été développés (Petrify [Cor96]) ou sont en cours de développement (CAST [Mar93], TAST [Bou05], BALSA [Bar00]) mais ne sont pas encore passés dans le domaine industriel. Toutefois, ce gap semble se résorber rapidement depuis cinq ans comme le démontre l'apparition des sociétés comme Fulcrum Microsystems [Ful06], qui s'appuie sur les solutions développées dans [Mar93], ou encore par handshake solutions [han06].

Pour contourner ce problème d'outil, les concepteurs de circuits asynchrones sont contraints d'utiliser au mieux les outils de conception de circuits synchrones, i.e. d'adopter les flots de conception à base de cellules pré-caractérisées. Cependant, ceci conduit à un autre écueil, à savoir : l'absence de bibliothèque de cellules contenant les primitives de base de la conception de circuits asynchrones comme par exemple les portes de Muller qui réalisent des rendez-vous électriques entre divers signaux.

Cet écueil peut être aisément surmonté en implantant la fonction rendez-vous à partir de cellules complexes traditionnelles. A titre d'illustration, la figure 29 montre comment réaliser le rendez-vous électrique de deux signaux à partir d'une porte complexe AO222. Si cette solution de secours, mentionnée dans [Ren00a] que nous désignerons par approche standard cell par la suite, permet de concevoir des circuits sans avoir à développer de cellules spécifiques, son utilisation conduit à des implantations matérielles sous optimales et plus particulièrement très chères en terme de surface et consommation. A titre d'exemple, lors de la conception d'un AES asynchrone [Bou05a], l'utilisation d'une bibliothèque spécifique ne comprenant principalement que des portes de Muller, a permis de réduire la surface de 20% par rapport à l'utilisation d'une bibliothèque fournie par un fondeur.

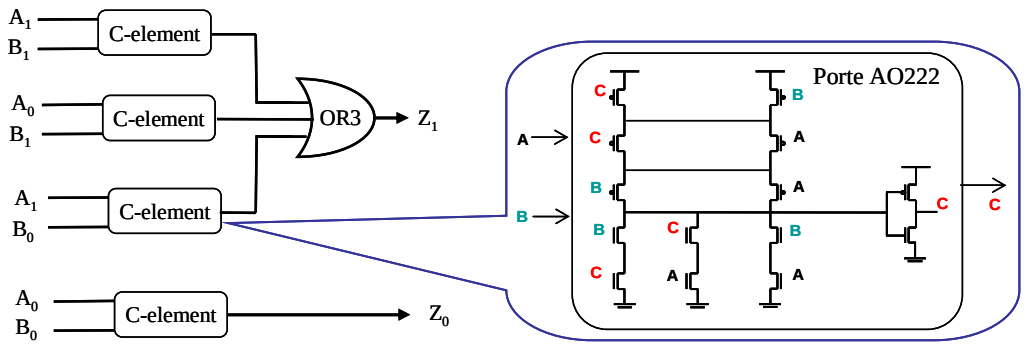


Fig. 29. Porte OR2 double rail réalisée à partir de cellules AO222 standard

En résumé, le principal handicap de la conception de circuits asynchrones double rail ou quasi insensible aux délais réside principalement dans l'absence d'outils de conception matures et de bibliothèques de cellules dédiées. Un second handicap est relatif quant à lui au coût excessif en surface des circuits asynchrones qui sont généralement 2.5 à 5 fois plus gros que leurs homologues synchrones en fonction du type de bibliothèque utilisé. Malgré ces deux handicaps, de nombreux travaux sont dévolus à la conception de circuits asynchrones sécurisés ; des prototypes ont d'ailleurs été réalisés et ont démontré que les circuits asynchrones sécurisés présentent une robustesse supérieure à celle de leurs homologues synchrones [Bou05a].

III-4- Les types d'implantations de cellules double rail

Si l'absence de signal d'horloge et surtout l'utilisation d'un codage double rail sont les deux principales origines de la robustesse à la DPA des circuits asynchrones quasiment insensible aux délais, ces caractéristiques ne sont pas suffisantes pour garantir qu'un circuit asynchrone double rail soit inattaquable. En effet, il a été expérimentalement démontré dans [Bou04], dans le cas d'un circuit DES asynchrone, que malgré l'utilisation de logique double rail, il subsiste des bits sur lesquels l'attaque DPA peut être effectuée avec succès. En d'autres termes malgré la minimisation des différences des profils en courant par rapport aux données manipulées, des canaux d'information subsistent et peuvent être exploités au moyen d'une analyse différentielle en courant.

Afin d'identifier clairement ces sources physiques d'information, il est nécessaire d'appréhender la manière dont sont implémentés ces circuits double rail et plus particulièrement les primitives double rail. Par exemple, les concepteurs du DES introduit

dans [Bou04] ont effectué la projection technologique de leur architecture en utilisant une bibliothèque de cellules simple rail spécifique à la conception de circuits double rail. En complément d'une bibliothèque fondeur.

Cette bibliothèque spécifique [Mau03] est essentiellement composée de cellules comme les portes de Muller et des fonctionnalités les plus souvent rencontrées dans les circuits asynchrones. Ces fonctionnalités sont par la suite appareillées pour réaliser une cellule double rail comme l'illustre la figure 30. Comme on peut le constater, l'appariement de ces cellules simple rail conduit à la réalisation de cellules visiblement dissymétriques et par voie de conséquence à des cellules dont la consommation est étroitement dépendante des données traitées. Cette vulnérabilité des primitives double rail réalisée à partir de TAL est essentiellement due au fait que TAL a été développée afin de limiter le surcoût en surface des circuits asynchrones double rail et donc qu'aucun effort de 'sécurisation' à la DPA n'a été consenti.

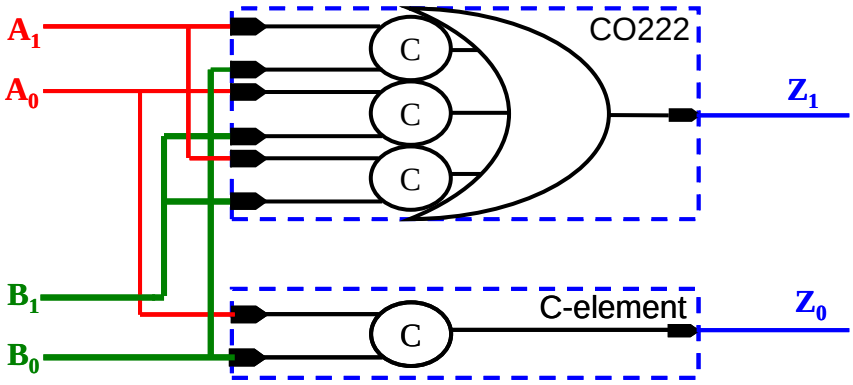


Fig. 30. Porte 'OR2' double rail réalisée avec des cellules simple rail CO222 et C-element

Partant de ce constat, et en considérant la robustesse des circuits QDI expérimentalement prouvée dans [Bou04], on peut se demander si en consentant un effort particulier au développement d'une bibliothèque de cellules double rail sécurisées, on peut rendre la consommation d'un circuit double rail quasiment indépendante des signaux appliqués sur ses entrées.

Des logiques sécurisées [Kul05, Raz04, Gui04, Tir02, Mac04] ont été introduites dans la littérature afin d'accroître la robustesse des circuits intégrés synchrones ou asynchrones aux attaques DPA. Si les portes logiques sécurisées résultantes se différencient radicalement par leurs topologies et/ ou leur style d'implantation (CMOS, differential precharged logic, Sense Amplifier based logic, Current Mode Logic), elles ont toutes une caractéristique commune, à

savoir : une consommation quasiment indépendante des entrées. Nous allons dans les paragraphes suivants étudier les caractéristiques de ces logiques, en gardant à l'esprit qu'il ne s'agit pas là d'une revue exhaustive et que d'autres schémas de cellules ont été proposés dans la littérature.

III-4-a-Logique SABL

La plus connue des logiques sécurisées est la SABL, pour Sense Amplifier Based Logic introduite dans [Tir02], héritière de nombreux travaux dévolus à l'optimisation de la logique DCVSL et de la logique DOMINO [Chu87, Som97]. C'est un des premiers styles de logique à être proposé dans la littérature comme une contre-mesure aux attaques différentielles en puissance.

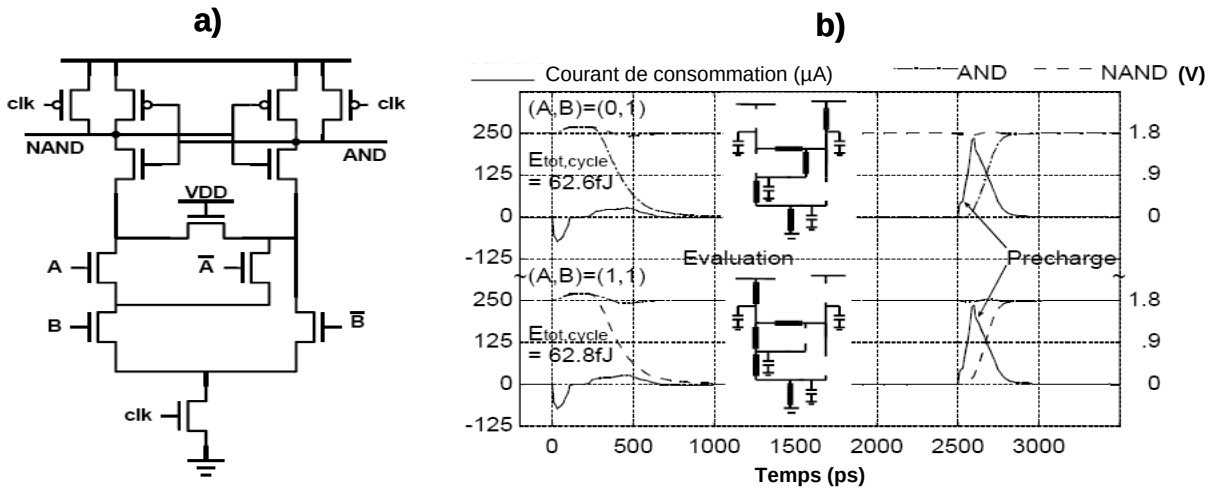


Fig. 31.a) Une porte NAND2/AND2 en logique SABL, b) Simulation temporelle des évènements de charge et décharge d'une porte NAND2/AND2 en logique SABL

Comme les autres logiques sécurisées, l'objectif de la logique SABL est clairement établi: obtenir une consommation indépendante des données traitées. Pour ce faire, la stratégie de conception des cellules proposée par les auteurs est de faire en sorte que quelque soit les événements (1-1,1-0,0-0,0-1) se produisant sur les entrées de la cellule, cette dernière charge ou décharge toujours la même charge capacitive, i.e. la même énergie. Pour ce faire, chaque cellule SABL est conçue de telle sorte que tous les nœuds internes (points noirs sur la figure 31) soient reliés à une sortie durant la phase d'évaluation et à un rail d'alimentation pendant la phase de précharge.

Si ce style d'implantation permet l'obtention de cellules compactes et robustes à priori aux attaques DPA, il présente toutefois un certains nombre de défauts à savoir: une forte

dissymétrie des plans N et P en terme de layout, une faible immunité au bruit (problème de partage de charges), et enfin son utilisation résulte en une surcharge de l'arbre d'horloge qui se traduit par des consommations instantanées très élevées.

III-4-b-Cellules CMOS double rail sécurisées

Ces défauts de la logique SABL justifient l'existence de solutions alternatives comme par exemple celles proposées dans [Gui04] qui adopte une approche CMOS statique pour concevoir ses cellules. Plus précisément, les auteurs de ces travaux optent pour une approche redondante (Figure 32) de la conception des cellules de manière à obtenir des chemins électriques de charge et décharge les plus symétriques possibles et donc une consommation indépendante des données traitées.

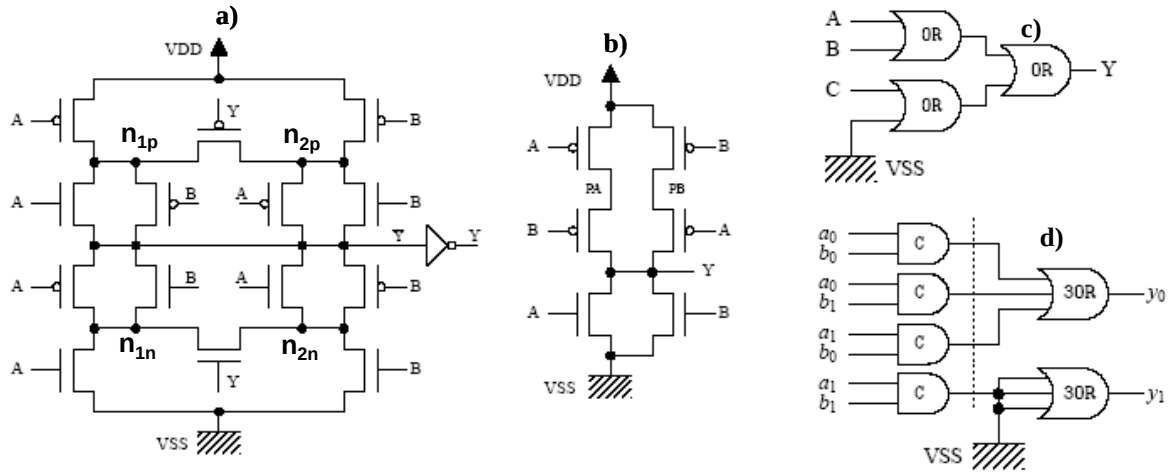


Fig. 32.a) C-element b) une porte NOR2 c) une porte OR3 d) une cellule AND2 double rail sécurisée

Comme on peut le constater sur la figure 32, cette approche résulte en des cellule double rail particulièrement coûteuse en terme de nombre de transistors et donc de surface. Toutefois, l'objectif des auteurs [discussion privée] n'est pas d'optimiser la logique double rail en soit mais de démontrer qu'il est possible de parfaitement équilibrer la consommation d'un circuit et donc de rendre inefficace l'attaque DPA.

Pour réduire l'impact de la position (dans les réseaux série de transistors) du transistor déclenchant la commutation de la porte, l'auteur préconise de dupliquer l'ensemble des réseaux série de transistors en effectuant des permutation circulaire des signaux d'entrées comme l'illustre la figure 32b dans le cas d'un réseau série de trois transistors. Comme on peut le constater, cette solution est coûteuse en terme de surface. A titre d'illustration,

considérons le cas d'une porte dont le plan P est constituée d'un unique réseau série de *n* transistors. Pour une telle porte, adopter cette approche revient à dupliquer *n*–1 fois le réseau P ce qui est terme de surface prohibitif au-delà de trois entrées.

Pour éviter le stockage de charge dans les réseaux série de transistors, chaque cellule est construite de telle façon à ce que, pour toutes les combinaisons possibles des entrées, tous les nœuds internes soient chargés/déchargés sur un cycle de commutation. Le cas de la porte Muller statique représentée sur la figure 32a illustre parfaitement cette approche. Comme on peut le constater, des transistors P (N) ont été rajoutés dans le plan N (P) afin de permettre la décharge des nœuds n_{1p}, n_{2p}, n_{1n}, n_{2n}.

Enfin, pour avoir des profils en courant complètement identiques en amplitude et en durée, il faut non seulement charger/décharger la même valeur de capacité mais aussi et surtout avoir des chemins de données équilibrés. Pour ce faire, les auteurs introduisent des portes redondantes d'un point de vue logique afin d'équilibrer topologiquement et électriquement leurs cellules double rail. La figure 32c illustre parfaitement cette stratégie coûteuse en terme de surface et consommation.

Des validations ont montré qu'individuellement les cellules double rail ont des profils en courant identiques indépendamment des données traitées. Cependant au niveau cellule élémentaire, les impacts au niveau des performances sont sévères: accroissement d'un facteur x 6 en consommation, x 12 en délai de propagation et x 25 en surface par rapport à de la logique CMOS conventionnelle.

III-5- Définition et caractérisation d'une bibliothèque double rail

Durant nos travaux de recherche nous avons passé en revue un grand nombre de styles d'implantation dédiés à la conception de circuits sécurisés double rail dont deux ont été présentés dans les paragraphes précédents. Suite à cette revue, il semble que la quasi-totalité des styles d'implantation présente des handicaps significatifs: surcharge de l'arbre d'horloge, consommation excessive, coût d'intégration élevé, etc. Par ailleurs, il existe très peu de méthodes simples de construction de cellules double rail à l'exception de celle proposée par Kris Tiri pour concevoir les cellules SABL [Tir02].

III-5-a- Méthode de conception de cellules double rail

Dans ce contexte, notre objectif est de proposer une méthode, simple de mise en œuvre, permettant la conception de cellules double rail offrant un meilleur compromis sécurité – performances (consommation – surface – vitesse) et pouvant être utilisée tant pour la conception de circuits synchrones que pour la conception de circuits asynchrones.

Pour introduire cette méthode, rappelons quelles sont les principales spécificités des cellules nécessaires à la conception de circuits double rail synchrones ou asynchrones communiquant selon un protocole quatre phases et adoptant le codage des données représenté sur la figure 22. Pour de tels circuits, le transfert des données entre un émetteur et un récepteur débute par l'émission d'un signal de requête encodé dans les données et s'achève, après un intervalle de temps Δt, par l'émission d'un signal d'acquittement par le récepteur. Durant cet intervalle de temps Δt, dont la durée est à priori inconnue, les données en sortie des cellules doivent être maintenues afin de garantir la propriété d'insensibilité aux délais.

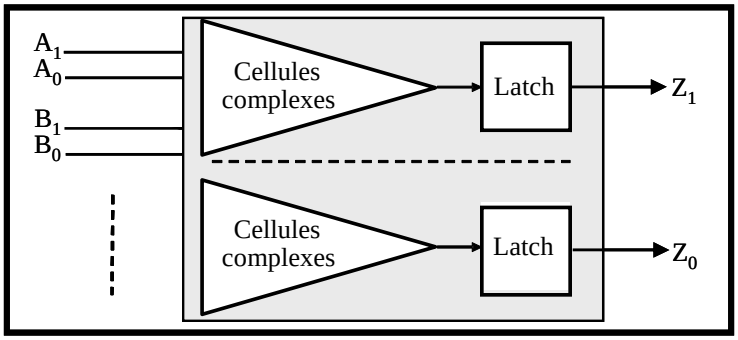


Fig. 33. Topologie d'une cellule double rail

Si le maintien des niveaux en sortie des cellules peut être assuré à l'aide de re-bouclage [Ren00a] entre les sorties et les entrées des cellules, il est généralement assuré par des latches. Dans ce cas, une cellule double rail peut être considérée comme la juxtaposition de deux portes CMOS complexes contrôlant chacune une latch comme le montre la figure 33.

Si l'on adopte pour topologie de cellule double rail celle représentée sur la figure 33, la conception d'une porte double rail se résume à la détermination des deux cellules complexes contrôlant les latches. Ceci peut être réalisé en six étapes successives qui constituent la méthode que nous proposons.

Etape n°1 :

A l'image du codage double rail représenté sur la figure 22, la première étape consiste à identifier les trois expressions booléennes représentatives de la fonction que l'on souhaite réaliser. Deux de ces expressions seront relatives aux conditions de mise à V_{DD} des rails z_1 et z_0 ; mises à V_{DD} qui indiquent respectivement que le bit de sortie de la cellule prend la valeur booléenne '1' ou '0'. La troisième de ces expressions étant elle, relative aux conditions de retour à Gnd des rails z_1 et z_0. Afin d'illustrer cette étape d'identification, considérons le cas où l'on souhaite réaliser une porte OR3 prenant trois entrées: A(a_1, a_0), B(b_1, b_0) et C(c_1, c_0). Les trois expressions booléennes représentatives à la fonction OR3 en logique double rail sont dans ce cas:

$$Z = A + B + C \tag{14}$$

$$\overline{Z} = \overline{A} \cdot \overline{B} \cdot \overline{C} \tag{15}$$

$$Z^I = A^I \cdot B^I \cdot C^I \tag{16}$$

Si les deux premières expressions, relatives aux mises à V_{DD} des rails z_1 et z_0 sont identiques à celles définissant le comportement d'une porte OR3 simple rail, la troisième est spécifique au codage double rail puisqu'elle définit les conditions de retour à Gnd des deux rails (retour à l'état invalide). Dans cette expression, le caractère 'I' est utilisé pour stipuler que les bits A, B et C sont dans un état invalide.

Etape n°2 :

A	**B**	**C**	**Z**
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Tableau 2. Table de vérité d'une OR3

L'identification des trois expressions caractéristiques du fonctionnement de la cellule double rail effectuée, la deuxième étape de la méthode consiste à réécrire ces trois expressions comme des sommes de monômes fonction de l'ensemble des bits d'entrée de la cellule. Cette

étape s'effectue aisément en considérant la table de vérité de la fonction simple rail correspondante. Dans le cas de notre exemple, la réécriture à partir de la table de vérité (Tableau 2), de trois expressions caractéristiques de la porte OR3 conduit à :

$$Z = A \cdot B \cdot C + A \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C \quad (17)$$

$$\overline{Z} = \overline{A} \cdot \overline{B} \cdot \overline{C} \quad (18)$$

$$Z^I = A^I \cdot B^I \cdot C^I \quad (19)$$

Etape n°3 :

La troisième étape de la méthode consiste à reformuler les trois expressions obtenues à l'étape n°2 de façon à ce que le codage double rail apparaisse de manière explicite. Cette étape de traduction en double rail s'effectue de manière très simple en considérant le tableau 3 qui définit les correspondances entre simple rail et double rail. Notez que dans ce tableau 3, inversement à Bit^I, la notation Bit^V signifie que la donnée Bit est dans un état valide. Dans le cas de notre cellule OR3, cette transformation conduit à obtenir les trois nouvelles expressions caractéristiques suivantes :

$$z_1 = a_1b_1c_1 + a_1b_1c_0 + a_1b_0c_1 + a_1b_0c_0 + a_0b_1c_1 + a_0b_1c_0 + a_0b_0c_1 \quad (20)$$

$$z_0 = a_0b_0c_0 \quad (21)$$

$$z^I = \overline{z}_1\overline{z}_0 = \overline{a}_1\overline{a}_0\overline{b}_1\overline{b}_0\overline{c}_1\overline{c}_0 \quad (22)$$

Notre méthode ayant pour cible la topologie représentée sur la figure 33, les expressions (20) et (21) vont permettre de bâtir de manière traditionnelle les réseaux N des deux portes complexes, alors que l'expression (22) permet elle de bâtir les réseaux P. Toutefois dans le cas de notre OR3 comme pour beaucoup de cellules double rail, la construction des réseaux P conduit à la mise en série d'un nombre excessif (>4) de transistors, ce qui est inacceptable en terme de performance. Afin de résoudre de problème, il est donc nécessaire d'introduire des variables intermédiaires. Ceci est réalisé lors de la quatrième étape.

	Bit	$\overline{Bit}$	Bit^V	Bit^I
A(a1,a0)	a_1	a_0	$a_1 \oplus a_0 \equiv a_1 + a_0$	$\overline{a_1} \bullet \overline{a_0}$
B(b1,b0)	b_1	b_0	$b_1 \oplus b_0 \equiv b_1 + b_0$	$\overline{b_1} \bullet \overline{b_0}$
C(c1,c0)	c_1	c_0	$c_1 \oplus c_0 \equiv c_1 + c_0$	$\overline{c_1} \bullet \overline{c_0}$
Z(z1,z0)	z_1	z_0	$Z_1 \oplus z_0 \equiv z_1 + z_0$	$\overline{z_1} \bullet \overline{z_0}$

Tableau 3. Table de correspondance simple rail - double rail

Etape n°4 :

S'il est possible d'insérer des variables intermédiaires quelconques afin d'implanter les réseaux de transistors P, il est également possible d'exploiter les spécificités du codage double rail. Parmi les principales caractéristiques de ce codage, deux sont particulièrement notables. La première d'entre elles est relative à l'existence d'un état interdit (a_0, a_1) = (1,1). La seconde de ces caractéristiques est elle relative au fait que les deux rails véhiculant la valeur booléenne d'un même bit sont mutuellement exclusifs. Ces deux caractéristiques confèrent une caractéristique intéressante à l'opération 'ou exclusif'.

a_1	a_0	A	$a_1 \oplus a_0$	$a_1 + a_0$
0	0	Invalide	0	0
0	1	Valide '0'	1	1
1	0	Valide '1'	1	1
1	1	Interdit	Inexistant	Inexistant

Tableau 4. Table de vérité de $a_1 \oplus a_0$

Considérons à titre d'illustration la table de vérité (Tableau 4) de l'opération consistant à effectuer le 'ou exclusif' des rails a_1 et a_0 véhiculant la valeur binaire de A. Comme illustré sur cette table de vérité, l'existence d'un état interdit permet de définir une bijection entre l'état de validité du bit A et la valeur de $a_1 \oplus a_0 \equiv a_1 + a_0$; bijection qui peut être exploitée pour redéfinir les conditions de retour à l'état invalide de la sortie des cellules double rail et qui justifie les correspondances simple rail double rail définies dans le tableau 3. Pour ce faire il faut, lors de la conception d'une cellule, définir '*n*' variables intermédiaires définies comme les 'ou exclusifs' des paires de rails véhiculant les '*n*' valeurs binaires d'entrées. Dans le cas de notre porte OR3, cela revient à définir trois valeurs intermédiaires comme suit :

$$\begin{aligned} U &= a_1 \oplus a_0 = a_1 + a_0 \\ V &= b_1 \oplus b_0 = b_1 + b_0 \\ W &= c_1 \oplus c_0 = c_1 + c_0 \end{aligned} \tag{23}$$

Ces trois valeurs intermédiaires étant définies, il est dès lors possible de simplifier les expressions (20) à (22) pour aboutir aux expressions suivantes :

$$z_1 = a_1 VW + a_0 b_1 W + a_0 b_0 c_1 \tag{24}$$

$$z_0 = a_0 b_0 c_0 \tag{25}$$

$$z^I = \overline{z}_1\overline{z}_0 = UVW \qquad (26)$$

qui peuvent aisément être implantées en considérant des limitations du nombre de transistors que l'on peut mettre en série à 3 ou 4.

Etape n°5 :

La cinquième étape consiste à dessiner le schéma des cellules complexes à partir des expressions obtenues lors de la troisième étape et ce en considérant que les valeurs intermédiaires sont fournies par l'environnement de la cellule. Ceci se fait de manière traditionnelle, et conduit dans le cas d'une porte OR3 au schéma de la figure 34.

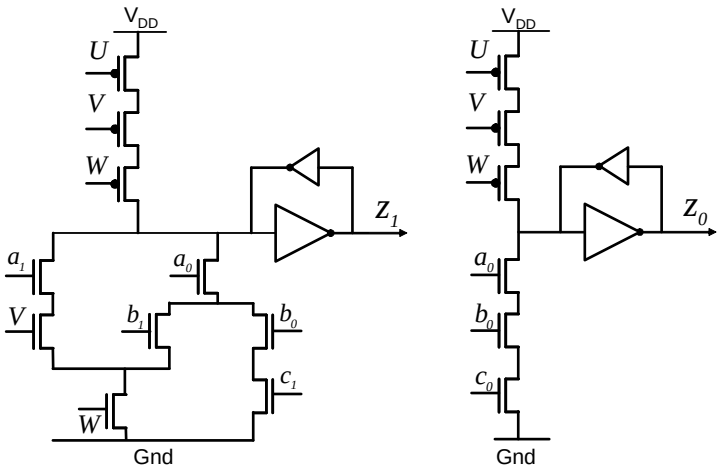


Fig. 34. Schéma partiel d'une cellule OR3 double rail

Etape n°6 :

L'étape finale permettant d'obtenir le schéma complet des cellules double rail consiste à implanter le calcul des valeurs intermédiaires, i.e. à réaliser les 'ou exclusifs' des paires de rails véhiculant les '*n*' valeurs binaires d'entrées. Cette fonction 'ou exclusif', étant équivalente à une simple fonction 'ou' du fait du codage, elle peut être réalisée à moindre coût à l'aide d'une simple porte OR2 simple rail de taille minimale. Dès lors, il est très facile d'obtenir le schéma complet (Figure 35) de la porte OR3 qui comporte 41 transistors. Ce schéma de la porte OR3 met en évidence que les signaux devront traverser au pire deux couches de transistors permettant l'évaluation de valeurs intermédiaires U, V et W, puis traverser l'une des deux portes complexes et enfin une des deux latches de sortie. Par conséquent, si les transistors sont proprement dimensionnés, la latence de la porte OR3 sera de l'ordre de quatre couches élémentaires de transistors, ce qui est peu par rapport aux implantations plus classiques de la OR3 comme nous le verrons dans la partie analyse des

performances.

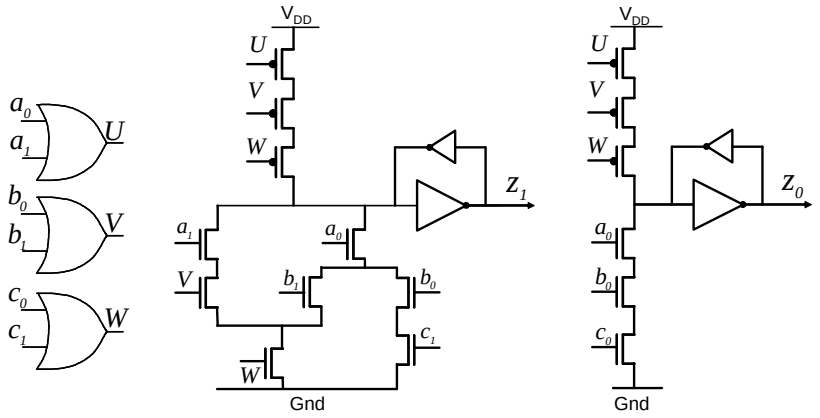


Fig. 35. Implantation pseudo statique d'une "OR3" double rail

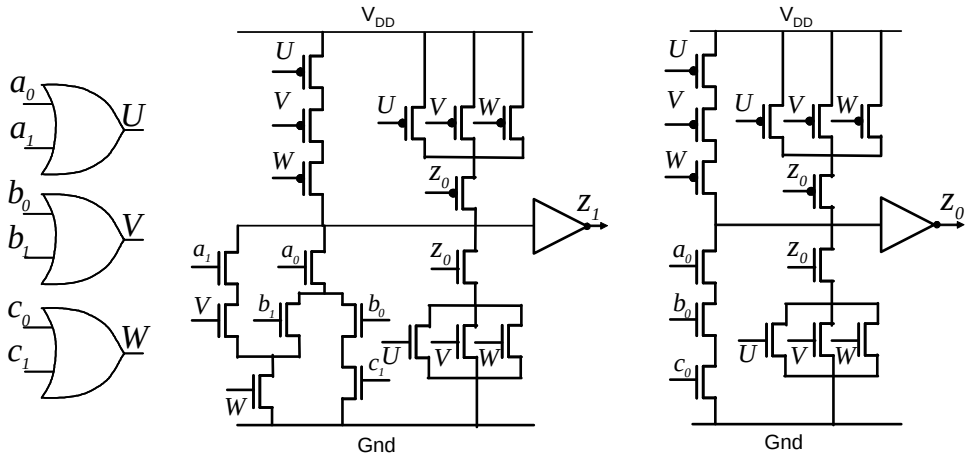


Fig. 36. Implantation statique d'une "OR3" double rail

Par la suite, cette implémentation pseudo statique de la fonctionnalité "OR3" peut également se mettre aisément sous forme statique comme l'illustre la figure 36. Cette transformation a cependant un prix qui est un accroissement du nombre de transistors. Toutefois, cet accroissement de surface peut amener sous certaines conditions des gains substantiels en vitesse et permettre de travailler avec des tensions d'alimentation plus faibles.

Des améliorations peuvent être apportées en terme de surface et consommation en éliminant les portes OR2 nécessaires à l'évaluation de l'état de validité. Dans ce cas, les signaux de validité doivent être fournis par les cellules en amont. Ces nouvelles considérations aboutissent à la définition d'une nouvelle topologie des cellules double rail qui est présentée sur la figure 37. On remarquera que les cellules complexes acceptent en entrée des bits codés sur deux rails Ai-1... Ti-1 et des bits codés sur un simples rail Ui-1 …Xi-1 respectivement où l'exposant donne le rang de la cellule générant le signal. Ces signaux U…X peuvent être soit

des signaux d'acquittement dans un environnement asynchrone, soit des signaux relatifs à la validité des bits Ai-1…Ti-1 générés par les portes nand2 des portes double rail amont.

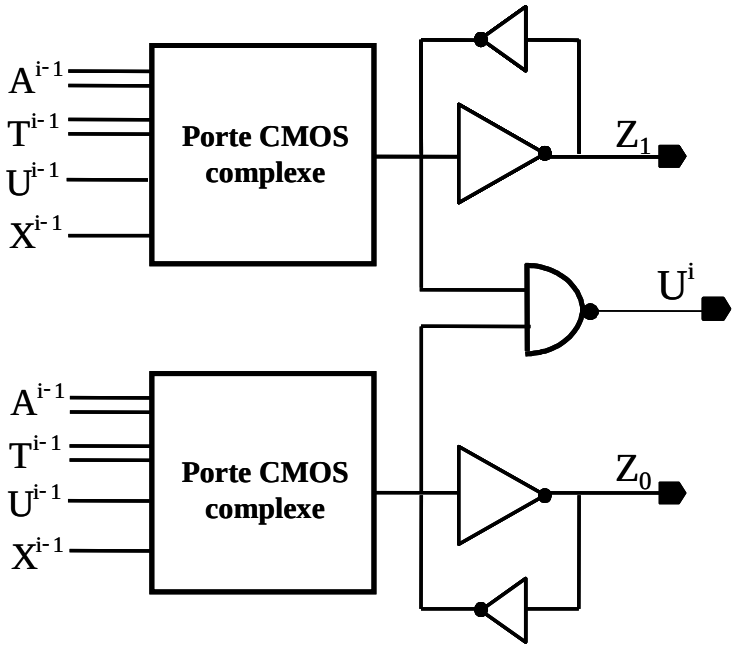


Fig. 37. Autre topologie d'une cellule double rail

Avec cette nouvelle topologie de la figure 37, la méthode de conception d'une cellule double rail reste identique à celle détaillée dans les paragraphes précédents à quelques différences près: suppression des portes OR2 pour l'état de validité des signaux et rajout d'une porte NAND2 pour la génération du signal de validité. Par voie de conséquence, une nouvelle version de l'expression (23) est définie, celle-ci est l'expression (27).

Pour introduire cette expression (27), les variables supplémentaires introduites dans l'expression (26) afin de réduire le réseau série de transistors P sont remplacées par les bits Ui-1…Xi-1 représentant l'état de validité des bits Ai-1…Ti-1 Les bits Ui-1…Xi-1 étant directement fournis par les portes nand2 (figure 38), des cellules amont double rail on peut écrire, dans le cas de la OR3 prise en considération:

$$\begin{aligned} U^{i-1} &= \overline{\left(\bar{a}_1 \cdot \bar{a}_0\right)} \\ V^{i-1} &= \overline{\left(\bar{b}_1 \cdot \bar{b}_0\right)} \\ W^{i-1} &= \overline{\left(\bar{c}_1 \cdot \bar{c}_0\right)} \end{aligned} \qquad (27)$$

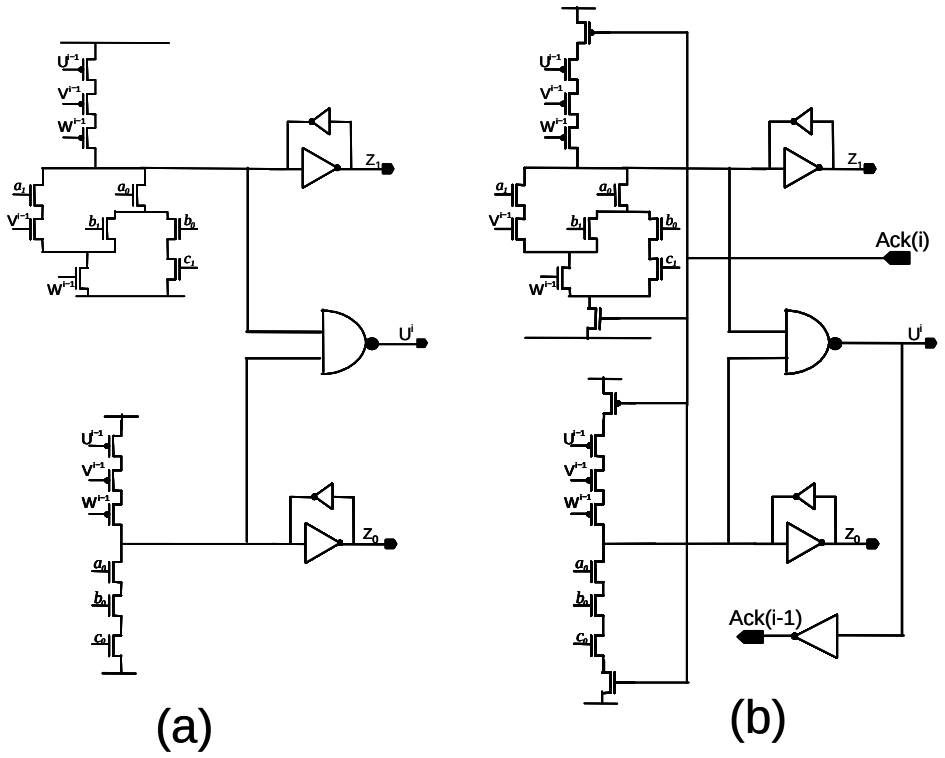


Fig. 38. (a) Implantation pseudo-statique d'une porte OR3 double rail (28 transistors)
(b) Half buffer complexe (OR3) (32 transistors)

La nouvelle topologie de la cellule OR3 obtenue, ainsi que celle de toutes les cellules que l'on obtient en utilisant la méthode proposée est très voisine de celle d'un half buffer. En d'autres termes ces cellules peuvent être modifiées comme l'illustre la figure 38b pour être utilisées comme des half-buffers complexes. A priori ceci devrait constituer un avantage important pour gérer les performances dans les circuits asynchrones double rail en permettant l'obtention de niveaux de pipeline très élevés [Sin00].

En conclusion, nous avons défini une méthode assez simple de mise en œuvre permettant la conception de cellules CMOS double rail pseudo-statiques et statiques. Par ailleurs, ces dernières peuvent être utilisées pour la conception de circuits synchrones et asynchrones.

III-5-b- Bibliothèque de cellules double rail: LISAL

En se basant sur la méthode de conception détaillée ci-dessus, nous avons constitué notre bibliothèque de cellules CMOS double rail: LISAL (Lirmm Synchronous Asynchronous Library). Par ailleurs, cette méthode offre aussi la possibilité d'effectuer une transcription rapide d'une bibliothèque standard en double rail.

Pour la constitution de la bibliothèque LISAL, nous avons procédé à un inventaire des

cellules les plus utilisées dans la réalisation des circuits sécurisés. Le tableau 5 présente une liste des fonctionnalités à implémenter en double rail.

AND2 / NAND2	Fonction « et »
AO22 / AO21	Fonction complexe « et - ou »
MUX21	Fonction « multiplexeur 2 vers 1 »
OR2 / NOR2	Fonction « ou »
XOR2 / XNOR2	Fonction « ou-exclusif »

Tableau 5. Quelques fonctionnalités de la bibliothèque LISAL

III-5-c-Analyse de performances

III-5-c-1-Performances en surface: estimation au 1er ordre

Afin de mieux quantifier le coût en nombre de transistors, nous avons dessiné les schémas de diverses fonctions élémentaires que l'on trouve traditionnellement dans les bibliothèques de fondeurs.

	[KUL05]	[RAZ04]	[GUI04]	SABL	Mace	AO222	Majorité	LISAL	Simple rail
[n]or2/[n]and2	19	37	112	18	19	64	56	22	6/4
[n]or3/[n]and3	36	64	224	24 / 36	25 / 38	128	112	28	8/6
[n]or4/[n]and4	54	101	336	28 / 54	29 /57	192	168	36	10/8
xor2/xnor2	19	42	80	18	19	68	60	24	12
xor3/xnor3	38	84	160	22 / 36	23 / 38	136	120	36	20
xor4/xnor4	57	126	240	26 / 54	27 / 57	204	180	52	32
AO21	36	64	336	24 / 36	25 / 38	128	112	29	8
AO22	54	101	336	28 / 54	29 / 57	192	168	38	10

Tableau 6. Tableau de comparaison des coûts d'intégration en nombre de transistors (surface)

Ce travail a été réalisé pour l'ensemble des styles d'implantation évoqués précédemment et notre bibliothèque LISAL de cellules double rail. Les résultats obtenus sont reportés dans le tableau 6 Il est à noter que dans certaines colonnes des cases sont grisées ; ceci permet d'identifier les schématiques de cellule comportant des réseaux série de 5 transistors et plus. Si de tels réseaux série de transistors étaient envisageables en technologie micronique, ce n'est plus le cas en technologie fortement submicronique. En effet, pour de telles technologies la tension d'alimentation est de l'ordre du volt voire moins, alors que les tensions de seuil

restent élevées afin de limiter les courants de fuite. C'est la raison pour laquelle nous avons également reporté dans les cases grisées le nombre de transistor nécessaire à la réalisation de ces fonctions à partir de portes à deux entrées

Ce tableau 6 met clairement en évidence que l'utilisation de logique double rail est coûteuse en terme de surface. En effet, l'intégration de fonctions booléennes simples, comme les [N]or et les [N]and, requiert entre 3 et 6 fois plus de transistors qu'en logique CMOS statique simple rail et ce même en utilisant les styles d'implantation double rail les plus compacts: [Kul05], SABL, LISAL. Ce coût de réalisation des primitives double rail se retrouve également au niveau circuit comme l'illustre la figure 39 qui reporte le nombre de transistors nécessaire à la réalisation de la structure de la figure 44 avec les différents styles d'implantation.

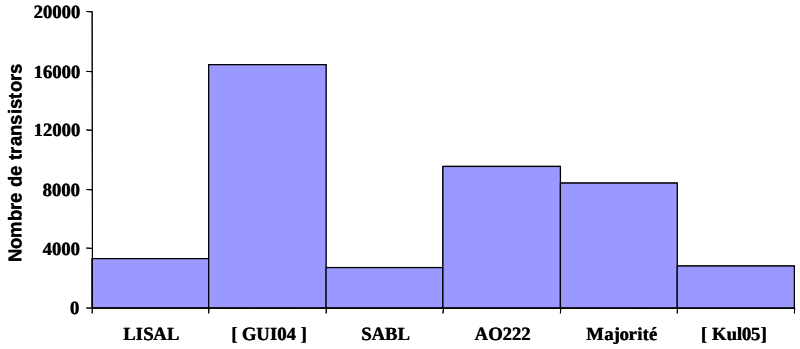


Fig. 39. Nombre de transistors nécessaires à la réalisation de la structure de la figure 44

La conclusion qu'on peut tirer de cette figure 39 est que le style de logique [Gui04] est celui le moins optimal en terme de surface tandis que les trois styles de logique: [Kul05], SABL, LISAL sont a priori les plus compactes. Toutefois, si ces trois styles d'implantation sont équivalents en terme de coût de réalisation, seules les cellules proposées dans [Kul05] et celles obtenues avec notre méthode sont adaptées à la conception de circuits asynchrones.

III-5-c-2-Délais de propagation et consommation

Il est difficile de comparer les performances en vitesse et consommation des divers styles de logique sans tomber dans le piège de la caractérisation. La méthode de comparaison que nous avons adopté s'appuie sur le protocole de simulation illustré sur la figure 40.

Ce protocole de simulation permet de capturer la sensibilité du délai de propagation et de la consommation à la rampe d'entrée et à la charge, qui sont les paramètres environnementaux les plus influents. Bien que simple, il présente des lacunes. En premier lieu, il néglige la

consommation additionnelle induite dans l'arbre d'horloge ou d'acquittement dans le cas des logiques à pré-charge. Cette consommation additionnelle est toutefois difficile à capturer au niveau cellule dans la mesure où elle dépend du nombre additionnel de buffers (taux d'activité de 1) utilisés dans l'arbre d'acquittement ou d'horloge, et donc dépend directement de la taille du circuit réalisé. Enfin ce protocole néglige le routage intra-cellulaire nécessaire pour interconnecter les portes AO222 (ou majorité) avec les portes OR et AND dans le cas des cellules double rail composites [Ren00a]. Toutefois les capacités de routage restent faibles si ces briques élémentaires constituant les cellules double rail sont placées les unes à proximité des autres.

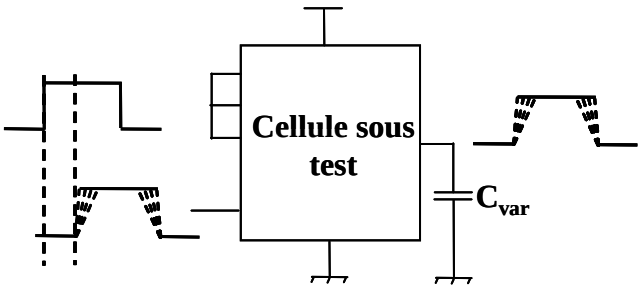


Fig. 40. Protocole de simulation

Les domaines de rampe et de charge que nous avons considéré lors de la comparaison des performances sont très supérieurs à ceux considérés dans [Mac04] puisque nous avons considéré des valeurs de rampes d'entrée et de charge en sortie comprises respectivement entre 10ps et 200ps et 5fF et 50fF.

La politique de dimensionnement utilisée pour réaliser les diverses fonctionnalités considérées ([N]OR2,3,4 ; X[N]OR2,3,4 ; AO21, AO22) a consisté à imposer, pour une même charge, des temps de transitions (C.V/I) identiques en sortie des cellules, i.e. à dimensionner les cellules de façon à ce qu'elles aient les mêmes possibilités en courant. Cette étape de dimensionnement nous a amené à utiliser les modèles analytiques de performances introduits dans [Auv00, Dag99].

Pour une meilleure appréciation des résultats, nous avons choisi de les présenter sous forme pire cas. Le pire cas est choisi en fonction de l'ordre d'arrivée des signaux d'entrée. Les figures 41 et 42 reportent les résultats que nous avons obtenus pour une porte 'OR2' double rail, sous différents styles d'implémentation et pour le passage de l'état invalide à un des deux états valides. Des résultats similaires ont été obtenus pour les autres fonctionnalités double rail.

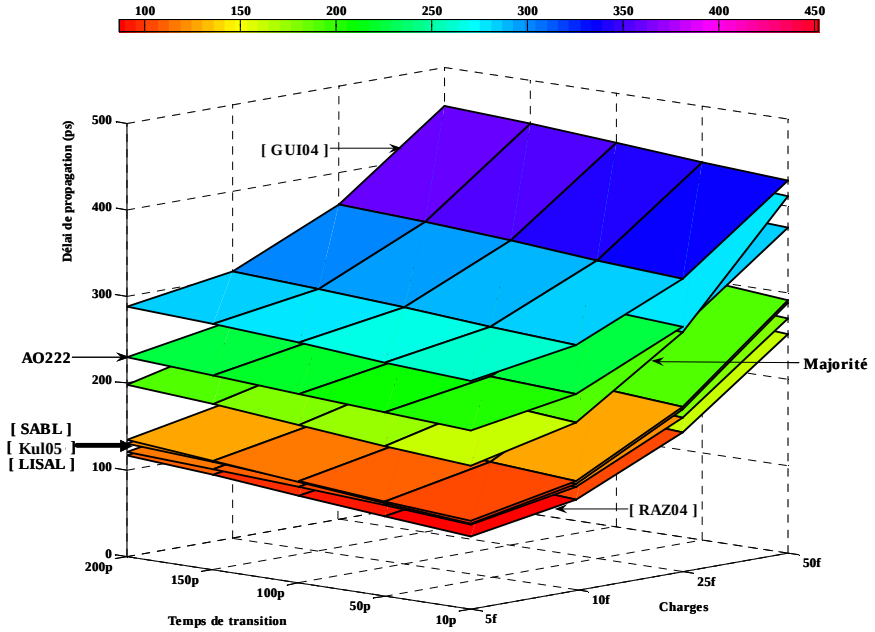


Fig. 41.Délais de propagation d'une porte 'OR2' double rail sous différents styles d'implémentation

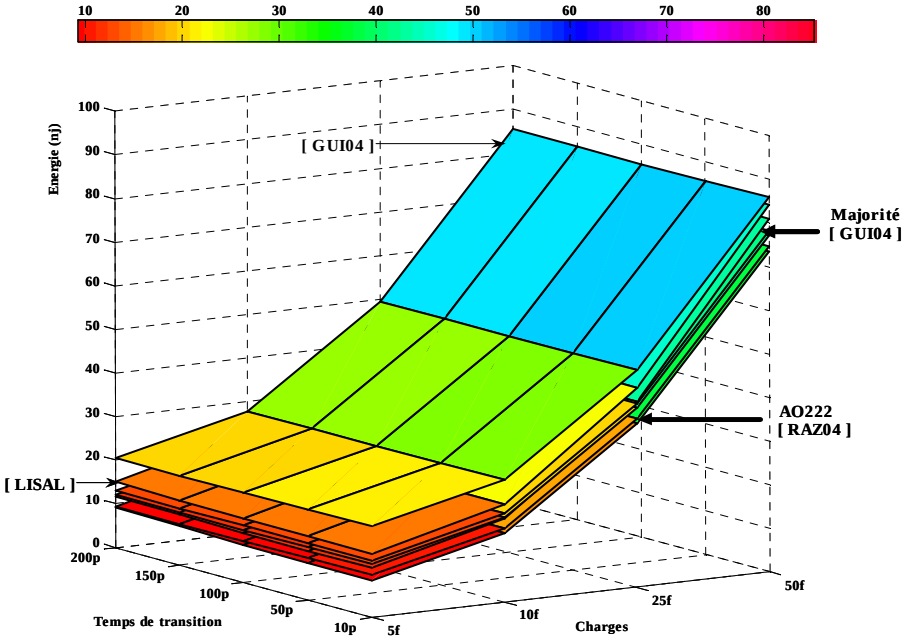


Fig. 42. Energie d'une porte 'OR2' double rail sous différents styles d'implémentation

En terme de délai de propagation, les cellules de LISAL sont 2 à 3 fois plus rapides que celles à base de porte complexe AO222. Toutefois, elles sont moins performantes par rapport à des cellules CMOS statiques mais restent acceptables dans la mesure où le critère de conception le plus important reste la sécurité. En terme de consommation, le gap est beaucoup moins important. En effet, comme illustré sur la figure 42, nos cellules double rail consomment

jusqu'à 1.5 fois plus que celles à base de porte complexe AO222.

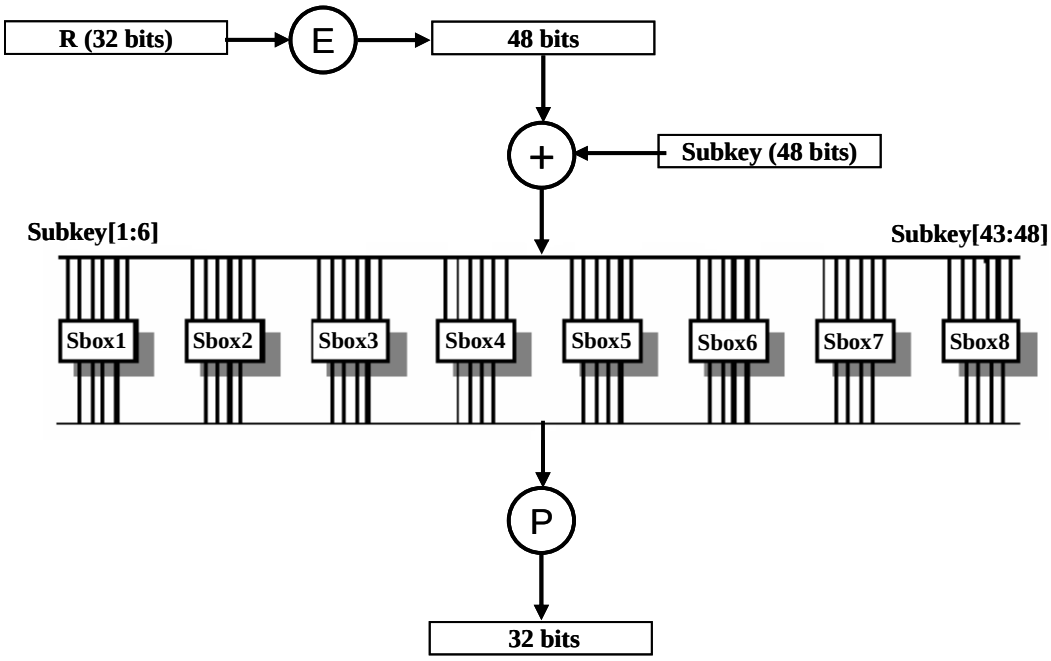


Fig. 43. La fonction F d'un algorithme DES

III-6- Analyse de robustesse

Dans cette partie nous allons présenter une évaluation de performance en terme de robustesse à l'attaque DPA de la bibliothèque LISAL. La même évaluation sera menée sur les autres styles d'implémentation.

III-6-a-Dispositif expérimental

Pour cette campagne d'évaluation, nous avons considéré un bloc sensible de l'algorithme de chiffrement symétrique DES: la fonction F (Figure 43). Rappelons que c'est en effet au niveau de ce module du DES que s'effectue le mélange entre les textes en clair et une partie de la clef secrète.

Toutefois pour des raisons de rapidité et de simplicité de mise en œuvre nous allons nous restreindre à un sous-ensemble de la fonction F notamment celui avec la Sbox1. Ainsi notre bloc expérimental se limite à celui présenté par la figure 44. On remarquera l'absence du module P. En effet, ce dernier étant exclusivement du câblage, il n'a aucun impact sur la consommation du circuit et donc sur les signatures DPA.

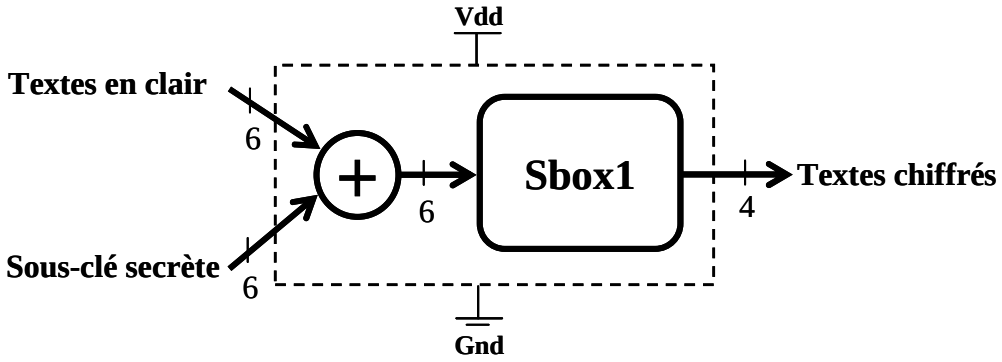


Fig. 44. Bloc expérimental pour le test de robustesse

III-6-b-Evaluations de robustesse et comparaisons

Dans ce paragraphe nous allons traiter la mise en œuvre de l'évaluation des différentes bibliothèques double rail en se basant sur le bloc expérimental de la figure 44. En l'absence d'outil de conception dédié nous avons adopté le flot de conception présenté par la figure 45. On remarquera que c'est un flot assez proche des flots standard avec quelques interventions manuelles entre les différentes étapes.

La première phase de l'implémentation du circuit consiste à définir sa description comportementale. Pour ce faire nous avons utilisé l'outil "Nclaunch" de Cadence. Une fois le fonctionnement du module vérifié, nous avons utilisé l'outil de synthèse "Ambit" pour arriver à une description au niveau porte (Verilog simple rail); c'est la phase de synthèse physique ou d'assignation technologique. Un analyseur syntaxique (parseur) a ensuite été développé en langage de programmation PERL pour la transcription du "Verilog simple rail": **1)** en Verilog double rail en vue d'un placement routage, **2)** en une netlist double rail compatible au simulateur "Eldo" et en technologie HCMOS9 130 nm de STMicroelectronics.

Il est important de noter que c'est au niveau de cette troisième phase de l'implémentation du circuit que s'effectue le choix de la bibliothèque double rail. Par exemple sur la figure 45, nous proposons l'utilisation de notre propre bibliothèque de cellules double rail: LISAL.

Pour l'évaluation de robustesse et de comparaison nous avons effectué des simulations d'attaque DPA sur les différents netlists générés précédemment. Notez que nous avons appliqué la même méthode de dimensionnement sur les différents styles d'implémentation et puis que nous n'avons pas pris en considération les capacités de routage.

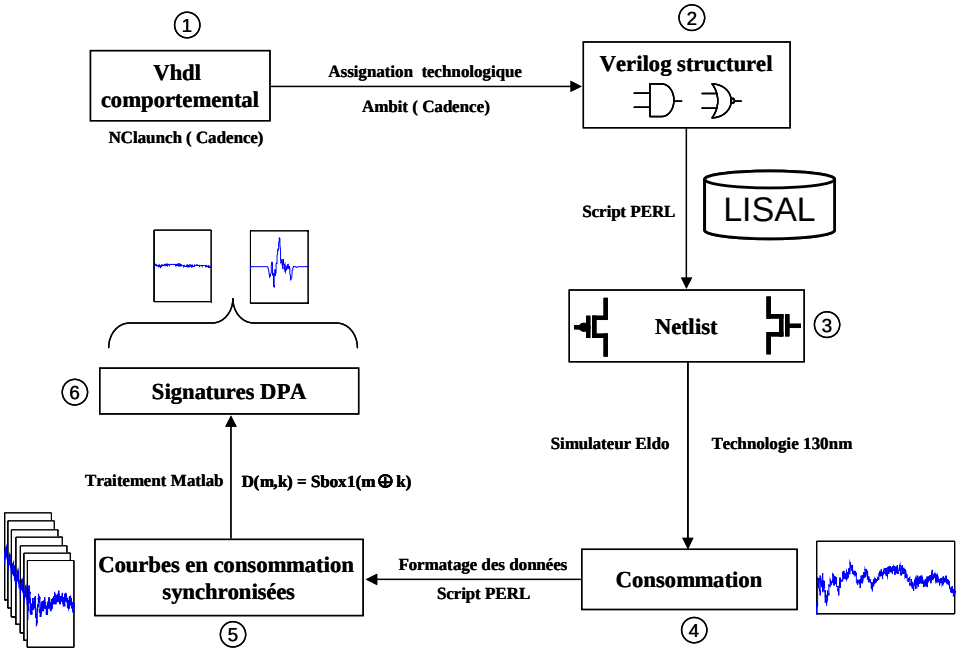


Fig. 45. Flot de l'analyse DPA

Logique "double rail" versus "simple rail"

Toutefois avant que de comparer les divers types de cellules double rail, nous avons réalisé par simulation des attaques DPA sur des implantations simple rail et double rail du bloc représenté sur la figure 44.

Comme l'illustre la figure 46, l'attaque DPA sur l'implémentation simple rail est une réussite tandis que celle menée sur l'implémentation double rail est un échec: la clef correspondante à la courbe dont l'amplitude est la plus importante n'est pas la bonne clef secrète. Comme attendu, il semble qu'une implantation double rail d'un circuit soit plus robuste à la DPA qu'une implantation simple rail.

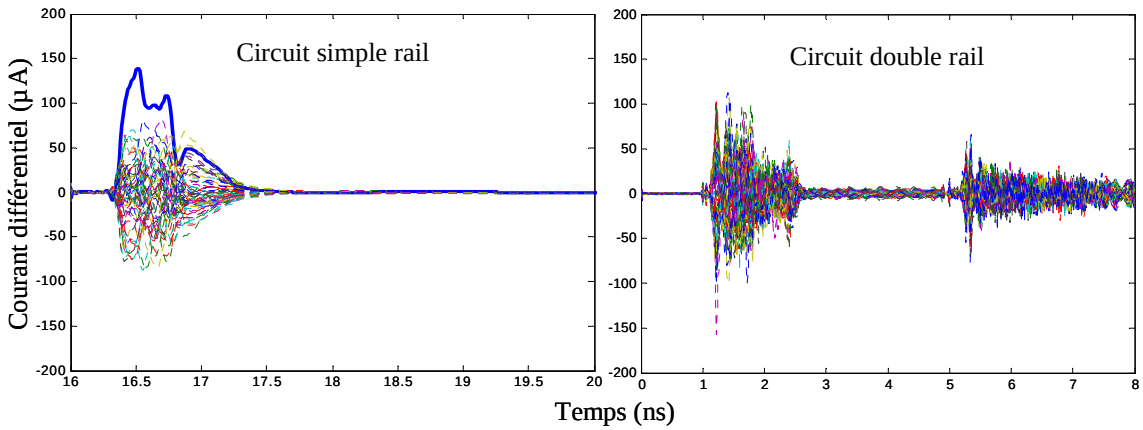


Fig. 46. Signatures DPA d'un circuit simple rail et double rail

Logique "double rail" versus "double rail"

Dans cette partie nous évaluons et comparons la robustesse à la DPA des différentes bibliothèques double rail dont la nôtre. Pour ce faire, nous avons effectué une attaque exhaustive sur les différents circuits: "64 messages en clair différents avec 64 propositions de clef".

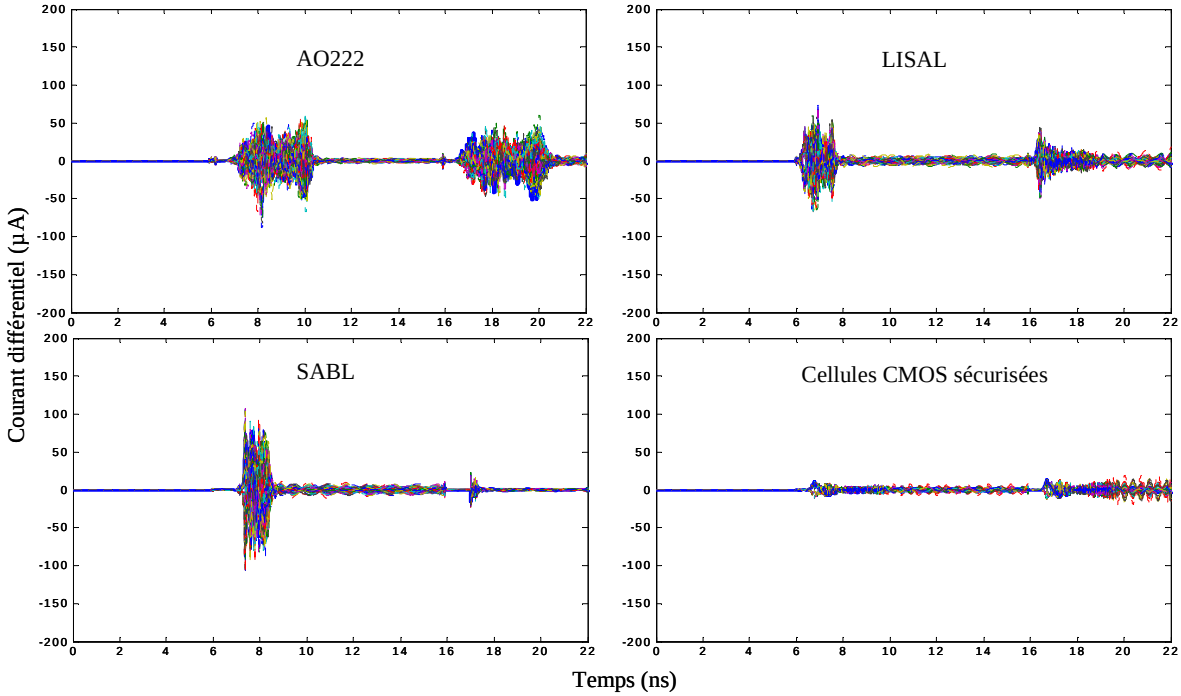


Fig. 47. Signatures DPA des différents styles d'implantation

La figure 47 reporte les signatures DPA obtenues en considérant les implantations de la structure figure 44 à base de portes complexe AO222, des cellules sécurisées introduites en III-4-b [Gui04], des cellules SABL et des cellules de LISAL.

Compte tenu des résultats obtenus, une première conclusion s'impose, peu importe la bibliothèque de cellules double rail utilisée, le circuit double rail semble être robuste aux diverses attaques DPA réalisées (sur les 4 bits de sortie). En effet, pour l'ensemble des quatre implantations considérées, la courbe correspondante à l'utilisation de la bonne clef secrète est complètement noyée à l'intérieur des autres courbes ce qui signifie que l'attaque DPA est un échec.

La deuxième conclusion concerne l'amplitude des signatures DPA qui est un des critères d'évaluation de robustesse des circuits sécurisés. En effet, plus les amplitudes de ces

signatures DPA sont faibles plus le circuit est robuste aux attaques DPA. De ce fait, avec une amplitude maximale de 30 µA le style d'implémentation à base de cellules sécurisées apparaît comme le circuit le plus robuste aux attaques DPA. Toutefois avec 75µA d'amplitude maximale, l'implémentation à base de notre bibliothèque LISAL a un niveau de robustesse satisfaisant. Bien que n'étant pas dédiée à la sécurité, l'implémentation à base de portes complexes AO222 se montre, au niveau schématique, également résistante aux attaques DPA.

III-7- Conclusion

Dans ce chapitre nous avons mis en évidence le problème des implémentations classiques de systèmes sécurisés vis-à-vis de la DPA qui est: "une consommation dépendante des données traitées". Nous avons ensuite justifié pourquoi la logique double rail apparaît comme une contre mesure intéressante aux attaques différentielles en courant.

Nous avons ensuite proposé une méthode de conception de cellules double rail permettant d'obtenir des cellules compactes, performantes et robustes à la DPA au niveau schéma. En effet, nous avons pu obtenir des cellules dont le coût d'intégration, évalué en nombre de transistors, est 2 à 4 moins élevé qu'avec une approche à base de cellule AO222. En terme d'énergie, les cellules proposées consomment jusqu'à 1.5 fois plus que les cellules à base de porte complexe AO222 et jusqu'à 2 fois moins que les cellules sécurisées de [Gui04].

Des évaluations de la robustesse à la DPA ont été effectuées et ont montré dans un premier temps que quelle que soit la bibliothèque double rail utilisée un circuit double rail est naturellement robuste aux attaques DPA et dans un deuxième temps que les circuits à base de cellules sécurisées [Gui04] affichent la meilleure robustesse aux attaques DPA avec des signatures DPA proches de zéro. Dans ce contexte, les cellules que nous proposons offrent un compromis sécurité – performance intéressant.

Toutefois, ces résultats et conclusions sont à nuancer dans la mesure où les capacités parasites de routage n'ont pas été prises en considération. En effet, nous le verrons dans le chapitre suivant, lorsqu'on les capacités parasites sont prises en compte, le niveau de robustesse d'une implémentation double rail aux attaques DPA diminue de manière significative.

Chapitre IV:

La logique double rail: impact de la synthèse physique et étude formelle

En l'absence d'outil spécifique, la phase de placement routage peut introduire des déséquilibres importants au niveau des nœuds différentiels ce qui peut faire perdre l'avantage de la logique double rail. A partir d'une modélisation analytique de la signature DPA, nous proposons dans ce chapitre, une étude formelle de la robustesse de la logique double rail en prenant en considération les capacités de routage. Par ailleurs, cette étude nous a permis de définir quelques métriques de robustesse à l'attaque DPA.

Chapitre IV : La logique double rail: impact de la synthèse physique et étude formelle

IV-1-Introduction

Les cellules CMOS simple rail, traditionnellement utilisées pour concevoir des circuits sécurisées, se caractérisent par des profils en consommation qui dépendent fortement des données manipulées.

Dans le chapitre III, nous avons établi que ce lien entre les données manipulées et le courant consommé par les cellules, constitue une des faiblesses qu'exploitent les analyses différentielles en courant. Nous avons également identifié que la logique double rail se positionne comme une alternative intéressante à la logique simple rail dans la mesure où ce lien entre les données et la consommation est plus ténu voire quasiment inexistant. En effet, le codage double rail associé à ce style de logique permet d'équilibrer de manière parfaite, et ce par conception, les profils en courant et donc de rendre les attaques DPA quasiment inopérantes.

Compte tenu de ce constat, nous avons donc défini une bibliothèque de cellules double rail orientée sécurité. Afin de valider celle-ci, mais également de quantifier la validité de l'approche double rail, nous avons réalisé, par simulation des attaques DPA sur un module sensible de l'algorithme DES et ce pour plusieurs types de cellules double rail.

A l'issue de ces attaques, il nous a été impossible d'identifier, de manière formelle, la clef secrète de chiffrement considérée et ce pour la plupart des logiques double rail considérées. Si ce résultat nous a permis de valider la pertinence de l'approche double rail, celui-ci doit toutefois être nuancé. En effet, ces attaques ont été réalisées en considérant des descriptions portes idéales, i.e. en négligeant les capacités parasites introduites lors de la phase de placement et routage des cellules.

Cette étape classique du flot de conception est une étape critique de la conception des circuits intégrés. En effet, les éléments parasites introduits lors de celle-ci, et notamment les capacités parasites de routage entre cellules, vont altérer de manière significative les performances des

circuits. Dans le contexte de la conception d'un circuit de chiffrement en logique double rail, cette étape est d'autant plus critique qu'elle peut introduire des déséquilibres de charge au niveau des sorties différentielles des cellules et par conséquent déséquilibrer au moins partiellement les profils en consommation des cellules, i.e. rendre leur consommation dépendante des données manipulées. L'impact du placement routage sur la robustesse à la DPA de la logique double rail constitue l'objet de ce chapitre.

Plus précisément, dans un premier temps, nous allons étudier l'impact que peut avoir le choix d'un style de cellule double rail sur le placement routage et donc sur la robustesse d'un circuit à la DPA. Dans un deuxième temps l'impact à proprement parler du placement routage sur la robustesse de la logique double rail à la DPA sera étudié. Cette étude de l'impact du placement routage sur la robustesse de la logique DR s'appuiera sur une modélisation au premier ordre des profils en courant des cellules CMOS qui nous permettra de dégager des métriques de conception mais également d'identifier l'espace de conception dans lequel la logique DR peut être considérée comme robuste à la DPA.

IV-2- Placement routage et choix de librairie

IV-2-a-Définition de placement routage

Lors de la conception d'un circuit intégré, l'étape de placement et routage est l'étape durant laquelle les cellules sont interconnectées entre elles de manière à réaliser la fonction désirée. Les interconnexions sont généralement réalisées avec des polygones de métaux qui relient physiquement des broches (Pins) d'entrée/sortie de cellules définies lors de la conception celles-ci, i.e. lors du développement des bibliothèques.

Cette étape de placement routage est généralement effectuée à l'aide d'outils de conception assistée par ordinateurs comme Silicon Ensemble ou First Encounter de la société Cadence [Cad06]. Les algorithmes de placement routage mis en œuvre par ce type d'outils visent généralement à réduire la distance d'interconnexion et donc la capacité de routage entre les différentes instances constituant le circuit et ce afin de minimiser l'impact du routage sur les performances temporelles des circuits ou encore leur consommation.

La conception de circuits double rail restant une manière marginale de concevoir les circuits, il n'existe pas de flots de conception industriels spécifiques et donc aucun outil de placement

routage ad hoc. Les concepteurs de tels circuits sont donc contraints de placer et router leurs circuits en utilisant les outils disponibles sur le marché.

Ceci est très pénalisant dans la mesure où ces outils ne permettent pas d'optimiser les circuits selon les directions souhaitées. C'est la raison pour laquelle des méthodes et outils de placement routage ont été proposées [Tir04, Bou05a]. Toutefois, ces méthodes, qui permettent d'améliorer la qualité du placement routage des circuits selon des métriques propres à la sécurité, sont généralement limitées et ne constituent que des solutions palliatives à l'existence d'outils de placement et routage dédiés.

IV-2-b-Choix de la librairie et placement routage

Malgré ce constat et dans le contexte du développement de circuits double rail avec un flot de conception classique, trois alternatives de placement et routage des cellules peuvent être identifiées selon les méthodes de conception et de synthèse envisagées. Plus précisément, ces trois alternatives sont intrinsèquement liées à la manière de concevoir les primitives double rail. Elles conduisent à des types de placement routage différents : éclaté, semi éclaté et non éclaté. Cette terminologie est adoptée dans le reste de ce chapitre pour différencier ces trois alternatives.

IV-2-b-1-Placement routage éclaté :

Quelle que soit la fonctionnalité devant être réalisée en double rail, celle-ci peut être réalisée à partir de cellules CMOS simple rail [Ren00a, Rig03]. A titre d'illustration, dans la figure 48, nous avons reporté le schéma d'une porte or2 double rail réalisée à partir de différentes cellules simple rail.

Il est donc envisageable dans une première approche, de concevoir des circuits double rail uniquement à partir de cellules simple rail et plus particulièrement à partir des bibliothèques développées par les fondeurs. A titre d'exemple, cela peut être réalisé en traduisant des netlists Verilog de circuits simple rail en des netlists Verilog de circuits double rail. Cette approche de la conception de circuits double rail présente certains avantages. Sa simplicité de mise en œuvre, dans la mesure où elle ne requiert pas le développement de cellules spécifiques, est l'un d'entre eux. Toutefois, elle conduit à des implantations matérielles des circuits particulièrement complexes et sous optimales ne serait ce qu'en terme de surface et consommation comme on peut le constater sur la figure 48.

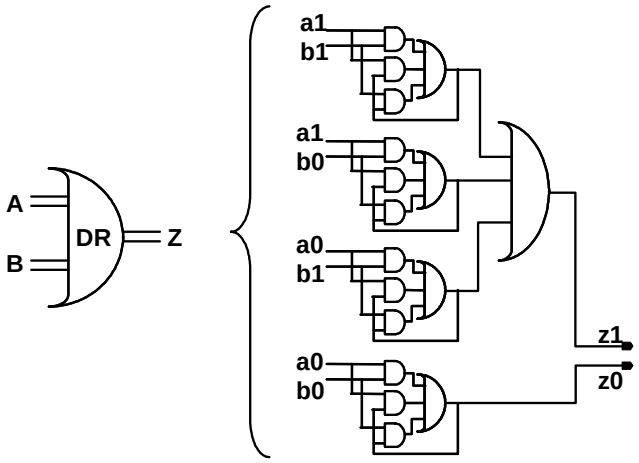


Fig. 48.Réalisation d'une porte OR2 double rail à partir de cellules simple rail

Par ailleurs, cet inconvénient majeur n'est pas le seul lorsque l'on se place dans le cadre de la conception de circuits sécurisés. En effet, adopter ce style d'implantation des cellules double rail conduit à des placements routages éclatés. La figure 49 représente de manière symbolique le placement routage de cellules simple rail permettant d'obtenir une cellule double rail. Comme on peut le constater, du fait de la répartition des primitives simple rail sur deux rangées, les interconnexions entre les broches de cellules sont de longueurs différentes et peuvent être réalisées en métal 1, 2 ou 3.

En terme de charges cela se traduit par une disparité des valeurs de capacités parasites introduites par l'outil de placement routage et ce au sein même du schéma de la cellule. En terme de sécurité cela pose problème dans la mesure où ces capacités parasites, dont on ne peut pas maîtriser les valeurs, peuvent déséquilibrer les profils de consommation de la cellule et les rendre caractéristiques des données manipulées.

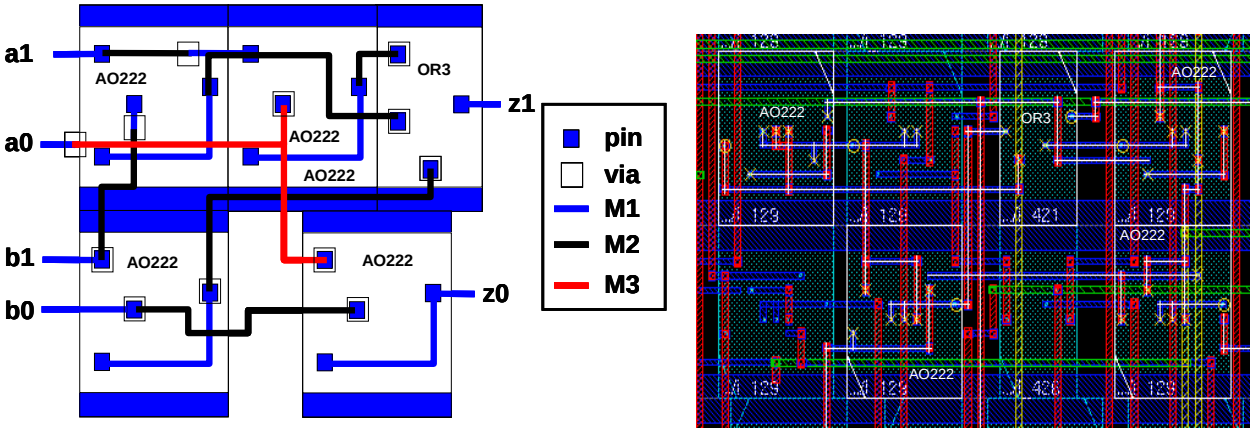


Fig. 49. Placement routage symbolique de cellules simple rail afin d'obtenir une cellule OR2 double rail et vue du layout

IV-2-b-2-Placement routage semi-éclaté :

Comme on peut le constater sur la figure 48, réaliser une cellule double rail avec des portes simple rail revient à définir deux cônes logiques disjoints de cellules simples. Afin de réduire les coûts de réalisation des cellules double rail, il est possible de développer des cellules complexes simple rail réalisant les cônes logiques les plus fréquemment utilisés [Raz03, Raz05].

Dans le cas de la fonctionnalité OR2 considéré précédemment cela revient à développer deux cellules simple rail (Muller 2 et COR222) dédiées à la conception de circuits double rail comme l'illustre la figure 50. Plus généralement, il est possible de développer des bibliothèques de cellules complexes réalisant les fonctionnalités les plus souvent utilisées pour la construction de primitives double rail [Raz03].

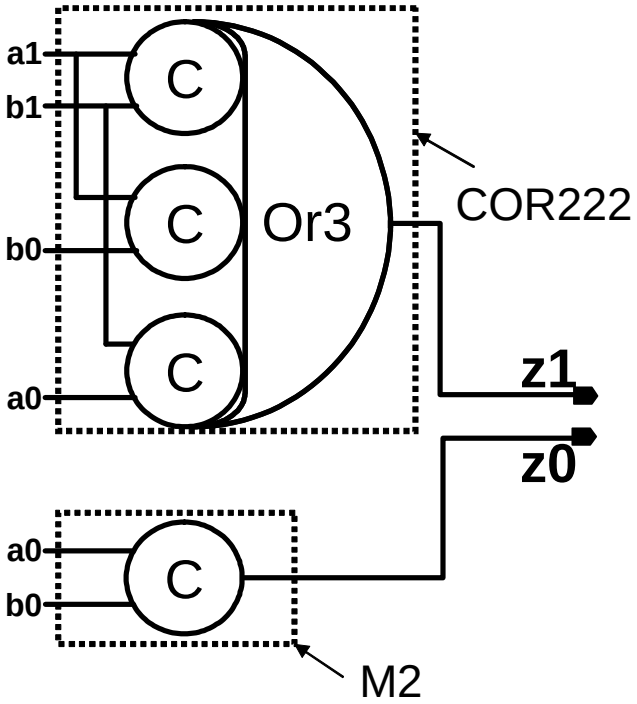


Fig. 50. Conception d'une cellule double rail à partir de cellules simple rail

Outre la réduction du coût d'intégration des principales primitives double rail, l'adoption de ce type de bibliothèques offre d'autres avantages et ce notamment dans le cadre de la conception de circuits robustes aux attaques différentielles en courant. En effet, l'adoption de ce type d'approche permet de réduire de manière significative l'impact du placement routage sur les profils en courant. Plus précisément, cette approche aboutit nécessairement à l'obtention de placement routage des cellules semi-éclaté. La figure 51 donne un exemple

caricatural de placement routage semi-éclaté. Plus précisément, ce schéma montre un placement routage possible des primitives simple rail COR222 et M2 (Muller) réalisant une porte OR2, tout comme dans le cas de la figure 49.

Comme on peut le constater en comparant les figures 49 et 51, le nombre d'interconnexions entre cellules est beaucoup plus faible dans le cas d'un placement routage semi-éclaté. De manière générale, cette approche conduit au niveau circuit à une réduction significative du nombre d'interconnexion et de la capacité totale de routage. A titre d'illustration, l'utilisation de la bibliothèque TAL [Mau03] pour concevoir un AES a permis de réduire la surface (surface du coeur) de 22% et de réduire le nombre d'interconnexions de 20% par rapport à une conception réalisée uniquement avec des cellules standard [Bou05a].

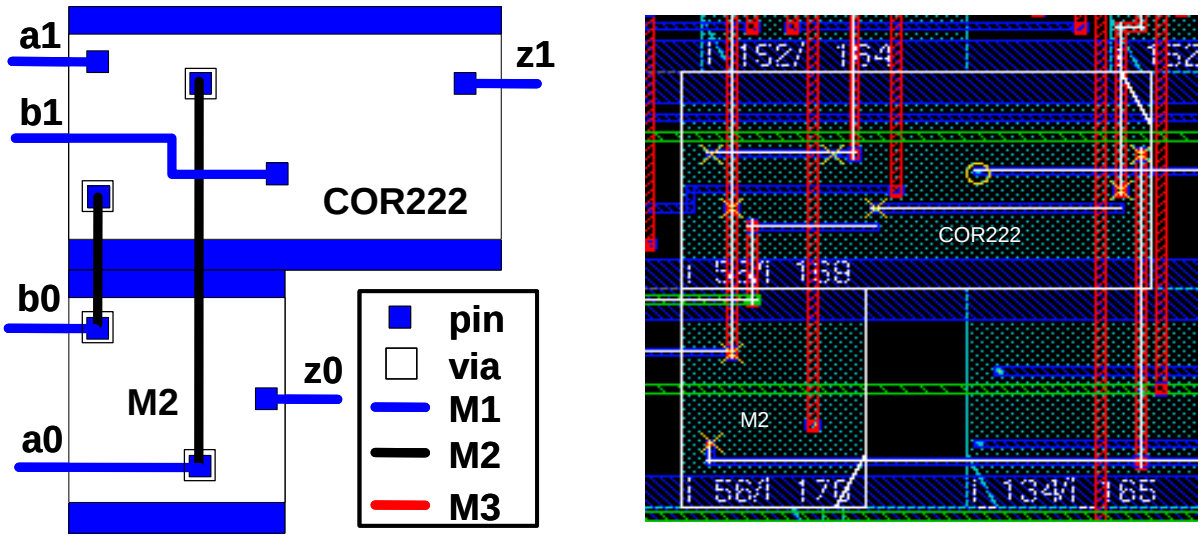


Fig. 51. Placement routage symbolique de cellules simple rail complexes afin d'obtenir une cellule OR2 double rail.

Dans le cadre de la conception de circuits sécurisés, la réduction du nombre d'interconnexion et de la surface constitue un avantage significatif. En effet, ceci réduit l'impact du placement routage sur les profils en courant et donc accroît la robustesse du circuit à la DPA. En d'autres termes, adopter cette approche permet de réduire l'impact introduit par le manque de maîtrise de la phase de placement routage en déportant une grande partie du problème au niveau de la conception des cellules.

IV-2-b-2-Placement routage non éclaté :

Le constat que nous venons de dresser à propos de l'utilisation de cellules simple rail pose bien évidement la question suivante : "Est-il finalement judicieux de concevoir des cellules double rail dédiées comme celles introduites dans ce manuscrit et celles proposées par Kris Tiri [Tir03] ou Konrad J. Kulikowski [Kul05]". En d'autres termes, est-il intéressant de réduire au mieux l'impact du placement routage en déportant le problème sur la phase de conception des cellules.

Dans la pratique, l'utilisation de bibliothèques de cellules double rail permet l'obtention de placement routage non-éclaté. La figure 52 donne, en guise d'illustration, un placement routage possible de trois cellules OR2 double rail.

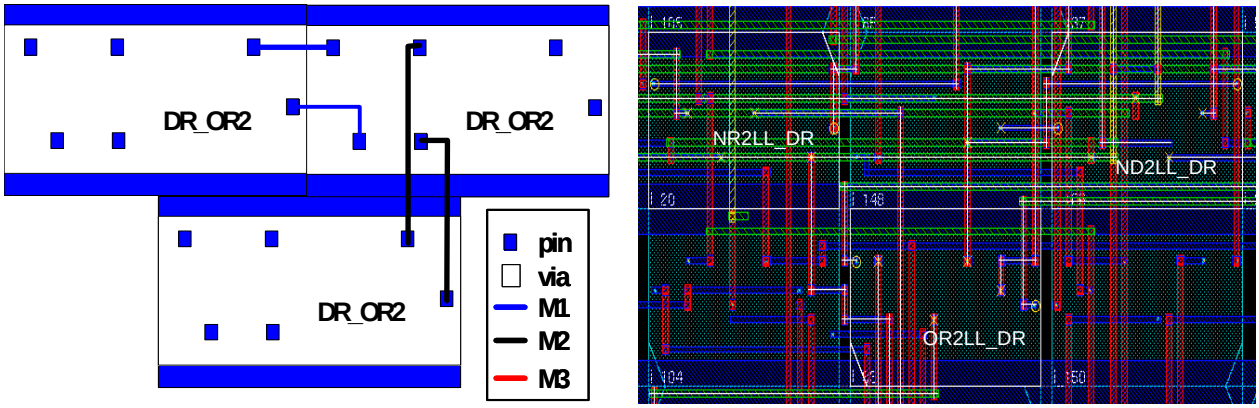


Fig. 52. Placement routage de cellules OR2 double rail.

Comme on peut le constater, cette approche permet de réduire encore un peu plus le nombre d'interconnexions par rapport à une approche 'semi éclaté'. Comme pour l'approche semi éclaté, cette réduction du nombre d'interconnexions limite l'impact du placement routage sur les profils en courant et plus précisément les écarts que l'on peut constater entre les profils en courant avant et après placement routage sont moins importants.

IV-2-b-3-Comparaisons des types de placement

Afin de quantifier plus précisément l'impact du choix de librairie sur le placement routage des circuits double rail, nous avons placé et routé, en utilisant First Encounter [Cad06], la structure de la figure 53. Pour ce faire, nous avons développé tous les gabarits des cellules (LEF) nécessaires au placement routage selon les trois approches considérées.

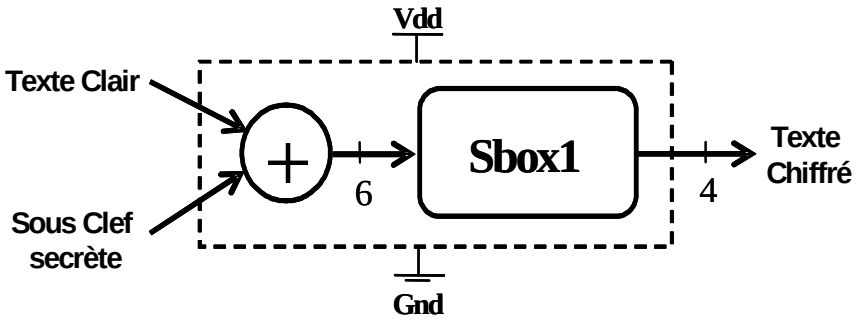


Fig. 53. Structure considérée pour la comparaison des approches eclatées, semi éclatées et non éclatées de placement routage.

A l'issue de ces étapes de placement routage, nous avons évalué, pour l'ensemble des paires différentielles de toutes les primitives double rail ou amalgames équivalents de cellules simple rail, la différence $|C_T-C_F|$ entre les capacités des équipotentielles véhiculant les valeurs logiques '1' et '0'. La figure 54 reporte la proportion cumulée du nombre de nœuds différentiels se caractérisant par un déséquilibre de charge inférieur ou égal à $|C_T-C_F|$.

Comme on peut le constater sur cette figure, dans le cas d'un placement routage de type non éclaté, le pourcentage de nœuds différentiels atteint 90% pour un déséquilibre de charge d'environ 1.3fF alors que ce pourcentage est atteint pour des déséquilibres de 2.5fF et 6.8fF pour des placement routage semi éclaté et éclaté respectivement. On notera toutefois que quelque soit l'approche considérée, il existe toujours quelques nœuds différentiels se caractérisant par des déséquilibres relativement importants, i.e. de l'ordre de 10fF. Afin de quantifier l'importance de ces déséquilibres, les valeurs moyennes et maximales des capacités des équipotentielles sont reportées dans le tableau 7.

	LISAL (P&R non-éclaté)	**TAL (P&R semi-éclaté)**	**DIMS (P&R éclaté)**	**AO222 (P&R éclaté)**
Moyenne des capacités de routage (fF)	**4.63**	**6.61**	**11.14**	**16.87**
Moyenne des rapports C_T/C_F	**1.82**	**1.66**	**1.52**	**1.49**
Ecart type des rapports C_T/C_F	**1.21**	**1.19**	**0.63**	**0.88**
Maximum parmi les rapports C_T/C_F	**7.15**	**9.88**	**3.80**	**7.75**
Moyenne des différences C_T-C_F (fF)	**0.82**	**1.21**	**2.21**	**2.92**
Ecart type des différences C_T-C_F	**0.39**	**1.14**	**2.46**	**2.79**
Maximum parmi les différences C_T-C_F (fF)	**2.04**	**6.49**	**17.07**	**16.38**

Tableau 7. Valeurs moyennes et maximales des capacités des équipotentielles

Ce résultat démontre clairement l'intérêt que l'on peut avoir à utiliser des cellules double rail plutôt que de réaliser les primitives double rail à partir de cellules simple rail, et ce d'autant plus que les nœuds de routage internes aux primitives double rail (placements semi-éclaté et éclaté) ne sont pas considérés au cours de cette analyse.

Malgré ce constat, la figure 54 montre que, quelque soit le style de placement routage adopté, il reste toujours des paires différentielles pour lesquelles le déséquilibre des charges est important.

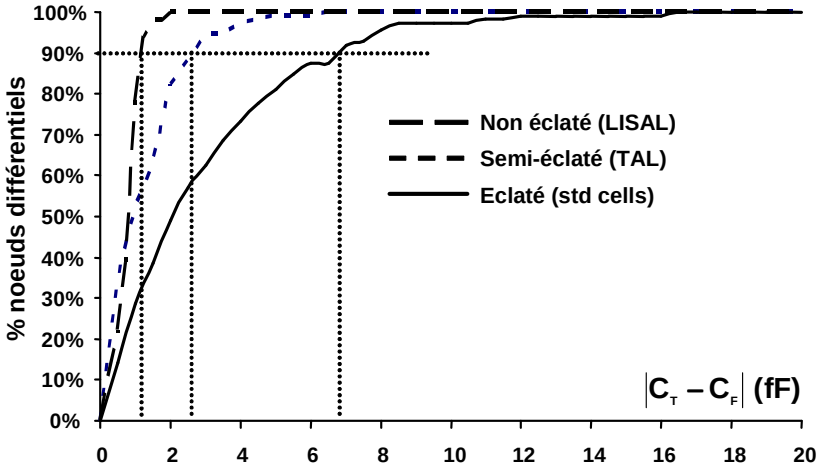


Fig. 54. Pourcentage cumulé de nœuds différentiels dont le déséquilibre de charge est inférieur ou égal à $|C_T-C_F|$.

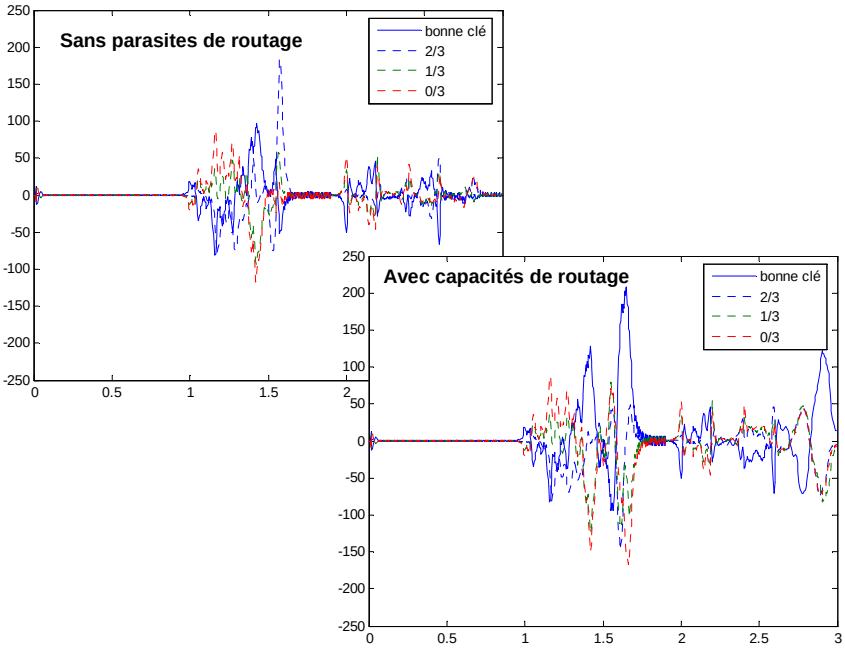


Fig. 55. Signatures DPA de la structure figure 57 avant et après placement routage 'éclaté'

IV-3- Méthodes de placement et routage spécifique

En l'absence d'outil de conception dédié, la phase de placement routage est susceptible d'introduire des déséquilibres de charge au niveau des nœuds différentiels. Comme l'illustre la figure 55, la considération des capacités de routage, avec des déséquilibres de charge importants, peut faire en sorte qu'un circuit double rail ne soit plus robuste aux attaques DPA. Afin d'apporter une solution à ce problème, de nombreuses méthodes de placement routage permettant de minimiser ces déséquilibres ont été proposées [Tir04, Tir05, Bou05a].

Dans [Bou05a], une stratégie de placement routage hiérarchique est proposée. Celle-ci consiste à contraindre le placement du circuit de sorte que les cellules simple rail, dont l'amalgame réalise des primitives double rail, soient placées à proximité les unes des autres et ce afin de limiter autant que possible les déséquilibres de charges.

Le principal avantage de cette solution réside dans le fait que sa mise en œuvre ne nécessite pas le développement d'outils spécifiques. Toutefois son efficacité est limitée puisqu'elle ne garantie pas l'obtention de paires différentielles parfaitement équilibrées. Il est donc nécessaire de s'assurer qu'aucun déséquilibre important de charges ne subsiste après routage détaillé du circuit, et le cas échéant d'apporter des modifications adéquates afin de les réduire. Ce post traitement du circuit nécessite donc le développement de critères permettant d'identifier les déséquilibres susceptibles d'être exploités lors d'attaques DPA.

Une autre approche a été introduite par Kris Tiri dans [Tir05] où un flot de conception de circuits sécurisés et plus particulièrement robuste aux attaques DPA est introduit. Les principales originalités de ce flot de conception sont l'utilisation de logique double rail (et plus spécifiquement la logique WDDL pour Wave Dynamic Differential Logic) et d'une méthode de routage innovante appelée routage différentiel.

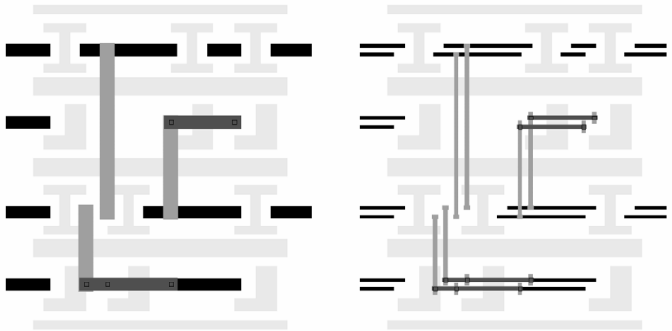


Fig. 56. Routage différentiel proposé par Kris Tiri

Plus précisément, après avoir réalisé un placement des cellules standard, le routage des nœuds différentiel est effectué en deux étapes. Dans une première étape, les paires différentielles sont traitées comme de simples équipotentielles que l'on route avec des pistes de métal larges (fat wires) comme cela est illustré sur la figure 56. Suite à cette première étape, ces larges pistes de routage, sont découpées en deux interconnexions distinctes. Toutefois, cette approche n'est pas suffisante, et des pistes blindage 'shielding' peuvent être insérées entre les paires différentielles afin de s'affranchir des variations de la capacité des lignes d'interconnexions induites par le phénomène de diaphonie. Si cette méthode de placement routage permet d'équilibrer de manière efficace les paires différentielles, elle est particulièrement coûteuse en terme de surface et en temps cpu. A titre d'illustration, dans [Bou05], il est reporté que le surcoût en surface est de l'ordre de 300% environ dans le cas d'un AES.

A ce stade de l'étude, nous pouvons conclure que la phase de placement routage des cellules a été identifiée comme une étape pouvant introduire des corrélations importantes entre la consommation d'un circuit et les données manipulées et ce même si l'on utilise un style de placement routage éclaté. Les corrélations introduites sont principalement dues à l'introduction de déséquilibres de charge au niveau des paires différentielles des cellules double rail. Afin de résoudre ce problème deux méthodes de placement routage ont été proposées : le placement contraint et le routage différentiel [Tir04] qui visent toutes deux à réduire l'impact du routage sur la robustesse du circuit et en essayant de équilibrer autant que possible les charges des sorties différentielles des cellules double rail. Ce constat nous soulève deux questions auxquelles nous allons apporter des débuts de réponses dans les paragraphes suivants :

- est-il vraiment nécessaire d'équilibrer de manière parfaite les charges présentes sur les nœuds différentiels des cellules? En d'autres termes, quels niveaux de déséquilibre peut-on tolérer ?
- quelle est la contribution d'une cellule à la signature DPA d'un circuit complet ?

IV-4- Etude formelle de la robustesse de la logique double rail

Afin d'apporter des éléments de réponse à ces deux questions, nous allons dans ce paragraphe, étudier la contribution d'une cellule à la signature DPA d'un bloc combinatoire, et modéliser l'impact du routage sur la contribution à la signature DPA d'une cellule et ce en

fonction du contexte dans lequel cette cellule opère. Pour ce faire, nous nous appuierons sur une modélisation analytique au premier ordre du courant de commutation des cellules CMOS.

IV-4-a-Contribution d'une cellule à la signature DPA

Afin d'identifier la contribution d'une cellule DR à la signature globale d'un bloc combinatoire, nous avons effectué une campagne de simulations. Plus précisément, nous avons effectué des attaques DPA de la structure représentée sur la figure 57 en se focalisant sur le bit cible Z (sortie du dispositif). On remarquera que cette structure sur la figure 57 est un modèle réduit d'une SBox que l'on rencontre dans les circuits DES.

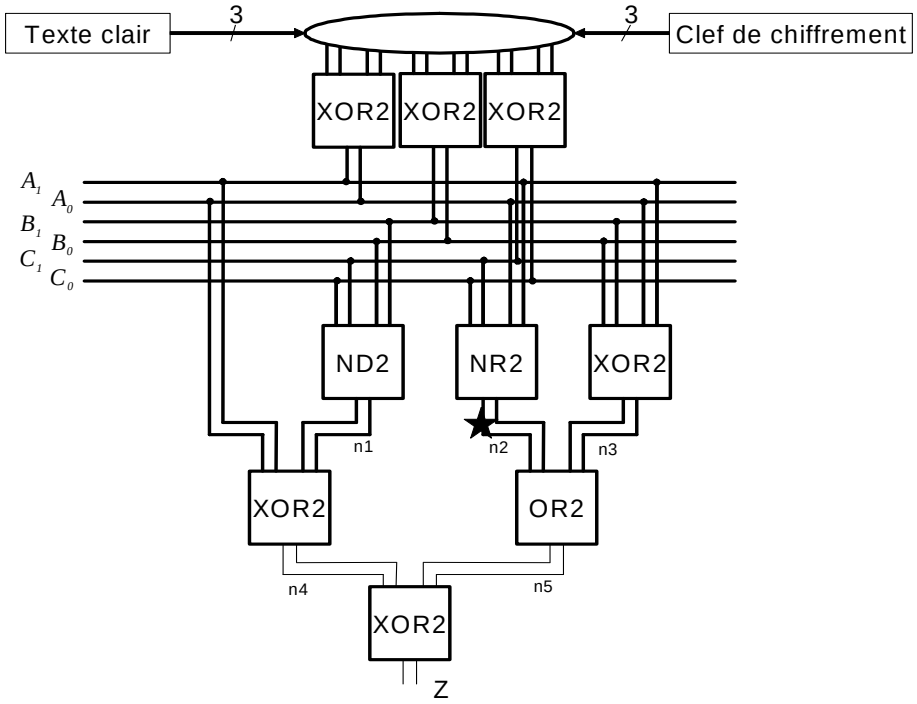


Fig. 57. Micro-circuit de chiffrement considéré

La démarche que nous avons adoptée pour identifier la contribution d'une cellule à la signature DPA d'un bloc complet a été la suivante :

- dans un premier temps, nous avons réalisé des attaques DPA sans qu'aucun déséquilibre n'ait été introduit.
- Dans un deuxième temps, des attaques DPA ont été effectuées après avoir déséquilibrer la paire différentielle n2, par simple introduction d'une capacité C sur un des noeuds de celle-ci.

Les valeurs de cette capacité C que nous avons considérées étaient comprises entre 1fF et

200fF. La figure 58 reporte les résultats que nous avons obtenus. Comme on le mentionne dans la légende, cette figure reporte deux type de courbes : la signature DPA du micro-circuit de chiffrement et les profils en courant différentiels Δi(t) de la cellule double rail NOR2 contrôlant la paire différentielle n2, et des portes OR2 et XOR2 situées en aval de n2. Par profil en courant différentiel, nous entendons ici la différence entre les courants de consommés par une cellule pour établir des valeurs logiques '1' et '0' respectivement sur sa sortie.

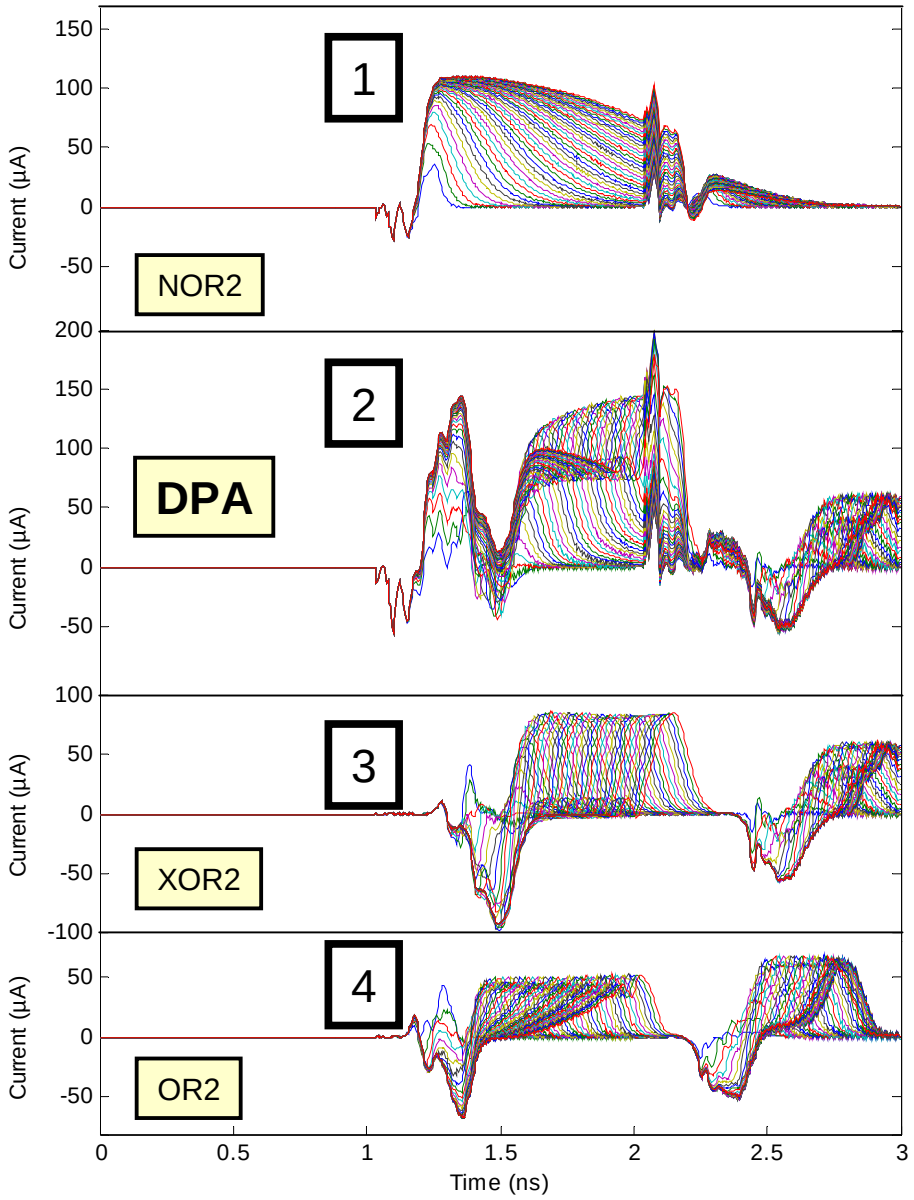


Fig. 58. Profils différentiels en courant des portes NOR2, XOR2 et OR2 et signature DPA complète du micro-circuit de chiffrement pour l'ensemble des valeurs de C considérées

L'analyse de la figure 58, on pourra s'en convaincre avec une règle et un crayon, met en

évidence que la signature DPA du micro-circuit de chiffrement $S_{DPA}(Z)$ n'est autre que la combinaison linéaire des profils différentiels en courant suivante :

$$S_{DPA}(Z) = \Delta i(t)_{NOR\,2} + \Delta i(t)_{XOR\,2} - \Delta i(t)_{OR\,2} \tag{28}$$

Cette observation est parfaitement cohérente avec les résultats présentés dans [Bou05a], et en s'appuyant sur ces derniers nous pouvons donc émettre l'hypothèse suivante : la signature DPA d'un bloc combinatoire n'est autre qu'une combinaison linéaire pondérée des profils différentiels en courant de ses éléments constitutifs. La signature DPA d'un bloc combinatoire doit donc pouvoir s'écrire sous la forme suivante :

$$S_{DPA}(Z) = \sum_{p \in P} \alpha_p \cdot \Delta i(t)_p \tag{29}$$

où P est l'ensemble des portes qui constitue le bloc combinatoire, α_P est un coefficient de pondération qui prend en compte l'activité de la porte lors de l'attaque et $\Delta i(t)_p$ est le profil différentiel des courants consommés par la porte p. $\Delta i(t)_p$ a donc pour expression

$$\Delta i(t)_p = i_{p,1}(t) - i_{p,0}(t) \tag{30}$$

où $i_{p,1}(t)$ et $i_{p,0}(t)$ sont respectivement les profils des courants consommés lorsque la porte p positionne une valeur '1' ou '0' sur sa sortie.

Afin de démontrer que cette hypothèse est correcte, considérons qu'une attaque DPA est effectuée selon le bit Z d'un bloc combinatoire, constitué de P portes, avec l'ensemble de vecteurs V. Parmi ces V vecteurs, considérons que T d'entre eux forcent Z à prendre la valeur logique '1' alors que V-T=F d'entre eux forcent la sortie à la valeur '0'. Avec ces définitions la signature DPA s'exprime selon :

$$S_{DPA}(Z) = \frac{1}{T} \cdot \sum_{v=1}^{T} I_V(t) - \frac{1}{F} \cdot \sum_{w=1}^{F} I_W(t) \tag{31}$$

$I_V(t)$ et $I_W(t)$ sont respectivement les profils en courant du bloc combinatoire lorsque les vecteurs v et w appartenant respectivement à T et F sont appliqués sur son entrée :

$$\begin{aligned} I_V(t) &= \sum_{p=1}^{P} i_p(t) \\ I_W(t) &= \sum_{p=1}^{P} i_p(t) \end{aligned} \tag{32}$$

où $i_p(t)$ est le profil du courant consommé par la porte p lorsque les vecteurs v et w sont

appliqués sur l'entrée du bloc considéré.

Compte tenu des définitions précédentes, et en définissant r_p^T, f_p^T (r_p^F et f_p^F) comme les nombres de vecteurs de T (F) qui forcent la sortie de la porte p respectivement à '1' et '0', il est possible d'exprimer la signature DPA sous la forme suivante :

$$S_{DPA}(Z) = \frac{1}{T} \cdot \sum_{p=1}^{P-1} \left(r_p^T \cdot i_{p,1}(t) + f_p^T \cdot i_{p,0}(t) \right) - \frac{1}{F} \cdot \sum_{p=1}^{P-1} \left(r_p^F \cdot i_{p,1}(t) + f_p^F \cdot i_{p,0}(t) \right) + \Delta i_Z(t) \qquad (33)$$

qui après simplification permet d'obtenir l'expression suivante de la signature DPA :

$$S_{DPA}(Z) = \sum_{p=1}^{P-1} \left(\frac{f_p^F}{F} - \frac{f_p^T}{T} \right) \cdot \Delta i_p(t) + \Delta i_Z(t) \qquad (34)$$

Comme on peut le constater, cette expression est bien une combinaison linéaire des profils différentiels en courant des cellules constituant le bloc attaqué, ce qui valide l'hypothèse que nous avons formulé auparavant. Outre cette confirmation, cette expression est intéressante dans la mesure où elle met en évidence de manière formelle le syndrome qu'exploite la DPA à savoir les dissymétries de courant consommé et plus spécifiquement les profils différentiels en courant des cellules.

Remarquons également que si fp^T et f_p^F sont semblables (ce qui est raisonnable si le nombre de vecteurs V est important), le premier terme de l'expression (34) devient négligeable devant le dernier. La signature DPA se résume alors au profil différentiel en courant de la cellule contrôlant le bit Z attaqué. Ceci démontre la finesse de l'analyse DPA.

IV-4-b-Sensibilité du profil différentiel en courant des cellules DR

Compte tenu que la signature DPA est une combinaison linéaire des profils différentiels en courant des cellules élémentaires composant le circuit attaqué, il semble judicieux de s'intéresser à la sensibilité de ces derniers aux paramètres de conception.

Afin d'évaluer la sensibilité du profils différentiels en courant des cellules double rail, mais également d'établir de manière formelle la robustesse de la logique double rail, nous avons développé une modélisation au premier ordre du courant de commutation des structures CMOS. Cette modélisation a été ensuite utilisée pour comprendre l'impact du routage, et de manière plus générale l'impact de l'environnement de la cellule (charge, temps de transition et d'arrivée des signaux) sur les profils différentiels des cellules double rail. Le reste de ce

paragraphe est structuré de la manière suivante. Dans une première étape, un modèle simplifié des portes double rail est introduit. En s'appuyant sur ce modèle, une modélisation du profil en courant des cellules double rail est établie dans un deuxième temps. Cette modélisation est ensuite mise en œuvre pour appréhender l'impact de l'environnement de fonctionnement des cellules double rail sur leur profil en courant dans une troisième étape. Enfin dans une dernière étape des critères de robustesse à la DPA sont identifiés et validés par simulation.

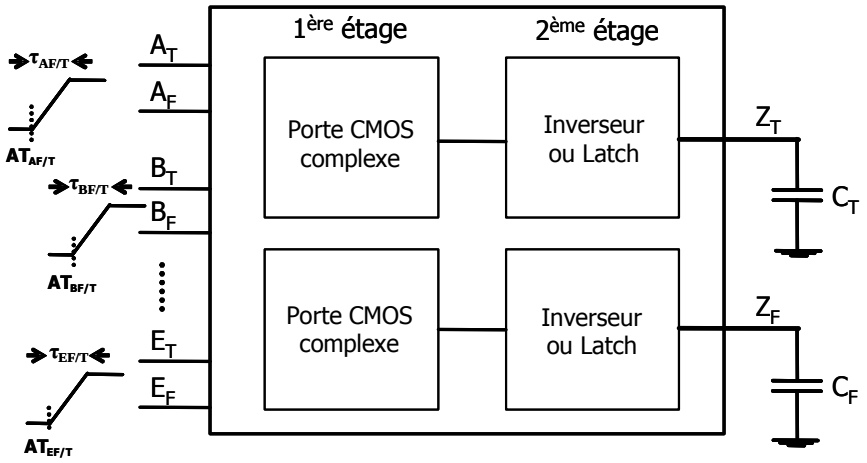


Fig. 59. Topologie générique d'une cellule double rail

IV-4-b-1-Modèle équivalent des cellules double rail

Une analyse en profondeur des cellules double rail proposées dans la littérature, a permis d'établir que la topologie générique (la plus fréquente) d'une cellule double rail est celle représentée sur la figure 59. Sur cette figure, $AT_{IF/T}$ et $\tau_{IF/T}$ sont respectivement les temps d'arrivée et les temps de transition des signaux appliqués sur l'entrée des cellules, alors que C_T (T pour True) et C_F (F pour False) sont respectivement les capacités présentent sur les nœuds de sortie de la cellule double rail.

En considérant que n'importe quel réseau série parallèle de transistors ou porte CMOS peut être réduite à un inverseur équivalent, la topologie de cellule représentée sur la figure 59 peut être ramenée à celle de la figure 60a. Bien que cette procédure de réduction soit extrêmement agressive, nous l'avons adopté dans la mesure où notre objectif est d'étudier :

- l'impact d'un éventuel déséquilibre des temps transition qui ne peut qu'affecter que le premier étage des cellules double rail,
- l'impact d'un éventuel déséquilibre des charges qui ne peut qu'altérer le comportement du dernier étage des cellules double rail,

- et enfin l'impact d'un éventuel déséquilibre des temps d'arrivée des signaux, qui affecte de manière similaire l'ensemble des étages des portes double rail.

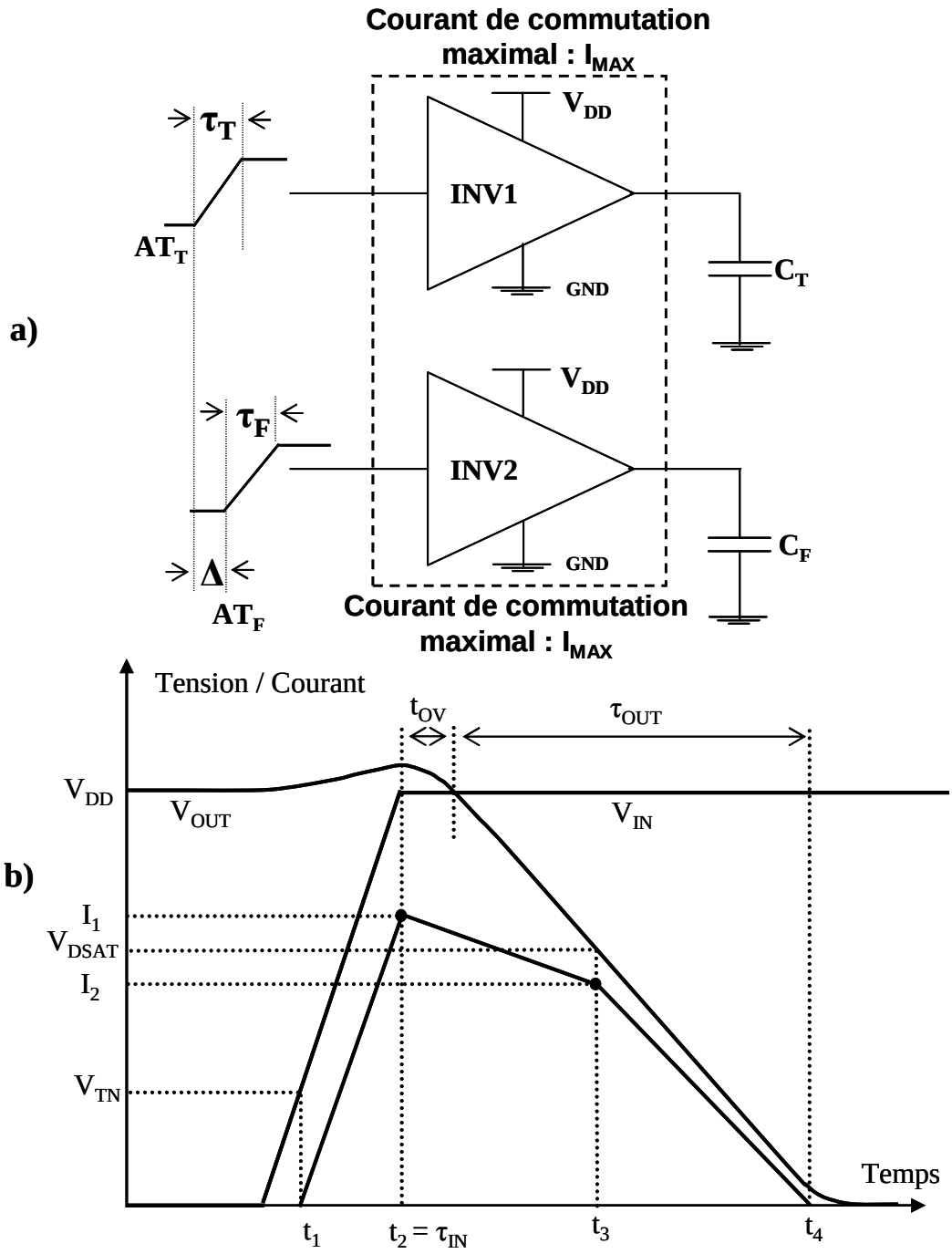


Fig. 60.(a) modèle équivalent réduit d'une porte double rail, **(b)** Allures typiques du courant de commutation et des tension d'entrée et de sortie d'une structure CMOS

IV-4-b-2-Modélisation du courant de commutation

En appliquant une telle technique de réduction, la modélisation du courant de commutation des cellules double rail se ramène à la modélisation du courant de commutation de n'importe quel inverseur CMOS. De nombreux travaux ont été dévolus à la modélisation du processus de commutation des inverseurs.

Pour des conditions de charge et de contrôle typiques (domaine des rampes d'entrée rapides [Mau01]), l'évolution du courant de commutation de n'importe quelle porte CMOS peut être modélisée par une fonction linéaire par morceaux comme cela est illustré sur la figure 60b. Sur cette dernière figure, I_1, I_2 et t_1, t_2, t_3 et t_4 sont des grandeurs devant être modélisées. I_1 et I_2 sont respectivement les valeurs maximales du courant que peuvent délivrer les transistors constituant l'inverseur lorsque leur tension drain source sont respectivement égales à V_{DD} et V_{DSAT}. En considérant un modèle de courant drain source propre au transistor à canal court, il est possible d'exprimer I_1 et I_2 selon :

$$I_1 = \frac{K}{D_W} \cdot \frac{W}{L} \cdot (V_{DD} - V_T) \cdot (1 + \lambda \cdot V_{DD}) \quad (35)$$

$$I_2 = \frac{K}{D_W} \cdot \frac{W}{L} \cdot (V_{DD} - V_T) \cdot (1 + \lambda \cdot V_{DSAT}) \quad (36)$$

où K, W, L, V_T sont respectivement le facteur de conduction, la largeur W, la longueur du canal et la tension de seuil du transistor N ou P ; D_W est le coefficient de réduction du courant, ou poids logique, induit par la phase de réduction des réseaux série parallèle de transistors.

Les instants t_1, t_2 caractéristiques de la commutation d'un inverseur, sont définis comme les instants auxquels le signal d'entrée V_{IN} atteint respectivement les tensions V_T et V_{DD}. t_1, t_2 peuvent donc être exprimés selon :

$$t_1 = \frac{V_{TN}}{V_{DD}} \cdot \tau_{IN} \quad (37)$$

$$t_2 = \tau_{IN} \quad (38)$$

Enfin, t_3 et t_4 correspondent respectivement à l'instant où la tension drain source (V_{DS}) du transistor N ou P selon le front considéré est égale V_{DSAT} et V_{DS}=0. Ainsi, nous obtenons les expressions (39) et (40) de t_3 et t_4.

$$t_3 = \tau_{ov} + \frac{V_{DSAT}}{V_{DD}} \cdot \tau_{OUT} \quad (39)$$

$$t_4 = \tau_{OV} + \tau_{OUT} \tag{40}$$

Notons que dans les expressions précédentes, τ_{ov} est la durée de la phase de surtension [Jep94] et τ_{out} correspond au temps de transition du signal de sortie. Dans le cas d'un fonctionnement en rampe rapide, l'expression de τ_{out} (41) peut être trouvée dans [Mau01] tandis que l'expression (42) de τ_{ov} peut être aisément retrouvée par la résolution de l'équation différentielle régissant le comportement d'un inverseur CMOS.

$$\tau_{OUT} = \frac{D_W \cdot C_L \cdot V_{DD}}{K \cdot \frac{W}{L} \cdot (V_{DD} - V_{TN})} \tag{41}$$

$$\tau_{OV} = \tau_{IN} + \frac{C_M}{I_1} \cdot V_{DD} \tag{42}$$

Dans les expressions (41) et (42), C_L et C_M correspondent respectivement à la charge de l'inverseur et à la capacité de couplage entrée-sortie. Toutes ces expressions ont ensuite été implémentées sous Matlab pour constituer notre modèle de profil en courant d'un inverseur CMOS.

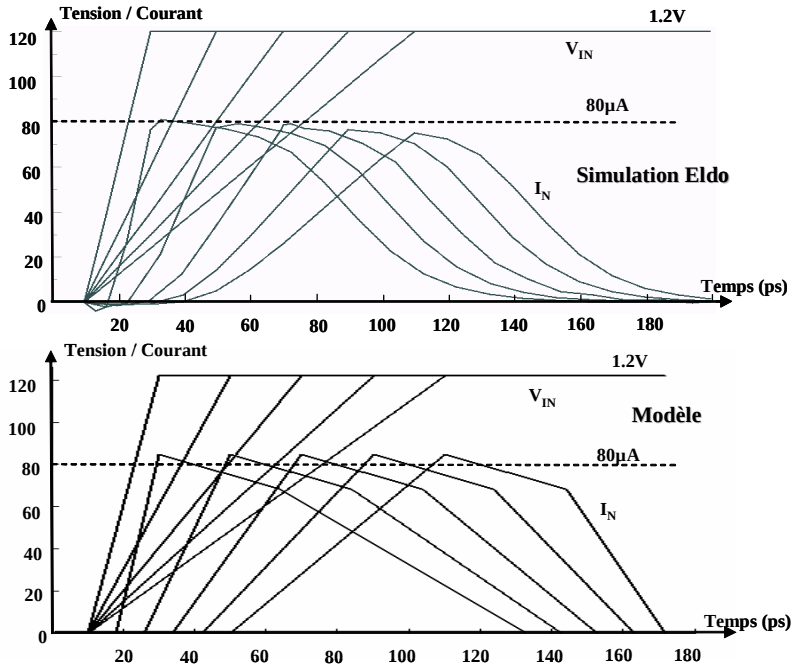


Fig. 61. Profils en courant calculés et simulés d'un inverseur CMOS

Afin de valider notre modèle de profil en courant de porte CMOS, nous allons effectuer des comparaisons entre les résultats de notre modèle et ceux d'un simulateur Eldo (Mentor).

Comme notre modèle prend en charge toutes les portes logiques simples, pour ces validations nous avons utilisé des inverseurs, des portes Nand2, Nor2 etc. Notons par ailleurs que ces simulations ont été menées sur une technologie CMOS 130 nm et pour différentes valeurs de rampe d'entrée (20 ps à 100 ps). Sur la figure 61, nous avons une illustration de ces validations dans le cas d'un inverseur CMOS.

Comme on peut l'observer sur cette figure 61, les profils de courant obtenus à partir de notre modèle sont très représentatifs de ceux obtenus par simulation avec l'outil Eldo. Par ailleurs, il en est de même pour les autres validations sur les autres types de portes logiques (Nor2, Nand2, etc.). En conclusion, la précision de notre modèle est satisfaisante. Il donc envisageable de l'exploiter pour évaluer la contribution d'une cellule double rail à la signature DPA d'un bloc combinatoire.

IV-4-b-3-Etude des profils différentiels en courant

Dans ce paragraphe nous allons étudier, en nous appuyant sur le modèle au premier ordre du courant de commutation des structures CMOS, l'impact sur les profils différentiels en courant d'éventuels déséquilibres de charge, de temps de transition et de temps d'arrivée.

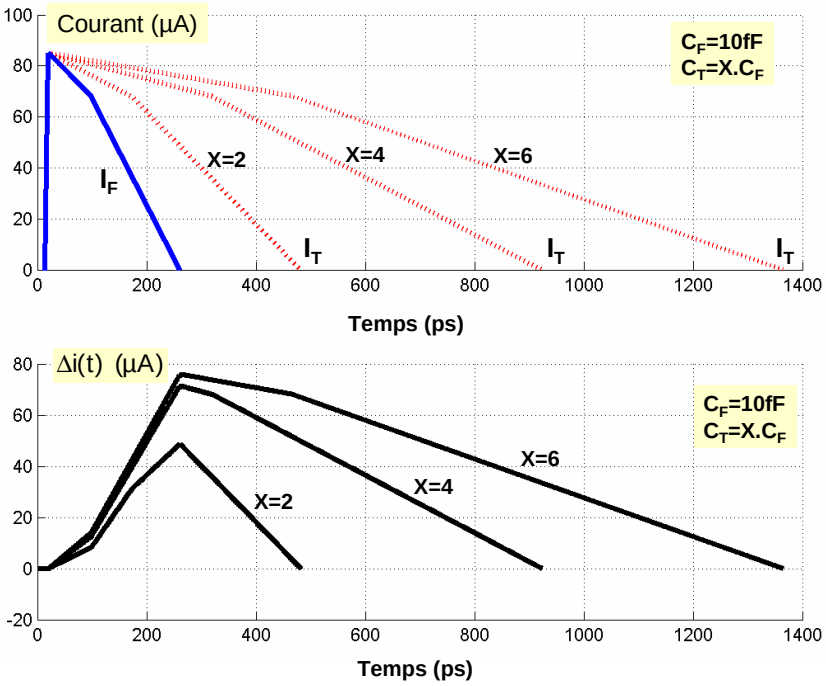


Fig. 62. Impact d'un déséquilibre de charge sur Δi(t)

- *Impact d'un déséquilibre de charge :*

Afin de rapidement quantifier l'impact d'un déséquilibre des charges présentes sur les sorties différentielles des cellules double rail, nous avons tracé à l'aide de scripts Matlab, les profils différentiels en courant de la structure représentée sur la figure 60a. Un exemple typique de profils obtenus est représenté sur la figure 62.

Comme on peut le constater sur cette figure, l'amplitude et la durée de Δi(t) s'accroissent rapidement lorsque l'on augmente le taux de déséquilibre $X=C_T/C_F$. Ces accroissements sont essentiellement induits par l'accroissement du temps que met la structure à décharger les nœuds de sortie.

Si l'on considère plus particulièrement l'amplitude de Δi(t), on remarque que des que le taux de déséquilibre atteint 4, celle-ci représente 90% du courant maximal que peut délivrer la porte double rail. Ceci met évidence qu'il existe une valeur limite de déséquilibre au-delà de laquelle la logique double rail n'est pas plus robuste que la logique simple rail. En d'autres termes, cela signifie que la logique double rail n'est pas inconditionnellement robuste à la DPA.

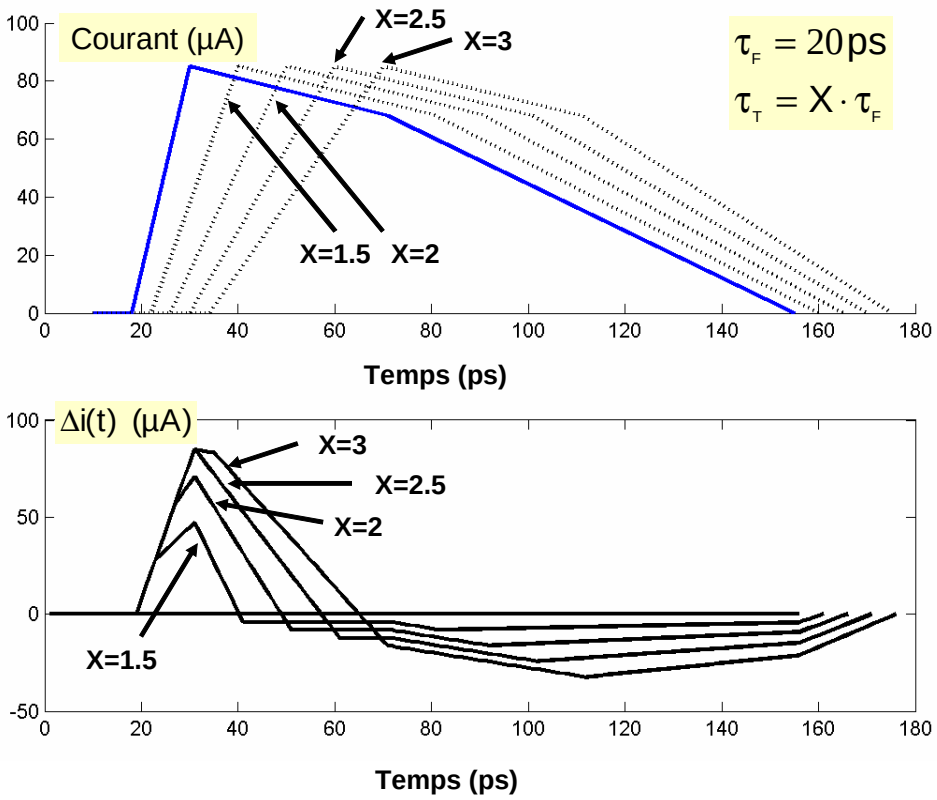


Fig. 63. Impact d'un déséquilibre de temps de transition sur Δi(t)

- *Impact d'un déséquilibre de temps de transition :*

De la même manière que nous avons évalué l'impact de déséquilibre des charges sur les profils différentiels de courant, nous avons évalué l'impact de déséquilibre de temps de transition. La figure 63 est un exemple typique des résultats que nous avons obtenus. Comme on peut le constater, l'amplitude de Δi(t) s'accroît rapidement avec le taux de déséquilibre $X=\tau_T/\tau_F$ jusqu'à atteindre une valeur maximale égale au courant maximum que peut fournir la structure pour des valeurs de X supérieures à 2. Cette observation démontre qu'il est nécessaire d'équilibrer les temps de transition des signaux sous peine de perdre tous les avantages de la logique double rail en terme de sécurité.

- *Impact d'un déséquilibre de temps d'arrivée des signaux :*

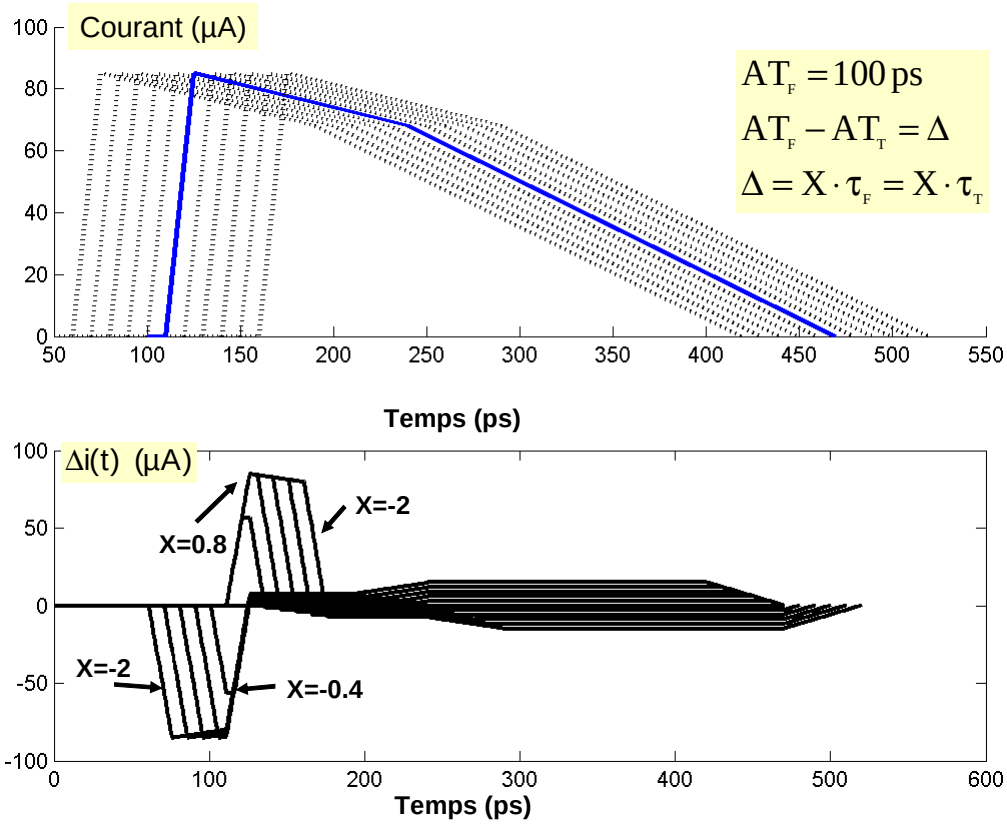


Fig. 64. Impact d'un déséquilibre de temps de transition sur Δi(t)

De la même manière que nous avons évalué l'impact, sur les profils différentiels de courant, de déséquilibre de charges et de temps de transition dans les paragraphes précédents, nous avons évalué l'impact d'un éventuel déséquilibre Δ de temps d'arrivée des signaux déclenchant la commutation. La figure 64 est un exemple typique des résultats que nous avons obtenus. Comme on peut le constater, dès que le taux déséquilibre X (exprimé en fonction du temps de transition $X=\Delta/\tau$) atteint 0.8, l'amplitude maximale de Δi(t) devient voisine du

courant maximal que peut fournir la porte. En d'autres termes, cette figure semble indiquer que la logique double rail ne peut être considérée comme robuste que si les délais de propagation sont parfaitement équilibrés et plus précisément que si les signaux pouvant déclencher la commutation d'une porte arrivent dans un intervalle de temps très restreint.

IV-4-b-4-Modélisation de l'amplitude de Δi(t)

Nous venons de mettre en évidence, sur un exemple très simple, que le profil différentiel en courant d'une cellule double rail, est fortement sensible à la présence d'éventuels déséquilibres de charges sur ses sorties, de temps transition de signaux appliqués sur ses entrées, ou encore de temps d'arrivée de ces derniers. Plus particulièrement, les analyses que nous avons effectuées démontrent que l'amplitude des profils différentiels peut, en présence de déséquilibres d'une certaine importance, atteindre l'amplitude du courant de commutation maximal de la porte. Ceci signifie qu'en présence de déséquilibres suffisamment importants, la logique double rail peut n'offrir aucun avantage sur la logique simple rail. Afin de pouvoir quantifier au premier ordre à partir de quel niveau de déséquilibre, la logique double rail ne peut plus être considérée comme robuste à la DPA, nous avons développé une modélisation simple de l'amplitude du profil différentiel en courant des cellules double rail et ce pour les trois types de déséquilibre considérés

IV-4-b-4-a-Amplitude maximum de Δi(t) en présence de déséquilibre de charge

Afin d'obtenir une expression analytique de I^S_{MAX}, l'amplitude maximale du profil différentiel en courant d'une cellule double rail, nous avons évalué la différence des profils en courant calculés lorsque la cellule force sa sortie à '0' et '1' respectivement et ce en considérant que des valeurs distinctes : $C_F<C_T$. A partir des expressions (35) et (36) nous avons obtenu les résultats suivants :

$$I^S_{MAX} = \frac{I_{MAX}\cdot(1-\beta)}{\left(\frac{V_{DD}}{V_{DSAT}}-1\right)}\cdot\left(\frac{C_T}{C_F}-1\right) \quad \text{if} \quad 1<\frac{C_T}{C_F}<\frac{V_{DD}}{V_{DSAT}} \tag{43}$$

$$I^S_{MAX} = I_{MAX}\cdot\left(1-\frac{\beta.V_{dd}}{V_{dsat}}\cdot\frac{C_F}{C_T}\right) \quad \text{if} \quad \frac{C_T}{C_F}>\frac{V_{DD}}{V_{DSAT}} \tag{44}$$

où I_{MAX} est le courant maximal ($V_{DS}=V_{DD}$) que peuvent fournir les inverseurs de la figure

60a, β le rapport entre le courant que fournissent ces inverseurs lorsque $V_{DS}=V_{DD}$ et $V_{DS}=V_{DSAT}$.

La figure 65 reporte les évolutions calculées et simulées de I^S_{MAX} en fonction du rapport C_T/C_F i.e. du taux de déséquilibre introduit sur la sortie différentielle de la cellule double rail. Les valeurs simulées ayant été obtenues en considérant la structure de la figure 60a, et plus particulièrement des inverseurs dont les principales caractéristiques sont W_n=0.15µm …. , W_p=2.9·Wn … et L=0.130µm. Les temps de transition des signaux appliqués sur les entrées respectives de ces deux inverseurs étaient de 50ps et la valeur de C_F considérée de 4fF.

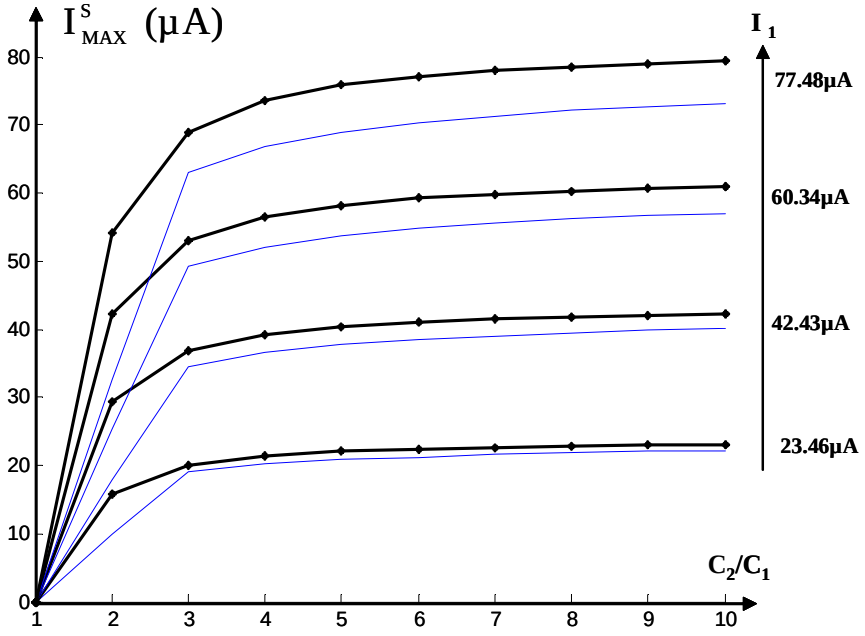


Fig. 65. I_S^{MAX} simulé et calculé en fonction C_T/C_F

Comme on peut le constater sur cette figure l'adéquation entre les résultats de simulation et de calcul est satisfaisante. Il est donc envisageable d'exploiter ces expressions pour appréhender la sensibilité de I^S_{MAX} au déséquilibre des charges.

Outre la validité des expressions (43-44) la figure 65 met en évidence que dès C_T/C_F est égal ou supérieur à environ 3.5, I^S_{MAX} est sensiblement égal à I_{MAX}. En d'autres termes cela suggère que la logique double rail n'est pas plus robuste à la DPA si le déséquilibre de charge du noeud différentiel attaqué (Z) est supérieur à 3.5.

IV-4-b-4-b-Amplitude maximum de Δi(t) en présence de déséquilibre des temps de transitions

Après avoir étudier l'impact d'un déséquilibre de charge sur l'amplitude de Δi(t), nous avons modéliser l'impact d'un éventuel déséquilibre des temps de transition sur l'amplitude maximale du profil différentiel en courant de la structure figure 60a et ce en adoptant la même démarche que précédemment. Nous avons obtenu l'expression suivante de I^S_{MAX} :

$$I^S_{MAX} = \mathbf{min}\left\{K \cdot \frac{W}{L \cdot D_W} \cdot V_{DD} \cdot \left(1 - \frac{\tau_F}{\tau_T}\right); I_{MAX}\right\} \qquad (45)$$

où τ_F et τ_T sont les temps de transition des signaux appliqués sur les entrées respectives des deux inverseurs. Afin d'estimer la validité de cette expression, nous avons comparé les évolutions, simulées et calculées, avec τ_T/τ_F de I^S_{MAX}. La figure 66 reporte les résultats obtenus, en considérant la même structure que précédemment et les valeurs suivantes des divers paramètres : $C_T = C_F = 4$ fF, $\tau_F = 50$ ps. Comme on peut le constater, l'expression (45) reproduit de manière très satisfaisante l'évolution obtenue par simulation.

Tout comme la figure 65, la figure 66 donne des indications quant à la robustesse à la DPA de la logique double rail. En effet, comme on peut le constater, I^S_{MAX} devient voisin de I_{MAX} (I_1) dès que τ_T/τ_F est égal ou supérieur à environ 3.5, ce qui signifie que la robustesse de la logique double rail à la DPA devient identique à celle de la logique simple rail.

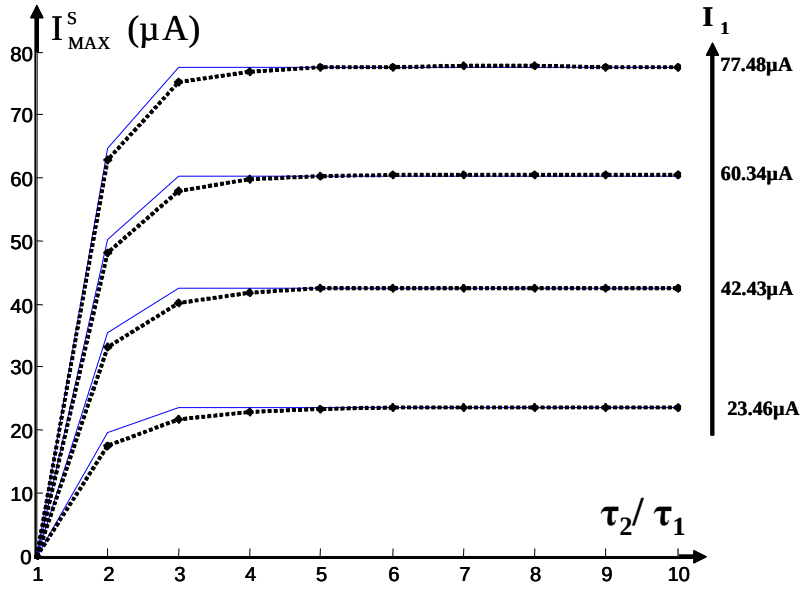


Fig. 66. I_S^{MAX} simulé et calculé en fonction τ_T/τ_F

IV-4-b-4-c-Amplitude maximum de Δi(t) en présence de déséquilibre des temps de d'arrivée des signaux

Tout comme nous avons modélisé l'amplitude maximale de Δi(t) en présence de déséquilibre de charge ou des temps de transition, nous avons modélisé I^S_{MAX} en présence d'un éventuel déséquilibre des temps d'arrivée des signaux. Nous avons obtenu l'expression suivante de I^S_{MAX} :

$$I^S_{MAX} = \mathbf{min}\left\{K \cdot \frac{W}{L \cdot D_W} \cdot \left(\frac{V_{DD} \cdot \Delta}{\tau}\right); I_{MAX}\right\} \qquad (46)$$

où Δ est la valeur absolue de la différence entre les temps d'arrivée des signaux contrôlant l'entrée des inverseurs. La figure 67 reporte les valeurs calculées et simulées de I^S_{MAX} en fonction du rapport Δ/τ où τ est le temps de transition des signaux ($\tau_F = \tau_T$). Comme on peut le constater l'expression (46) modélise de manière satisfaisante l'impact d'un déséquilibre des temps d'arrivée sur le profil différentiel en courant.

Par ailleurs, la figure 67 met en évidence que l'amplitude de I^S_{MAX} est très sensible à un déséquilibre des temps d'arrivée des signaux. En effet, dès que ce dernier atteint environ 0.8 à 0.9 fois la valeur du temps de transition des signaux, I^S_{MAX} devient sensiblement égal I_{MAX}. C'est très peu compte tenu que le déséquilibre des temps d'arrivée des signaux T et F peut s'accroître lorsque la donnée se propage dans la structure.

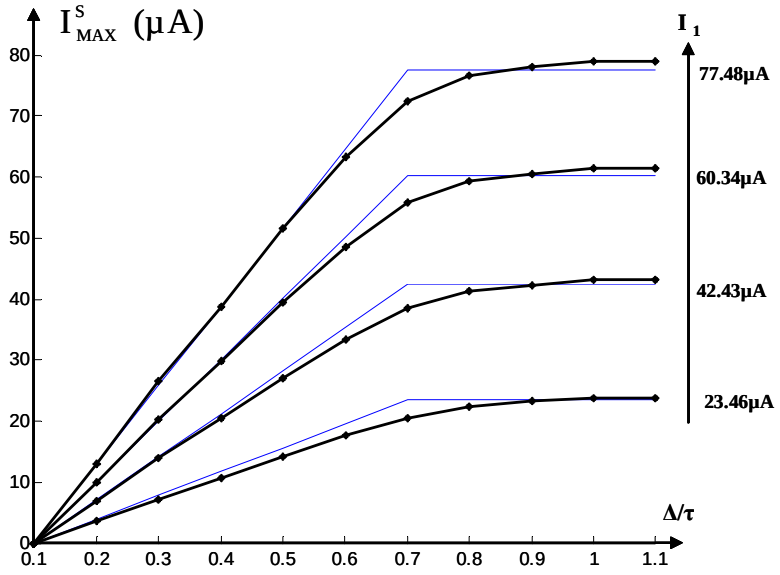


Fig. 67. I_S^{MAX} simulé et calculé en fonction Δ/ τ

IV-4-c-Métriques de robustesse à la DPA de structures double rail

A ce stade, nous avons dans ce chapitre démontré que la signature DPA d'un bloc

combinatoire est une combinaison linéaire des profils différentiels en courant des cellules qui le constituent, mis en évidence et modélisé la sensibilité des profils différentiels des cellules à aux éventuels déséquilibres de charge, de temps de transition et de temps d'arrivée des signaux introduits par la phase de placement routage. Dans ce paragraphe nous allons établir des métriques permettant d'évaluer la dangerosité d'une cellule double rail ou encore d'un nœud différentiel à une potentielle attaque DPA.

Pour ce faire, définissons I_{TH} comme le plus petit déséquilibre en courant qui peut être détecté avec un nombre N donné d'acquisition de la consommation d'un circuit et ce en accord avec la définition du rapport signal sur bruit SNR :

$$SNR = \frac{I_{TH}}{\sigma} \cdot \sqrt{N} \qquad (47)$$

σ: Variance du bruit

En s'appuyant sur cette définition, il est possible de quantifier les déséquilibres C_F/C_T, τ_T/τ_F ou encore Δ/τ qui peuvent être tolérés.

IV-4-c-1- Métrique de robustesse au déséquilibre de charge

En égalisant les expressions (43) et (44) à I_{TH}, il est en effet possible de définir la valeur critique $(C_F/C_T)_{Crit}$ du rapport C_F/C_T au-delà de laquelle, l'amplitude du profil différentiel en courant d'une cellule devient supérieure à I_{TH} et peut donc être, selon l'expression du rapport signal sur bruit (47), capturé par une attaque DPA :

$$\left.\frac{C_F}{C_T}\right|_{Crit} = \max\left\{ \frac{I_{TH}}{I_{MAX}} \cdot \frac{\frac{V_{DD}}{V_{DSAT}} - 1}{(1-\beta)} + 1 ; \left(\frac{V_{DSAT}}{\beta \cdot V_{DD}} \left(1 - \frac{I_{TH}}{I_{MAX}} \right) \right)^{-1} \right\} \qquad (48)$$

La figure 68 reporte les évolutions simulées et calculées de $(C_F/C_T)_{Crit}$ en fonction de I_{MAX} et ce pour différentes valeur de I_{TH}. Ces valeurs simulées ont été obtenues en considérant que le dernier étage des cellules double rail pouvait être constitué par des inverseurs, ou de portes Nand, ou de portes Nor ou encore de portes latches. Comme l'illustre la figure 68, la précision des résultats obtenus est satisfaisante.

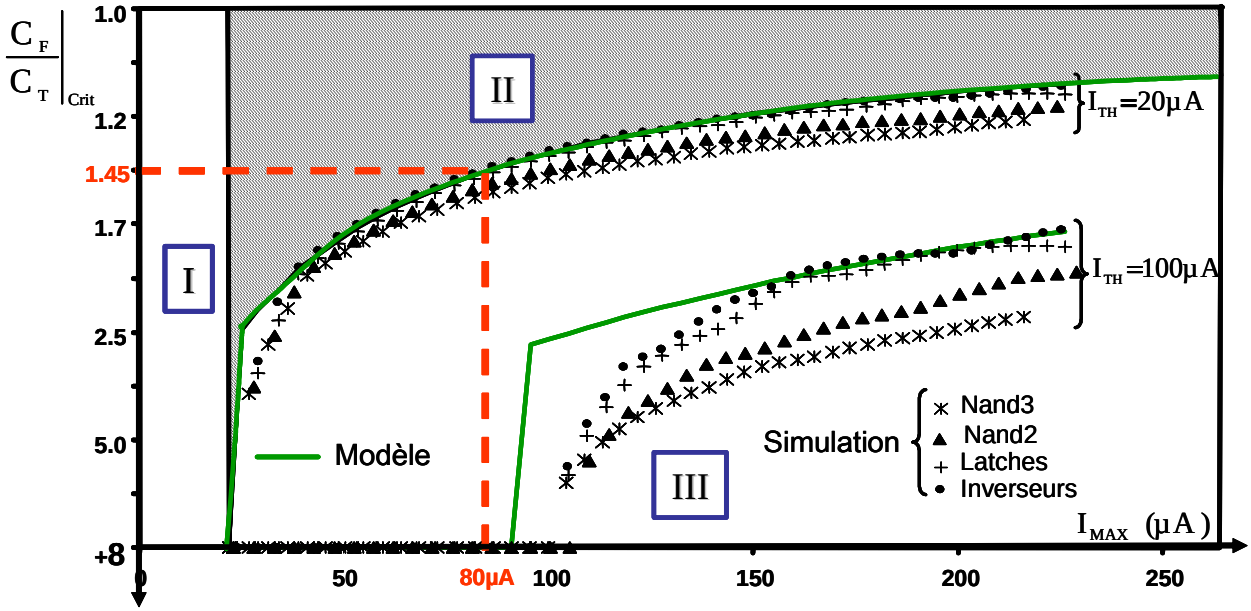


Fig. 68. Evolutions simulées et calculées de $(C_F/C_T)_{Crit}$ en fonction de I_{MAX}

Outre ce point, la figure 68 peut être découpée en plusieurs régions distinctes. La première région correspond à une zone de conception robuste à la DPA dans la mesure où la valeur de I_{MAX} est inférieure à I_{TH} et ce que l'on utilise de la logique simple rail ou double rail.

La région II (partie grisée de la figure) quant à elle correspond à une zone dans laquelle seule la logique double rail peut être considérée comme robuste à la DPA dans la mesure où son profil différentiel en courant n'excède pas I_{TH}. La partie hachurée de cette figure matérialise donc l'apport, en terme de robustesse, de la logique double rail sur la logique simple. La troisième et dernière région de cette figure se situe en dessous de la région II. Dans cette zone, la logique double rail ne peut plus être considérée comme robuste dans la mesure où, à cause des déséquilibres introduits lors de l'étape de placement routage, l'amplitude du profil différentiel de la cellule excède I_{TH}.

Outre la matérialisation de l'espace de conception dans lequel une cellule double rail est considérée comme étant robuste à la DPA, cette figure met en évidence que plus la valeur de I_{MAX} caractérisant une cellule est élevée, plus la valeur de $(C_F/C_T)_{Crit}$ doit être proche de 1 pour que la cellule reste dans la zone sécurisée (II)

L'expression (48) et la figure 68 constituent donc de bons moyens pour évaluer rapidement la robustesse à la DPA d'une cellule double rail et ce en fonction de son environnement.

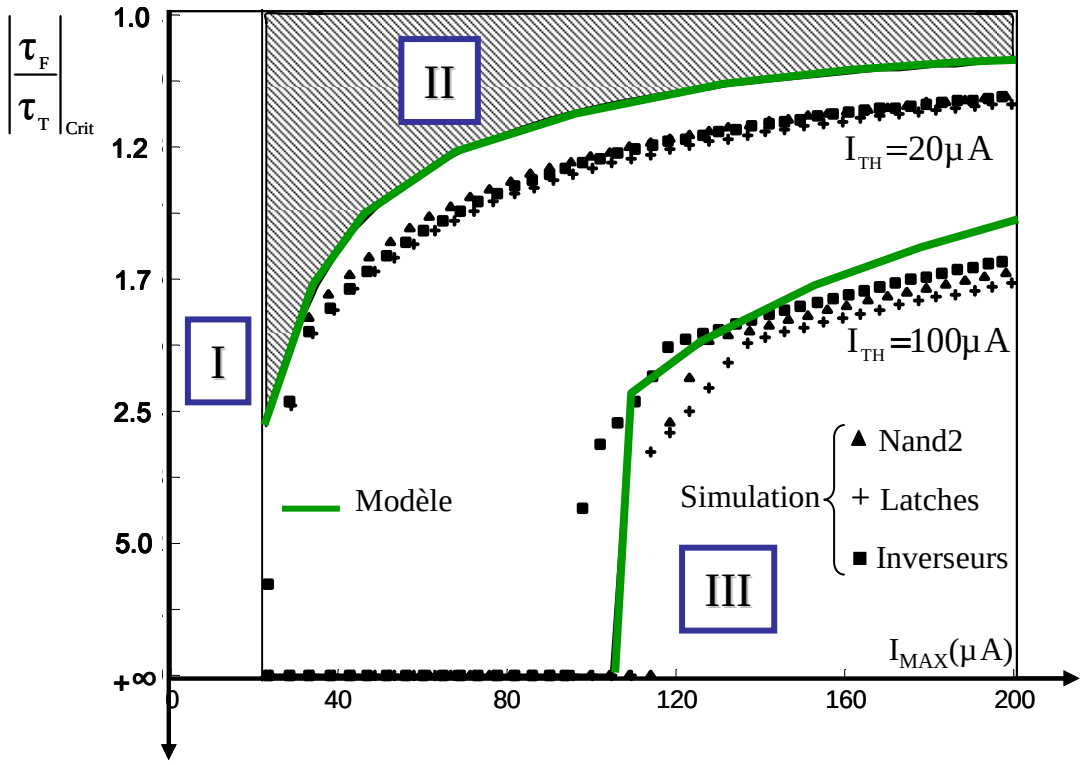


Fig. 69. Evolutions simulées et calculées de $(\tau_F/\tau_T)_{Crit}$ en fonction de I_{MAX}

IV-4-c-2- Métrique de robustesse au déséquilibre des temps de transition

En procédant comme nous l'avons fait pour les déséquilibres de charges, il est possible de définir une métrique de robustesse des cellules double rail au déséquilibre des temps de transition :

$$\left.\frac{\tau_T}{\tau_F}\right|_{Crit} = 1 - \frac{I_{TH}.(V_{DD} - V_T)}{I_{MAX}.V_{DD}} \text{ if } I_{MAX} > I_{TH} \qquad (49)$$

Afin de valider cette expression, nous avons comparé, pour différentes valeurs de I_{MAX}, les valeurs simulées et calculées de $(\tau_T/\tau_F)_{Crit}$. La figure 69 reporte les évolutions obtenues. Comme on peut le constater, il y a une bonne adéquation entre le calcul et la simulation et comme nous l'avons fait précédemment, nous pouvons conclure que plus une cellule délivre du courant et moins celle-ci sera robuste à une attaque DPA.

IV-4-c-3- Métrique de robustesse au déséquilibre des temps d'arrivée

En adoptant la même démarche que dans les deux précédents paragraphes, nous avons obtenu la métrique de robustesse, $|\Delta/\tau|_{Crit}$, au déséquilibre des temps d'arrivée suivante :

$$\left|\frac{\Delta}{\tau}\right|_{Crit} = \frac{I_{TH}\cdot(V_{DD}-V_T)}{I_{MAX}\cdot V_{DD}} \text{ if } I_{MAX}>I_{TH} \qquad (50)$$

La figure 70, qui reporte les évolutions simulées et calculées de $|\Delta/\tau|_{Crit}$ en fonction de I_{MAX} démontre la validité de l'expression obtenue.

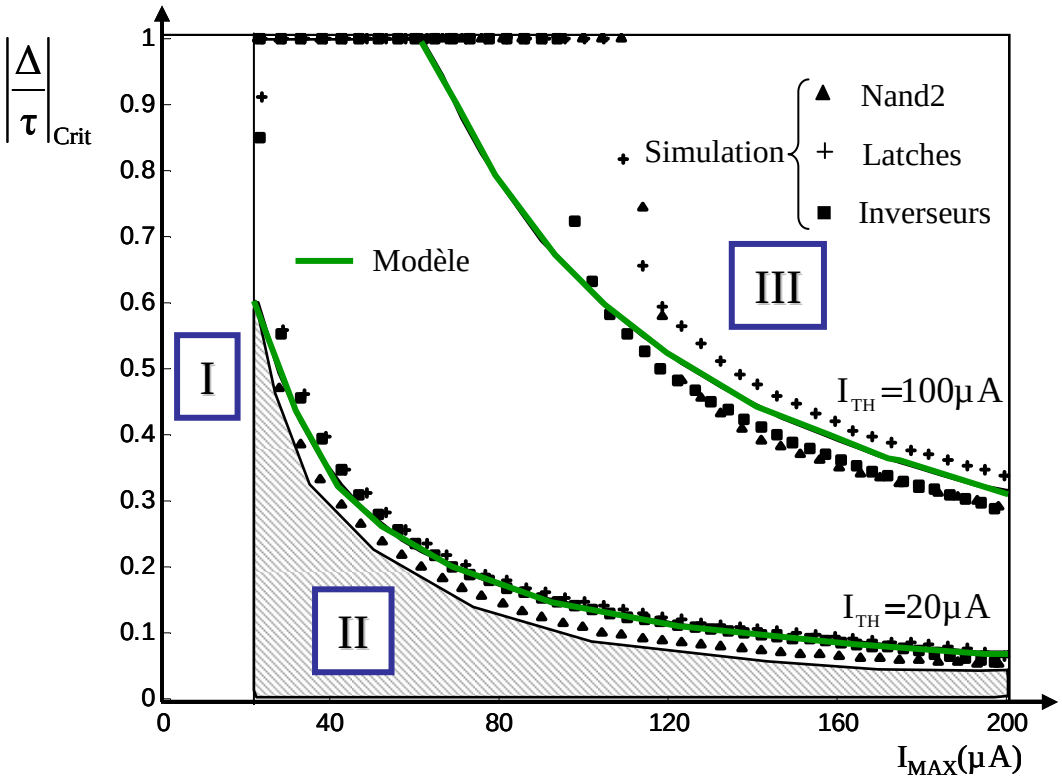


Fig. 70. Evolutions simulées et calculées de $|\Delta/\tau|_{Crit}$ en fonction I_{MAX}

Cette figure met également en évidence que la logique double rail est particulièrement sensible aux éventuels déséquilibres des temps d'arrivées des signaux introduits par le placement routage. En effet, si l'on considère une porte double rail caractérisée par une valeur I_{MAX} de 80 µA, l'abaque de la figure 70 nous permet d'affirmer que son profil différentiel en courant aura une amplitude supérieure à 20 µA si $|\Delta/\tau|>0.17$. Cela est très peu si l'on considère que les temps de transition sont généralement compris entre 20 ps et 300 ps.

IV-5- Conclusion

Dans ce chapitre nous avons mis en évidence l'impact que peut avoir la phase de placement routage sur la robustesse à la DPA d'un circuit double rail. Cette constatation nous a conduit à formaliser mathématiquement ce qu'est la signature DPA d'un circuit double rail à savoir une combinaison linéaire des profils différentiels en courant des cellules le constituant.

Compte tenu de ce résultat, nous avons analysé et modélisé la sensibilité de ces profils différentiels en courant aux paramètres de conception habituels que sont les temps d'arrivée et de transition des signaux et enfin la charge.

Cette étude nous a permis d'identifier de manière formelle l'espace de conception dans lequel la logique double rail est effectivement robuste. Plus particulièrement, nous avons pu montrer que la sensibilité des profils différentiels en courant des cellules double rail est faible, i.e. qu'il n'est pas nécessaire de symétriser de manière parfaite les interconnexions entre cellules. Par contre, nous avons également montré qu'une cellule double rail n'est robuste que si les signaux, pouvant potentiellement déclencher sa commutation, arrivent tous dans un intervalle de temps particulièrement réduit.

Afin de contourner ce problème, nous allons dans le chapitre suivant, introduire une logique ne présentant pas cette limitation.

Chapitre V:

La logique triple rail sécurisée

A partir des métriques de robustesse, nous avons clairement établi qu'une cellule double rail n'est robuste que si les signaux déclanchant sa commutation, arrivent tous dans un intervalle de temps particulièrement réduit. Pour satisfaire cette contrainte, nous proposons dans ce chapitre, une variante de la logique double rail qui est la logique triple rail sécurisée. Au niveau physique, les cellules triple rail sécurisées ont la particularité d'avoir un temps de calcul quasi-constant et par conséquent, indépendant des données.

Chapitre V : La logique triple rail sécurisée

V-1-Introduction

A partir des résultats du chapitre précédent, il apparaît qu'il existe un espace de conception dans lequel une cellule double rail peut être considérée comme robuste aux attaques DPA. Pour la technologie 130nm, cet espace de conception sécurisé met en évidence non seulement la capacité des cellules double rail à minimiser la corrélation entre la consommation et les données manipulées mais aussi l'extrême faiblesse de ces cellules aux décalages temporels. En effet, une cellule double rail n'est robuste que si les signaux déclanchant sa commutation, arrivent tous dans un intervalle de temps particulièrement réduit.

Notre objectif dans ce chapitre, est d'élargir l'espace de conception par rapport aux décalages temporels. Avec cette intention, nous proposons une variante de la logique double rail que nous appelons la logique triple rail sécurisée (Secure Triple Track Logic, STTL). En terme de sécurité à la DPA, un circuit sécurisé à base de la STTL est attendu pour avoir les propriétés intéressantes telles que:

- Une consommation indépendante des données manipulées
- Un temps de calcul indépendant des données manipulées

V-2 Fondamentaux de la STTL

V-2-a-Pourquoi la STTL?

Reformulé autrement, notre objectif dans ce chapitre est de rendre les cellules double rail quasi insensibles aux décalages temporels. La manière dont nous envisageons de traiter ce problème est dans un premier temps une brève analyse de la sensibilité du profil différentiel en courant au décalage temporel d'une cellule double rail et puis dans un deuxième temps une proposition de solution adéquate.

Pour une meilleure compréhension de l'influence du décalage temporel sur le profil différentiel en courant d'une cellule double rail, nous allons nous aider de la figure 71.

Comme énoncé dans le chapitre précédent (IV-4-b-1), la topologie d'une cellule double rail peut être ramenée à celle de la figure 71a. Sur cette figure, $AT_{T/F}$ représente le temps d'arrivée du signal, τ le temps de transition des signaux et enfin Δt le décalage temporel. Comme l'illustre la figure 71b, le profil différentiel en courant est la différence entre les profils en courant de INV1 et de INV2.

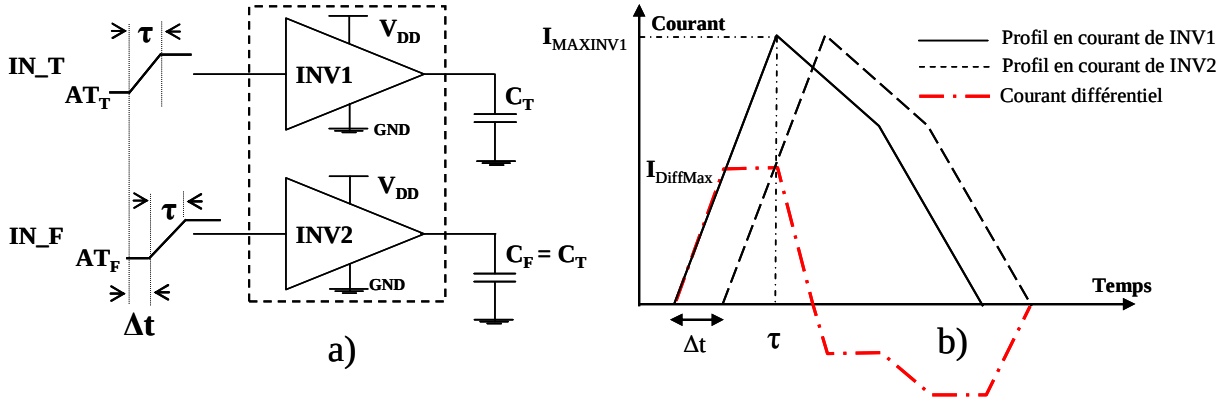


Fig. 71. Impact du décalage temporel sur le courant différentiel

Comme on peut le constater, un décalage temporel de Δt au niveau des entrées résulte en un décalage du même ordre au niveau des profils en courant. Par conséquent, pendant cette durée Δt, le profil différentiel en courant est égal au profil en courant de INV1 et dont l'amplitude maximale peut atteindre $I_{MAXINV1}$ pour $\Delta t = \tau$.

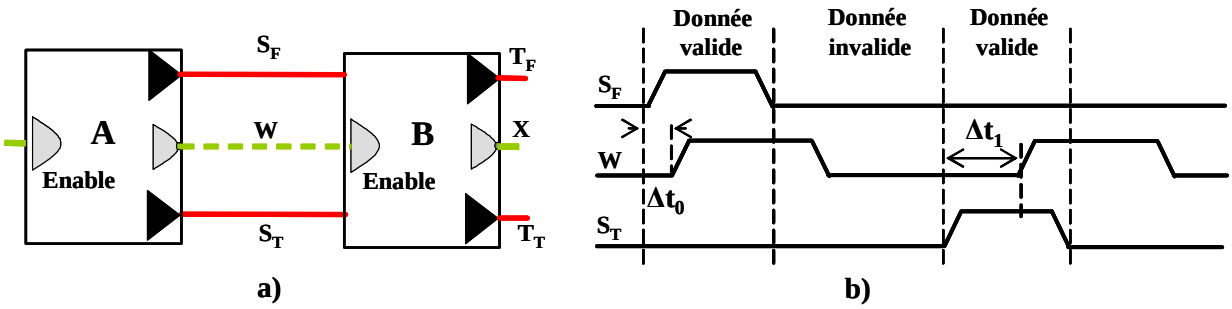


Fig. 72.a) Structures insensibles au décalage temporel, **b)** Mode de fonctionnement des structures insensibles au décalage temporel

Dans ce contexte, notre objectif est de faire en sorte que ce décalage temporel Δt ne soit pas répercuté au niveau des profils en courant de la cellule double rail. Comme l'illustre la figure 72, on peut facilement imaginer des cellules double rail avec "un signal lent" qui déclencherait le calcul au bout d'un temps $t \geq \Delta t$ modulable après que les entrées soient dans un état valide stable. Si on se réfère à la figure 72, tant que le décalage temporel entre les entrées S_F et S_T n'excède pas Δt_i, il n'aura aucune incidence sur les profils en courant du

module B. En d'autres termes, les profils en courant du module B sont attendus pour être quasi-superposés et insensibles au décalage temporel par voie de conséquence.

Comme l'approche décrite ci-dessus nous paraît réaliste, nous avons mené des investigations un peu plus poussées, ce qui a abouti à la définition d'un nouveau style de logique: la logique triple rail sécurisée ou Secure Triple Track Logic (STTL). Dans l'ensemble, c'est une logique assez proche de la logique double rail, toutefois avec un codage des données particulier et un modèle de fonctionnement différent.

V-2-b-Codage des données

A la différence de la logique double rail, la STTL utilise trois fils au lieu de deux pour coder un bit de donnée. Deux fils serviront toujours à coder les données tandis que le troisième fil servira à identifier l'état de validité de la donnée. La figure 73 présente le codage des données adopté par la STTL.

Comme on peut le constater, ce n'est pas tout à fait du codage triple rail (1 parmi 3) dans la mesure où l'information véhiculée par le troisième fil S^V est redondante et correspond à la validité de la donnée (S_0S_1). La conséquence directe est que dans un cycle de traitement, ce troisième fil commutera systématiquement à V_{DD} avant de revenir à 0 (Gnd). Ayant une activité complètement indépendante des données traitées, ce troisième fil n'aura aucune incidence particulière sur les signatures DPA.

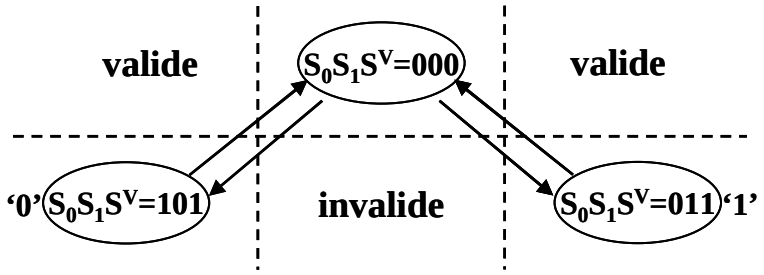


Fig. 73. Codage des données utilisé par la STTL

V-2-c-Méthode de conception

A partir du codage des données associé à ce style de logique STTL, nous avons tout naturellement défini la topologie de la figure 74 comme étant celle des cellules STTL pseudo statique. Du reste, une stratégie de dimensionnement adaptée sera mise en œuvre pour s'assurer que les signaux de validité arrivent après les données.

Au premier abord, cette topologie de la figure 74 paraît très proche de celle de nos cellules double rail définie dans le paragraphe III-5-a. Toutefois, à la différence de nos cellules double rail, la validité de toutes les données aux entrées d'une cellule STTL ne suffit pas à débuter l'évaluation de la cellule. En effet, le début de calcul est conditionné par l'arrivée des signaux de validité (A^V, B^V) qui, pour filtrer l'effet du décalage temporel, doivent arriver au bout d'un temps Δt après que les données (A_0, A_1, B_0, B_1) soient dans un état valide stable.

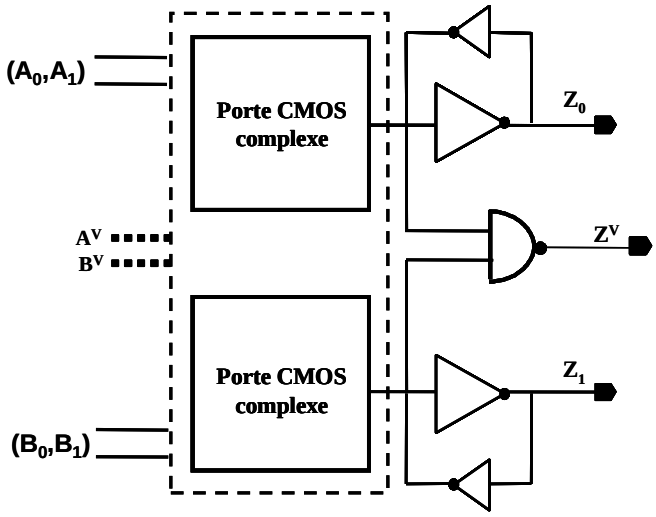


Fig. 74. Topologie d'une cellule STTL

Si l'on considère la topologie de la figure 74, la conception d'une cellule STTL se résume à la détermination des portes CMOS complexes et d'un dispositif qui permettra de conditionner le début de calcul par l'arrivée de tous les signaux de validité.

La détermination des portes CMOS complexes est quasiment identique à celle détaillée dans le paragraphe III-5-a. La seule différence est qu'il va falloir rajouter une circuiterie supplémentaire pour conditionner le début de l'évaluation de la cellule.

Pour l'implémentation de cette circuiterie nous nous appuierons sur la logique DCVS [Erd84, Chu86] dédiée à la conception de circuit asynchrone. En effet, ce style de logique utilise un signal de requête (Req=1) pour commencer l'évaluation d'une cellule. Ainsi, comme dans le cas d'une cellule DCVSL, nous allons insérer un transistor d'autorisation/interdiction entre les réseaux série de transistors N et la masse (Gnd). Ce transistor sera contrôlé par une porte "And" qui elle-même sera contrôlée par les signaux de validité. A titre d'exemple, dans le cas d'une cellule XOR2 STTL, la détermination des portes CMOS complexes avec le dispositif qui permettra de conditionner le début de calcul par l'arrivée de tous les signaux de validité

aboutit au schéma de la figure 75a.

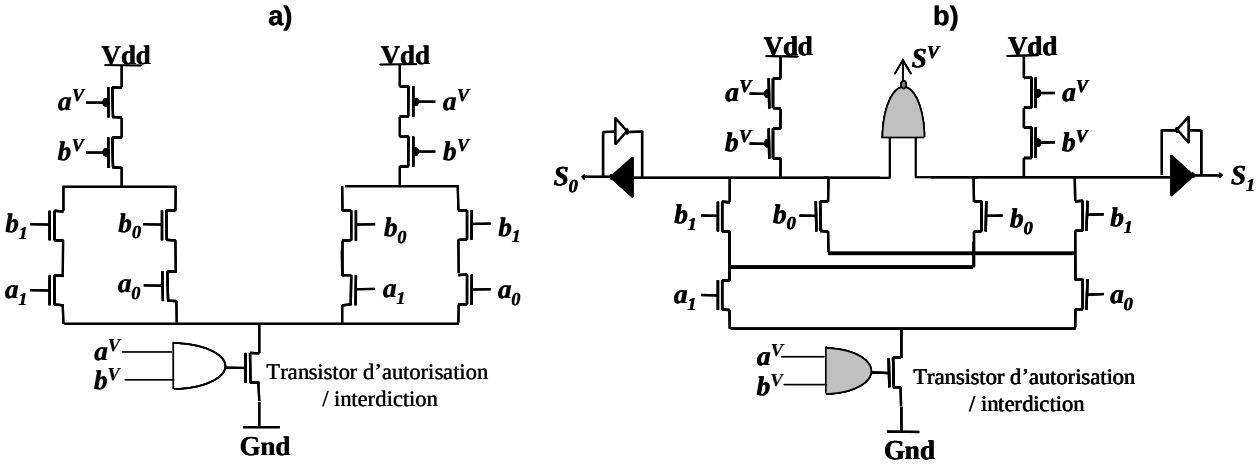


Fig. 75.a) Les portes complexes d'une porte XOR2 STTL, **b)** Schéma complet d'une porte XOR2 STTL pseudo statique

Toutefois, des optimisations de surface restent possibles. Des transistors peuvent être mis en commun tout en respectant les opérations logiques de la structure. Enfin, pour compléter le schéma de la porte XOR2 STTL, il faut se référer à la topologie de la figure 74 et rajouter les sous blocs manquants à savoir: les latches pour le maintien des données et la porte "Nand2" pour le calcul du signal de validité. La figure 75b présente le schéma final d'une porte XOR2 STTL pseudo statique.

Par la suite, cette implémentation pseudo statique peut également se mettre aisément sous forme statique comme l'illustre la figure 76. Cette transformation a cependant un coût qui est un accroissement du nombre de transistors. Toutefois, cette augmentation du nombre de transistors peut amener sous certaines conditions des gains substantiels en vitesse.

Cependant des précautions particulières doivent être prises lors de la conception des cellules STTL afin de garantir le type de fonctionnement désiré qui est que les signaux de validité (a^V, b^V, S^V) doivent arriver au bout d'un temps Δt après les données (A_0, A_1, B_0, B_1). Pour satisfaire cette contrainte de fonctionnement, tous les signaux de validité sont délivrés par des portes à faible courant de commutation tandis que les signaux de donnée sont délivrés par des portes à fort courant de commutation. En d'autres termes, les portes délivrant les signaux de validité ont des délais de propagation largement supérieurs à ceux des portes délivrant les signaux de donnée. Au niveau circuit, ces caractéristiques des portes peuvent être aisément obtenues par un dimensionnement adéquat des transistors. A titre d'exemple considérons la

structure composée de trois cellules STTL de la figure 77.

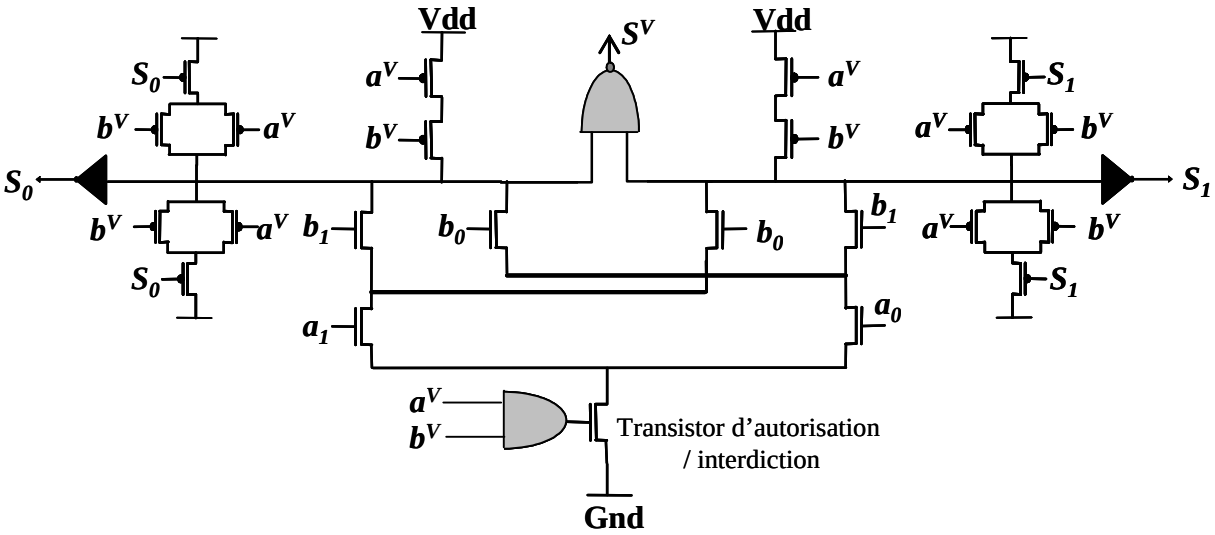


Fig. 76. Schéma complet d'une porte XOR2 TTL statique

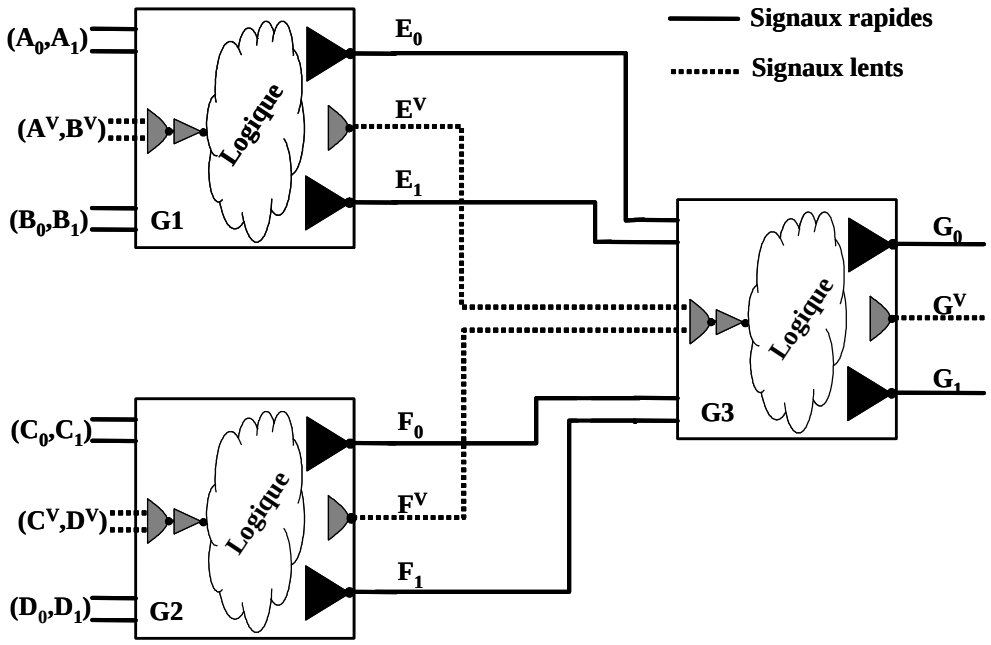


Fig. 77. Structure composée de cellules STTL

Sur cette figure 77, les portes à fort courant de commutation sont celles de couleur noire tandis que celles de couleur grise sont les portes à faible courant de commutation. Ainsi, un signal est qualifié de rapide et de lent selon le type de porte le contrôlant. En effet, sur cet exemple on considère que pour les trois cellules, le délai de propagation entrée/sortie des données est de ΔT_D et des signaux de validité de ΔT_V avec $(\Delta T_D/\Delta T_V) > 1$. Pour une meilleure compréhension du fonctionnement de cette structure nous avons dessiné le chronogramme de

la figure 78. Sur cette figure 78, on assume que les signaux de validité A^V, B^V, C^V et D^V arrivent à t = 0 ou encore que le début d'évaluation des modules G1 et G2 commence à t = 0. Notez que la fenêtre d'incertitude du temps d'arrivée des données (E_0, E_1, F_1, F_0, G_1, G_0) est représentée par une zone grisée (U_{AT}). En clair, cette zone correspond à un éventuel décalage temporel qui peut être introduit soit par des déséquilibres de charge générés par la phase de placement routage, soit par des déséquilibres structurels des portes logiques.

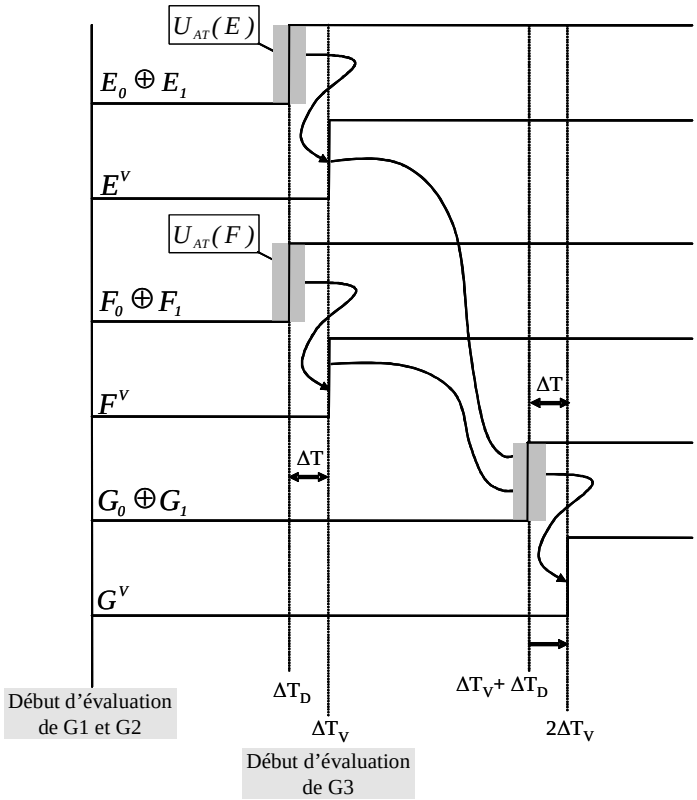


Fig. 78. Chronogramme associé au fonctionnement de la figure 77

Une première remarque intéressante: malgré l'incertitude au niveau du temps d'arrivée des données ($E_0 \oplus E_1$, $F_0 \oplus F_1$), le module G3 commence invariablement son évaluation à t = ΔT_V. En d'autres termes, le temps de calcul du module est complètement indépendant des données manipulées. Toutefois, cette propriété est valide tant que la fenêtre temporelle d'incertitude U_{AT} n'excède pas $\Delta T = \Delta T_V - \Delta T_D$.

Autrement dit, dans un circuit STTL, le début d'évaluation d'une cellule élémentaire se produit toujours au même moment par rapport au début du cycle si et seulement si $U_{AT} < \Delta T$. Par ailleurs, la valeur de ΔT est modulable. Comme $\Delta T = \Delta T_V - \Delta T_D$, la valeur de ΔT peut être réglée en dimensionnant correctement la porte qui contrôle le signal de validité. Dans un flot

de conception de circuit sécurisé, on peut imaginer un algorithme qui en fonction de l'importance de la fenêtre temporelle d'incertitude des données, va aller adapter la valeur de la tolérance ΔT pour chaque cellule élémentaire.

En conclusion, malgré l'existence de décalages temporels au niveau des temps d'arrivée des signaux, une structure à base de cellules STTL a un temps de calcul constant et par conséquent indépendant des données manipulées. Ainsi en terme de sécurité à la DPA, un circuit sécurisé à base de cellules STTL est attendu pour avoir les propriétés suivantes:

- Insensibilité aux déséquilibres des charges introduits par le placement routage.
- Temps de cycle indépendant des données manipulées.

V-3 Analyse des performances

Dans ce paragraphe nous proposons une analyse des performances de nos cellules STTL. Pour ce faire, quelques schémas des portes ont été dessinés afin d'en évaluer la surface, la vitesse et enfin la consommation. Pour effectuer des comparaisons, les mêmes évaluations ont été menées sur d'autres styles de logique dédiés à la conception de circuit sécurisé. La technologie utilisée est la HCMOS9 130nm.

V-3-a-Performance en surface

Dans cette partie, nous proposerons une évaluation de surface en terme de nombre de transistors. Comme annoncé précédemment, nous avons dessiné les schémas de diverses fonctions élémentaires que l'on trouve traditionnellement dans les bibliothèques de fondeurs. Le tableau ci-dessous (Tableau 8) donne le nombre de transistors.

Fonctionnalités	STTL pseudo statique	STTL statique	AO222	[GUI04]	[KUL05]	[TIR02]
Nand2/Nor2/And2/Or2	27	35	64	112	18	14
Nand3/Nor3/And3/Or3	29	37	128	224	26	28
Xor2/Xnor2	29	37	68	80	19	18
Xor3/Xnor3	35	43	136	160	38	36
AO21/ AOI21	39	47	128	224	36	28
AO22/ AOI22	42	50	192	336	54	42

Tableau 8. Comparaison des coûts de réalisation en nombre de transistors

Il apparaît que les styles de logique à précharge [Kul05, Tir02] sont les moins complexes et les plus compactes. Toutefois à propos de ces estimations, la surface de l'arbre à précharge n'est pas prise en considération ce qui fait que ces chiffres ne sont pas très représentatifs de la

réalité physique. Après les cellules de [Kul05] et Tir02], nos cellules STTL sont les plus optimisées en surface. Par rapport aux cellules à base de la porte complexe AO222, nous avons des réductions en nombre de transistors allant de 57% à 78% et des réductions allant de 63% à 87% par rapport aux cellules proposées dans [Gui04].

Pour une analyse plus réaliste de la surface, nous avons mené une estimation de surface d'un circuit après la phase de placement routage et selon les différents styles de logique. Pour ce faire, nous proposons la réalisation de la structure de la figure 79, qui à la base est destinée à l'évaluation de robustesse aux attaques DPA.

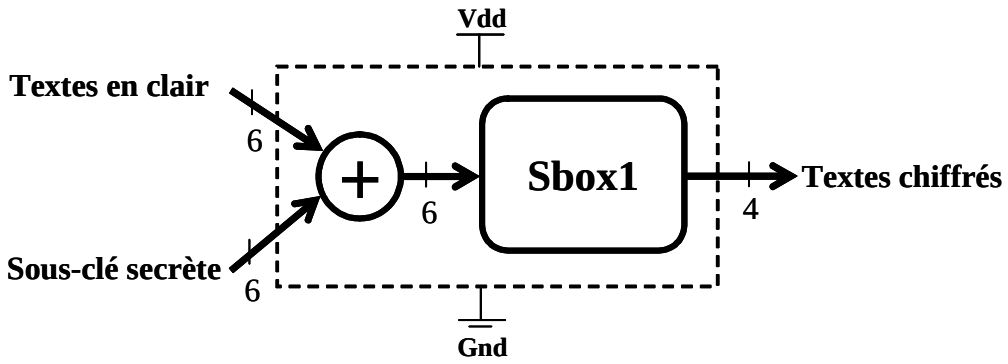


Fig. 79. Structure d'évaluation

Pour des raisons de simplicité, nous nous sommes restreints à l'utilisation de cellules à deux entrées pour la réalisation de cette structure.

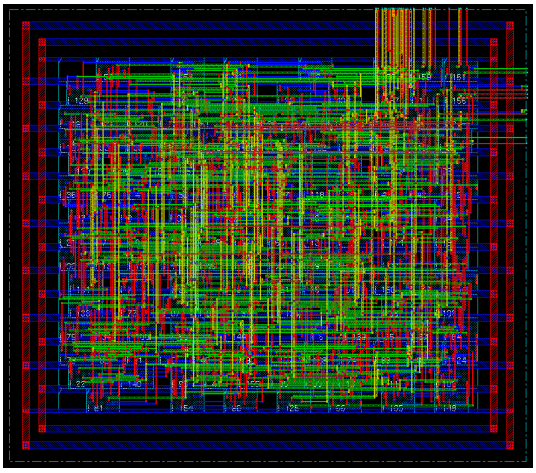
Pour l'étape de placement routage en général, il faut un certain nombre de fichiers à savoir: les fichiers liés à la technologie, le fichier Verilog structurel (description au niveau porte), le fichier LEF (délimitation physique des portes).

Tout d'abord pour le fichier Verilog structurel double rail, nous avons développé un analyseur syntaxique capable de le générer à partir d'un fichier Verilog structurel classique (simple rail) obtenu après synthèse. Typiquement, nous avons réalisé une spécification VHDL de la structure de la figure 79 que nous avons par la suite synthétisé avec l'outil "Ambit" (Cadence).

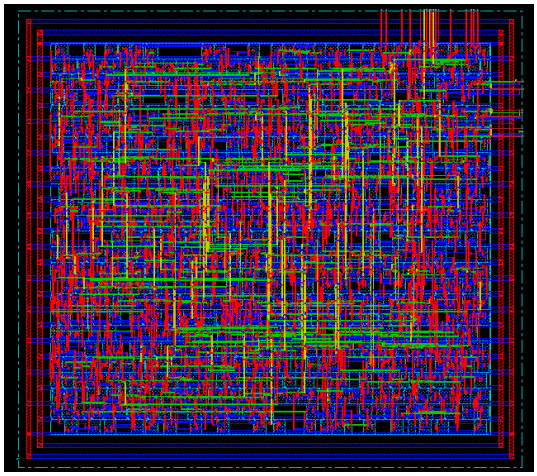
Quant à la création du fichier LEF, nous avons effectué des estimations de surface en prenant comme référence une latch de la bibliothèque standard. En clair, nous supposons que le rapport "nombre de transistors / surface physique" d'une latch est identique à celui des cellules des différents styles de logique considérés précédemment dans ce manuscrit. Le tableau 9 présente ces estimations de surface des cellules selon le style d'implémentation considéré.

Portes	STTL	_AO222	[TIR02]	[GUI04]
EOLL_DR	34,29 µm²	92,79 µm²	22,18 µm²	108,43 µm²
MUX21NLL_DR	46,39 µm²	318,71 µm²	22,18 µm²	108,43 µm²
ND2LL_DR	34,29 µm²	84,72 µm²	16,13 µm²	156,84 µm²
NR2LL_DR	34,29 µm²	84,72 µm²	16,13 µm²	156,84 µm²
OR2LL_DR	34,29 µm²	84,72 µm²	16,13 µm²	156,84 µm²

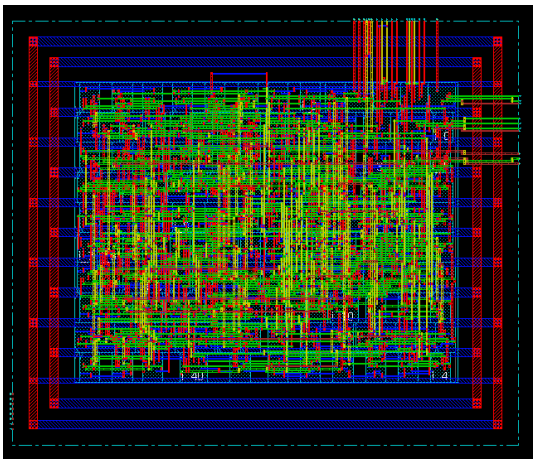
Tableau 9. Estimation de surface physique des cellules



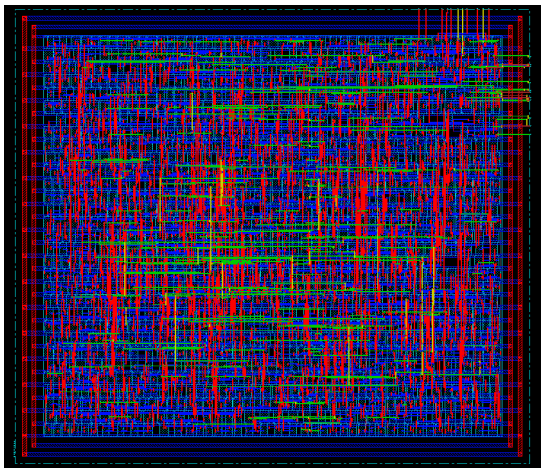
[Circuit STTL] surface estimée: 10121 µm²



Circuit à base de AO222 surface estimée: 22711 µm²



[Tir02] surface estimée: 5756 µm²



[Gui04] surface estimée: 31753 µm²

Fig. 80.Résultats de placement routage

Pour la phase de placement routage, nous avons utilisé l'outil "First Encounter" [Cad06] de chez Cadence. Pour des raisons d'équité, nous avons adopté les mêmes conditions d'environnement pour l'ensemble des différents styles d'implémentation lors de la phase de placement routage comme par exemple le taux d'occupation du cœur qui est de 80%. Les résultats sont présentés par la figure 80, et comme on peut le constater, notre circuit STTL est deux fois plus gros que celui à base des cellules proposées dans [Tir02] mais largement plus

petit que les circuits à base de la porte complexe AO222 et des cellules proposées dans [Gui04].

V-3-b-Délais de propagation et consommation

Pour l'évaluation des performances en vitesse et consommation des différents styles d'implémentation, nous proposons l'utilisation du protocole de simulation proposé dans le chapitre III et illustré par la figure 81. Typiquement, ce protocole de simulation va nous permettre de capturer les délais de propagation et la consommation (l'énergie) d'une porte en fonction de la rampe d'entrée et de la charge qui sont les paramètres environnementaux les plus influents.

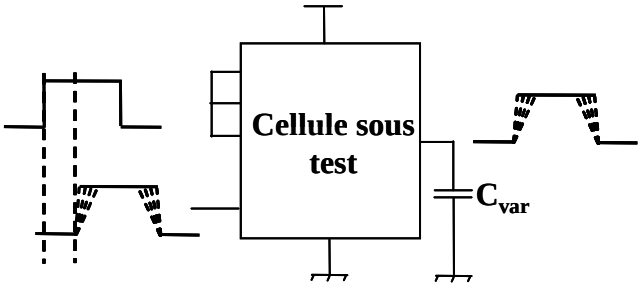


Fig. 81. Protocole de simulation

Pour cette campagne de simulation, nous avons dessiné les schémas de quelques fonctionnalités de base (N/OR2, XOR2, AO22, etc.) et selon les différents styles d'implémentation dont le nôtre (STTL), celui à base de la porte complexe AO222 et ceux proposés dans [Tir02] et [Gui04]. Cependant, ces évaluations présentent quelques carences: **1)** les capacités parasites engendrées par le layout et par le routage des cellules composites (AO222) ne sont pas prises en considération; **2)** dans le cas d'une logique dynamique ([Tir02]), la consommation de l'arbre d'horloge n'est pas extraite. Toutefois, les capacités parasites restent faibles avec une conception soignée et si toutes les composantes des cellules composites sont placées les unes à proximité des autres.

Pour rappel, la stratégie de dimensionnement des transistors consiste à imposer, pour toutes les fonctionnalités et pour une même valeur de charge, des temps de transition identiques en sortie des cellules. En d'autres termes, les cellules ont été dimensionnées de telle façon à ce qu'elles aient les mêmes possibilités en courant. Par ailleurs, le choix des domaines de rampe et de charge a été effectué de façon à couvrir le plus d'espace de conception à savoir des valeurs de rampes d’entrée et de charge en sortie comprises respectivement entre 10ps et

200ps et 5fF et 50fF. Enfin pour une meilleure appréciation des résultats, nous avons choisi de les présenter sous forme pire cas. Le pire cas est choisi en fonction de l'ordre d'arrivée des signaux d'entrée.

Les figures 82 et 83 reportent les résultats que nous avons obtenus pour une porte 'OR2', sous différents styles d'implémentation et pour le passage de l'état invalide à un des deux états valides. Toutefois, des résultats identiques ont été obtenus pour les autres fonctionnalités.

En terme de délai de propagation, comme l'illustre la figure 82, les cellules STTL sont jusqu'à 2.5 fois plus rapides que celles à base de cellules proposées dans [Gui04] et jusqu'à 1.5 fois plus rapides que celles à base de porte complexe AO222 [Ren00a]. Toutefois, elles sont moins performantes par rapport à des cellules LISAL proposées dans le chapitre III et à celles proposées dans [Tir02]. Par ailleurs, ces gains et pertes en performance au niveau des délais de propagation sont prévisibles dans la mesure où ils sont directement liés à la complexité et à la profondeur logique de la cellule considérée.

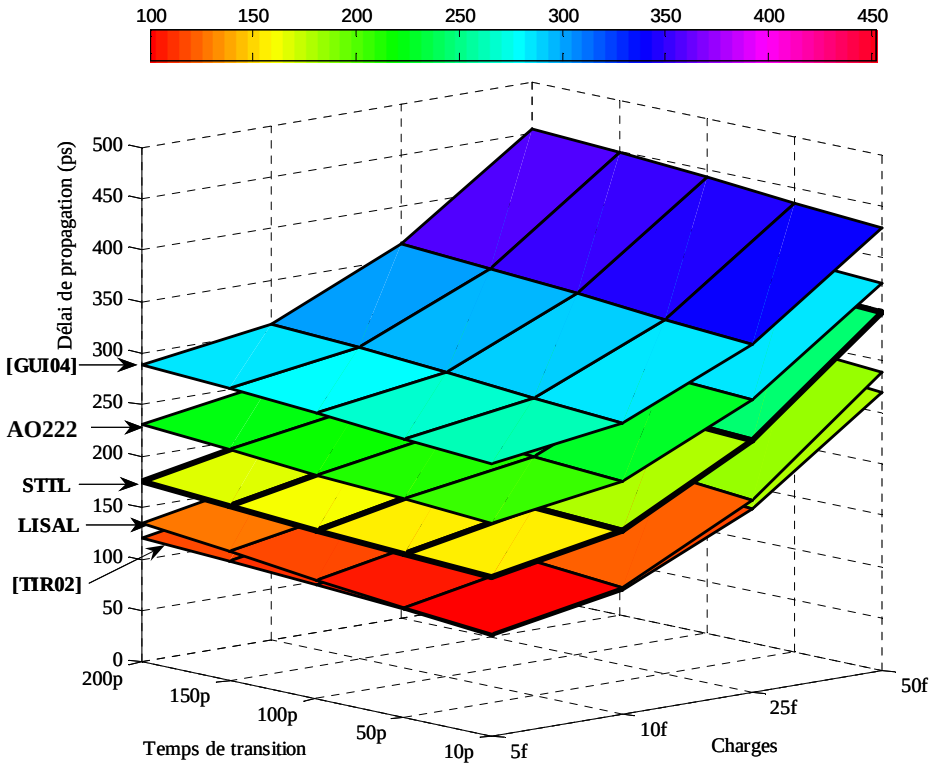


Fig. 82.Délais de propagation d'une porte 'OR2' sous différents styles d'implémentation

En terme de consommation, comme le montre la figure 83, le gap entre les différents styles d'implémentation est beaucoup moins important. En effet, les cellules STTL consomment jusqu'à 1.7 fois plus que celles à base de porte complexe AO222 et jusqu'à 0.15 moins que

celles à base de cellules proposées dans [Gui04].

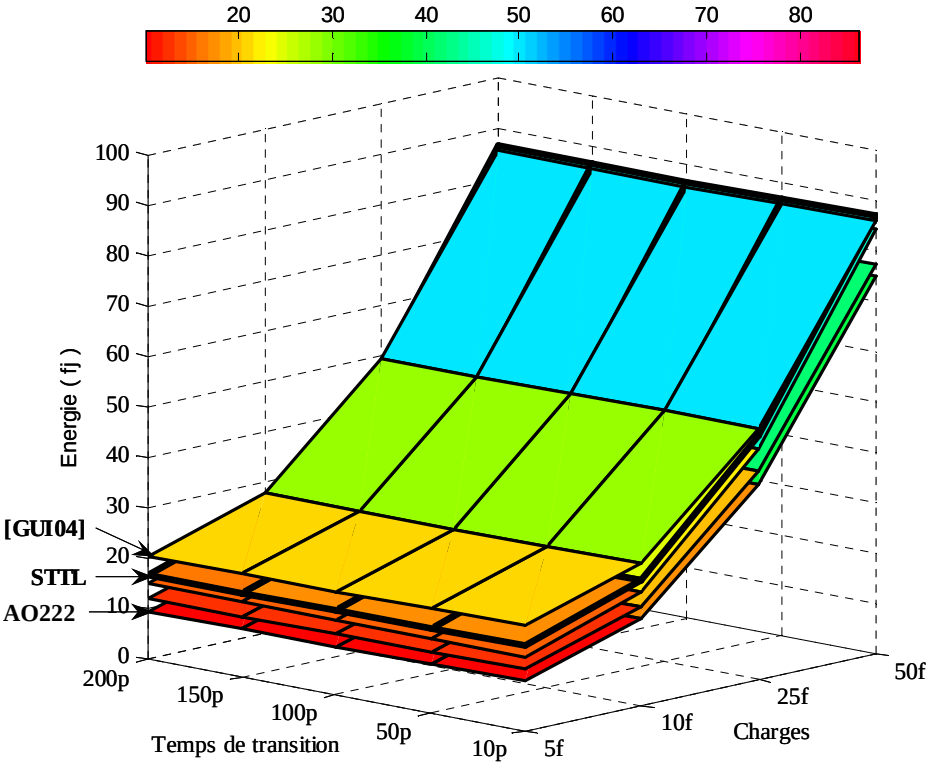


Fig. 83. Consommation d'une porte 'OR2' sous différents styles d'implémentation

V-4 Validations expérimentales

Dans ce paragraphe, nous présentons des validations et des évaluations de nos cellules STTL. Pour les différentes validations qui vont suivre, nous nous sommes basés sur la structure de la figure 79 qui représente un bloc sensible de l'algorithme de chiffrement DES. Plus précisément, nous avons utilisé notre bibliothèque de cellules STTL pour implémenter ce circuit. Par ailleurs, ce même circuit a été implémenté à partir d'autres bibliothèques dont celle à base de AO222 et celle proposée dans [Tir02] et [Gui04] pour pouvoir effectuer des comparaisons.

Dans le paragraphe V-2-b, nous avons annoncé qu'un circuit à base des cellules STTL est attendu pour avoir certaines propriétés intéressantes pour se prémunir de la DPA comme l'insensibilité aux capacités parasites et la constance du temps de cycle indépendamment des données traitées. Par conséquent, dans les deux paragraphes qui vont suivre, nous procédons à une vérification de ces propriétés des circuits STTL.

V-4-a-Insensibles aux déséquilibres de charge

Pour vérifier cette insensibilité aux déséquilibres de charge introduits par le placement routage, nous avons mené des attaques DPA sur la structure de la figure 79 à base des cellules STTL. Pour pouvoir visualiser les impacts des capacités parasites, les simulations d'attaque DPA sont d'abord réalisées sur une netlist idéale c'est-à-dire sans les capacités parasites et sur une netlist avec les capacités parasites. Ces mêmes expériences sont réalisées pour les autres circuits (à base de AO222, Gui04, Tir02]. Pour des raisons d'équité, nous avons adopté la même politique de dimensionnement qui est spécifiée dans le paragraphe V-3-b et utilisé les mêmes paramètres de placement routage avec "First Encounter".

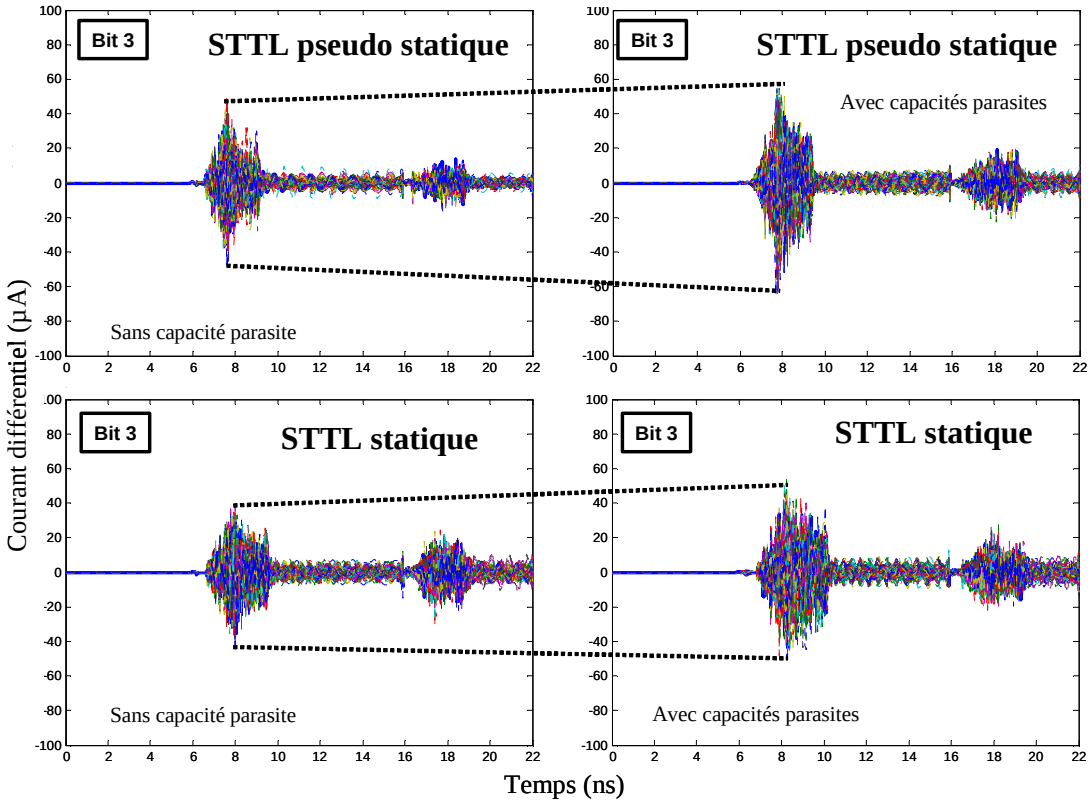


Fig. 84.Résultats des attaques DPA menées sur le circuit STTL

Pour une analyse la plus complète qui soit, nous avons mené des attaques DPA exhaustives (i.e avec les 64 propositions de clef) sur l'ensemble des différents circuits (STTL, SABL, etc.): pseudo statique (avec et sans les capacités parasites), statique (avec et sans les capacités parasites). La figure 84 présente les résultats des attaques menées sur le bit 3.

A partir de cette figure, deux conclusions peuvent être tirées: **1)** une attaque DPA exhaustive menée sur le bit 3 mais aussi sur les autres bits n'a pas permis de retrouver la bonne valeur de

la clef secrète. Ainsi, dans ces conditions, notre style de logique STTL offre aux circuits, une bonne résistance aux attaques DPA. **2)** Comme attendu, "**la logique STTL est quasi insensible aux déséquilibres de charge introduits par la phase de placement routage**" dans la mesure où les signatures DPA sur un circuit idéal et avec capacités parasites n'affichent qu'une légère différence en amplitude et en temps. Cette conclusion est d'autant plus vraie pour le circuit statique car on peut le noter sur la figure 84, les résultats DPA sont quasi-identiques. Cette robustesse aux capacités parasites est en effet une conséquence directe d'une des propriétés intéressantes de la logique STTL qui est d'avoir un temps de calcul indépendant des données. Pour effectuer une comparaison de performance par rapport aux impacts des capacités parasites, nous avons mené la même expérience sur un circuit à base des cellules sécurisées proposées dans [Gui04] qui à notre connaissance offre la meilleure résistance aux attaques DPA. La figure 85 présente les résultats obtenus. Tout d'abord, la bonne résistance de ces cellules aux attaques DPA est confirmée dans la mesure où les amplitudes des signatures en courant sont très faibles et proches de zéro [-19µA +19µA]. Toutefois, après ajout des capacités parasites, ces amplitudes en courant deviennent plus importantes et sont maintenant comprises entre [-45µA +30µA]. En conclusion, l'introduction des capacités parasites a largement modifié les signatures DPA (en amplitude et en temps) allant même jusqu'à rendre le circuit non résistant aux attaques. En effet, la figure 85 montre qu'avec les capacités parasites, l'attaque DPA est un succès.

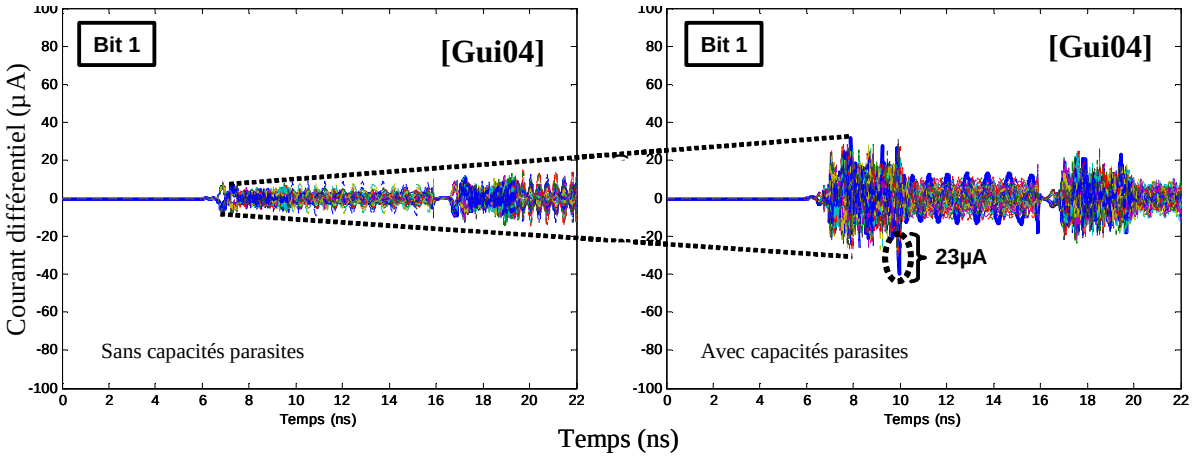


Fig. 85.Résultats des attaques DPA menées sur un circuit à base des cellules proposées dans [Gui04]

V-4-a-Temps de cycle indépendant des données manipulées

Dans cette partie, nous procédons à la vérification d'une autre caractéristique de la logique

STTL qui est d'avoir un temps de calcul indépendant des données manipulées. Pour ce faire, nous nous sommes appuyés sur les simulations électriques des différents styles d'implémentation (STTL, à base de AO222 [Gui04, Tir02]) incluant les capacités parasites de la structure de la figure 79. Plus précisément, nous avons capturé le temps de propagation des signaux des entrées vers les sorties et ce pour un grand nombre de vecteur d'entrée. Tout naturellement, la mesure des délais de propagation de la structure commence à '50%*V_{IN}' et s'arrête à '50%*V_{OUT}', V_{IN} et V_{OUT} étant respectivement les signaux d'entrée et sortie.

	Bit cible	**<T_1> (ps)**	**<T_0> (ps)**	**|<T_1>-<T_0>| (ps)**
STTL	**S_1**	3547	3528	19
	S_2	3748	3727	21
	S_3	3663	3649	14
	S_4	3803	3792	11
[Tir02]	**S_1**	2194	2226	27
	S_2	2224	2288	64
	S_3	2238	2330	93
	S_4	2237	2432	195
[Gui04]	**S_1**	3959	3943	17
	S_2	4071	4060	12
	S_3	3911	3908	3
	S_4	3981	3970	11
[Mau03]	**S_1**	2211	2171	40
	S_2	2317	2266	51
	S_3	2251	2212	39
	S_4	2311	2285	26
[Raz05]	**S_1**	1766	1703	63
	S_2	1758	1705	53
	S_3	1782	1735	48
	S_4	1870	1811	11

Tableau 10. Tableau reportant les délais de propagation des différents circuits (STTL, [Gui04, Tir02, Mau03, Raz05])

Il est clair que le délai de propagation varie en fonction de la sortie considérée et le type d'évènement, aussi dans le tableau 10, nous proposons les valeurs moyennes des délais de propagation en fonction du bit cible et selon si la donnée manipulée est un '1' (<T_1>) ou un '0' (<T_0>). Pour apprécier effectivement la constance du temps de calcul notre style de logique indépendamment de la valeur calculée ('1' ou '0'), nous proposons également une troisième métrique qui est la différence absolue entre les deux moyennes: | <T_1>-<T_0>|.

La première conclusion par rapport au tableau ci-dessus, est que la différence entre le temps

nécessaire au calcul d'un '1' et d'un '0' est dépendante du style de logique considéré. En effet, cette différence peut aller de quelques picosecondes à quelques centaines de picosecondes. La deuxième conclusion qui nous intéresse plus particulièrement est le degré de constance des délais de propagation. Visiblement, après le style de logique proposé dans [Gui04], notre style de logique STTL apparaît comme la logique ayant les plus petites valeurs et variations de cette différence entre le temps nécessaire entre la manipulation d'un '1' et d'un '0'. En somme, notre style de logique STTL affiche un temps de calcul quasi-indépendant des données manipulées pour tous les vecteurs d'entrée possibles.

Pour simplifier la compréhension de ces résultats et conforter notre conclusion du paragraphe précédent, nous avons reporté directement les délais de propagation sur un même graphe comme l'illustre la figure 86. On notera que ces graphes de distribution ont été calculés uniquement pour le bit cible 3 et ce pour l'ensemble des différents style de logique.

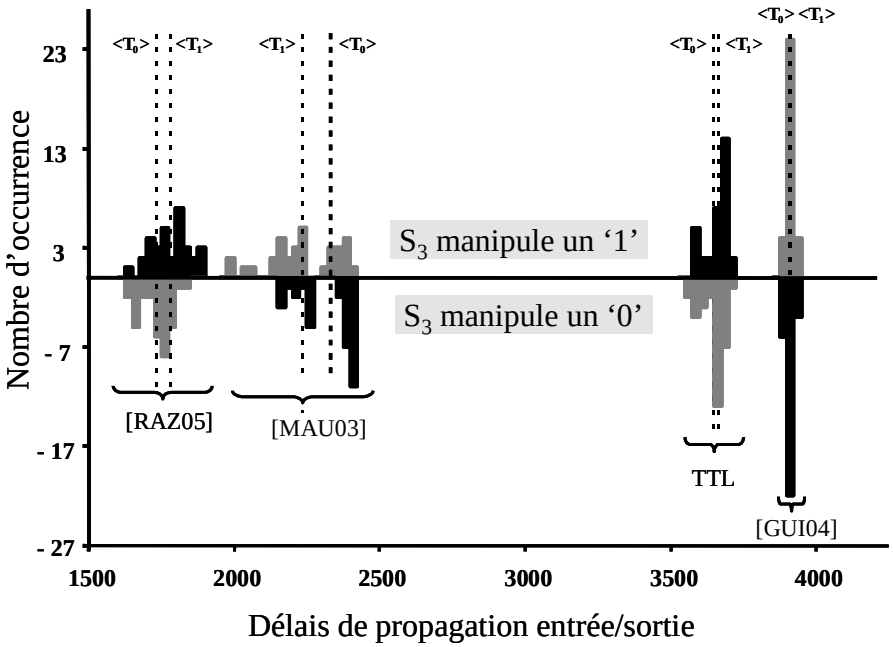


Fig. 86. Variation des délais de propagation entrée/sortie

Cette nouvelle représentation des résultats nous permet d'une part d'apprécier visuellement les variations au niveau des délais de propagation des signaux selon si la valeur calculée est un '1' ou un '0' et d'autre part d'apprécier les performances en vitesse des différentes structures. Par ailleurs, cette nouvelle représentation peut être un bon indicateur du niveau de robustesse d'un circuit à l'attaque DPA. En effet, plus les distributions des délais sont symétriques et étroites, plus le circuit a un comportement en vitesse indépendant des données manipulées et plus il est

robuste aux attaques DPA.

Comme prévu, le circuit à base de cellules sécurisées proposées dans [Gui04] affiche un temps de calcul le plus constant qui soit dans la mesure où d'une part, les distributions des délais correspondantes respectivement à la manipulation d'un '1' et d'un '0' sont quasi-symétriques par rapport à l'axe horizontal, et d'autre part, les distributions affichent un faible taux de dispersion. Chose surprenante, le circuit à base de nos cellules STTL (spécialement conçues pour avoir un temps de calcul le plus constant qui soit) affiche un taux de dispersion des distributions des délais plus important que celui du circuit à base des cellules proposées dans [Gui04]. Cet état de fait peut s'expliquer en partie par le fait que ces cellules sécurisées ont une profondeur logique très importante ([4-6] couches logiques) et que cela puisse masquer les effets des capacités de routage. Toutefois, par rapport aux autres styles de logique (autre que [Gui04]), ayant les distributions des '1' et des '0' quasi-symétriques, notre style de logique STTL est celui qui a un temps de calcul le plus décorrélé des données. En conclusion, on peut affirmer qu'un circuit STTL a effectivement un temps de cycle quasi-indépendant des données manipulées.

V-4-c- Comparaison de robustesse aux attaques DPA

Nous proposons, dans ce paragraphe, une comparaison de robustesse aux attaques DPA entre les différents types de circuit. Parmi ces circuits, on notera celui à base de porte complexe AO222, de cellules STTL, de cellules proposées dans [Gui04], dans [Tir02], et dans [Mau03]. Comme dans le paragraphe V-4-a, nous avons effectué des attaques DPA par simulation sur des schémas électriques incluant les capacités parasites. La figure 87 présente l'ensemble des résultats obtenus.

Une première conclusion est que les attaques DPA menées sur les circuits STTL n'ont pas permis de retrouver la valeur de la clef secrète. En effet, la courbe correspondante à la manipulation de la bonne clef (courbe épaisse bleu) est complètement noyée dans le tas. Une deuxième conclusion est qu'avec des amplitudes en signatures DPA se trouvant dans l'intervalle [-40 µA - +45 µA], notre circuit STTL statique affiche un niveau de sécurité comparable voire meilleur à celui du circuit à base de cellules sécurisées proposées dans [Gui04]. En effet, malgré des signatures en courant comprises dans l'intervalle [-40 µA - +38 µA], la version [Gui04] laisse fuir de l'information sur la valeur de la clef secrète. En d'autres termes, l'attaque DPA sur cette version [Gui04] est une réussite. Par ailleurs, les attaques DPA

réalisées sur les autres styles d'implémentation (à base de AO222, [Mau03], [Tir02]) ont respectivement permis de retrouver la clef secrète. D'une part, sur les versions à base de AO222 et [Mau03], ces résultats sont prévisibles dans la mesure où ces styles de logiques n'étaient pas conçus pour la conception de circuit sécurisé.

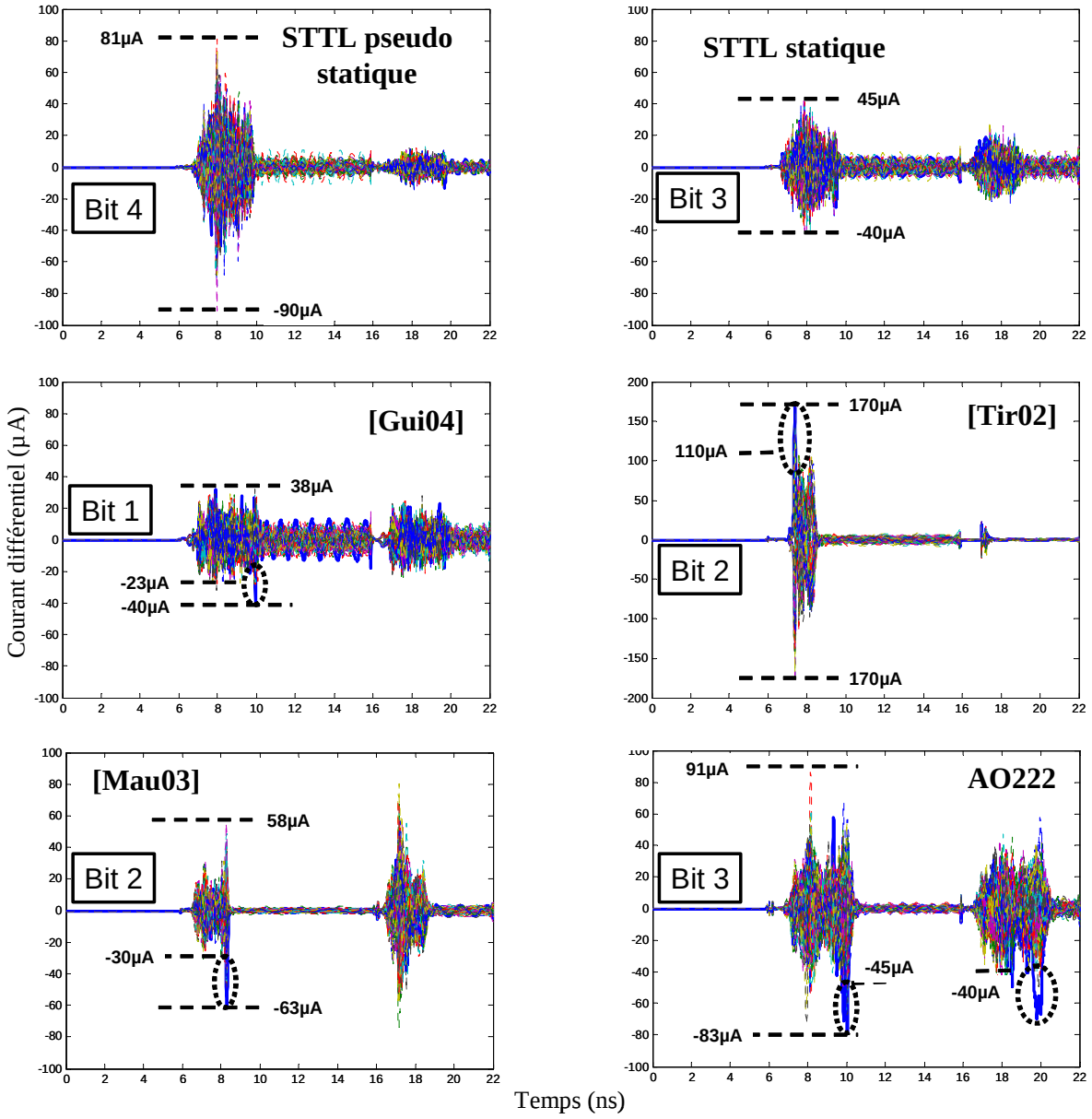


Fig. 87.Résultats des attaques DPA menées sur les différents circuits

D'autre part, sur la version [Tir02], ces résultats nous ont quelque peu surpris à partir du moment où ce style de logique (SABL) a été spécialement conçu pour contrecarrer les attaques DPA. L'absence d'une gestion appropriée des signaux de précharge/évaluation peut être en partie une explication à ces résultats.

V-5- Conclusion

Dans le chapitre précédent, nous avons clairement identifié que l'espace de conception dans lequel nos cellules double rail (LISAL) peuvent être considérées comme robuste aux attaques DPA est assez restreint et plus particulièrement par rapport aux décalages temporels.

Pour affaiblir la sensibilité des cellules à ces décalages temporels et élargir ainsi l'espace de conception sécurisé, nous avons proposé dans ce chapitre une variante de la logique double rail qui est la logique triple rail sécurisée: STTL. Les cellules STTL proposées ont été conçues de telle sorte que "des décalages temporels au niveau des données d'entrée n'aient aucune influence sur leurs profils en courant" ou encore que "le temps de calcul soit indépendant des données".

Une méthode de conception simple de mise en œuvre a été proposée pour la réalisation de cellule STTL ce qui nous a permis de développer notre bibliothèque de cellules STTL. Des validations expérimentales ont été menées et nous ont permis de confirmer qu'effectivement un circuit à base de cellules STTL a un temps de calcul quasi-indépendant des données manipulées et est par conséquent insensible aux déséquilibres de charges introduits par le placement routage.

Enfin, des comparaisons de robustesse aux attaques DPA ont montré qu'un circuit STTL est très résistant et a un niveau de sécurité comparable à un circuit à base de cellules sécurisées proposées dans [Gui04]. Toutefois, les cellules STTL sont largement plus compactes et consomment beaucoup moins que ces cellules proposées dans [Gui04]. Dans ce contexte, un circuit STTL apparaît comme un excellent compromis "robustesse aux attaques DPA – surface – consommation".

Conclusion

Conclusion

Dans le domaine de la cryptographie, il est aujourd'hui clairement identifié que le talon d'Achille des applications sécurisées réside dans leur implantation matérielle. C'est peut être la raison pour laquelle les attaques matérielles ou « Side Channel Attacks », comme par exemple les analyses différentielles de la consommation (DPA) et des émissions électromagnétiques (SEMA, DEMA), se sont généralisées. Ces attaques, et notamment l'attaque DPA, sont maintenant reconnues comme les attaques les plus dangereuses dans le sens où elles permettent à moindre frais et avec un faible niveau de compétences d'obtenir les clefs de chiffrement des algorithmes de chiffrement standard comme le DES et l'AES qu'utilisent nos cartes à puces.

Dans ce contexte, les technologies asynchrones constituent une alternative à la technologie synchrone pour la conception de circuits intégrés sécurisés, en particulier pour les cartes à puces. Des méthodes de conception ad hoc permettent ainsi de concevoir des fonctions logiques dont l'activité électrique est indépendante des données traitées rendant moins efficaces les attaques différentielles en puissance (DPA).

Ainsi, dans le cadre d'un projet de collaboration tripartite entre le LIRMM, le TIMA et la division SmartCard de STM Rousset dont l'objectif à long terme est la mise en place d'un flot de synthèse de circuit asynchrone, ce travail de thèse a consisté à définir une bibliothèque de cellules sécurisées avec une stratégie d'utilisation afin d'éviter les fuites en consommation.

Dans une première partie de la thèse, nous nous sommes focalisé sur les attaques par canaux cachés ou "side channel attacks" et plus particulièrement sur les attaques DPA et les contre-mesures à base de logique spécifique. L'utilisation de la logique double rail, largement utilisée dans la conception de circuits asynchrones, apparaît comme une approche intéressante dans la mesure où elle offre la possibilité d'équilibrer la consommation. Cependant, en l'absence de flot de conception industriel, la conception des circuits double rail est largement handicapée par un coût en surface très important. La pertinence de la solution double rail étant clairement établie, le premier objectif de ces travaux de thèse était de développer une bibliothèque de cellules double rail dont les principales caractéristiques sont: **1)** consommation indépendante des données, **2)** optimisées en surface, **3)** compatibles avec une

bibliothèque standard, **4**) utilisables dans un environnement asynchrone et synchrone. Pour la réalisation de telles cellules, nous avons développé une méthode de conception de cellules double rail assez simple de mise en œuvre. Un bloc sensible de l'algorithme de chiffrement DES a été implémenté à partir de nos cellules double rail mais aussi à partir d'autres styles d'implémentation qui existent dans la littérature. Des attaques DPA par simulation ont été mises en œuvres sur les différents circuits. En conclusion, par rapport aux autres styles d'implémentation, le circuit à base de nos cellules double rail affiche une surface moins importante et offre un niveau de robustesse meilleur sinon équivalent.

Cependant, les attaques DPA ont été menées dans des conditions idéales, c’est-à-dire sans considération des capacités parasites internes aux cellules et des capacités de routage qui, rappelons-le, peuvent localement être plus importantes que les capacités d’entrée des portes. Tout naturellement, si aucune précaution particulière n’est prise lors de la phase de placement-routage des circuits double rail, des déséquilibres importants peuvent être introduits au niveau des noeuds différentiels et faire perdre le principal avantage des cellules double rail face aux attaques DPA, qui est d'avoir une consommation indépendante des données manipulées.

Dans ce contexte, nous nous sommes focalisés sur les impacts de la synthèse physique sur la robustesse des circuits double rail. Plus particulièrement, nous nous sommes focalisés sur les impacts que peuvent avoir les dissymétries, induites par l’introduction de capacités de routage, au niveau des noeuds différentiels sur les signatures DPA. Afin de pouvoir analyser finement ces impacts, nous avons développé, dans un premier temps, un modèle analytique des signatures en courant des cellules double rail. Dans un deuxième temps, ce modèle a été exploité pour l'analyse formelle de la sensibilité des signatures en courant par rapport aux dissymétries des nœuds différentiels, des temps de transition et des temps d'arrivée des signaux. Cette analyse de sensibilité a conduit à l’identification d'un espace de conception sécurisé dans lequel une cellule double rail peut être considérée comme étant robuste aux attaques DPA. En d'autres termes, en fonction d'un certain nombre de paramètres technologiques et environnementaux, cet espace de conception sécurisé définit un seuil de dissymétrie tolérable au niveau des nœuds différentiels, des temps de transitions et des temps d'arrivée des signaux. L'étude de cet espace de conception sécurisé a permis de conclure qu’il n’est pas nécessaire de symétriser de manière parfaite les interconnexions entre les cellules

élémentaires et qu'une cellule double rail n’est robuste que si les signaux, pouvant potentiellement déclencher sa commutation, arrivent tous dans un intervalle de temps particulièrement réduit.

Pour réduire la sensibilité des cellules à ces décalages temporels et élargir ainsi l'espace de conception sécurisé, nous avons proposé une variante de la logique double rail qui est la logique triple rail sécurisée ou STTL (Secure Triple Track Logic). En clair, les cellules STTL ont été conçues de telle sorte que "des décalages temporels au niveau des données d'entrée n'aient aucune influence sur leurs profils en courant" ou encore que "le temps de calcul soit indépendant des données". Des validations par simulation ont permis de conforter la robustesse des circuits STTL aux décalages temporels et par conséquent aux capacités parasites. Par rapport aux autres styles d'implémentation dédiés à la conception de circuits sécurisés, un circuit STTL affiche un niveau de robustesse supérieur ou équivalent tout en offrant une certaine compacité. Il offre donc un très bon compromis sécurité / compacité, la surface constituant un des handicaps importants des circuits double rail et des circuits asynchrones

Références bibliographiques

Bibliographie

[Abr91] D.G. Abraham, G.M. Dolan, G.P. Double, and J.V. Stevens, "Transaction security systems", IBM Systems Journal, 30(2) :206-229, 1991.

[Akk01] M. L. Akkar and C. Giraud, "An Implementation of DES and AES Secure Against Some Attacks," presented at Cryptographic Hardware and Embedded Systems (CHES2001), pp. 309-318, LNCS 2162, Paris, France, May 14-16, 2001.

[Arc03] D. Agrawal, B. Archambeault, S. Chari, J. R. Rao and P. Rohatgi. Advances in Side-Channel Cryptanalysis. RSA Laboratories Cryptobytes, Vol.6, number 1, pages 20-32, 2003.

[Auv00] D. Auvergne, J. M. Daga, and M. Rezzoug, "Signal Transition Time Effect on CMOS Delay Evaluation," IEEE Transactions on Circuits and Systems, vol. 47 N°9, pp. 1362-1369, 2000.

[Bar00] A. Bardsley, D. A. Edwards. The Balsa Asynchronous Circuit Synthesis System. FDL 2000, Sept. 2000.

[Bec98] F. Beck: Integrated Circuit Failure Analysis: A Guide to Preparation Techniques. JohnWiley & Sons, 1998.

[Bie00] I. Biehl, B. Meyer and V. Müller. Differential fault attacks on elliptic curve cryptosystems. In M. Bellare, editor, Advances in Cryptology: Proceedings of CRYPTO'00, number 1880 of LNCS, pages 131-146,Springer-Verlag.

[Bih90] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like cryptosystems (Extanded abstract)," presented at Advances in Cryptology Crypto'90, Springer-Verlag 1991 ed, vol. 537 of LNCS, pp. 2-21, 1990.

[Bih97] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," presented at 17th annual International Cryptology Conference on Advances in Cryptology CRYPTO'97, vol. 1294, pp. 513-525, California, USA, 1997.

[Bih99] E. Biham and A. Shamir, "Power Analysis of the Key Scheduling of the AES Candidates," presented at Second AES Candidate Conference (AES2),Rome, Italy, March 22-23, 1999.

[Bon97] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," presented at EUROCRYPT'97, vol. 1233, pp. 37-51, Berlin, 1997.

[Bou04] F. Bouesse, M. Renaudin, B. Robisson, E. Beigne, P. Y. Liardet, S. Prevosto, and J. Sonzogni, "DPA on Quasi Delay Insensitive Asynchronous Circuits: Concrete Results," presented at XIX Conference on Design of Circuits and Integrate

[Bou05] F. Bouesse, M. Renaudin, A. Witon, and F. Germain, "A Clock-less low-voltage AES crypto-processor," presented at 31st European Solid-State Circuits Conference (ESSCIRC2005), pp. 403-406, Grenoble, France, 2005.

[Bou05a] F. Bouesse, "Contribution à la conception de circuits intégrés sécurisés: l'alternative asynchrone" thèse soutenue le 01 décembre 2005, à l'Institut National Polytechnique de Grenoble, France, 2005. http://tima.imag.fr/Publications/files/th/csd_221.pdf#search=%22Fraidy%20Bouesse%20%26%20these%22

[Cad06] http://www.cadence.com

[Chu86] K.M. Chu et D. Pulfrey, "Design procedures for differential cascode voltage switch circuits", IEEE journal of solid-state circuits, Volume SC-21, N° 6, pp 1082 -1087, Dec. 1986.

[Chu87] K. M. Chu, D. L. Pulfrey, “A Comparison of CMOS Circuit Techniques: Differential Cascode Voltage Switch Logic versus Conventional Logic”, IEEE Journal of Solid State Circuits, SC-22n n°4, August 1987

[Cor00] J. S. Coron and L. Goubin, "On Boolean and Arithmetic Masking against Differential Power Analysis," presented at Cryptographic Hardware and Embedded Systems (CHES2000), pp. 231-

237, LNCS 1965, Worcester, MA, USA, August 17-18, 2000.

[Cor96] Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, Alex Yakovlev, "Petrify: A Tool for Manipulating Concurrent Specifications and Synthesis of Asynchronous Controllers", in IEICE Transactions on Information and Systems.

[Dae99] J. Daemen and V. Rijmen. Resistance against implementation attacks:A comparative study of the AES proposals. In Proceedings of the Second Advanced Encryption Standard (AES) Candidate Conference, March 1999, http://csrc.nist.gov/CryptoToolkit/aes/round1/conf2/aes2conf.htm.

[Dag99] J. Daga, D. Auvergne, "A comprehensive delay macro modelling for submicrometer CMOS logics", IEEE Journal of Solid-State Circuits, Vol.34, n°1, January 1999.

[Dhe98] J. F. Dhem, F. Koeune, P. A. Leroux, P. Mestre, J. J. Quisquater, and J. L. Willems, "A Pratical Implementation of the Timing Attack," presented at The Third International Conference on Smard Card Research and Applications (CARDIS'

[Dif76] W. DIFFIE & M. HELLMAN, New directions in cryptography, IEEE TIT, vol. 22, (1976), pp 644--654

[Erd84] C.H. Erdelyi, W.R. Griffin et R.D. Kilmoyer, "Cascode Voltage Switch Logic Design", VLSI design 84, pp. 78-86, Oct. 1984.

[FIPS140-2] http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[FIPS186] NIST, "DIGITAL SIGNATURE STANDARD (DSS)," National Institut of Standard and Technology FIPS PUB 186-2, January 2000

[FIPS197] NIST, "Advanced Encryption Standard (AES)," National Institut of Standard and Technology, FIPS PUBS 197, November 2001

[FIPS46a] NIST, "Data Encryption Standard (DES)," National Institut of Standard and Technology FIPS PUB46-2, December 1993

[FIPS46b] NIST, "Data Encryption Standard (DES and Triple-DES)," National Institut of Standard and Technology FIPS PUB46-3, October 1999

[Ful06] EE Times: Fulcrum IC heats asynchronous design debate, http://www.eetimes.com/story/OEG20020819S0031

[Fur03] Z. C. Yu, S. B. Furber and L. A. Plana, "An Investigation into the Security of Self-Timed Circuits", in Proceedings Async 2003, pp 206-215, 2003

[Gan01] K. Gandolfi, C. Mourtel and F. Olivier. Electromagnetic Analysis: Concrete Results. In Ç. K. Koç, D. Naccache and C. Paar, editors, Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES), number 2162 of LNCS, pages

[Gem01] GEMPLUS, "Single Power Analysis," presented at GEMPLUS, Workshop on Cryptography and Security, March 13, 2001. http://:www.cs.uku.fi/kurssit/ads/09_20-_20SPA.pdf.

[Gou01] L. Goubin. A sound method for switching between boolean and arithmetic masking. In Ç. K. Koç, David Naccache and C. Paar, editors, Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES), number 2162 of LNCS, pages 3-15, 2001, Springer-Verlag.

[Gou99] L. Goubin and J. Patarin. DES and Differential Power Analysis ``The Duplication'' Method. In Ç. K. Koç and C. Paar, editors, Proceedings of 1st International Workshop on Cryptographic Hardware and Embedded Systems(CHES), number 1717 of LNCS, pages 158-172, 1999, Springer-Verlag.

[Gui04] S. Guillet et al. "CMOS structures suitable for secured hardware", Design, Automation and Test in Europe (DATE) Conference and Exposition, 2004.

[Han06] www.handshakesolutions.com

[Jep94] Kjell O. Jeppson, "Modeling the influence of the transistor gain ratio and the input to output coupling capacitance on the cmos inverter delay", IEEE, Solid-State Circuits, Vol. 29, N°6, June

1994.

[Kar01] R. Karri et al, "Fault-Based Side-Channel Cryptanalysis Tolerant Rijndael Symmetric Block Cipher Architecture", IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'01), IEEE, 2001

[Kar04] Robust Protection against Fault-Injection Attacks on Smart Cards Implementing the Advanced Encryption Standard, International Conference on Dependable Systems and Networks (DSN'04), p. 93, 2004.

[Kes01] J. Kessels, A. Peeters The Tangram framework: asynchronous circuits for low power. Proceedings of ASP-DAC, pp. 255–260, 2001.

[Koc96] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman,RSA, DSS, and Others Systems," presented at 16th International Cryptology Conference on Advances in Cryptology (CRYPTO'96), vol. 1109, pp. 104-113, Santa Barbara, Calif

[Koc99] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," presented at the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 99), Lecture Notes In Computer Sciences ed. Springer-Verlag, vol.

[Koc99a] P. C. Kocher, J. Jaffe, and B. Jun, "Introduction to Differential Power Analysis and Related Attacks", www.cryptography.com

[Koe99] F. Koeune and J. J. Quisquater, "Timing Attack against Rijndael," UCL Crypto Group, Louvain la Neuve June 10 1999.http://web.engr/oregonstate.edu/~aciicmez/osutass/data/Koeune99.dpf.

[Köm99] O. Kömmerling and M. G. Kuhn, "Design Principles for Tamper-Resistant Smartcard Processors," presented at USENIX Workshop on Smartcard Technoloy (Smartcard 99), pp. 9-19, Chicago, Illinois, USA,, May,10-11, 1999.

[Kul05] Konrad J. Kulikowski, Ming Su, Alexander B. Smirnov, Alexander Taubin, Mark G. Karpovsky, Daniel MacDonald, "Delay Insensitive Encoding and Power Analysis: A Balancing Act." Proceedings of ASYNC'2005, pp 116-125

[Lee93] T.W. Lee, S.V. Pabbisetty (eds.): Microelectronic Failure Analysis, Desk Reference. 3rd edition, ASM International, Ohio, 1993, ISBN 0-87170-479-X.

[Mac04] F. Mace and al. “A dynamic current mode logic to counteract power analysis attacks”, DCIS 2004.

[Mar90] A. J. Martin, Programming in VLSI: from communicating processes to delay-insensitive circuits, in C.A.R. Hoare, ed,Developments in Concurrency an Communication, UT Year of Programming Series: Addison-Wesley, 1990.

[Mar93] CAST Alain J. Martin, Drazon Borkovi'c, et al. CAST, Caltech Asynchronous Synthesis Tools: The first release. Technical Report Caltech-CS-TR-93-11, Computer Science Department, California Institute of Technology, 1993.

[Mau01] P. Maurine, "Modélisation et Optimisation des Performances de la logique Statique en Technologie CMOS Submicronique Profond," in Electronique, Optronique et Systèmes. Montpellier: Université Montpellier II, 2001.

[Mau03] P. Maurine, J. B. Rigaud, F. Bouesse, G. Sicard, M. Renaudin, "TAL : une bibliothèque de cellules pour le design de circuits asynchrones QDI", (FTFC'03), 4èmes journées d'études Faible Tension, Faible Consommation, 15-16 Mai 2003,

[Mca92] A.J. Mcaulley, "Four state asynchronous architectures", IEEE transactions on computers, volume 41, N°2, pp 129-142, February 1992.

[Mes00] T. S. Meserges, "Securing the AES Finanlist Against Power Analysis Attacks (FSE2000)," presented at Fast Software Encryption Workshop,New york, USA, April, 2000.

[Mes00a] T. S. Meserges, "Using Second-Order Power Analysis to Attack DPA Resistant Software," presented at Cryptographic Hardware and Embedded Systems (CHES2000), pp. 238-251, LNCS1965, Worcester, MA, USA, August 17-18, 2000.

[Mes99] T. S. Meserges, E. A. Dabbish, and R. H. Sloan, "Investigations of Power Analysis Attacks on Smartcards," presented at USENIX Workshop on Smarthcard Technology,Chicago, Illinois, USA, May 10-11, 1999.

[Moo02] Moore Simon, Ross Anderson, Cunningham Paul, Mullins Robert, and Taylor George, "Improving Smart Card Security using Self-timed Circuits," presented at Eighth International Symposium on Asynchronous Circuits and systems (ASYNC 2002)

[Mul01] D. Muller and N. P. Smart, "Random Registre Renaming to Foil DPA," presented at Cryptographic Hardware and Embedded Systems (CHES2001),Paris, France, 2001.

[Nik99] S. Nikolaidis and A. Chatzigeorgiou, "Analytical Estimation of Propagation Delay and Short-circuit Power Dissipation in CMOS Gates," International Journal of Circuit Theory and Applications, pp. 375-392, 1999.

[Pig97] C. Piguet, "Synthesis of asynchronous CMOS circuits with negative gates", Journal of Solid State Devices and Circuits (Brazil), 5(2):12-20, July 1997.

[Pig98] C. Piguet, J. Zhand "Electrical Design of Dynamic and Static Speed Independent CMOS Circuits from Signal Transistion Graphs" PATMOS '98, pp. 357-366, 1998.

[Qui00] J. J. Quisquater and D. Samyde, "A new tool for non-intrisuve analysis of smart cards based on electro-magnetic emissions, the SEMA and DEMA methods," presented at the rump session of EUROCRYPT'2000,Bruges, Belgium, May 14-18, 2000.

[Qui01] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): Measures and countermeasures for smard cards. In I. Attali and T. Jensen, editors, Proceedings of Smart Card Programming and Security (E-smart2001), number 2140 of LNCS, pages 200-210, 2001, Springer-Verlag.

[Qui02] J.-J. Quisquater, Math RiZK, François Koeune, "State-of-the-art regarding side channel attacks", from CRYPTREC, 2002, www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1047_Side_Channel_report.pdf

[Rak01] P. Rakers and T. Collins, "Secure Contactless SmartCard ASIC with DPA Protection," IEEE Journal of Solid-State Circuits, vol. 36, pp. 559-565, 2001.

[Raz**] "Voir bibliographie personnelle"

[Ren00] M. Renaudin, "Asynchronous Circuits and System: a Promising Design Alternative," Microlectronics for Telecommunications: Managing High Complexity and Mobility (MIGAS2000), special issue Microelectronics-Engineering Journal, vol. 54.

[Ren00a] Marc Renaudin, "Etat de l'art sur la conception des circuits asynchrones: perspectives pour l'integration des systemes complexes", Janvier. 2000, Internal Report, page 82 http://tima.imag.fr/publications/files/rr/eac_168.pdf#search=%22Marc%20Renaudin%20%26%20Etat%20de%20l'art%22

[Rig02] J. B. Rigaud, "Spécification de Bibliothèques pour la Synthèse de Circuits Asynchrones," in Institut National Polytechnique de Grenoble. Grenoble, France, 2002.

[Ros01] A. Ross, "Security Engineering:A Guide to Building Dependable Distributed Systems," vol. 1, Wiley Computer publishing ed. United States of America, 2001.

[Sam02] D. Samyde, S. Skorobogatov, R. Anderson, and J. J. Quisquater, "On a New Way to read Data from Memory," presented at First International IEEE Security in Storage Workshop,Greenbelt Maryland, December 11-11, 2002.

[Sch01] B. Schneier, Cryptographie Appliquée, 2 ed: Vuibert Informatique, 2001. ISBN 2-7117-8676-5.

[Sha00] A. Shamir, "Protecting Smart Card from Passive Power Analysis with Detached Power Supplies," presented at Cryptographic Hardware and Embedded Systems (CHES2002), pp. 71-77, LNCS 1965, San Francisco, USA, 2000.

[Sin00] Montek Singh and Steven M. Nowick. Fine-grain pipelined asynchronous adders for high-speed DSP applications. In Proceedings of the IEEE Computer Society Workshop on VLSI, pages 111-118. IEEE Computer Society Press, Apri12000.

[Sko02] Sergei Skorobogatov, Ross Anderson, Optical Fault Induction Attacks, Cryptographic Hardware and Embedded Systems Workshop (CHES-2002), LNCS, Vol. 2523, Springer-Verlag, 2002, pp. 2–12

[Som97] D. Somasekhar, K. Roy, “LVDCSL: Low Voltage Differential Current Switch Logic, A Robust Low Power DCSL family”, Proceedings of the International Symposium Low Power Electronics and Design, 1997, 18-20 Aug 1997, pp. 18- 23

[Sti01] D. Stinson: " Cryptographie Théorie et pratique", ISBN 2-7117-8675-7, vuibert, 2001

[Sti96] D. Stinson, Cryptographie: Théorie et Pratique, International Thomson Publishing ed. Paris: International Thomson Publishing France, 1996. 2-84180-013. pp. 106

[Sum97] Rita C. Summers, "Secure Computing Threats and Safeguards", McGraw-Hill, 1997.

[Sut89] I. Sutherland, "Micropipelines," Comm. ACM, Vol. 32 No. 6, ACM Press, New York, June 1989.

[Sze02] S.M. Sze, Semiconductor Devices: Physics and Technology, John Wiley and Sons, 2002

[Tir02] K. Tiri and I. Verbauwhede, "Adynamic and Differential CMOS Logic with Signal Indepent Power Consumption to Withstand Differential Power Analisys on Smart Cards," presented at 28th European Solid-State Circuits Conference (ESSCIR200

[Tir03] K. Tiri and I. Verbauwhede. Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In C. Walter, Ç.K. Koç and C. Paar, editors, Proceedings of 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES), number 2779 of LNCS, page 125-136, 2003, Springer-Verlag.

[Tir04] Kris Tiri, and Ingrid Verbauwhede, "Place and Route for Secure Standard Cell Design", 6th International Conference on Smart Card Research and Advanced Applications (CARDIS 2004), pp. 143-158, August 2004.

[Tir05] Kris Tiri, and Ingrid Verbauwhede, "A VLSI Design Flow for Secure Side-Channel Attack Resistant ICs", Design, Automation and Test in Europe Conference (DATE 2005), pp. 58-63, March 2005.

[Yen02] S.-M. Yen, S. J. Moon and J. C. Ha. Hardware fault attacks on RSA with CRT revisited. In P. J. Lee and C. H. Lim, editors, Proceedings of the 5th International Conference on Information Security and Cryptology - (ICISC), number 2587 of LNCS, pages 374-388

[Yen03] S.-M. Yen, S. J. Moon and J. C. Ha. Permanent fault attack on the parameters of RSA with CRT. In R. Safavi-Naini and J. Seberry, editors, Proceedings of the 8th Australasian Conference on Information Security and Privacy (ACISP), number 2727 of LNCS

Bibliographie personnelle

Bibliographie personnelle

1. Revues internationales avec actes et comité de lecture

[Raz05] A. RAZAFINDRAIBE, M. ROBERT, P MAURINE, " Compact and secured primitives for the design of asynchronous circuits ", JOLPE Journal of Low Power Electronics , 2005, Vol. 1, n° 1, pp. 20-26.

2. Conférences internationales avec actes et comité de lecture

[Raz04] A. RAZAFINDRAIBE, M. ROBERT, P MAURINE, F. BOUESSE, B. FOLCO, M. RENAUDIN, "Secured Structures for Secured Asynchronous QDI Circuits", DCIS'04, XIX Design of Circuits and Integrated Systems Conference, Bordeaux, France, 24-26, Novembre 2004

[Raz05a] A. RAZAFINDRAIBE, M. ROBERT, P MAURINE, "Design of Compact Dual Rail Asynchronous Primitives", DCIS'05, XX Design of Circuits and Integrated Systems Conference, Lisbonne, Portugal, 23-25, Novembre 2005.

[Raz06] A. RAZAFINDRAIBE, M. ROBERT, P MAURINE, "Evaluation of the robustness of dual-rail logic against DPA", ICICDT'06, International Conference on IC Design and Technology, Padova, Italie, 24-26 Mai 2006.

[Raz06a] A. RAZAFINDRAIBE, M. ROBERT, M. RENAUDIN, P MAURINE, "Security evaluation of dual rail logic against DPA attacks", VLSI-SOC 2006, International Conference on Very Large Scale Integration Systems, Nice, France, October 16-18, 2006.

3. Workshop internationaux avec Comité de Lecture

[Raz05b] A. RAZAFINDRAIBE, M. ROBERT, M. RENAUDIN, P MAURINE, "A Method to Design Compact DUAL-RAIL Asynchronous Primitives", PATMOS'05 : 15th International Worshop on Power and Timing Modeling Optimization and Simulation, Louvain, Belgium, September 2005.

[Raz06b] A. RAZAFINDRAIBE, M. ROBERT, M. RENAUDIN, P MAURINE, "Formal evaluation of the robustness of dual rail logic against DPA attacks PATMOS'06, 16th International Worshop on Power and Timing Modeling Optimization and Simulation, Montpellier, France, September 2006.

4. Colloques Nationaux avec Comité de lecture

[Raz05c] A. RAZAFINDRAIBE, M. ROBERT, P MAURINE, "La technologie asynchrone QDI pour la sécurité des cryptosystèmes", JNRDM'2005, 8ème édition des Journées Nationales du Réseau Doctoral de Microélectronique, 10-12 Mai 2005, Paris, France.

[Raz05d] A. RAZAFINDRAIBE, M. ROBERT, P MAURINE, "Méthode de conception de primitives asynchrones double rail", FTFC'2005, 5ème journées Faible Tension Faible Consommation, 18-20 Mai 2005, Paris, France.

[Raz05e] A. RAZAFINDRAIBE, M. ROBERT, M. RENAUDIN, P MAURINE, "Asynchronous Dual rail Cells to Secure Cryptosystem Against Side Channel Attacks", SAME'2005, Sophia Antipolis MicroElectronics Forum, 8th Edition, October 5-6 2005, Sophia Antipolis, France

RESUME

Dans le domaine de la conception de circuits sécurisés (cartes à puce) et plus particulièrement des circuits robustes aux attaques différentielles en puissance (DPA), la logique double rail apparaît comme une alternative intéressante à la logique statique CMOS. En effet, le codage associé à ce style de logique offre la possibilité d'équilibrer la consommation rendant ainsi impossible les attaques DPA. Partant de ce constat, dans cette thèse, nous nous sommes focalisés sur l'analyse des atouts et faiblesses de la logique double rail et surtout à son amélioration. Dans un premier temps, nous avons montré qu'un circuit double rail est nettement plus résistant aux attaques DPA que son homologue simple rail. Dans un deuxième temps, après une étude approfondie de l'impact de la synthèse physique sur la robustesse de la logique double rail, nous avons abouti à la conclusion qu'en présence de déséquilibres des capacités de charge, des temps de transition et des temps d'arrivée, les circuits double rail peuvent perdre leur avantage et devenir vulnérables aux attaques DPA. Cette étude a permis de définir quelques métriques de robustesse aux attaques DPA à partir desquelles nous avons clairement établi qu'une cellule double rail n'est robuste que si les signaux la contrôlant arrivent tous dans un intervalle de temps particulièrement réduit. Afin d'éliminer cette faiblesse résiduelle de la logique double rail, nous avons finalement proposé une amélioration simple mais efficace de la logique double rail. La logique résultante a été appelée STTL (Secured Triple Track Logic). La mise en œuvre de cette logique a permis de montrer que la logique STTL permet d'obtenir des circuits dont les temps de calcul et la consommation sont indépendants des données.

MOTS-CLES: Cryptographie, Carte à puce, Attaque par canaux cachés, Attaques DPA, Logique double rail, Circuits asynchrones, Méthode de conception, Bibliothèque de cellules CMOS double rail.

DISCIPLINE : Microélectronique

TITLE

ANALYSIS AND IMPROVEMENT OF DUAL-RAIL LOGIC FOR DESIGNING SECURE CIRCUITS

ABSTRACT

In the area of secure circuits design and more particularly of (Differential Power Analysis) DPA-resistant ones, dual-rail logic looks like an interesting alternative to static CMOS logic. Indeed, the encoding style associated with this logic offers the opportunity to make power consumption balanced thus making DPA attacks impossible. In this context, we focused ourselves on the analysis of the assets and weaknesses of dual-rail logic and especially to its improvement. Firstly we showed that a dual-rail circuit is distinctly more resistant to DPA attacks than its counterpart single-rail. Secondly, after a thorough study of the physical synthesis impact on the robustness of dual-rail circuits, we arrived at the conclusion that in the presence of loads, input transition times and arrival times imbalances, dual-rail circuits can lose their advantage and become vulnerable to DPA attacks. This study made it also possible to define some metric robustness with respect to DPA attacks, from which we clearly established that a dual-rail cell is DPA-resistant if and only if every signals controlling it arrive in a particularly reduced interval time. In order to eliminate this residual weakness from dual-rail logic, we finally proposed a simple but effective improvement. The resulting logic was called STTL (Secured Triple Track Logic). At last, the implementation of this logic made it possible to show that STTL logic enables us to obtain circuits with running times and power consumption which are data independent.

KEYWORDS: Cryptography, Smartcard, Side channel attacks, DPA attacks, Dual-rail logic, Asynchronous circuits, Design method, Library of CMOS dual-rail cells.

Université de Montpellier II: Sciences et Techniques du Languedoc
LIRMM : Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier
161 Rue Ada - 34392 Montpellier Cedex 5