Sécurité cryptographique par la conception spécifique de circuits intégrés.

Fabien Germain

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

ÉCOLE POLYTECHNIQUE

SGDN/DCSSI — **École Polytechnique**

Laboratoire des Technologies de l'Information — **Département de Physique**

Sécurité cryptographique par la conception spécifique de circuits intégrés

THÈSE

présentée et soutenue publiquement le 23 juin 2006

pour l'obtention du

Doctorat de l'École Polytechnique

par

Fabien Germain

Composition du jury

Président :	Bernard Drévillon	École Polytechnique
Rapporteurs :	Jean-Jacques Quisquater	Université catholique de Louvain
	Jean-Pierre Schœllkopf	STMicroelectronics, Crolles
Examinateurs :	Alain Greiner	École Polytechnique et Université Paris VI
	Antoine Joux	DGA et Université de Versailles
Invité :	Marc Renaudin	TIMA, Grenoble
Directeur de thèse :	François Anceau	CNAM et Université Paris VI, (Ex-École Polytechnique)

Travaux effectués au Laboratoire des Technologies de l'Information de la DCSSI
Direction Centrale de la Sécurité des Systèmes d'Information
51, boulevard de La Tour-Maubourg 75700 PARIS 07 SP

Mis en page avec la classe thloria.

Remerciements

Je souhaite tout d'abord remercier Florent Chabaud et Antoine Joux qui m'ont donné l'opportunité de faire une thèse dans le cadre exceptionnel de la DCSSI au sein du laboratoire des technologies de l'information. La pluridisciplinarité de ce laboratoire et les compétences si diverses de chacun de ses membres en font sa richesse permettant d'aborder la sécurité des systèmes d'information dans tous ses aspects. La mise en place des activités mircroélectroniques et des projets de recherche dont j'ai la charge n'en sont qu'une facette.

Cette thèse me permet de remercier mes collègues passés et présents : Florent, détenteur du record de vitesse mx5 sur le circuit Fort d'Issy-École Polytechnique avec lunettes de soleil par temps nuageux, Philippe, pour sa qualité d'écoute et sa perspicacité cryptographique commençant toujours par "oui, mais ...", Louis, toujours formel dans ses propos y compris ses boutades, Éric, un formel soutien de Louis et grand voyageur, Véronique, experte en photo numérique, Frédéric, diplomate protocolaire, Vincent, geek probable descendant de Darth Vador et gourmet de pizzas, Olivier, M>aître geek expert du niveau standard, Albert, Francis et pour finir, notre star internationale, Loïc, grand découvreur de l'attaque par sèche-cheveux des microprocesseurs. L'ambiance dans le laboratoire des technologies de l'information y est excellente, merci à tous. Je voudrais aussi remercier mes autres collègues de la DCSSI qui, par les échanges que j'ai eus avec eux, ont contribué à enrichir mes réflexions.

L'activité de recherche du laboratoire des technologies de l'information est avant tout possible grâce au soutien de la direction de la DCSSI. Je remercie donc les précédents membres de la direction, l'ingénieur général des télécommunications H. Serres et A. Esterle qui fut l'un de ses directeurs adjoints. Je tiens aussi à remercier notre nouveau directeur central, P. Pailloux, ainsi que le général J. Novacq, directeur adjoint. Travailler à la DCSSI et y faire une thèse est une chance exceptionnelle. Cet environnement de travail permet d'appréhender la sécurité des systèmes d'information dans tous ses aspects pour les produits gouvernementaux et civils.

Je tiens à remercier personnellement chacun des membres du jury pour leur contribution. Je remercie Bernard Drévillon pour avoir accepté d'être président du jury. Je remercie Jean-Jacques Quisquater et Jean-Pierre Schœllkopf qui ont accepté d'être rapporteurs, ce qui leur a demandé un important investissement que démontrent leur rapports très pertinents, très complets et très détaillés. Je remercie Alain Greiner pour les échanges scientifiques que nous avons eus et pour son important investissement. Je remercie Antoine Joux pour tout le soutien qu'il m'a apporté à la DCSSI. Je remercie Marc Renaudin avec qui je travaille depuis de nombreuses années et dont j'apprécie les qualités tant professionnelles qu'humaines. Et pour finir, je remercie François Anceau qui a bien voulu devenir mon directeur de thèse et qui a su me guider tout au long de ces trois années pour réussir. François est une mine d'information en microélectronique, une encyclopédie qui va de la création du silicium à la dernière technologie extraterrestre.

Je remercie particulièrement Jean-Michel Henriet pour avoir avoir relu ma thèse et Danielle Henriet qui m'a apporté un soutien logistique dans l'organisation de la soutenance.

Mes dernières pensées sont pour Laurence pour son amour et son soutien et pour ma fille Clotilde.

à mon épouse Laurence pour son amour et
le soutien qu'elle m'a apporté ;

à ma fille Clotilde qui m'a donné tant de
joie pendant ma dernière année de thèse avec ses si nombreux rires et sourires.

Table des matières

Table des figures

Introduction générale

Les circuits intégrés sont très souvent employés pour augmenter le niveau de sécurité des systèmes d'information. De façon générale, ces circuits intègrent un ou plusieurs cryptoprocesseurs et du logiciel spécifique afin de pouvoir fournir des garanties suffisantes pour une application de sécurité donnée. Les conditions d'emploi définissent des attaques que le système doit contrer. Des contre-mesures matérielles et logicielles sont aussi intégrées aux différentes fonctions afin de protéger efficacement le système d'information. Ces circuits intégrés manipulent et contiennent donc des secrets. Dans le cas de protocoles de signature numérique, la confidentialité et l'intégrité de la clé de signature doivent être garanties. Dans le cas d'une communication confidentielle, le protocole permet l'échange de clés de session de chiffrement et de déchiffrement communes. Non seulement les clés servant à l'échange doivent rester confidentielles et intègres mais les clés de session doivent aussi le rester pendant leur utilisation. Bon nombre de systèmes d'information demandent à leurs utilisateurs un mot de passe. Des primitives cryptographiques permettent d'en extraire une clé secrète qui permet, dans certains cas, de déchiffrer une clé système utilisateur. Les circuits intégrés, qui détiennent et manipulent des secrets, deviennent de fait un maillon clé de la chaîne de la sécurité.

Dans un tel contexte, un attaquant a pour objectif de déterminer les secrets protégés par le circuit intégré comme les clés secrètes pour les algorithmes symétriques, les clés privées pour les algorithmes asymétriques, les algorithmes gouvernementaux confidentiels, les mots de passe, etc.

Les attaques sur les composants peuvent être classées en deux grandes familles. Une première est dite intrusive parce que la mise en œuvre technique de l'attaque nécessite de modifier physiquement le circuit intégré. Dans le cas d'un algorithme cryptographique, la modification d'intégrité de fonctions logiques permet de provoquer des fautes et par la différence de résultat de déterminer la ou les clés utilisées. L'inhibition de fonctions de sécurité permet de tenter une attaque qui initialement était impossible à réaliser. De plus, les données secrètes sont manipulées en clair dans le circuit intégré. La lecture des données circulant en interne sur certaines équipotentielles permet aussi de retrouver des secrets.

Une seconde famille d'attaques est dite non intrusive parce que leur mise en œuvre technique ne nécessite pas de modification physique du circuit intégré. Parmi les attaques non intrusives se trouvent les exemples suivants. Certains processus cryptographiques peuvent avoir une exécution temporelle qui n'est pas constante. Si cette variation temporelle est une fonction du secret alors l'analyse fine des temps de calcul peut fournir suffisamment d'information pour déterminer tout ou en partie le secret [23]. Chaque opération exécutée par un circuit intégré provoque une consommation du composant. Cette consommation peut être caractérisée par la variation temporelle du courant et est propre à l'opération en cours. Elle est assimilable à une signature électrique de l'opération. Dans le cas d'un algorithme cryptographique asymétrique, l'opération d'exponentia-

tion est une suite de carrés et de multiplications optionnelles. Par la simple lecture du courant de consommation sur un oscilloscope, l'attaquant peut déterminer si une multiplication est exécutée et donc lire la clé privée [2]. En supposant que des contre-mesures soient employées pour lutter efficacement contre ces attaques, la signature électrique d'une opération est aussi modulée par la valeur des opérandes utilisées. Un algorithme cryptographique utilise toujours un message et une clé comme opérandes. Dans certains cryptosystèmes, il est possible qu'une même clé secrète soit utilisée pour un grand nombre de messages. A chaque message correspond une consommation différente bien que la clé soit constante. Cette analyse différentielle de la puissance consommée, appelée aussi DPA [24, 25], peut être exploitée pour retrouver la clé secrète. Cette attaque est opérationnelle sur des composants synchrones ou asynchrones ([10, 36], figure 1).

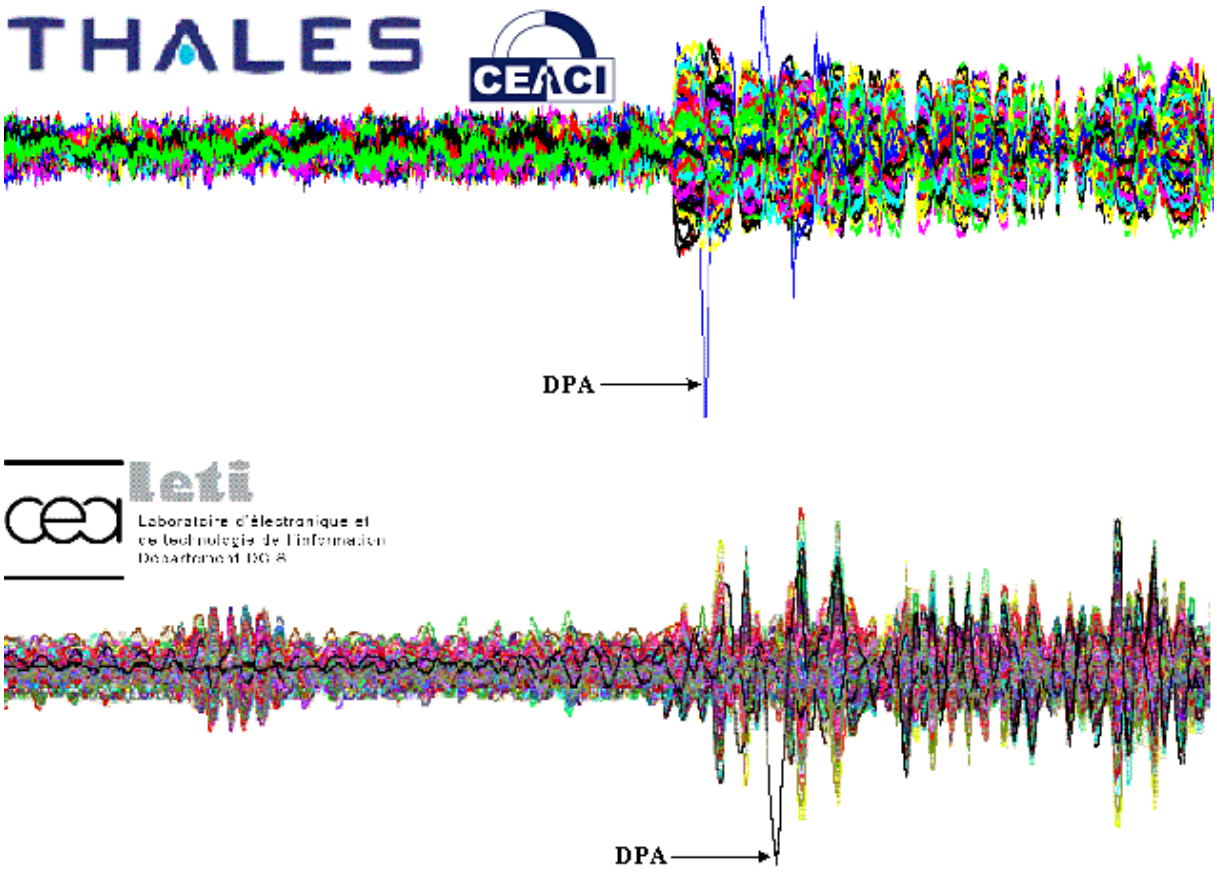


FIG. 1 – Attaque DPA réalisée avec succès par les centres d'évaluation de la sécurité des systèmes d'information du CEA LETI et de THALES Security Systems lors d'un chiffrement réalisé par un composant cryptoprocesseur AES en technologie 120nm - Chaque courbe représente une hypothèse de l'octet de sous-clé - Les pics DPA indiquent avec certitude la valeur de l'octet de sous-clé utilisée lors du chiffrement

De nombreuses contre-mesures anti-DPA ont été proposées sous forme logicielle [1, 3, 16, 47] ou sous forme matérielle [5, 14, 18, 20, 22, 26, 41, 42, 43, 44, 45, 46, 49]. Cependant et quelle que soit la solution proposée dans cette littérature, elles échouent toutes à fournir une contre-mesure réellement efficace et des attaques nouvelles ou améliorées apparaissent [13, 17, 28, 29, 40]. En règle générale, elles sont conçues dans l'objectif de décourager l'attaquant de mener l'attaque

DPA en augmentant sa complexité. Quand il ne s'agit pas d'augmenter la difficulté de l'attaque, les solutions essaient de contrer la DPA de façon globale. Or, la technologie CMOS consomme sur transition. Á chaque période d'horloge, une porte logique à deux entrées est soumise à 1 évènement parmi 16 représentant toutes les transitions sur chaque signal d'entrée. Le problème revient donc à contrer de façon globale un nombre de transitions électriques supérieur à l'espace des clés quel que soit l'algorithme cryptographique. La difficulté du problème va croissante avec la prise en compte d'autres paramètres tels que l'indépendance de la contre-mesure vis à vis de l'architecture matérielle.

Cette thèse s'inscrit donc dans une logique très précise pour contrer la DPA. Tout d'abord, un simple constat mondial montre que la grande majorité des circuits intégrés utilise la technologie CMOS. Cette thèse s'appuie donc sur la conception CMOS afin d'apporter une réponse à la faisabilité d'une contre-mesure anti-DPA basée sur cette technologie. Pour être véritablement efficace, une contre-mesure doit être indépendante des algorithmes, des technologies et des architectures matérielles des circuits intégrés. Le fait qu'une contre-mesure globale ait un espace des transitions électriques plus grand que l'espace des clés d'un algorithme cryptographique, définit la logique de cette thèse.

Il est possible de considérer comme élément de base pour la conception d'un circuit intégré une porte logique à deux entrées et à une sortie. Cette thèse va donc se focaliser sur la conception de telles portes logiques insensibles à la DPA. L'indépendance vis à vis des algorithmes, des technologies et des architectures sera ainsi acquise. Avec de telles portes, il devient théoriquement possible de concevoir un circuit intégré résistant intrinsèquement à la DPA. Cependant, les outils de conception peuvent introduire des biais pendant les phases de synthèse et de placement-routage qui seront exploitables lors de la mise en œuvre de l'attaque. La conception de portes logiques insensibles à la DPA est la première étape de la logique de conception de la contre-mesure et certainement la plus difficile actuellement. La solution proposée fournira un schéma électrique à base de transistors de portes insensibles à la DPA. Elle ne couvrira pas le dessin des cellules qui est dépendant des technologies ni la nécessaire adaptation des outils de conception. Ces deux étapes nécessaires à la mise en œuvre de cette thèse seront couvertes par un travail futur.

Afin d'atteindre cet objectif, les première et deuxième parties de la thèse explicitent respectivement, la dispersion instantanée des caractéristiques électriques des portes logiques CMOS sur un plan théorique général et la mise en œuvre théorique de la DPA. La troisième partie de la thèse peut alors expliciter les sources microélectroniques de la DPA quels que soient les architectures et les algorithmes sur les circuits intégrés. Il s'agit de se donner la base nécessaire avant toute proposition de contre-mesure. La quatrième et dernière partie développe une proposition de solution contrant les sources microélectroniques de la DPA.

Cette thèse s'adresse à deux catégories de personnes toutes deux concernées par la DPA, Les cryptologues et les microélectroniciens, mais qui, la plus part du temps, travaillent trop rarement ensemble. Les cryptologues trouveront en annexe des rappels de microélectronique. Ces annexes sont destinées à leur permettre de comprendre les différents phénomènes qui font que la DPA fonctionne. La compréhension des sources microélectroniques de la DPA appliquée à plusieurs types d'architecture implantant un algorithme cryptographique devrait leur permettre d'améliorer les attaques, par exemple en amplifiant les différences. Mais pour cela il devient nécessaire de prendre en compte les aspects physiques en plus des séparation logiques de la DPA.
Ensuite les microélectroniciens trouveront en annexe une introduction à la cryptographie ainsi

que la description des algorithmes illustrant la DPA dans cette thèse. Ces annexes ont pour but de leur permettre de comprendre la mise en œuvre de l'attaque DPA sur un circuit intégré quelconque. La connaissance fine des sources microélectroniques de la DPA leur montrera que la DPA fonctionne toujours quel que soit le circuit intégré et l'absolue nécessité de prendre en compte les attaques matérielles dès les spécifications des composants.

Première partie

Différence instantanée de la variation de la consommation des portes logiques

Chapitre 1

Dispersion instantanée des caractéristiques électriques des réseaux de transistors

Sommaire

1.1 Introduction

Chacune des portes logiques (annexe A) des librairies technologiques fournies par les fondeurs est décrite par des caractéristiques dynamiques de temps de propagation entre les entrées et les sorties et de charges pouvant être induites en entrée (fanin) ou supportées en sortie (fanout). Une table de vérité définit la fonction logique de chacune de ces portes. Lors de la synthèse d'un composant - étape transformant une description VHDL ou Verilog en portes logiques - toutes ces caractéristiques sont utilisées afin de garantir que tous les signaux électriques soient stables avant le front d'horloge suivant. Cette description est suffisante pour un flot de conception d'une fonction numérique mais en ce qui concerne les attaques matérielles - dont fait partie la DPA - elle n'apporte aucune information expliquant son fonctionnement détaillé. Ce chapitre introduit toutes les notions microélectroniques nécessaires à l'explication théorique de l'attaque DPA et de ses sources. Le comportement électrique de tout circuit électronique dépend des grandeurs appliquées à ses entrées, de ses connexions en sortie et de ses états internes. Il se trouve que le comportement analogique d'une fonction logique peut varier y compris lorsque la même opération donnant le même résultat logique est réexécutée. Afin d'expliquer ce phénomène - qui sera exploité par la DPA - il devient nécessaire de recourir à un modèle analogique du comportement du transistor prenant en compte les phénomènes capacitifs et résistifs. Un transistor peut se trouver dans trois états principaux qui sont déterminés par la tension grille-source V_{GS}. Selon la tension appliquée le transistor peut être dans l'état d'accumulation, de déplétion ou de forte inversion. Lorsque le transistor est dans l'état de forte inversion, le transistor est en mode de fonctionnement dit saturé. En dehors de cet état, il est dit en mode de fonctionnement non saturé ou ohmique. La grille du transistor se comporte comme une capacité. Le signal électrique commandant la grille peut être considéré comme l'entrée d'un circuit RC où C est la capacité grille-source C_{GS} du transistor. La consommation provoquée par le signal qui commande la grille du transistor dépend donc fortement de cette capacité.

1.2 Capacité de grille d'un transistor en mode saturé

La charge présente en l'abscisse y dans le canal du transistor est donnée par l'équation A.15 en annexe A. Le courant traversant le canal sur la longueur dy est donné par l'équation A.19 en annexe A. Il est à noter que la capacité d'oxyde C_{ox} varie selon l'état du transistor. C_{ox} représente donc dans chaque équation qui suit, la valeur associée à l'état. Par exemple, la capacité d'oxyde diminue lorsqu'une zone de déplétion apparaît sous le canal entre le drain et la source parce que celle-ci augmente l'épaisseur d'isolant. La charge présente dans le canal d'un transistor de longueur effective L_{eff} et de largeur W est égale à :

$$Q = \int_0^{L_{eff}} W.Q(y).dy = C_{ox}.W.\int_0^{L_{eff}} (V_{GS} - V(y) - V_{THN}).dy \tag{1.1}$$

D'après l'équation A.19,

$$dy = \frac{\mu_n.C_{ox}.W}{I_{DS}}.(V_{GS} - V(y) - V_{THN}).dV(y) \tag{1.2}$$

En substituant dy dans l'équation 1.1 par l'équation 1.2, l'équation 1.1 de la charge Q devient :

$$Q = \int_0^{V_{GS}-V_{THN}} C_{ox}.W.(V_{GS} - V(y) - V_{THN}).\frac{\mu_n.C_{ox}.W}{I_{DS}}.(V_{GS} - V(y) - V_{THN}).dV(y)$$

$$
\begin{aligned}
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \int_0^{V_{GS}-V_{THN}} (V_{GS} - V(y) - V_{THN})^2.dV(y) \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \int_0^{V_{GS}-V_{THN}} (V_{GS}^2 + V(y)^2 + V_{THN}^2 - 2V_{GS}V(y) - 2V_{GS}V_{THN} + \\
&\quad 2V(y)V_{THN}).dV(y) \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \Bigg((V_{GS}^2 + V_{THN}^2 - 2V_{GS}V_{THN})[V(y)]_0^{V_{GS}-V_{THN}} - \\
&\quad 2(V_{GS} - V_{THN}) \left[\frac{V(y)^2}{2}\right]_0^{V_{GS}-V_{THN}} + \left[\frac{V(y)^3}{3}\right]_0^{V_{GS}-V_{THN}} \Bigg) \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} (V_{GS} - V_{THN})^2 (V_{GS} - V_{THN}) - 2(V_{GS} - V_{THN}) \frac{(V_{GS} - V_{THN})^2}{2} + \\
&\quad \frac{(V_{GS} - V_{THN})^3}{3} \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \frac{(V_{GS} - V_{THN})^3}{3}
\end{aligned}
\tag{1.3}
$$

Le courant drain-source I_{DS} en mode saturé est donné par l'équation A.31. En substituant I_{DS} dans l'équation 1.3, l'expression de la charge Q devient :

$$
\begin{aligned}
Q &= \frac{\mu_n.(C_{ox}.W)^2.(V_{GS} - V_{THN})^3}{\frac{3.\mu_n.C_{ox}.W.(V_{GS} - V_{THN})^2}{2.(L - 2.L_{diff} - X_{dl})}. \left(1 + \left(\frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{DS}}\right).(V_{DS} - V_{DS,sat})\right)} \\
&= \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}).\frac{V_{GS} - V_{THN}}{1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{DS}}.(V_{DS} - V_{GS} + V_{THN})}
\end{aligned}
\tag{1.4}
$$

La capacité grille-source C_{GS} en mode saturé est déterminée par la variation de la charge Q en fonction de la tension grille-source V_{GS} (Équation 1.5).

$$
\begin{aligned}
C_{GS} &= \frac{\partial Q}{\partial V_{GS}} \\
&= \frac{\frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl})}{\left(1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{DS}}.(V_{DS} - V_{GS} + V_{THN})\right)^2}.\Bigg(1 + \\
&\quad \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{DS}}.(V_{DS} - V_{GS} + V_{THN}) - \\
&\quad (V_{GS} - V_{THN})\frac{-1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{DS}}\Bigg) \\
&= \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}).\frac{1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{DS}}.V_{DS}}{1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{DS}}.(V_{DS} - V_{GS} + V_{THN})}
\end{aligned}
\tag{1.5}
$$

Lorsque la différence de potentiel drain-source V_{DS} s'approche de 0V et devient négligeable par rapport à la tension grille-source V_{GS}, la variation du canal $\frac{dX_{dl}}{dV_{DS}}$ devient aussi négligeable et proche de 0. Nous retrouvons l'expression simplifiée $C_{GS} = \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl})$.

1.3 Capacité de grille d'un transistor en mode non saturé

La charge présente à l'abscisse y dans le canal du transistor et le courant traversant ce canal sur la longueur dy sont respectivement donnés par les équations A.15 et A.19. En mode non saturé, le canal du transistor n'est pincé en aucune région et sa longueur effective est donc égale à la longueur totale dessinée L. La charge présente dans le canal de longueur effective L est donnée par l'équation 1.6.

$$Q = \int_0^L W.Q(y).dy = C_{ox}.W.\int_0^L (V_{GS} - V(y) - V_{THN}).dy \tag{1.6}$$

En substituant à nouveau dy dans l'équation 1.6 par l'équation 1.2, l'équation 1.6 de la charge Q devient :

$$\begin{aligned}
Q &= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \int_0^{V_{DS}} (V_{GS} - V(y) - V_{THN})^2.dV(y) \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \int_0^{V_{DS}} (V_{GS}^2 + V(y)^2 + V_{THN}^2 - 2V_{GS}V(y) - 2V_{GS}V_{THN} + \\
&\quad 2V(y)V_{THN}).dV(y) \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \left((V_{GS}^2 + V_{THN}^2 - 2V_{GS}V_{THN})[V(y)]_0^{V_{DS}} - 2(V_{GS} - V_{THN}) \left[\frac{V(y)^2}{2}\right]_0^{V_{DS}} + \right. \\
&\quad \left. \left[\frac{V(y)^3}{3}\right]_0^{V_{DS}} \right) \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \left(V_{DS}(V_{GS}^2 + V_{THN}^2 - 2V_{GS}.V_{THN}) - V_{DS}^2(V_{GS} - V_{THN}) + \frac{V_{DS}^3}{3} \right) \\
&= \frac{\mu_n.(C_{ox}.W)^2}{I_{DS}} \left(V_{DS}(V_{GS} - V_{THN})^2 - V_{DS}^2(V_{GS} - V_{THN}) + \frac{V_{DS}^3}{3} \right)
\end{aligned} \tag{1.7}$$

Le courant drain-source I_{DS} en mode non saturé est donné par l'équation A.21. En substituant I_{DS} dans l'équation 1.7, l'expression de la charge Q devient :

$$\begin{aligned}
Q &= \frac{\mu_n.(C_{ox}.W)^2 \left(V_{DS}(V_{GS} - V_{THN})^2 - V_{DS}^2(V_{GS} - V_{THN}) + \frac{V_{DS}^3}{3} \right)}{\mu_n.C_{ox}.\frac{W}{L} \left((V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2} \right)} \\
&= W.L.C_{ox}.\frac{V_{DS}(V_{GS} - V_{THN})^2 - V_{DS}^2(V_{GS} - V_{THN}) + \frac{V_{DS}^3}{3}}{(V_{GS} - V_{THN})V_{DS} - \frac{V_{DS}^2}{2}}
\end{aligned} \tag{1.8}$$

La capacité grille-source C_{GS} en mode non saturé est déterminée par la variation de la charge Q en fonction de la tension grille-source V_{GS} (Équation 1.9).

$$\begin{aligned}
C_{GS} &= \frac{\partial Q}{\partial V_{GS}} \\
&= \frac{W.L.C_{ox}.(2.V_{DS}(V_{GS}-V_{THN})-V_{DS}^2)\left((V_{GS}-V_{THN})V_{DS}-\frac{V_{DS}^2}{2}\right)}{\left((V_{GS}-V_{THN})V_{DS}-\frac{V_{DS}^2}{2}\right)^2} - \\
&\quad \frac{W.L.C_{ox}.V_{DS}\left(V_{DS}(V_{GS}-V_{THN})^2-V_{DS}^2(V_{GS}-V_{THN})+\frac{V_{DS}^3}{3}\right)}{\left((V_{GS}-V_{THN})V_{DS}-\frac{V_{DS}^2}{2}\right)^2} \\
&= \frac{W.L.C_{ox}.2.\left((V_{GS}-V_{THN})V_{DS}-\frac{V_{DS}^2}{2}\right)^2}{\left((V_{GS}-V_{THN})V_{DS}-\frac{V_{DS}^2}{2}\right)^2} - \\
&\quad \frac{W.L.C_{ox}.\left(V_{DS}^2(V_{GS}-V_{THN})^2-V_{DS}^3(V_{GS}-V_{THN})+\frac{V_{DS}^4}{3}\right)}{V_{DS}^2(V_{GS}-V_{THN})^2-(V_{GS}-V_{THN})V_{DS}^3+\frac{V_{DS}^4}{2}} \\
&= W.L.C_{ox}\left(2-\frac{1-\frac{V_{DS}}{V_{GS}-V_{THN}}+\frac{V_{DS}^2}{3(V_{GS}-V_{THN})^2}}{1-\frac{V_{DS}}{V_{GS}-V_{THN}}+\frac{V_{DS}^2}{2(V_{GS}-V_{THN})^2}}\right)
\end{aligned} \tag{1.9}$$

Dans le cas où la tension drain-source V_{DS} est considérée comme négligeable par rapport à la tension grille-source V_{GS}, nous retrouvons l'expression simplifiée $C_{GS} = W.L.C_{ox}$ qui correspond à la capacité totale de grille en mode ouvert.

1.4 Capacité de grille d'un transistor en mode ouvert

En mode ouvert la tension grille-source V_{GS} est nulle. La capacité grille-source C_{GS} est égale à la capacité totale induite par la surface de grille $W.L$ (Équation 1.10).

$$C_{GS} = W.L.C_{ox} \tag{1.10}$$

1.5 Résistance drain-source d'un transistor en mode saturé

L'expression du courant drain-source I_{DS} en mode saturé est donnée par l'équation A.31. La résistance drain-source R_{DS} est donnée par l'inverse de la variation du courant drain-source I_{DS}

en fonction de la tension drain-source V_{DS} (Équation 1.11).

$$\begin{aligned} R_{DS} &= \frac{1}{\frac{\partial I_{DS}}{\partial V_{DS}}} \\ &= \frac{1}{\frac{\partial}{\partial V_{DS}}\left(\frac{\mu_n.C_{ox}.W.(V_{GS}-V_{THN})^2}{2.(L-2.L_{diff}-X_{dl})}.\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{DS}}.(V_{DS}-V_{DS,sat})\right)\right)} \\ &= \frac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{DS}}.(V_{GS}-V_{THN})^2\right)} \end{aligned} \tag{1.11}$$

En mode saturé, le transistor se comporte comme un générateur de courant I_{DS} en série avec une résistance drain-source R_{DS} qui varie en fonction de la tension grille-source V_{GS} et de la tension drain-source V_{DS}.

1.6 Résistance drain-source d'un transistor en mode non saturé

L'expression du courant drain-source I_{DS} en mode non saturé est donnée par l'équation A.21. La résistance drain-source R_{DS} se calcule de la façon suivante (Équation 1.12) :

$$\begin{aligned} R_{DS} &= \frac{1}{\frac{\partial I_{DS}}{\partial V_{DS}}} \\ &= \frac{1}{\frac{\partial}{\partial V_{DS}}\left(\mu_n.C_{ox}.\frac{W}{L_{eff}}.\left((V_{GS}-V_{THN}).V_{DS}-\frac{{V_{DS}}^2}{2}\right)\right)} \\ &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{GS}-V_{THN})-V_{DS})} \end{aligned} \tag{1.12}$$

En mode non saturé, le transistor se comporte aussi comme un générateur de courant I_{DS} en série avec une résistance drain-source R_{DS} qui est non seulement une fonction de la tension grille-source V_{GS} mais aussi fonction de la tension drain-source V_{DS}.

1.7 Résistance drain-source d'un transistor en mode ouvert

Le courant drain-source I_{DS} en mode ouvert est considéré nul. La résistance drain-source R_{DS} est alors égale à l'infini (Équation 1.13).

$$R_{DS} = \infty \tag{1.13}$$

1.8 Évènements sur une porte à deux entrées a et b

Un évènement est défini par un couple de transitions dont l'une est sur le signal a et l'autre sur le signal b. Une transition peut être vue comme un changement de valeur d'un signal sur un front d'horloge. Elle correspond soit à un front montant du signal, soit à un front descendant du signal, soit à un signal qui reste constant. Pour une porte à deux entrées, 16 évènements au total sont possibles (Figure 1.1).

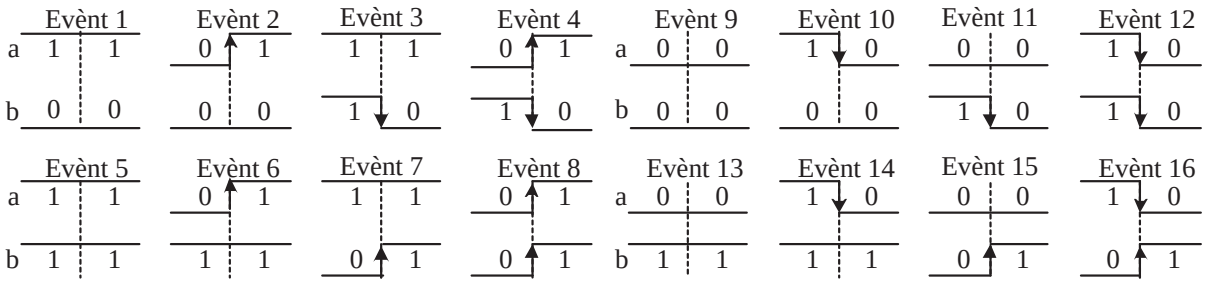


FIG. 1.1 – Définition des évènements pour une porte à deux entrées

Tous les évènements définis précédemment partent d'un état électrique initial et arrivent dans un état électrique final. Tous les états initiaux et finaux des évènements appartiennent au même ensemble de 4 états électriques qui correspondent aux 4 valeurs logiques que les signaux a et b représentent de façon statique, 00, 01, 10 et 11. Les évènements 1, 5, 9 et 13 ne représentent aucune activité sur les signaux a et b. L'absence de transition n'entraîne aucune variation instantanée des caractéristiques électriques du réseau de transistors et donc aucune variation instantanée de la consommation. Ces 4 évènements ne seront plus considérés dans la suite.

1.9 Réseau de transistors connectés en série

Un réseau de transistors en série est utilisé dans la construction d'une porte logique afin de réaliser une fonction *et* de conduction entre les signaux commandant leurs grilles. Il s'agit d'une interprétation de la logique des entrées qui correspond à une technologie hybride transformant des tensions en entrée en une conduction en sortie. Les réseaux série sont composés soit de transistors de type NMOS pour une fonction pull-down, soit de transistors de type PMOS pour une fonction pull-up. Le plus petit réseau série n'est composé que de deux transistors (Figure 1.2).

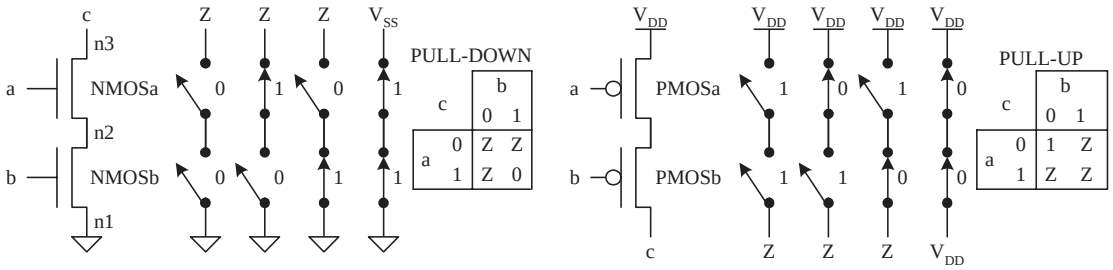


FIG. 1.2 – Réseaux de transistors en série

Un réseau de deux transistors en série passe par un ensemble de combinaisons de modes de fonctionnement des transistors lors des transitions des entrées. La valeur de la capacité grille-source du transistor d'une entrée paramètre la consommation du circuit générant ladite entrée (Figure 1.3).

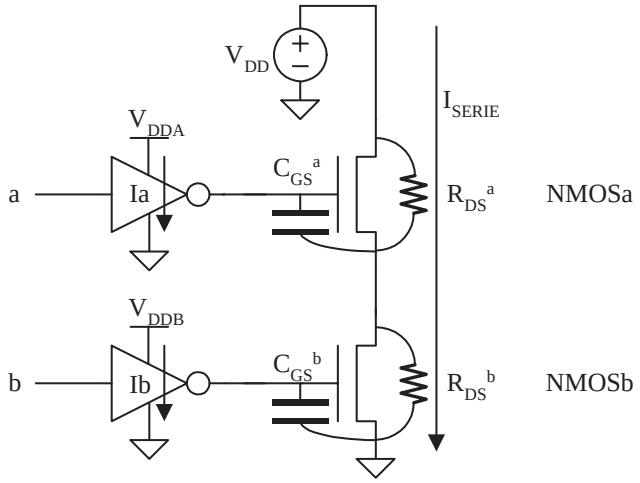


FIG. 1.3 – Circuit de simulation de l'influence des capacités grille-source sur la consommation des circuits d'entrée et de l'influence de la résistance drain-source sur la consommation du réseau de transistors série

La valeur de la résistance drain-source du transistor d'une entrée paramètre la consommation du réseau série commandé par ladite entrée. L'état initial des signaux définit le mode de fonctionnement des transistors dans l'état initial. Lors d'un évènement sur les signaux d'entrée, les transistors changent d'état de fonctionnement en passant par un état transitoire. Les capacités

grille-source C_{GS} et les résistances drain-source R_{DS} sont modifiées de manière précise en passant de l'état initial à l'état transitoire des transistors. De même, l'évènement sur les signaux d'entrée détermine l'état final des transistors ainsi que la façon dont sont modifiées les capacités grille-source et les résistance drain-source en passant du mode de fonctionnement transitoire au mode de fonctionnement final du transistor. L'influence de la variation des capacités grille-source sur la consommation des circuits générant les entrées et l'influence de la variation des résistances drain-source sur la consommation du réseau série est illustrée en utilisant le circuit de la figure 1.3. Un inverseur est utilisé comme circuit d'entrée pour les signaux a et b, chacun ayant sa propre alimentation. Les simulations qui suivent illustrent à chaque fois les variations des tensions drain-source qui modifient les capacités grille-source et donc la consommation des circuits inverseurs sur les deux entrées à travers I_a et I_b. Elles illustrent aussi l'impact des tensions drain-source qui modifient les résistances drain-source et donc la consommation des réseaux série de transistor à travers I_{SERIE}.

1.10 États électriques initiaux et finaux d'un réseau série de deux transistors

Dans l'état électrique noté 00S, les transistors NMOSa et NMOSb sont ouverts. Le transistor NMOSa est soumis à l'effet de substrat. Les tensions $V(n3)$, $V(n2)$ et $V(n1)$ (Figure 1.2) sont donc différentes et les tensions de seuil V_{THNa} et V_{THNb} le sont aussi. La longueur du canal est constante pour les deux transistors, c'est à dire que $\frac{dX_{dl}}{dV_{n3n2}}$ et $\frac{dX_{dl}}{dV_{n2n1}}$ sont nuls. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont égales à :

$$\begin{array}{rcl} C^a_{an2} & = & W.L.C_{ox} \\ C^b_{bn1} & = & W.L.C_{ox} \\ R^a_{n3n2} & = & \infty \\ R^b_{n2n1} & = & \infty \\ \text{avec} & & V(n3) \neq V(n2) \neq V(n1) \\ & & V_{THNa} \neq V_{THNb} \end{array} \tag{1.14}$$

Dans l'état électrique noté 01S, le transistor NMOSa est ouvert et le transistor NMOSb est saturé. La tension $V(n3)$ a une valeur quelconque constante et les tensions $V(n2)$ et $V(n1)$ sont égales. La source du transistor NMOSa étant connectée à la masse via le transistor NMOSb, il n'est pas soumis à l'effet de substrat. Les tensions de seuil V_{THNa} et V_{THNb} sont donc égales. La longueur du canal est constante pour les deux transistors, c'est à dire que $\frac{dX_{dl}}{dV_{n3n2}}$ et $\frac{dX_{dl}}{dV_{n2n1}}$sont nuls. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont égales à :

$$\begin{array}{rcl} C^a_{an2} & = & W.L.C_{ox} \\ C^b_{bn1} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ R^a_{n3n2} & = & \infty \\ R^b_{n2n1} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \end{array}$$

$$\begin{array}{ll} \text{avec} & V(n3) \neq V(n2) = V(n1) \\ & V_{THNa} = V_{THNb} \end{array} \tag{1.15}$$

Dans l'état électrique noté 10S, le transistor NMOSa est saturé et le transistor NMOSb est ouvert. La différence de potentiel drain-source V_{n3n2} du transistor NMOSa est donc toujours nulle et sa longueur de canal reste constante ($\frac{dX_{dl}}{dV_{n3n2}} = 0$). Le transistor NMOSb étant ouvert fait que la source du transistor NMOSa n'est pas connectée à la masse. Il apparaît une différence de potentiel entre la source et le substrat du transistor NMOSa dit effet de substrat. Or le substrat est connecté à la masse tout comme le nœud $n1$. De fait, la différence de potentiel drain-source V_{n2n1} du transistor NMOSb peut être légitimement considérée non nulle. De plus, l'apparition d'une différence de potentiel source-substrat V_{n2-sub} du transistor NMOSa modifie aussi sa tension de seuil de fonctionnement V_{THNa} la rendant différente de la tension de seuil V_{THNb} du transistor NMOSb. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont alors égales à :

$$\begin{array}{rcl} C^a_{an2} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ C^b_{bn1} & = & C_{ox}.W.L \\ R^a_{n3n2} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \\ R^b_{n2n1} & = & \infty \\ \text{avec} & & V(n3) = V(n2) \neq V(n1) \\ & & V_{THNa} \neq V_{THNb} \end{array} \tag{1.16}$$

Dans l'état électrique noté 11S, les transistor NMOSa et NMOSb sont saturés. Les différences de potentiel drain-source V_{n3n2} pour le transistor NMOSa et $V(n2n1)$ pour le transistor NMOSb sont nulles. La variation de la longueur de canal du transistor NMOSa $\frac{dX_{dl}}{dV_{n3n2}}$ et la variation de la longueur de canal du transistor NMOSb $\frac{dX_{dl}}{dV_{n2n1}}$ sont nulles. La source du transistor NMOSa est connectée à la masse via le transistor NMOSb ce qui fait que le transistor NMOSa n'est pas soumis à l'effet de substrat. La tension de seuil V_{THNa} du transistor NMOSa est égale à la tension de seuil V_{THNb} du transistor NMOSb. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont alors égales à :

$$\begin{array}{rcl} C^a_{an2} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ C^b_{bn1} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ R^a_{n3n2} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \\ R^b_{n2n1} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \\ \text{avec} & & V(n3) = V(n2) = V(n1) \\ & & V_{THNa} = V_{THNb} \end{array} \tag{1.17}$$

1.11 Dispersion instantanée des caractéristiques électriques d'un réseau de deux transistors en série

1. Évènement 2
   L'état initial (Figures 1.1 et 1.4) correspond à l'état électrique 00S. Lors du front montant

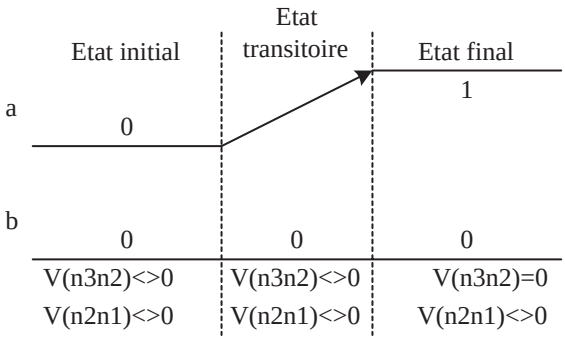


FIG. 1.4 – Évènement 2

du signal a, le transistor NMOSa passe dans l'état non saturé. La tension $V(n2)$ est modifiée parce que $V(n3)$ tend vers $V(n2)$. Ceci entraîne aussi une variation de la différence de potentiel drain-source $V(n2n1)$ du transistor NMOSb. De plus, le transistor NMOSa reste soumis à l'effet de substrat parce que sa source n'est pas connectée à la masse via le transistor NMOSb. Les tensions de seuil V_{THNa} et V_{THNb} des transistors NMOSa et NMOSb sont donc différentes. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa})-V_{n3n2})}\\
R^b_{n2n1} &= \infty\\
\text{avec} \quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned} \tag{1.18}$$

L'état électrique final correspond à l'état électrique 10S.

2. Évènement 3
   L'état initial correspond à l'état électrique 11S. Dans l'état transitoire, lors du front descen-

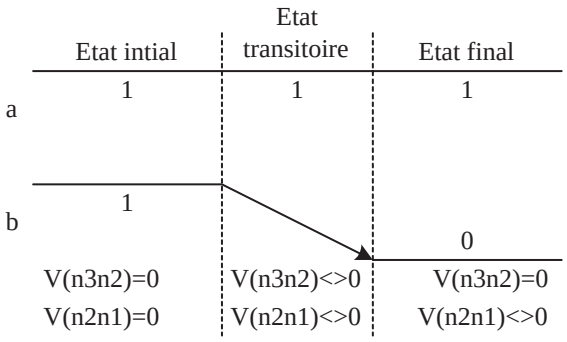


FIG. 1.5 – Évènement 3

dant du signal b, le transistor NMOSb passe dans l'état non saturé. Le transistor NMOSa reste saturé. La source du transistor NMOSa commence à ne plus être plus connectée à la masse via le transistor NMOSb. Il est soumis à l'effet de substrat qui entraîne une différence de potentiel source-substrat V_{n2-sub} non nulle du transistor NMOSa. Or le substrat étant connecté à la masse tout comme le nœud $n1$ du transistor NMOSb, il apparaît une différence de potentiel drain-source V_{n2n1} non nulle pour le transistor NMOSb. La tension de seuil V_{THNa} du transistor NMOSa devient de fait différente de la tension de seuil V_{THNb} du transistor NMOSb. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$
\begin{aligned}
C^a_{an2} &= \frac{\frac{2}{3}.C_{ox}.W.(L-2.L_{diff}-X_{dl}).\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n3n2}}.V_{n3n2}\right)}{1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n3n2}}.(V_{n3n2}-V_{an2}+V_{THNa})} \\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right) \\
R^a_{n3n2} &= \frac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n3n2}}.(V_{an2}-V_{THNa})^2\right)} \\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W(V_{bn1}-V_{THNb}-V_{n2n1})} \\
& V_{THNa}=V_{THNb} \\
\text{avec} \quad & V(n3)=V(n2)\neq V(n1) \\
& V_{THNa}\neq V_{THNb}
\end{aligned}
\tag{1.19}
$$

L'état final correspond à l'état électrique 10S.

3. Évènement 4

L'état initial correspond à l'état électrique 01S. Lors du front montant du signal a et

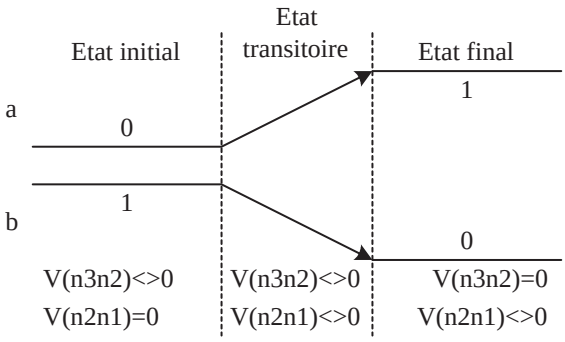


FIG. 1.6 – Évènement 4

du front descendant du signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. La tension $V(n3)$ est modifiée et tend vers $V(n2)$. La source du transistor NMOSa commence à ne plus être connectée à la masse via le transistor NMOSb parce que celui-ci devient non saturé. Le transistor NMOSa est soumis à l'effet de substrat. La différence de potentiel source-substrat V_{n2-sub} devient non nulle entraînant un différence de potentiel drain-source V_{n2n1} non nulle du transistor NMOSb. Ceci entraîne une variation des différences de potentiel drain-source $V(n3n2)$ et $V(n2n1)$ des transistors NMOSa et NMOSb. Les tensions de seuil V_{THNa} et V_{THNb} des deux transistors deviennent différentes pour la même raison. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$
\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa})-V_{n3n2})}\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THNb})-V_{n2n1})}\\
\text{avec}\quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned}
\tag{1.20}
$$

L'état final correspond à l'état électrique 10S.

4. Évènement 6

L'état initial de l'évènement 6 correspond à l'état électrique 01S. Lors du front montant

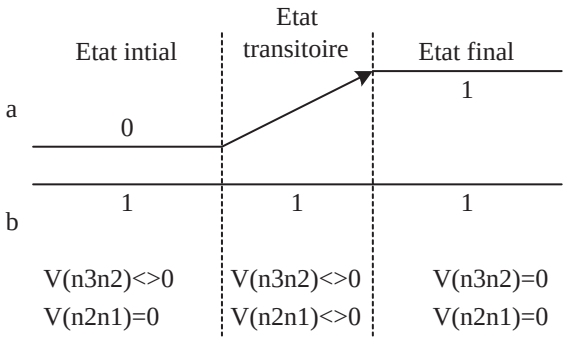


FIG. 1.7 – Évènement 6

du signal a (Figure 1.7), le transistor NMOSa passe dans l'état non saturé. Le transistor NMOSb reste dans l'état saturé. La tension $V(n3)$ est modifiée et commence à tendre vers $V(n2)$. La source du transistor NMOSa est toujours connectée à la masse via le transistor NMOSb. Le transistor NMOSa n'est jamais soumis à l'effet de substrat, les tensions de seuil V_{THNa} et V_{THNb} restent égales. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$
\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(2-\frac{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= \frac{\dfrac{2}{3}.C_{ox}.W.(L-2.L_{diff}-X_{dl}).\left(1+\dfrac{1}{L-2.L_{diff}-X_{dl}}.\dfrac{dX_{dl}}{dV_{n2n1}}.V_{n2n1}\right)}{1+\dfrac{1}{L-2.L_{diff}-X_{dl}}.\dfrac{dX_{dl}}{dV_{n2n1}}.(V_{n2n1}-V_{bn1}+V_{THNb})}\\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa})-V_{n3n2})}\\
R^b_{n2n1} &= \frac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\dfrac{1}{L-2.L_{diff}-X_{dl}}.\dfrac{dX_{dl}}{dV_{n2n1}}.(V_{bn1}-V_{THNb})^2\right)}\\
\text{avec}\quad & V(n3)\neq V(n2)=V(n1)\\
& V_{THNa}=V_{THNb}
\end{aligned}
\tag{1.21}
$$

L'état final correspond à l'état électrique 11S.

5. Évènement 7
   L'état initial de l'évènement 7 (correspond à l'état électrique 10S. Dans l'état transitoire

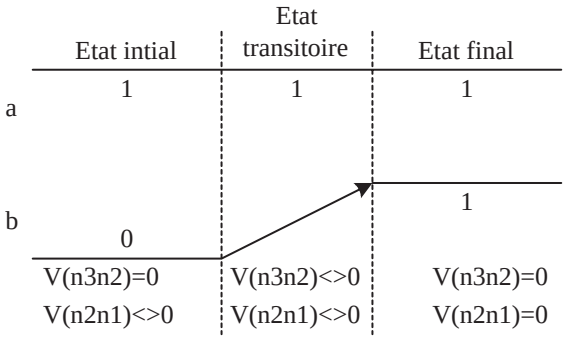


FIG. 1.8 – Évènement 7

lors du front montant du signal b (Figure 1.8), le transistor NMOSb passe dans l'état non saturé. Le transistor NMOSa reste dans l'état saturé. La source du transistor NMOSa commence à être connectée à la masse via le transistor NMOSb qui devient non saturé. L'effet de substrat commence à disparaître mais les tensions de seuil V_{THNa} et V_{THNb} des deux transistros restent transitoirement différentes. La tension $V(n2)$ est modifiée et commence à tendre vers $V(n1)$. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistor NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an2} &= \frac{\frac{2}{3}.C_{ox}.W.(L-2.L_{diff}-X_{dl}).\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n3n2}}.V_{n3n2}\right)}{1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n3n2}}.(V_{n3n2}-V_{an2}+V_{THNa})} \\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right) \\
R^a_{n3n2} &= \frac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n3n2}}.(V_{an2}-V_{THNa})^2\right)} \\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THNb})-V_{n2n1})} \\
\text{avec} \quad & V(n3)=V(n2)\neq V(n1) \\
& V_{THNa}\neq V_{THNb}
\end{aligned} \tag{1.22}$$

L'état final de l'évènement 7 correspond à l'état électrique 11S.

6. Évènement 8

L'état initial correspond à l'état électrique 00S. Lors du front montant du signal a et du

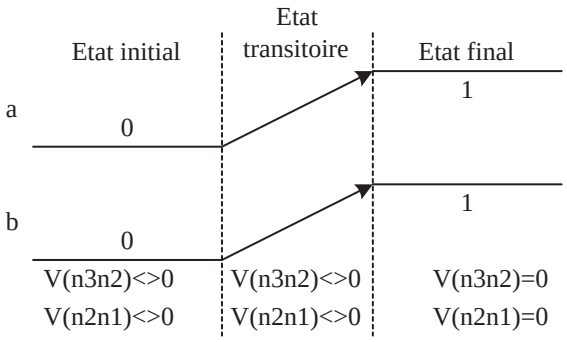


FIG. 1.9 – Évènement 8

signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. La tension $V(n3)$ est modifiée et tend vers $V(n2)$ qui est aussi modifiée et tend vers $V(n1)$. Ceci entraîne une variation des différences de potentiel drain-source $V(n3n2)$ et $V(n2n1)$ des transistors NMOSa et NMOSb. Tant que la source du transistor NMOSa n'est pas connectée à la masse via le transistor NMOSb, les tension de seuil V_{THNa} et V_{THNb} restent différentes. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa})-V_{n3n2})}\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THNb})-V_{n2n1})}\\
\text{avec}\quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned} \tag{1.23}$$

L'état final de l'évènement 8 correspond à l'état électrique 11S.

7. Évènement 10
   L'état initial de l'évènement 10 correspond à l'état électrique 10S. Lors du front descendant

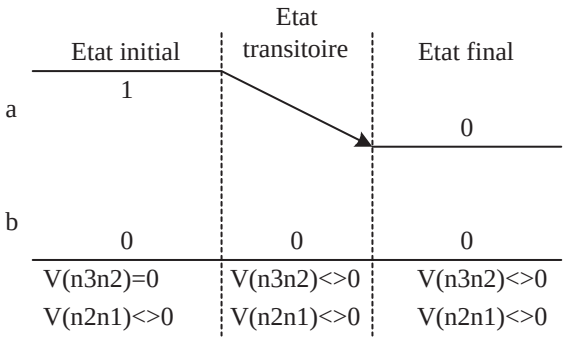


FIG. 1.10 – Évènement 10

du signal a, le transistor NMOSa passe dans l'état non saturé. Le transistor NMOSa est toujours soumis à l'effet de substrat, les tensions $V(n2)$ et $V(n1)$ sont différentes ainsi que les tensions de seuil V_{THNa} et V_{THNb}. Le nœud $n3$ n'étant plus connecté au nœud $n2$, la tension $V(n3)$ devient différente de la tension $V(n2)$. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa})-V_{n3n2})}\\
R^b_{n2n1} &= \infty\\
\text{avec}\quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned} \tag{1.24}$$

L'état final de l'évènement 10 correspond à l'état électrique 00S.

8. Évènement 11

L'état initial de l'évènement 11 correspond à l'état électrique 01S. Lors du front descen-

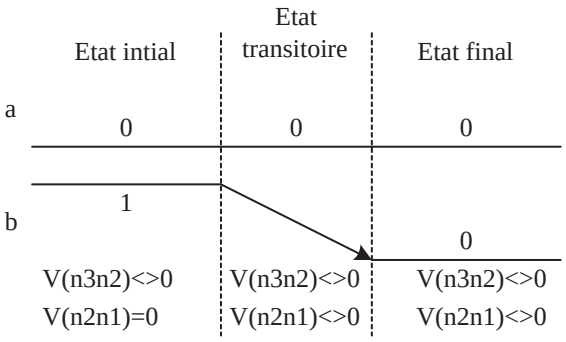


FIG. 1.11 – Évènement 11

dant du signal b, le transistor NMOSb passe dans l'état non saturé. Le transistor NMOSa commence à être soumis à l'effet de substrat parce que sa source n'est plus connectée à la masse via le transistor NMOSb. La différence de potentiel drain-source $V(n2n1)$ du transistor NMOSb devient donc non nulle et les tensions de seuil V_{THNa} et V_{THNb} des deux transistors deviennent différentes. Le nœud $n3$ n'étant plus connecté au nœud $n2$, la tension $V(n3)$ devient différente de la tension $V(n2)$. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \infty\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THNb})-V_{n2n1})}\\
\text{avec} \quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned} \tag{1.25}$$

L'état final correspond à l'état électrique 00S.

9. Évènement 12
   L'état initial de l'évènement 12 correspond à l'état électrique 11S. Lors du front descendant

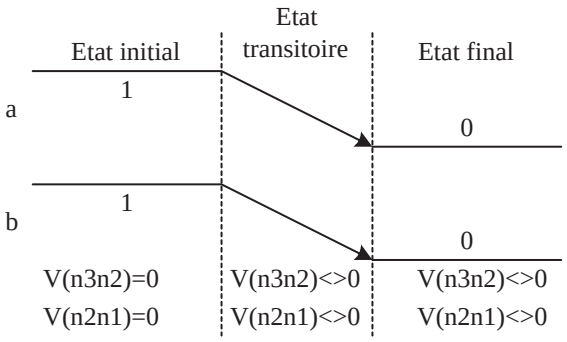


FIG. 1.12 – Évènement 12

du signal a et du signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. La source du transistor NMOSa n'est plus connectée à la masse via le transistor NMOSb et le transistor NMOSa commence à être soumis à l'effet de substrat. La tension $V(n2)$ du nœud $n2$ devient différente de la tension $V(n1)$ du nœud $n1$. De plus, les tensions de seuil V_{THNa} et V_{THNb} des deux transistors deviennent différentes pour la même raison. Le nœud $n3$ n'étant plus connecté au nœud $n2$ parce que le transistor NMOSa devient non saturé, la tension $V(n3)$ est différente de la tension $V(n2)$. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$
\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa})-V_{n3n2})}\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THNb})-V_{n2n1})}\\
\text{avec}\quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned}
\tag{1.26}
$$

L'état final de l'évènement 12 correspond à l'état électrique 00S.

10. Évènement 14
    L'état initial de l'évènement 14 correspond à l'état électrique 11S. Lors du front descendant

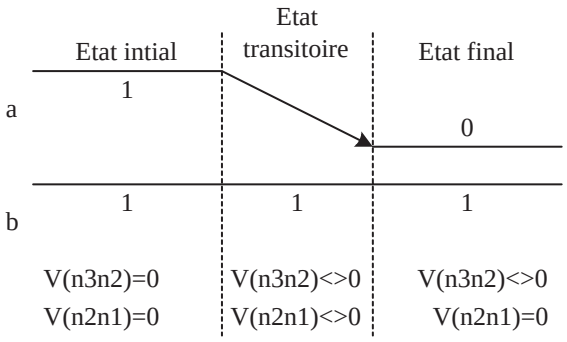


FIG. 1.13 – Évènement 14

du signal a, le transistor NMOSa passe dans l'état non saturé. Le transistor NMOSb reste saturé. La source du transistor NMOSa est toujours connectée à la masse via le transistor NMOSb. Le transistor NMOSa n'est pas soumis à l'effet de substrat. La différence de potentiel drain-source $V(n2n1)$ du transistor NMOSb est nulle et les tensions de seuil V_{THNa} du transistor NMOSa et V_{THNb} du transistor NMOS restent égales. La tension V_{n3} n'est plus connectée à la source du transistor NMOSa, celle-ci peut donc être différente. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$
\begin{aligned}
C^a_{an2} &= W.L.C_{ox}(2 - \frac{1 - \frac{V_{n3n2}}{V_{an2}-V_{THNa}} + \frac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1 - \frac{V_{n3n2}}{V_{an2}-V_{THNa}} + \frac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}) \\
C^b_{bn1} &= \frac{\frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}).\left(1 + \frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.V_{n2n1}\right)}{1 + \frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{n2n1} - V_{bn1} + V_{THNb})} \\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa}) - V_{n3n2})} \\
R^b_{n2n1} &= \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W.\left(1 + \frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{bn1} - V_{THNb})^2\right)} \\
\text{avec} \quad & V(n3) \neq V(n2) = V(n1) \\
& V_{THNa} = V_{THNb}
\end{aligned}
\tag{1.27}
$$

L'état final correspond à l'état électrique 01S.

11. Évènement 15

L'état initial de l'évènement 15 correspond à l'état électrique 00S. Lors du front montant

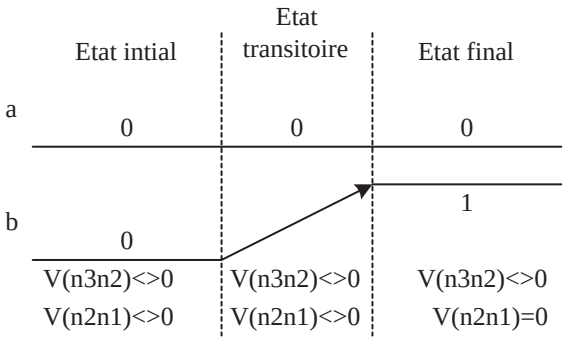


FIG. 1.14 – Évènement 15

du signal b, le transistor NMOSb passe dans l'état non saturé. La tension $V(n2)$ est modifiée parce qu'elle tend vers $V(n1)$. Ceci entraîne aussi une variation de la différence de potentiel drain-source $V(n3n2)$ du transistor NMOSa. De plus, l'effet de substrat, qui initialement affectait le transistor NMOSa, tend à disparaître au fur et mesure que $V(n2)$ tend vers $V(n1)$. Les tensions de seuil V_{THNa} et V_{THNb} des transistors NMOSa et NMOSb sont transitoirement différentes mais tendent vers la même valeur. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\frac{V_{n3n2}}{V_{an2}-V_{THNa}}+\frac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \infty\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THNb})-V_{n2n1})}\\
\text{avec} \quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned} \tag{1.28}$$

L'état final correspond à l'état électrique 01S.

12. Évènement 16
    L'état initial de l'évènement 16 correspond à l'état électrique 10S. Lors du front descendant

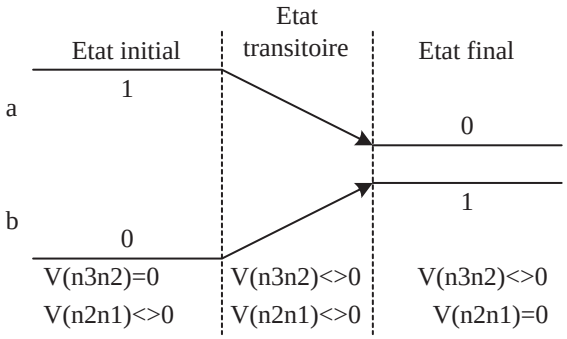


FIG. 1.15 – Évènement 16

du signal a et du front montant du signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. La source du transistor NMOSa n'est pas encore connectée à la masse via le transistor NMOSb qui devient non saturé. Le transistor NMOSa est encore soumis à l'effet de substrat. Les tensions de seuil V_{THNa} et V_{THNb} des deux transistors sont transitoirement différentes tant que la source du transistor NMOSa n'est pas connectée à la masse. La tension $V(n2)$ est modifiée et tend vers $V(n1)$. Ceci entraîne une variation des différences de potentiel drain-source $V(n3n2)$ et $V(n2n1)$ des transistors NMOSa et NMOSb. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$
\begin{aligned}
C^a_{an2} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{3(V_{an2}-V^2_{THNa})}}{1-\dfrac{V_{n3n2}}{V_{an2}-V_{THNa}}+\dfrac{V^2_{n3n2}}{2(V_{an2}-V_{THNa})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right)\\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THNa})-V_{n3n2})}\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THNb})-V_{n2n1})}\\
\text{avec}\quad & V(n3)\neq V(n2)\neq V(n1)\\
& V_{THNa}\neq V_{THNb}
\end{aligned}
\tag{1.29}
$$

L'état final de l'évènement 16 correspond à l'état électrique 01.

1.12 Exemple d'observation de la dispersion instantanée des caractéristiques électriques

La dispersion instantanée des caractéristiques électriques d'un réseau série de deux transistors est observable à travers, par exemple, sa consommation en courant. Les évènements 3 et 14 sont pris en exemple afin de démontrer qu'un front montant sur le signal a n'est pas équivalent, sur le plan électrique et donc de la consommation instantanée, à un front montant du signal b pour un réseau série de transistors.

Les états initiaux des évènements 3 et 14 sont identiques et correspondent à l'état électrique 11S. Dans l'état initial ils présentent tous deux des caractéristiques électriques identiques puisque les deux entrées a et b sont égales à 1 dans les deux cas (Équation 1.17)

Dans l'état transitoire de l'évènement 3, le transistor NMOSb passe dans l'état non saturé lors du front descendant du signal b. Le transistor NMOSa reste saturé. La source du transistor NMOSa commence à ne plus être plus connectée à la masse via le transistor NMOSb. Il est soumis à l'effet de substrat qui entraîne une différence de potentiel source-substrat V_{n2-sub} non nulle du transistor NMOSa. Or le substrat étant connecté à la masse tout comme le nœud $n1$ du transistor NMOSb, il apparaît une différence de potentiel drain-source V_{n2n1} non nulle pour le transistor NMOSb. La tension de seuil V_{THNa} du transistor NMOSa devient de fait différente de la tension de seuil V_{THNb} du transistor NMOSb. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à

$$\begin{array}{rcl}
C^a_{an2} & = & \dfrac{\frac{2}{3}.C_{ox}.W.(L-2.L_{diff}-X_{dl}).\left(1+\dfrac{1}{L-2.L_{diff}-X_{dl}}.\dfrac{dX_{dl}}{dV_{n3n2}}.V_{n3n2}\right)}{1+\dfrac{1}{L-2.L_{diff}-X_{dl}}.\dfrac{dX_{dl}}{dV_{n3n2}}.(V_{n3n2}-V_{an2}+V_{THNa})} \\
C^b_{bn1} & = & W.L.C_{ox}\left(2-\dfrac{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THNb})}}{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THNb}}+\dfrac{V^2_{n2n1}}{2(V_{bn1}-V_{THNb})^2}}\right) \\
R^a_{n3n2} & = & \dfrac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\dfrac{1}{L-2.L_{diff}-X_{dl}}.\dfrac{dX_{dl}}{dV_{n3n2}}.(V_{an2}-V_{THNa})^2\right)} \\
R^b_{n2n1} & = & \dfrac{L_{eff}}{\mu_n.C_{ox}.W(V_{bn1}-V_{THNb}-V_{n2n1})} \\
\text{avec} & & V(n3)=V(n2)\neq V(n1) \\
 & & V_{THNa}\neq V_{THNb}
\end{array} \tag{1.30}$$

Dans l'état transitoire de l'évènement 14, le transistor NMOSa passe dans l'état non saturé lors du front descendant du signal a. Le transistor NMOSb reste saturé. La source du transistor NMOSa est toujours connectée à la masse via le transistor NMOSb. Le transistor NMOSa n'est pas soumis à l'effet de substrat. La différence de potentiel drain-source $V(n2n1)$ du transistor NMOSb est nulle et les tensions de seuil V_{THNa} du transistor NMOSa et V_{THNb} du transistor

NMOS restent égales. La tension V_{n3} n'est plus connectée à la source du transistor NMOSa, celle-ci peut donc être différente. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n3n2} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à

$$\begin{aligned}
C^a_{an2} &= W.L.C_{ox}(2 - \frac{1 - \frac{V_{n3n2}}{V_{an2} - V_{THNa}} + \frac{V^2_{n3n2}}{3(V_{an2} - V^2_{THNa})}}{1 - \frac{V_{n3n2}}{V_{an2} - V_{THNa}} + \frac{V^2_{n3n2}}{2(V_{an2} - V_{THNa})^2}}) \\
C^b_{bn1} &= \frac{\frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}).\left(1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.V_{n2n1}\right)}{1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{n2n1} - V_{bn1} + V_{THNb})} \\
R^a_{n3n2} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2} - V_{THNa}) - V_{n3n2})} \\
R^b_{n2n1} &= \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W.\left(1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{bn1} - V_{THNb})^2\right)} \\
\text{avec} \quad & V(n3) \neq V(n2) = V(n1) \\
& V_{THNa} = V_{THNb}
\end{aligned} \tag{1.31}$$

Lors du passage dans l'état transitoire, l'évènement 3, qui correspond à un front descendant du signal b pendant que le signal a reste à 1, présente des différences de caractéristiques électriques avec l'évènement 14 qui correspond à un front descendant du signal a pendant que le signal b reste à 1. Ces différences portent sur les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances R^a_{n3n2} et R^b_{n2n1} qui dépendent des tensions d'interconnexion V(n3), V(n2) et V(n1) et des tensions de seuil V_{THNa} et V_{THNb} différentes entre les deux évènements (Équations 1.30 et 1.31). Ces différences de variations instantanées des caractéristiques électriques entraînent alors une différence de consommation instantanée entre les deux évènements lors du passage dans l'état transitoire.

L'état final de l'évènement 3 correspond à l'état électrique 10S. Le transistor NMOSa est saturé et le transistor NMOSb est ouvert. La source du transistor NMOSa n'est plus connectée à la masse via le transistor NMOSb, il est soumis à l'effet de substrat entraînant une différence de potentiel drain-source V_{n2n1} non nulle du transistor NMOSb et une tension de seuil V_{THNa} du transistor NMOSa différente de la tension de seuil V_{THNb} du transistor NMOSb. L'état final de l'évènement 14 correspond à l'état électrique 01S. Le transistor NMOSa est ouvert et le transistor NMOSb est saturé. La source du transistor NMOSa est connectée à la masse via le transistor NMOSb. Le transistor NMOSa n'est pas soumis à l'effet de substrat. La différence de potentiel drain-source V_{n2n1} du transistor NMOSb est donc nulle et les tensions de seuil V_{THNa} du transistor NMOSa et V_{THNb} du transistor NMOS sont égales. La tension V_{n3} n'étant pas connectée à la source du transistor NMOSa, celle-ci peut donc être différente.

Les deux évènements continuent donc à se différencier lors du passage dans l'état final. Les différences portent à nouveau sur les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances R^a_{n3n2}

et R^b_{n2n1} qui dépendent des tensions d'interconnexion V(n3), V(n2) et V(n1) et des tensions de seuil V_{THNa} et V_{THNb} encore différentes entre les deux évènements (Équations 1.15 et 1.16). A nouveau, cela se traduit par une consommation instantanée différente entre les évènement 3 et 14 lors du passage dans l'état final. Il est possible d'estimer théoriquement la différence en courant en se plaçant en un point où pour l'évènement 3 les conditions sont $C_{ox} = \frac{1.75fF}{\mu m^2}$, $L = 0.6\mu m$, $W = 0.8\mu m$, $V_{THN} = 0.8 + 0.3V$, $V^a_{an2} = 2.5V$ et $V_{n3n2} = 0.1V$ et où, pour l'évènement 14, les condition sont $C_{ox} = \frac{1.75fF}{\mu m^2}$, $L = 0.6\mu m$, $W = 0.8\mu m$, $V_{THN} = 0.8V$, $V^b_{bn1} = 2.5V$ et $V_{n2n1} = 0.1V$. Le calcul donne la différence de capacités (Équation 1.32) entre les deux évènements en ce point. La différence de variation de la capacité de grille pour une tension milieu de 2.5V pendant une picoseconde représente une différence de courant théorique entre les circuits des entrées a et b (Équation 1.32). Une simulation SPICE confirme cette approche théorique avec un ordre de grandeur similaire pour le courant (Figure 1.16).

$$\begin{aligned}
C^a_{an2} &= 0.84 \text{ fF} \cdot \left(2 - \frac{1 - \frac{0.1}{5-1.1} + \frac{0.1^2}{3(5-1.1^2)}}{1 - \frac{0.1}{5-1.1} + \frac{0.1^2}{2(5-1.1)^2}} \right) \simeq 0.839970 \text{ fF} \\
C^b_{bn1} &= 0.84 \text{ fF} \cdot \left(2 - \frac{1 - \frac{0.1}{5-0.8} + \frac{0.1^2}{3(5-0.8^2)}}{1 - \frac{0.1}{5-0.8} + \frac{0.1^2}{2(5-0.8)^2}} \right) \simeq 0.839944 \text{ fF} \\
\Delta C &\simeq 2.551\ 10^{-5} \text{ fF} \\
i &\simeq 2.551\ 10^{-5}\ 10^{-15} \frac{5}{10^{-12}} \simeq 0.64\ \mu\text{A}
\end{aligned} \tag{1.32}$$

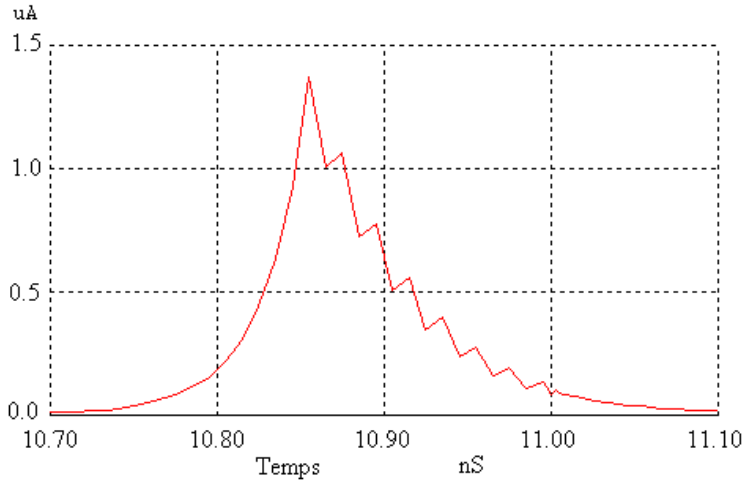


FIG. 1.16 – Différence de consommation instantanée entre les circuits d'entrée des évènements 3 et 14 induite par la différence de variation des capacités grille-source

1.13 Observation de la dispersion instantanée des caractéristiques électriques d'un réseau série

L'étude précédente du réseau série de deux transistors a défini les variations des capacités grille-source et des résistances des transistors pour chacune des 16 combinaisons d'évènements possibles sur les deux entrées. Elle a aussi défini les variations des tensions des nœuds d'interconnexion du réseau série de deux transistors. Les variations des capacités grille-source entraînent des variations de consommation en courant des circuits générant les entrées. Les variations des résistances des transistors entraînent une variation de consommation en courant du réseau série de transistors. Ces résultats expliquent théoriquement la différence instantanée de la consommation entre deux évènements. L'analyse complète de tous les évènements deux à deux avec ce modèle démontre théoriquement que tous les évènements présentent une consommation instantanée toujours différente, soit à cause d'une différence de capacité grille-source, soit à cause d'une différence de résistance drain-source, soit à cause d'une différence de tension des nœuds internes. Cela permet de conclure que deux évènements différents sur un réseau série de transistors ont toujours une consommation différente. Le tableau 1.1 résume ce résultat de façon qualitative où les indices a, b, c, d et e représentent un ordre de grandeur croissant de la différence de consommation. Par exemple, les évènements 3, 10, 14, etc provoquent les plus faibles différences de consommation entre eux (supérieure au micro ampère) et les évènements 4, 14, 15, etc provoquent les plus grandes différences de consommation entre eux.

	E3	E4	E6	E7	E8	E10	E11	E12	E14	E15	E16
E2	b	d	c	b	d	b	b	c	b	b	d
E3		e	c	b	e	a	a	c	a	b	d
E4			c	c	b	d	d	b	e	e	c
E6				c	c	c	c	c	c	c	d
E7					d	b	b	c	b	b	c
E8						d	c	c	e	e	c
E10							a	c	a	b	c
E11								c	a	b	d
E12									c	c	b
E14										b	d
E15											c

TAB. 1.1 – Différence instantanée de consommation entre évènements sur un réseau série de deux transistors - $\mu A <$ les plus faibles a < b < c < d < e les plus grandes

L'étude de la différence de consommation instantanée entre deux transistors connectés en série est généralisable. Il est tout à fait possible de remplacer chacun des transistors par un réseau de transistors. Ces réseaux sont alors connectés en série. L'effet de substrat s'applique au nœud d'interconnexion qui provoque une dispersion instantanée des caractéristiques électriques entre les deux réseaux y compris s'ils sont identiques. Le tableau 1.1 est aussi applicable à des réseaux connectés en série qu'ils soient composés de transistors NMOS (Figure 1.17-a) ou de transistors PMOS (Figure 1.17-b).

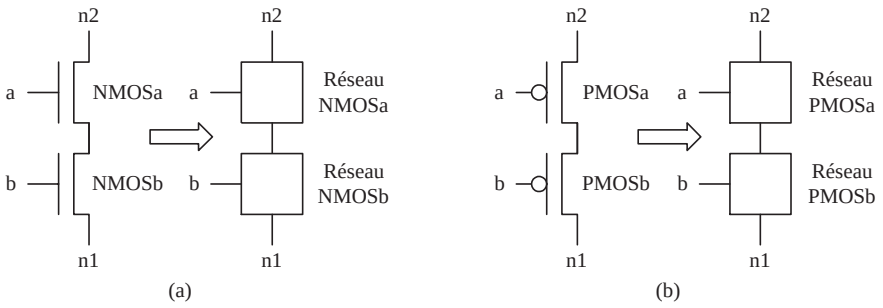


FIG. 1.17 – Généralisation aux réseaux de réseaux connectés en série

Toute connexion de transistors ou de réseaux de transistors en série entraîne une dispersion instantanée des caractéristiques électriques du circuit résultant entre deux évènements.

Une dispersion instantanée des caractéristiques électriques d'un circuit est observables à travers de grandeurs mesurables telles que la consommation en courant.

1.14 Réseau de transistors connectés en parallèle

Un réseau de transistors en parallèle est aussi utilisé dans la construction d'une porte logique et permet de réaliser une fonction *ou* entre les signaux commandant leurs grilles. Ils sont aussi composés de transistors de type NMOS pour une fonction pull-down ou de transistors de type PMOS pour une fonction pull-up. Le plus petit réseau parallèle n'est composé que de deux transistors (Figure 1.18). Un réseau de deux transistors en parallèle passe par un ensemble de

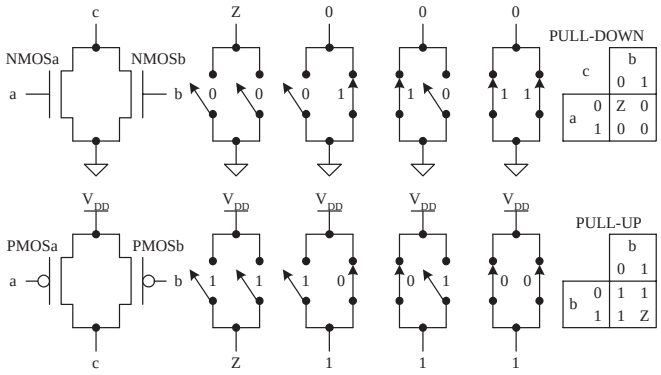


FIG. 1.18 – Réseaux de transistors en parallèle

combinaisons de modes de fonctionnement des transistors lors des transitions des entrées. La valeur de la capacité grille-source d'une entrée paramètre la consommation du circuit générant ladite entrée. L'état initial des signaux définit le mode de fonctionnement des transistors dans l'état initial. Lors d'un évènement sur les signaux d'entrée, les transistors changent d'état de fonctionnement et les capacités grille-source sont modifiées de manière précise qui dépend de l'état initial et du mode de fonctionnement transitoire des transistors. De même, l'évènement sur les signaux d'entrée détermine l'état final des transistors ainsi que la façon dont sont modifiées les capacités grille-source qui dépend de l'état transitoire et du mode de fonctionnement final du transistor. L'influence de la variation des capacités grille-source sur la consommation des circuits générant les entrées est illustrée en utilisant le circuit de la figure 1.19. Il consiste à placer un

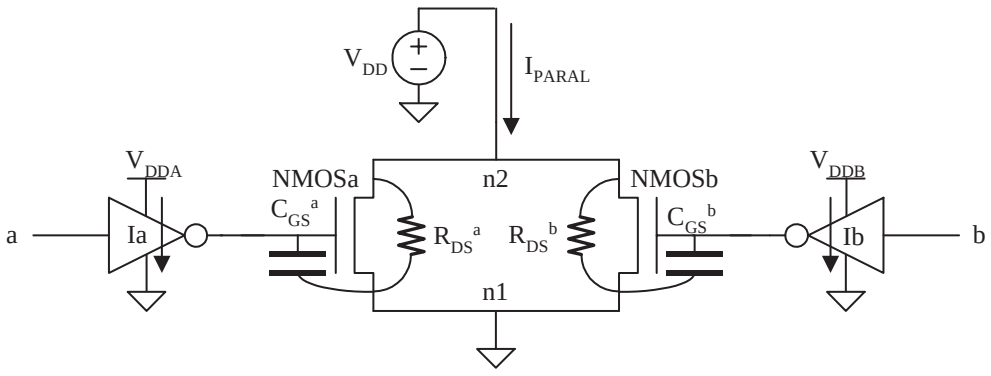


FIG. 1.19 – Circuit de simulation de l'influence des capacités grille-source sur la consommation des circuits d'entrée

inverseur comme circuit d'entrée pour les signaux a et b, chacun ayant sa propre alimentation. Les simulations qui suivent illustrent à chaque fois les variation des capacités drain-source qui

modifient les capacités grille-source et donc la consommation des circuits inverseurs sur les deux entrées. La différence de consommation entre les deux circuits inverseurs générant les entrée a et b est illustrée en parallèle avec les variations des capacités drain-source. Les transistors connectés en parallèle ont en permanence leurs sources respectives connectées soit à la masse, pour les transistors NMOS, soit à la tension d'alimentation pour les transistors PMOS. L'effet de substrat ne s'applique pas et les deux transistors conservent leurs caractéristiques électriques identiques avec l'égalité de leurs tensions de seuil V_{THNa} et V_{THNb}, notée V_{THN}, quel que soit l'évènement.

1.15 États électriques initiaux et finaux d'un réseau parallèle de deux transistors

Dans l'état électrique noté 00P, les transistors NMOSa et NMOSb sont ouverts, ce qui fait que les différences de potentiel drain-source V_{n2n1} des transistors NMOSa et NMOSb sont considérées quelconques et non nulles. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont alors égales à :

$$\begin{array}{rcl} C^a_{an1} & = & C_{ox}.W.L \\ C^b_{bn1} & = & C_{ox}.W.L \\ R^a_{n2n1} & = & \infty \\ R^b_{n2n1} & = & \infty \\ \text{avec} & & V(n2) \neq V(n1) \\ & & V_{THNa} = V_{THNb} = V_{THN} \end{array} \tag{1.33}$$

Dans l'état électrique noté 01P, le transistor NMOSb est saturé. La différence de potentiel drain-source V_{n2n1} du transistor NMOSb est nulle ainsi que la variation de la longueur du canal $\frac{dX_{dl}}{dV_{n2n1}}$. Le transistor NMOSa est ouvert, mais la différence de potentiel drain-source V_{n2n1} du transistor NMOSa est nulle puisque que le transistor NMOSb est saturé. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont alors égales à :

$$\begin{array}{rcl} C^a_{an1} & = & C_{ox}.W.L \\ C^b_{bn1} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ R^a_{n2n1} & = & \infty \\ R^b_{n2n1} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \\ \text{avec} & & V(n2) = V(n1) \\ & & V_{THNa} = V_{THNb} = V_{THN} \end{array} \tag{1.34}$$

Dans l'état électrique noté 10P, le transistor NMOSa est saturé. La différence de potentiel drain-source V_{n2n1} du transistor NMOSa est nulle ainsi que la variation de la longueur du canal $\frac{dX_{dl}}{dV_{n2n1}}$. Bien que le transistor NMOSb soit ouvert, le fait que le transistor NMOSa soit saturé implique une différence de potentiel drain-source V_{n2n1} du transistor NMOSb nulle. Les capacités

grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont alors égales à :

$$\begin{array}{rcl} C^a_{an1} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ C^b_{bn1} & = & C_{ox}.W.L \\ R^a_{n2n1} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \\ R^b_{n2n1} & = & \infty \\ \text{avec} & & V(n2) = V(n1) \\ & & V_{THNa} = V_{THNb} = V_{THN} \end{array} \tag{1.35}$$

Dans l'état électrique noté 11P, les transistors NMOSa et NMOSb sont saturés. Les différences de potentiel drain-source V_{n2n1} des transistors NMOSa et NMOSb sont nulles ainsi que la variation de la longueur du canal $\frac{dX_{dl}}{dV_{n2n1}}$. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont alors égales à :

$$\begin{array}{rcl} C^a_{an1} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ C^b_{bn1} & = & \frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}) \\ R^a_{n2n1} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \\ R^b_{n2n1} & = & \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W} \\ \text{avec} & & V(n2) = V(n1) \\ & & V_{THNa} = V_{THNb} = V_{THN} \end{array} \tag{1.36}$$

1.16 Dispersion instantanée des caractéristiques électriques d'un réseau de deux transistors en parallèle

1. Évènement 2
   L'état initial (Figures 1.1 et 1.20) correspond à l'état électrique 00P. Lors du front mon-

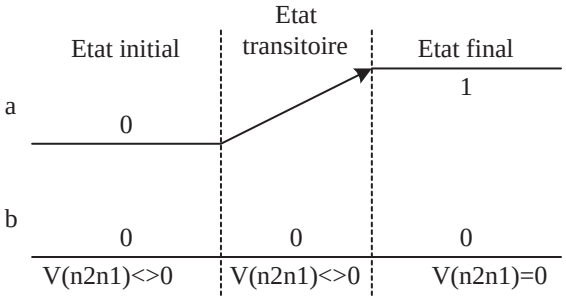


FIG. 1.20 – Évènement 2

tant du signal a, le transistor NMOSa passe dans l'état non saturé. La tension $V(n2)$ est modifiée parce qu'elle tend vers $V(n1)$. Ceci entraîne aussi une variation de la différence de potentiel drain-source $V(n2n1)$ du transistor NMOSb. Les capacités grille-source C^a_{an2} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^a_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n2n1}}{V_{an1}-V_{THN}}+\dfrac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\dfrac{V_{n2n1}}{V_{an1}-V_{THN}}+\dfrac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THN}}+\dfrac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THN}}+\dfrac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an1}-V_{THN})-V_{n2n1})}\\
R^b_{n2n1} &= \infty\\
\text{avec} \quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.37}$$

L'état électrique final correspond à l'état électrique 10P.

2. Évènement 3
   L'état initial (Figure 1.1) correspond à l'état électrique 11P. Dans l'état transitoire lors du

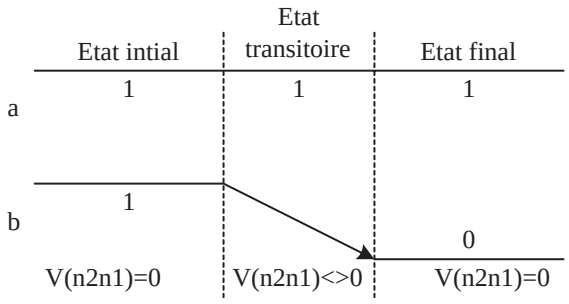


FIG. 1.21 – Évènement 3

front descendant du signal b, le transistor NMOSb passe dans l'état non saturé. Le transistor NMOSa reste saturé. Etant donné que la tension $V(n2)$ est initialement égale à la tension $V(n1)$, le passage du transistor NMOSb en mode non saturé modifie très peu la tension $V(n2)$. Puisque $V(n2)$ est très peu modifiée par l'état non saturé du transistor NMOSb, la variation de la différence de potentiel drain-source $V(n2n1)$ du transistor NMOSa et du transistor NMOSb est très faible. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= \frac{\frac{2}{3}.C_{ox}.W.(L-2.L_{diff}-X_{dl}).\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.V_{n2n1}\right)}{1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{n2n1}-V_{an1}+V_{THN})} \\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right) \\
R^a_{n2n1} &= \frac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{an1}-V_{THN})^2\right)} \\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W(V_{bn1}-V_{THN}-V_{n2n1})} \\
\text{avec} \quad & V(n2)\neq V(n1) \\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.38}$$

L'état électrique final correspond à l'état électrique 10P.

3. Évènement 4

L'état initial correspond à l'état électrique 01P. Lors du front montant du signal a et du

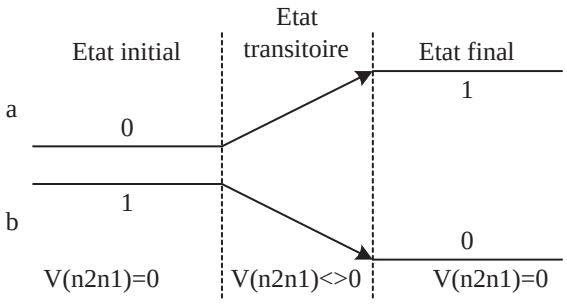


FIG. 1.22 – Évènement 4

front descendant du signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. La tension $V(n2)$ est modifiée tout en tendant vers $V(n1)$. Ceci entraîne une variation des différences de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n2n1}}{V_{an1}-V_{THN}}+\dfrac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\dfrac{V_{n2n1}}{V_{an1}-V_{THN}}+\dfrac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^a_{bn1} &= W.L.C_{ox}\left(\frac{2-\dfrac{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THN}}+\dfrac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\dfrac{V_{n2n1}}{V_{bn1}-V_{THN}}+\dfrac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an1}-V_{THN})-V_{n2n1})}\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THN})-V_{n2n1})}\\
\text{avec}\quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.39}$$

L'état électrique final correspond à l'état électrique 10P.

4. Évènement 6
   L'état initial correspond à l'état électrique 01P. Lors du front montant du signal a (Figure

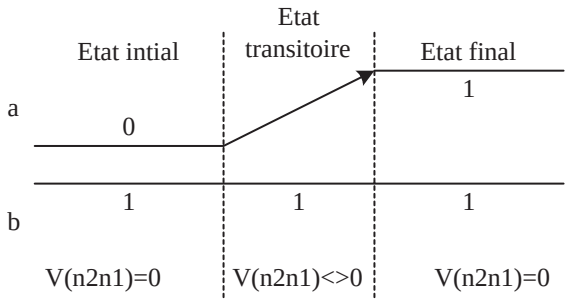


FIG. 1.23 – Évènement 6

1.23), le transistor NMOSa passe dans l'état non saturé. La tension $V(n2)$ est modifiée tout en tendant vers $V(n1)$. Ceci entraîne une variation des différences de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}(2 - \frac{1 - \frac{V_{n2n1}}{V_{an1} - V_{THN}} + \frac{V^2_{n2n1}}{3(V_{an1} - V^2_{THN})}}{1 - \frac{V_{n2n1}}{V_{an1} - V_{THN}} + \frac{V^2_{n2n1}}{2(V_{an1} - V_{THN})^2}}) \\
C^b_{bn1} &= \frac{\frac{2}{3}.C_{ox}.W.(L - 2.L_{diff} - X_{dl}).\left(1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.V_{n2n1}\right)}{1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{n2n1} - V_{bn1} + V_{THN})} \\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an1} - V_{THN}) - V_{n2n1})} \\
R^b_{n2n1} &= \frac{2(L - 2.L_{diff} - X_{dl})}{\mu_n.C_{ox}.W.\left(1 + \frac{1}{L - 2.L_{diff} - X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{bn1} - V_{THN})^2\right)} \\
\text{avec} \quad & V(n2) \neq V(n1) \\
& V_{THNa} = V_{THNb} = V_{THN}
\end{aligned} \tag{1.40}$$

L'état électrique final correspond à l'état électrique 11P.

5. Évènement 7
   L'état initial correspond à l'état électrique 10P. Dans l'état transitoire lors du front mon-

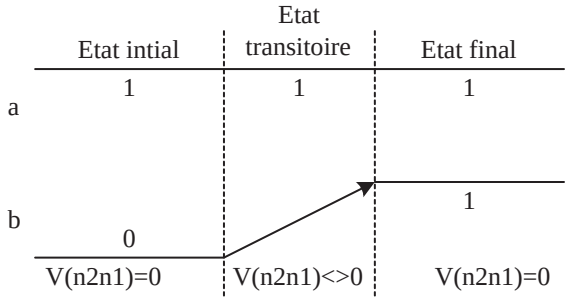


FIG. 1.24 – Évènement 7

tant du signal b (Figure 1.24), le transistor NMOSb passe dans l'état non saturé. La tension $V(n2)$ est modifiée tout en tendant vers $V(n1)$. Ceci entraîne une variation des différences de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} du transistor NMOSa et du transistor NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= \frac{\frac{2}{3}.C_{ox}.W.(L-2.L_{diff}-X_{dl}).\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.V_{n2n1}\right)}{1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{n2n1}-V_{an1}+V_{THN})} \\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right) \\
R^a_{n2n1} &= \frac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{an1}-V_{THN})^2\right)} \\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THN})-V_{n2n1})} \\
\text{avec} \quad & V(n2)\neq V(n1) \\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.41}$$

L'état électrique final correspond à l'état électrique 11P.

6. Évènement 8

L'état initial correspond à l'état électrique 00P. Lors du front montant du signal a et du

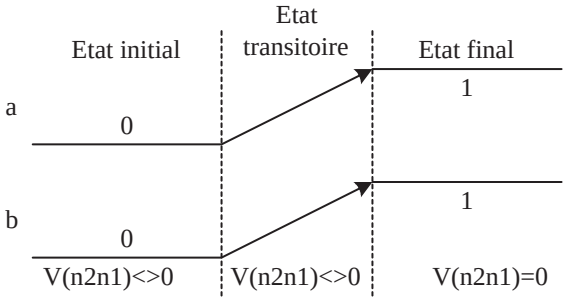


FIG. 1.25 – Évènement 8

signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. La tension $V(n2)$ est modifiée et tend vers $V(n1)$. Ceci entraîne une variation des différences de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an1}-V_{THN})-V_{n2n1})}\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THN})-V_{n2n1})}\\
\text{avec}\quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.42}$$

L'état électrique final correspond à l'état électrique 11P.

7. Évènement 10

L'état initial correspond à l'état électrique 10P. Lors du front descendant du signal a,

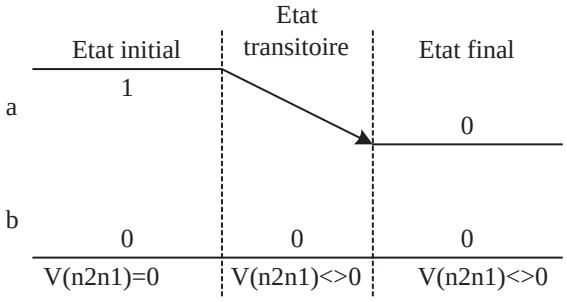


FIG. 1.26 – Évènement 10

le transistor NMOSa passe dans l'état non saturé. Etant donné que la tension $V(n2)$ est initialement égale à $V(n1)$, elle est peu modifiée par l'état non saturé du transistor NMOSa. La variation des différences de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb est très faible. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{an2}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an1}-V_{THN})-V_{n2n1})}\\
R^b_{n2n1} &= \infty\\
\text{avec} \quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.43}$$

L'état électrique final correspond à l'état électrique 00P.

8. Évènement 11

L'état initial correspond à l'état électrique 01P. Lors du front descendant du signal b,

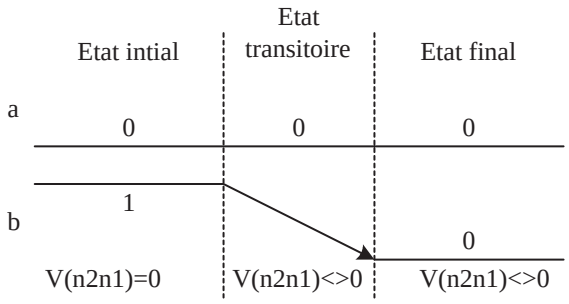


FIG. 1.27 – Évènement 11

le transistor NMOSb passe dans l'état non saturé. Etant donné que la tension $V(n2)$ est initialement égale à $V(n1)$, elle est très peu modifiée par l'état non saturé du transistor NMOSb. La variation des différences de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb est très faible. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \infty\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THN})-V_{n2n1})}\\
\text{avec} \quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.44}$$

L'état électrique final correspond à l'état électrique 00P.

9. Évènement 12

L'état initial correspond à l'état électrique 11P. Lors du front descendant du signal a

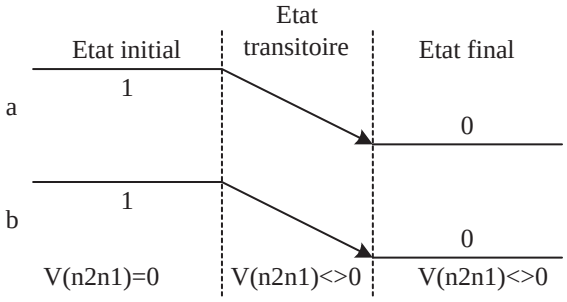


FIG. 1.28 – Évènement 12

et du signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. Etant donné que les tensions $V(n2)$ et $V(n1)$ sont initialement égales, le passage en mode non saturé des deux transistors ne les modifie que très peu. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an1}-V_{THN})-V_{n2n1})}\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THN})-V_{n2n1})}\\
\text{avec} \quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.45}$$

L'état électrique final correspond à l'état électrique 00P.

10. Évènement 14

L'état initial correspond à l'état électrique 11P. Lors du front descendant du signal a,

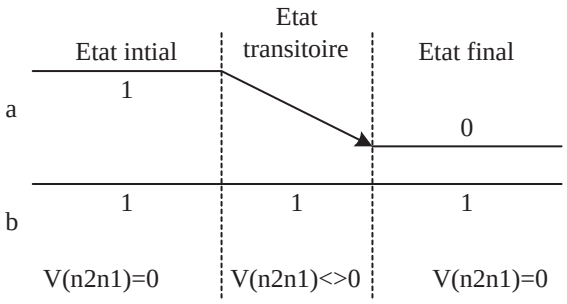


FIG. 1.29 – Évènement 14

le transistor NMOSa passe dans l'état non saturé. Etant donné que la tension $V(n2)$ est initialement égale à la tension $V(n1)$, le passage du transistor NMOSa en mode non saturé modifie très peu la tension $V(n2)$. Puisque $V(n2)$ est très peu modifiée par l'état non saturé du transistor NMOSa, la variation de la différence de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb est très faible. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= \frac{\frac{2}{3}.C_{ox}.W.(L-2.L_{diff}-X_{dl}).\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.V_{n2n1}\right)}{1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{n2n1}-V_{bn1}+V_{THN})}\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an1}-V_{THN})-V_{n2n1})}\\
R^b_{n2n1} &= \frac{2(L-2.L_{diff}-X_{dl})}{\mu_n.C_{ox}.W.\left(1+\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{n2n1}}.(V_{bn1}-V_{THN})^2\right)}\\
\text{avec}\quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.46}$$

L'état électrique final correspond à l'état électrique 01P.

11. Évènement 15

L'état initial correspond à l'état électrique 00P. Lors du front montant du signal b, le

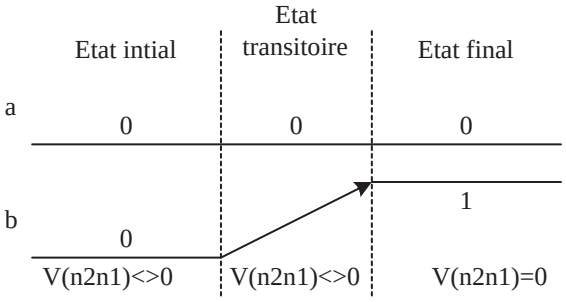


FIG. 1.30 – Évènement 15

transistor NMOSb passe dans l'état non saturé. La tension $V(n2)$ est modifiée. Ceci entraîne aussi une variation de la différence de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \infty\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THN})-V_{n2n1})}\\
\text{avec} \quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.47}$$

L'état électrique final correspond à l'état électrique 01P.

12. Évènement 16

L'état initial correspond à l'état électrique 00P. Lors du front descendant du signal a

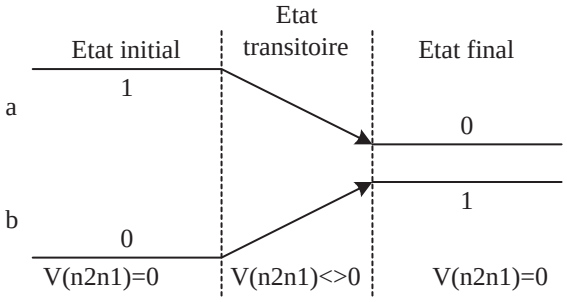


FIG. 1.31 – Évènement 16

et du front montant du signal b, les transistors NMOSa et NMOSb passent dans l'état non saturé. La tension $V(n2)$ est modifiée et tend toujours vers $V(n1)$. Ceci entraîne une variation des différences de potentiel drain-source $V(n2n1)$ des transistors NMOSa et NMOSb. Les capacités grille-source C^a_{an1} et C^b_{bn1} et les résistances drain-source R^a_{n2n1} et R^b_{n2n1} des transistors NMOSa et NMOSb sont transitoirement égales à :

$$\begin{aligned}
C^a_{an1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{an1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{an1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{an1}-V_{THN})^2}}\right)\\
C^b_{bn1} &= W.L.C_{ox}\left(2-\frac{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{3(V_{bn1}-V^2_{THN})}}{1-\frac{V_{n2n1}}{V_{bn1}-V_{THN}}+\frac{V^2_{n2n1}}{2(V_{bn1}-V_{THN})^2}}\right)\\
R^a_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{an2}-V_{THN})-V_{n2n1})}\\
R^b_{n2n1} &= \frac{L_{eff}}{\mu_n.C_{ox}.W((V_{bn1}-V_{THN})-V_{n2n1})}\\
\text{avec}\quad & V(n2)\neq V(n1)\\
& V_{THNa}=V_{THNb}=V_{THN}
\end{aligned} \tag{1.48}$$

L'état électrique final correspond à l'état électrique 01P.

1.17 Observation de la dispersion instantanée des caractéristiques électriques d'un réseau parallèle

La différence instantanée de la consommation entre les évènements 3 et 14 (Figure 1.1) peut être à nouveau reprise afin d'illustrer l'observation de la dispersion instantanée des caractéristiques électriques d'un réseau parallèle de transistors. Dans l'état initial, ils présentent les mêmes caractéristiques électriques (Équation 1.36). Lors du passage dans l'état transitoire, l'évènement 3, qui correspond à un front descendant du signal b pendant que le signal a reste à 1, présente les mêmes caractéristiques électriques que l'évènement 14 qui correspond à un front descendant du signal a pendant que le signal b reste à 1. Ceci vient du fait que les transistors NMOSa et NMOSb sont connectés en parallèle et que la tension drain-source V_{n2n1} est identique pour les deux. Il en est de même lors du passage dans l'état final. Les deux évènements entraînent les mêmes variations des caractéristiques électriques du réseau parallèle de transistors et donc ne provoquent aucune différence de consommation entre eux. A la différence du réseau série de transistors, certains évènements, qui avaient une consommation différente, ont cette fois-ci une consommation identique.

L'étude du réseau parallèle de deux transistors a défini les variations des capacités grille-source et des résistances des transistors pour chacune des 16 combinaisons d'évènements possibles sur les deux entrées. Elle a aussi défini les variations des tensions des nœuds d'interconnexion du réseau parallèle de deux transistors. Ces résultats expliquent théoriquement la différence de consommation entre deux évènements. L'analyse complète de tous les évènements deux à deux avec ce modèle démontre théoriquement que certains évènements présentent une consommation différente à cause soit d'une différence de capacité grille-source, soit d'une différence de résistance drain-source alors que d'autres ont une consommation identique. Le tableau 1.2 résume ce résultat. Un 0 indique qu'un même circuit ne présente pas de dispersion instantanée de ses caractéristiques électriques pour les deux évènements.

	E3	E4	E6	E7	E8	E10	E11	E12	E14	E15	E16
E2	b	b	b	b	c	b	b	b	a	0	b
E3		c	b	b	d	a	a	b	0	b	c
E4			c	c	b	c	c	b	c	b	0
E6				0	d	b	b	c	b	b	c
E7					c	b	b	c	b	b	c
E8						d	d	c	d	c	b
E10							0	c	a	b	c
E11								c	a	b	c
E12									c	b	b
E14										b	c
E15											b

Tab. 1.2 – Différence instantanée de consommation entre évènements sur un réseau parallèle de deux transistors - $\mu A <$ les plus faibles a < b < c < d les plus grandes

L'étude de la différence de consommation instantanée entre deux transistors connectés en paral-

lèle est généralisable. Chacun des deux transistors peut être remplacé par un réseau de transistors. Ces réseaux sont alors connectés en parallèle. L'effet de substrat ne s'applique pas au nœud d'interconnexion. Lorsque les deux réseaux sont identiques, il n'y a aucune dispersion instantanée des caractéristiques électriques. Le tableau 1.2 est aussi applicable à des réseaux connectés en parallèle qu'ils soient composés de transistors NMOS (Figure 1.32-a) ou de transistors PMOS (Figure 1.32-b).

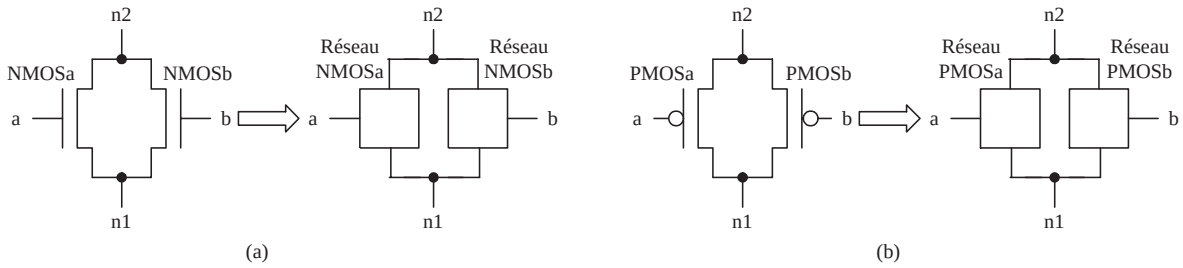


FIG. 1.32 – Généralisation aux réseaux de réseaux connectés en parallèle

Chapitre 2

Dispersion instantanée des caractéristiques électriques des portes logiques

Sommaire

2.1 La porte $Nand2$

La porte $Nand2$ est composée d'un réseau série de deux transistors NMOS et d'un réseau parallèle de deux transistors PMOS (Figure A.4 en annexe A). La différence de comportement analogique du réseau série de transistors NMOS lors des évènements sur les entrées a et b est définie par le tableau 1.1. La différence de comportement analogique du réseau parallèle de transistors PMOS lors des évènements sur les entrées a et b est définie par le tableau 1.2. La superposition des deux tableaux traduit la différence instantanée de comportement analogique de la porte $Nand2$ lorsque les deux réseaux sont simultanément soumis au même évènement. Les évènements, qui n'entraînaient pas de dispersion instantanée des caractéristiques électriques d'un circuit parallèle de transistors, la provoquent tout de même à cause du réseau série de transistors de la porte. Tous les évènements entraînent une consommation instantanée différente de la porte $Nand2$.

2.2 La porte $Nor2$

La porte $Nor2$ est composée d'un réseau parallèle de deux transistors NMOS et d'un réseau série de deux transistors PMOS (Figure A.5 en annexe A). La différence de comportement analogique du réseau parallèle de transistors NMOS lors des évènements sur les entrées a et b est définie par le tableau 1.2. La différence de comportement analogique du réseau série de transistors PMOS lors des évènements sur les entrées a et b est définie par le tableau 1.1. La superposition des deux tableaux traduit la différence instantanée de comportement analogique de la porte $Nor2$ lorsque les deux réseaux sont simultanément soumis au même évènement. Tous les évènements entraînent une consommation différente de la porte $Nor2$.

2.3 La porte $Xor2$

La porte $Xor2$ est composée de deux réseaux en parallèle de deux transistors en série NMOS et de deux réseaux en série de deux transistors PMOS en parallèle (Figure A.6 en annexe A). La différence de comportement analogique du réseau série de transistors NMOS lors des évènements sur les entrées a et b est définie par le tableau 1.1. La différence de comportement analogique du réseau parallèle de transistors PMOS lors des évènements sur les entrées a et b est définie par le tableau 1.2. La superposition des deux tableaux traduit la différence de comportement analogique de la porte $Xor2$ lorsque les deux réseaux des entrées a et b sont simultanément soumis au même évènement.

La porte $Xor2$ a aussi $\overline{a}$ et $\overline{b}$ comme entrées. Lorsque les signaux a et b sont soumis à l'évènement 1, les entrées $\overline{a}$ et $\overline{b}$ sont soumises à l'évènement 13. Dans le cas des entrées complémentées, la correspondance des évènements est donnée par le tableau 2.1.

Évènement	E1	E2	E3	E4	E5	E6	E7	E8
Évènement complémenté	E13	E14	E15	E16	E9	E10	E11	E12

TAB. 2.1 – Correspondance des évènements pour les entrées complémentées

La différence instantanée de comportement analogique du réseau série de transistors NMOS lors des évènements sur les entrées complémentées $\overline{a}$ et $\overline{b}$ est aussi définie par le tableau 1.1.

La différence instantanée de comportement analogique du réseau parallèle de transistors PMOS lors des évènements sur les entrées complémentées $\overline{a}$ et $\overline{b}$ est de même définie par le tableau 1.2.

La superposition des deux tableaux traduit la différence instantanée de comportement analogique de la porte $Xor2$ lorsque les deux réseaux des entrées $\overline{a}$ et $\overline{b}$ sont simultanément soumis au même évènement sur les entrées complémentées.

Tous les évènements entraînent une consommation différente de la porte $Xor2$.

2.4 La porte $Nand2$ cvsl

La porte $Nand2$ cvsl est composée d'un réseau série de deux transistors NMOS pour les entrées a et b et d'un réseau parallèle de deux transistors NMOS pour les entrées $\overline{a}$ et $\overline{b}$ (Figure 2.1). Ce type de circuiterie permet de générer une fonction et son complémentaire.

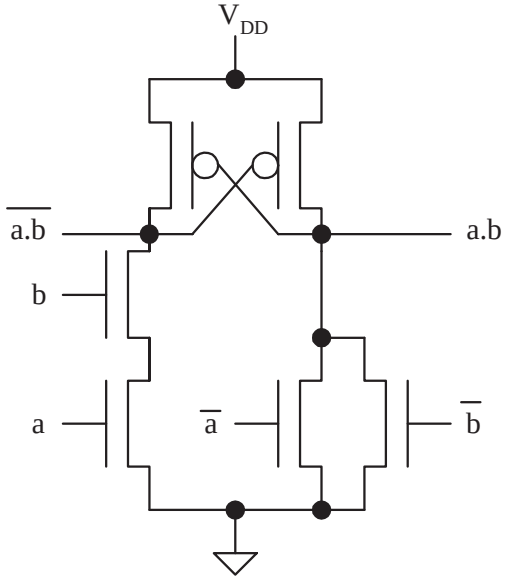


FIG. 2.1 – Schéma de la porte $Nand2cvsl$

La différence de comportement analogique du réseau série de transistors NMOS lors des évènements sur les entrées a et b est définie par le tableau 1.1. La différence de comportement analogique du réseau parallèle de transistors NMOS lors des évènements sur les entrées $\overline{a}$ et $\overline{b}$ est définie par le tableau 1.2. La superposition des deux tableaux traduit la différence de comportement analogique de la porte $Nand2$ cvsl lorsque le réseau des entrées a et b et le réseau des entrées $\overline{a}$ et $\overline{b}$ sont simultanément soumis au même évènement. Tous les évènements entraînent une consommation différente de la porte $Nand2$ cvsl.

2.5 La porte $And - Nand2$ SABL

La porte $And - Nand2$ SABL est une proposition de contre-mesure anti-DPA [26]. Elle est très similaire à la technologie cvsl précédente parce qu'elle génère f et $\overline{f}$ en deux temps en utilisant un transistor $M1$ qui retarde la commande afin d'éviter un conflit avec la valeur préchargée. Elle est composée, entre autres, d'un réseau série de deux transistors NMOS pour les entrées a et b et d'un réseau parallèle de deux transistors NMOS pour les entrées $\overline{a}$ et $\overline{b}$ (Figure 2.2).

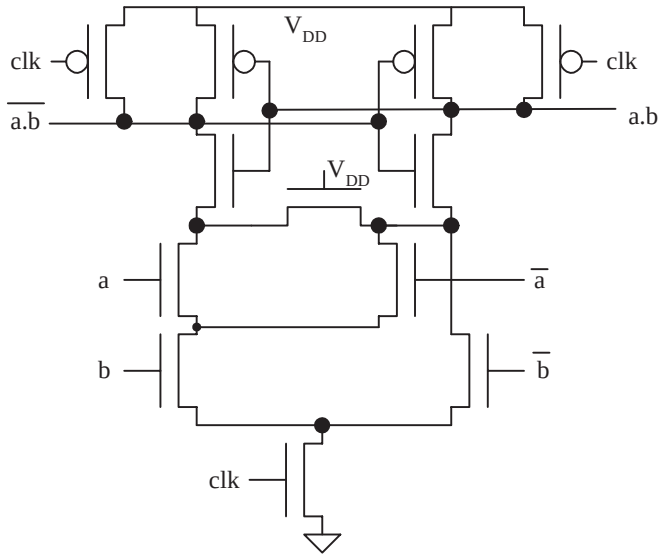


FIG. 2.2 – Schéma de la porte $And - Nand2SABL$

La différence de comportement analogique du réseau série de transistors NMOS lors des évènements sur les entrées a et b est définie par le tableau 1.1. La différence de comportement analogique du réseau parallèle de transistors NMOS lors des évènements sur les entrées $\overline{a}$ et $\overline{b}$ est définie par le tableau 1.2. La superposition des deux tableaux traduit la différence de comportement analogique de la porte $And - Nand2$ SABL lorsque le réseau des entrées a et b et le réseau des entrées $\overline{a}$ et $\overline{b}$ sont simultanément soumis au même évènement. Tous les évènements entraînent une différence instantanée de la consommation de la porte $And-Nand2$ SABL.

2.6 La porte $And2$ asynchrone

La porte $And2$ asynchrone (Figure 2.3-a) utilise dans l'exemple un codage double rail. Chaque bit a et b est respectivement codé par deux rails ($a.f$,$a.t$) et ($b.f$,$b.t$). Lorsque que le bit a a pour valeur logique 0, le rail $a.f$ prend pour valeur logique 1 et le rail $a.t$ prend pour valeur logique 0. Lorsque que le bit a a pour valeur logique 1, le rail $a.f$ prend pour valeur logique 0 et le rail $a.t$ prend pour valeur logique 1. La porte $And2$ asynchrone utilise aussi un protocole 4-phase. Dans ce protocole, les données passent par deux états. Un premier est dit valide lorsque l'un des rails prend la valeur logique 1 et un second est dit non valide lorsque les deux rails sont simultanément à la valeur logique 0. La porte $And2$ asynchrone est composée d'un étage de portes de Muller (Figure 2.3-b noté C sur le schéma). La porte C est une fonction de rendez-vous qui permet l'attente de données valides en entrée. Lorsque toutes les données en entrée sont valides, la porte asynchrone peut exécuter le calcul logique de la sortie. Une fonction *ou* CMOS classique réalise

la logique du rail $c.f$ alors que le rail $c.t$ est la sortie d'une porte de Muller.

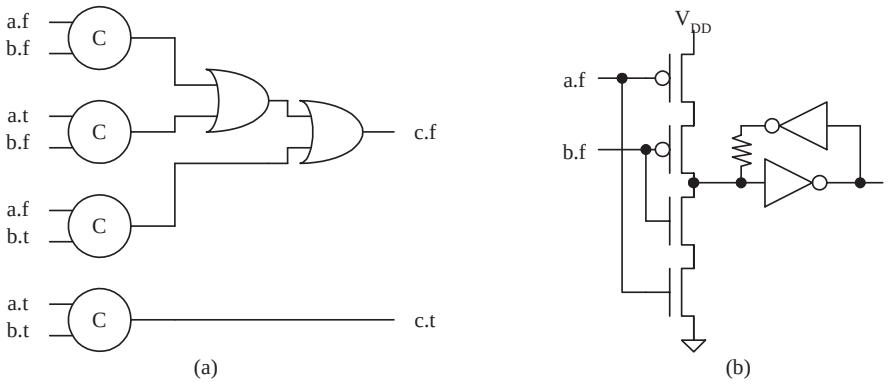


FIG. 2.3 – Schéma de la porte *And*2 asynchrone

Du fait du protocole asynchrone double rail 4-phase, les évènements à considérer sont différents de ceux présentés aux chapitre 1 (Figure 2.4). Cependant, ils correspondent au sous-groupe 2,8,10,11,12,15 des évènements 1 à 16 étudiés précédemment. Le tableau des différences entre ces évènements sur un réseau série de transistors est alors un sous-tableau 2.2 du tableau 1.1.

	E8	E10	E11	E12	E15
E2	d	b	b	c	b
E8		d	c	c	e
E10			a	c	b
E11				c	b
E12					c

TAB. 2.2 – Différence instantanée de consommation entre évènements asynchrones sur un réseau série de deux transistors

Dans l'exemple, la porte de Muller est composée d'un réseau de deux transistors NMOS en série et d'un réseau de deux transistors PMOS en série. La différence instantanée de comportement analogique des deux réseaux série de transistors NMOS et PMOS lors des évènements sur les entrées $a.f$ et $b.f$ est définie par le tableau 2.2. La superposition des deux démontre que tous les évènements asynchrones ont un comportement analogique différent entre eux sur une porte C de Muller. Cependant, 4 portes C de Muller sont nécessaires pour réaliser la fonction *And*2, chacune prenant en entrée deux rails des bits a et b. Lorsque les deux bits a et b sont valides, les 4 portes C de Muller ont toujours simultanément pour entrées les 4 évènements (2,8,9,15). L'évènement 9, bien que représentant une donnée restant à 0 est à considérer dans ce protocole. Selon le tableau 2.2, les évènements ont tous un comportement analogique différent. De même, lorsque les deux bits a et b deviennent à nouveau non valides, les 4 portes C de Muller ont toujours simultanément pour entrées les 4 évènements (9,10,11,12). Et selon le tableau 2.2 ils ont aussi tous un comportement analogique différent. La valeur logique est transportée par deux rails pour

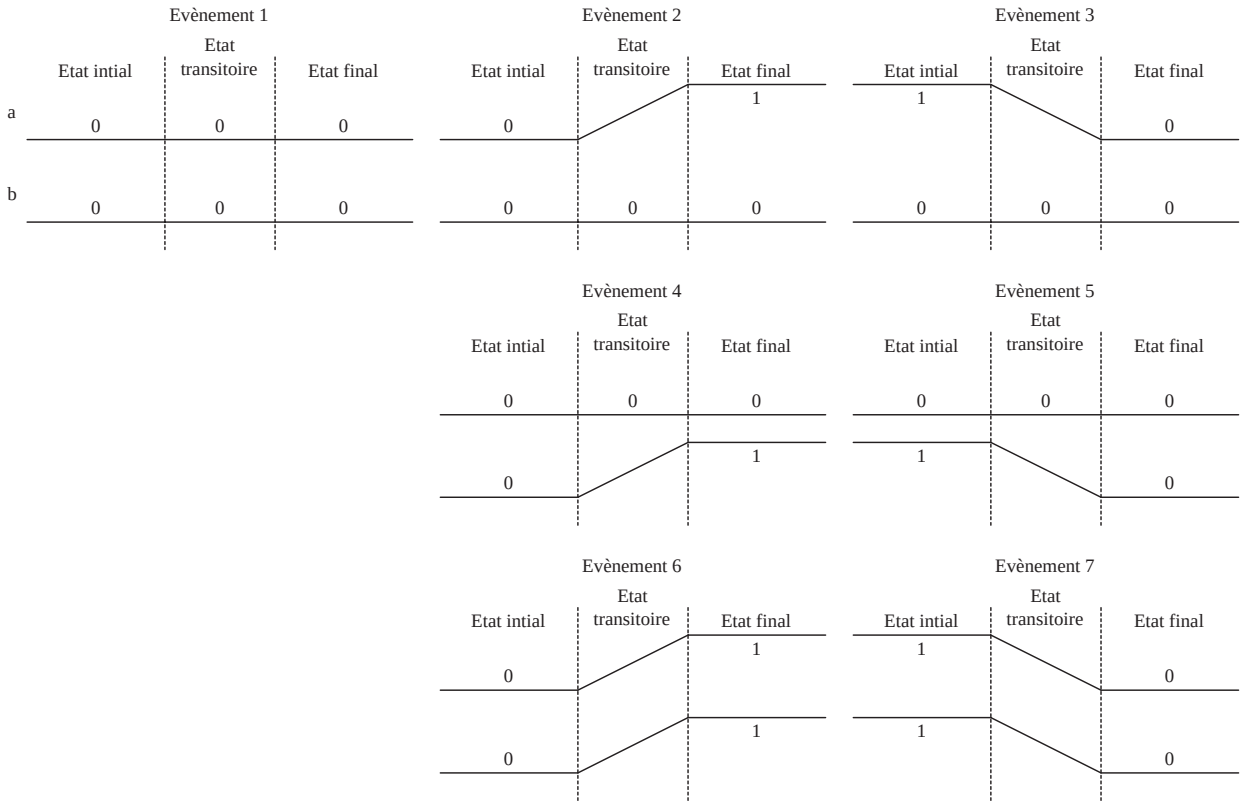


FIG. 2.4 – Évènements asynchrones

chacun des bits. Leurs différentes combinaisons sur chaque porte C-Muller est à l'origine de la dispersion instantanée des caractéristiques électriques. Par exemple, les noeuds d'interconnexion entre les transistors dont les grilles sont commandées par $a.f$ et $b.f$ sont sujets à l'effet de substrat. En conséquence, leurs tensions de seuil diffèrent instantanément. L'étage d'entrée d'une porte asynchrone composé de portes C de Muller a donc toujours un comportement analogique instantané différent qui dépend des valeurs logiques appliquées.

La logique générant les deux rail $c.t$ et $c.f$ du bit c en sortie est construite à partir de portes CMOS. Dans le cas de la porte $And2$ asynchrone il s'agit de portes $Or2$ CMOS. D'après le résultat au paragraphe 2.2, tous les évènements générés par les portes C de Muller ont tous un comportement analogique instantané différent sur une porte $Or2$ CMOS. L'étage de logique réalisant la sortie d'une porte asynchrone a toujours un comportement analogique instantané différent qui dépend des valeur logiques en entrée.

2.7 Conclusion

Toute connexion de transistors ou de réseaux de transistors en série entraîne une dispersion instantanée des caractéristiques électriques du circuit entre deux évènements. En conséquence, toutes les portes CMOS présentent une dispersion instantanée de leurs caractéristiques électriques selon les valeurs logiques. Des technologies à poids de Hamming constant construites à partir d'une structure CMOS, comme l'asynchrone, présentent aussi une dispersion instantanée de leurs caractéristiques électriques selon les valeurs logiques. Des technologies à pré-charge construites

à partir d'une structure CMOS, comme SABL, présentent aussi une dispersion instantanée de leurs caractéristiques électriques selon les valeurs logiques.

Deuxième partie

DPA

Chapitre 3

Fonctions de sélection DPA

Sommaire

3.1 Introduction

L'implémentation d'un algorithme cryptographique repose sur l'utilisation d'un microprocesseur ou d'un circuit intégré dédié. Dans tous les cas, l'exécution de l'algorithme entraîne l'activation des transistors réalisant les fonctions logiques. Les algorithmes cryptographiques sont étudiés pour résister à la cryptanalyse. Cependant, l'implémentation matérielle de ceux-ci comporte généralement des failles susceptibles d'être exploitées par d'autres types d'attaques.

Certaines attaques matérielles sont dites non intrusives parce qu'elles mettent en œuvre des techniques ne nécessitant aucune modification du circuit intégré. Lorsqu'un circuit intégré est activé, celui-ci consomme et rayonne en fonction des données manipulées. Les attaques non intrusives exploitent ces propriétés. D'autres sont dites intrusives dans la mesure où pour les mettre en œuvre il est nécessaire de procéder à des modifications du circuit intégré. Dans les deux cas, elles ont toutes pour objectif de permettre à l'attaquant de retrouver un secret contenu dans le matériel, par exemple une clé de chiffrement. Dans la suite de ce document, les attaques viseront uniquement à retrouver la clé secrète dans le cadre d'algorithmes de chiffrement par bloc.

L'attaque DPA (Differential Power Analysis) est une attaque non intrusive. La DPA correspond à une analyse différentielle de consommation. Elle utilise la dépendance entre la consommation du circuit intégré et la valeur de la clé secrète lorsque celle-ci est utilisée pendant l'exécution de l'algorithme cryptographique. Grâce à cette dépendance, la clé sera retrouvée. L'attaque DPA repose donc sur l'hypothèse physique selon laquelle le matériel consomme différemment en fonction des données manipulées.

L'étude de l'algorithme est donc primordiale pour déterminer où appliquer l'attaque. Le but est de trouver un lien mathématique entre quelques bits de clé et quelques bits de donnée qui sera appelé fonction de sélection. Les bits de donnée sont connus, ils proviennent soit des chiffrés dans le cas d'une attaque sur le dernier tour de l'algorithme, soit des clairs lorsque l'attaque porte sur le premier tour. La connaissance du lien mathématique entre les quelques bits de clé et les quelques bits de donnée permet de séparer les chiffrés ou les clairs en deux groupes. Cette séparation traduit aussi un comportement matériel différent pour chacun des deux groupes qui est visible à travers la différence de leurs moyennes de consommation.

Le fait de ne s'attaquer qu'à quelques bits de la clé permet l'étude exhaustive de toutes les valeurs de ces bits afin d'en déterminer la valeur exacte. L'attaque porte successivement sur différents paquets de bits de clé. Ceci revient à faire une exploration de celle-ci paquet par paquet. Dans le cas du DES dont la clé a une longueur de 56 bits, une attaque frontale nécessite 2^{56} essais alors que la DPA permet de déterminer 48 bits de cette clé par paquet de 6 bits. Les 8 bits restants sont ensuite calculés par recherche exhaustive. Cette approche nécessite $2^6 * 8 + 2^8 = 768$ essais. L'intérêt de cette attaque est immédiat, le nombre d'essais est très réduit en regard de la protection cryptographique assurée par la longueur de la clé.

Pour mener l'attaque, la clé de chiffrement est fixée à une valeur quelconque. L'attaquant génère N clairs aléatoires et réalise leur chiffrement avec la clé fixée pour obtenir les N chiffrés et leurs consommations respectives. Il s'attaque ensuite à chaque paquet de bits de clé dont il choisit successivement les valeurs à essayer. Pour chaque supposition il répartit les observations en deux groupes dont il calcule les moyennes de consommation respectives. Si la valeur proposée pour le paquet de bits de clé est exacte, la séparation des données est aussi exacte et celle de

leurs consommations respectives est significative. Dans ce cas, les moyennes de consommation des deux ensembles sont significativement différentes. Si la valeur du paquet de bits de clé ne correspond pas à la valeur employée lors du chiffrement, la séparation des données et celle de leurs consommations respectives est aussi erronée. Alors les deux moyennes de consommation tendent statistiquement vers la même valeur.

3.2 Définitions

3.2.1 Fonctions f

Les fonctions f considérées sont telles que tout élément x appartenant à l'ensemble A, l'image $f(x)$ appartient à l'ensemble A (Équation 3.1).

$$f : \forall x \in A,\ f(x) \in A \tag{3.1}$$

La distance maximale est définie par la plus grande distance entre tous les éléments de l'ensemble A (Équation 3.2).

$$\forall (x', x) \in (A \times A),\ d_{max} = max(d(x', x)) \tag{3.2}$$

La boule de rayon r centrée sur x est définie par l'ensemble des x' appartenant à A tels que la distance entre x' et x est égale à r (Équation 3.3).

$$B_r^x = \{x' \in A : d(x', x) = r\} \tag{3.3}$$

L'ensemble A peut être défini comme l'union de toutes les boules centrées sur l'un de ses éléments (Équation 3.4).

$$A = \{x' \in A : d(x', x) \leq d_{max}\} = B_0^x \cup B_1^x \cup ... \cup B_{d_{max}}^x \tag{3.4}$$

La fonction f est dite continue si pour tout x' appartenant à la boule unité centrée sur x, B_1^x, l'image par f de x', $f(x')$, appartient à la boule unité centrée sur $f(x)$ (Équation 3.5). Autrement dit, pour tout déplacement d'une distance unité de x, son image par f est déplacée d'une distance unité autour de $f(x)$.

$$\text{f est continue si } \forall x' \in B_1^x \text{ alors } f(x') \in B_1^{f(x)} \tag{3.5}$$

La fonction f est dite discontinue s'il existe x' appartenant à la sphère unité centrée sur x tel que son image par f de x', $f(x')$, appartient à une boule de rayon $r \neq 1$ centrée sur $f(x)$ (Équation 3.6).

$$\text{f est discontinue si } \forall x' \in B_1^x \exists! I \subset \{0, ..., d_{max}\} \text{ tel que F } = \cup B_I^{f(x)} \subset A \text{ alors } f(x') \in F \tag{3.6}$$

La fonction f est dite b-chaotique si pour tout x' appartenant à la boule unité centrée sur x, l'image par f de x', $f(x')$, appartient à l'union des boules centrées sur $f(x)$ (Équation 3.7). Tout déplacement d'une distance unité de x entraîne un déplacement potentiel de son image par f d'une distance allant de 0 à d_{max}, c'est à dire dans tout l'ensemble A.

$$\text{f est b-chaotique si } \forall x' \in B_1^x \text{ alors } f(x') \in A \tag{3.7}$$

3.2.2 Propriétés des bits des fonctions f

Dans la suite, les nombres seront exprimés en binaire $A = (a_{i-1}, ..., a_0)$, $B = (b_{j-1}, ..., b_0)$ et $C = (c_{k-1}, ..., c_0)$ et seront reliés entre eux par une fonction f telle que $f(A, B) = C$. Cette fonction sera recherchée avec des propriétés particulières.

Tout d'abord, les fonctions f intéressantes doivent avoir certaines propriétés lorsque l'opérande B est fixé. En supposant que l'opérande B soit fixé à la valeur Y, cela définit la fonction f suivante (Équation 3.8).

$$\begin{cases} B = Y \\ f(A, B = Y) = f_Y(A) = C \end{cases} \tag{3.8}$$

En parcourant l'espace des valeurs que peut prendre l'opérande A (Figure 3.1), il serait intéressant que pour chaque bit de C pris séparément, la répartition statistique entre le nombre de fois qu'il est égal à 0 et le nombre de fois qu'il est égal à 1 soit identique (Équation 3.9).

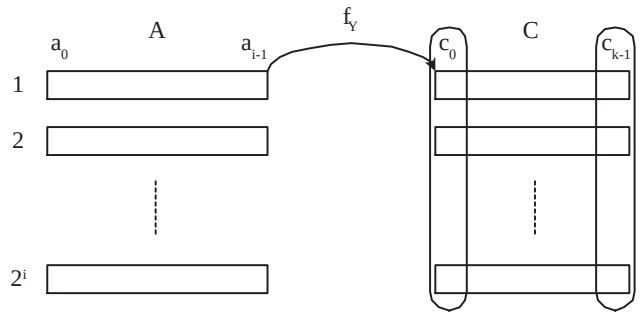


FIG. 3.1 – Propriété de f lorsque $B = Y$ sur chaque bit de C

$$\sum_{m=1}^{2^i} (c_0^m = 0) = \sum_{m=1}^{2^i} (c_0^m = 1) = ... = \sum_{m=1}^{2^i} (c_{k-1}^m = 0) = \sum_{m=1}^{2^i} (c_{k-1}^m = 1) \tag{3.9}$$

Puisque $\sum_{m=1}^{2^i} (c_0^m = 0) = \sum_{m=1}^{2^i} (c_0^m = 1) = \frac{2^i}{2}$, statistiquement la probabilité que le bit c_0 soit égal à 0 est donc égale à la probabilité que le bit c_0 soit égal à 1 (Équation 3.10).

$$p(c_0 = 0) = \frac{\sum_{m=1}^{2^i} (c_0^m = 0)}{2^i} = p(c_0 = 1) = \frac{\sum_{m=1}^{2^i} (c_0^m = 1)}{2^i} = \frac{1}{2} \tag{3.10}$$

Ce résultat reste vrai pour tous les bits de C. Lorsque l'opérande B est fixé, tous les bits de C prennent les valeurs 0 et 1 avec la même probabilité égale à $\frac{1}{2}$. Il s'agit de la première propriété recherchée pour les fonctions f (Équation 3.11).

$$\text{Pour } B = Y \text{ et } \forall A \text{ alors } p(c_0 = 0) = p(c_0 = 1) = ... = p(c_{k-1} = 0) = p(c_{k-1} = 1) = \frac{1}{2} \tag{3.11}$$

La seconde propriété recherchée est l'extension de la première propriété à tout l'espace des valeurs de l'opérande B. En parcourant l'espace des valeurs que peut prendre l'opérande B (Figure 3.2), il serait aussi intéressant que pour chaque bit de C pris séparément, la répartition statistique entre le nombre de fois qu'il est égal à 0 et le nombre de fois qu'il est égal à 1 soit identique (Équation 3.12).

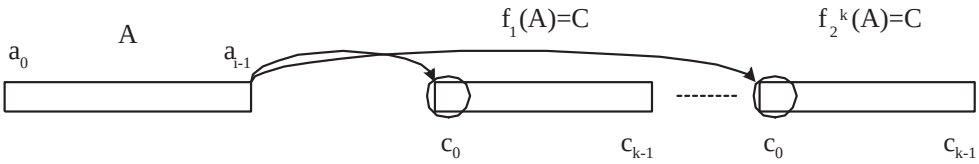


FIG. 3.2 – Propriété de f pour chaque B sur chaque bit de C

$$\sum_{n=1}^{2^k}(c_0^n=0)=\sum_{n=1}^{2^k}(c_0^n=1)=...=\sum_{n=1}^{2^k}(c_{k-1}^n=0)=\sum_{n=1}^{2^k}(c_{k-1}^n=1) \tag{3.12}$$

De même, puisque $\sum_{n=1}^{2^k}(c_0^n=0)=\sum_{n=1}^{2^k}(c_0^n=1)=\frac{2^k}{2}$, la probabilité que le bit c_0 soit égal à 0 est aussi égale à la probabilité que le bit c_0 soit égal à 1 (Équation 3.13).

$$p(c_0=0)=\frac{\sum_{n=1}^{2^k}(c_0^n=0)}{2^k}=p(c_0=1)=\frac{\sum_{n=1}^{2^k}(c_0^n=1)}{2^k}=\frac{1}{2} \tag{3.13}$$

Ce résultat est vrai pour tous les bits de C. Lorsque l'espace des valeurs de l'opérande B est parcouru, tous les bits de C prennent les valeurs 0 et 1 avec la même probabilité égale à $\frac{1}{2}$. Il s'agit de la seconde propriété recherchée pour les fonctions f (Équation 3.14).

$$\text{Pour } A \text{ fixé et } \forall B \text{ alors } p(c_0=0)=p(c_0=1)=...=p(c_{k-1}=0)=p(c_{k-1}=1)=\frac{1}{2} \tag{3.14}$$

Les fonctions f recherchées dans la suite seront des fonctions ayant la propriété définie par l'équation 3.15.

$$\forall A \text{ et } \forall B \text{ alors } p(c_0=0)=p(c_0=1)=...=p(c_{k-1}=0)=p(c_{k-1}=1)=\frac{1}{2} \tag{3.15}$$

3.2.3 Méthodes d'évaluation des fonctions f

La première méthode considérée est l'utilisation de la différence booléenne définie par l'équation 3.16. Elle permet de montrer la divergence entre deux expressions obtenues lorsqu'un bit quelconque est fixé dans un premier temps à 0 et dans un second temps à 1.

$$f_1 \triangle f_0 = f_1 \cap \overline{f_0} \cup \overline{f_1} \cap f_0 \tag{3.16}$$

Chaque bit c_p de C peut être écrit sous la forme d'une équation booléenne du type $c_p = f(a_{i-1},...,a_0,b_{j-1},...,b_0)$. Il est possible alors de définir deux équations booléennes. La première

correspond au bit c_p lorsque le bit b_n, par exemple, est égal à 0 (Équation 3.17).

$$f_0 = f(a_{i-1}, ..., a_0, b_{j-1}, ..., b_{n+1}, 0, b_{n-1}, ..., b_0) \tag{3.17}$$

La seconde équation correspond au bit c_p lorsque le bit b_n est égal à 1 (Équation 3.17).

$$f_1 = f(a_{i-1}, ..., a_0, b_{j-1}, ..., b_{n+1}, 1, b_{n-1}, ..., b_0) \tag{3.18}$$

Le résultat de la différence booléenne $f_1 \triangle f_0$ entre les fonctions f_0 et f_1 représente la divergence du bit c_p par rapport à la valeur du bit b_n traduisant la probabilité $\frac{1}{2}$. Les fonctions f qui sont recherchées divergent, c'est à dire lorsque la différence booléenne $f_1 \triangle f_0$ est définie par l'ensemble des bits de A et de B moins le bit b_n considéré (Équation 3.19).

$$f_1 \triangle f_0 = f(a_{i-1}, ..., a_0, b_{j-1}, ..., b_{n+1}, b_{n-1}, ..., b_0) \tag{3.19}$$

La seconde méthode considérée est l'analyse statistique de la fonction f. Pour la première caractéristique recherchée, le fait de fixer B à une valeur Y et que la fonction f soit déterministe fixe le calcul $f_Y(A) = C$ pour toutes les valeurs de A. Il est alors possible d'analyser le comportement d'un bit de C, c_p par exemple.

Lorsque B est égal à la valeur de référence Y et puisque le calcul est déterministe, le bit c_p sera dit calculé avec une probabilité égale à 1 (Équation 3.20).

$$\begin{cases} B = Y \text{ et } \forall A \text{ tel que } c_p = 0 \text{ alors } p(c_p = 0) = 1 \\ B = Y \text{ et } \forall A \text{ tel que } c_p = 1 \text{ alors } p(c_p = 1) = 1 \end{cases} \tag{3.20}$$

En revanche et d'après la propriété définie par l'équation 3.14, dès que la valeur de B est choisie aléatoirement et différente de la valeur de référence Y, la probabilité que le calcul du bit c_p donne le même résultat est égale à $\frac{1}{2}$ (Équation 3.21).

$$\begin{cases} B \neq Y \text{ et } \forall A \text{ tel que } c_p = 0 \text{ alors } p(c_p = 0) = \frac{1}{2} \\ B \neq Y \text{ et } \forall A \text{ tel que } c_p = 1 \text{ alors } p(c_p = 1) = \frac{1}{2} \end{cases} \tag{3.21}$$

La figure 3.3 présente la première caractéristique recherchée par analyse sur un bit quelconque c_p. Dès que la valeur de B est différente de la valeur de référence Y alors la probabilité que le calcul du bit c_p donne la même valeur est de $\frac{1}{2}$.

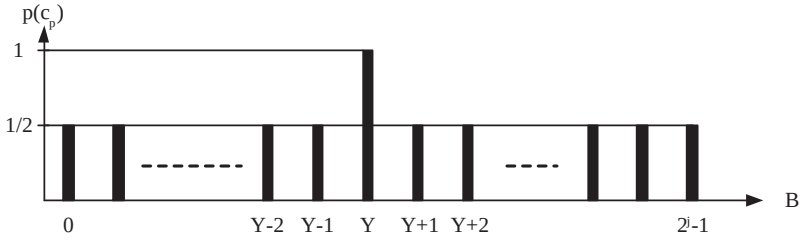


FIG. 3.3 – Probabilité que c_p avec $B \neq Y$ soit égal à c_p avec $B = Y$

La seconde caractéristique est l'extension de la première à tous les bits de C (Figure 3.4). Dès que la valeur de B est différente de la valeur de référence Y et ce quelle que soit la distance exprimée en bit, alors la probabilité que le calcul d'un bit avec $B \neq Y$ donne le même résultat que le calcul de ce même bit avec $B = Y$ est de $\frac{1}{2}$.

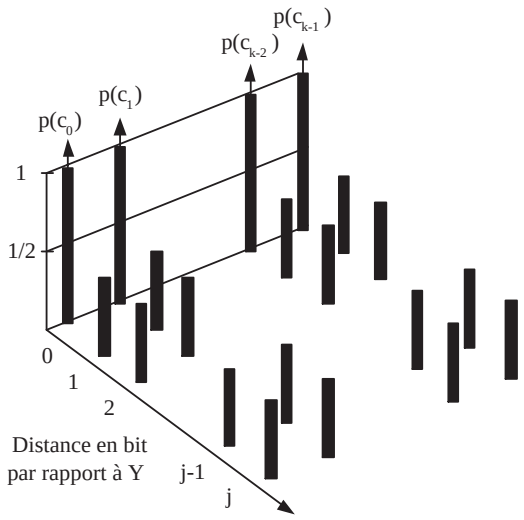


FIG. 3.4 – Probabilité pour tout bit de C que le calcul avec $B \neq Y$ soit égal au calcul avec $B = Y$

3.3 Décomposition itérative des algorithmes DES et AES

3.3.1 Approche globale d'algorithme cryptographique

Dans le cas d'un algorithme cryptographique public, le secret est défini par la connaissance de la clé. La force de l'algorithme repose premièrement sur l'espace des clés qui contient un très grand nombre de valeurs possibles pour celle-ci et deuxièmement sur l'espace des messages clairs et chiffrés. L'observation générale d'algorithmes de chiffrement permet de les noter de la façon suivante :

$$\text{chiffré}_M \quad = \quad f(clair_N, \text{clé}_P) \quad \text{où} \quad 2^M,\ 2^N \textit{ et } 2^P \textit{ sont de grands nombres} \tag{3.22}$$

M représente le nombre de bits du message chiffré, N le nombre de bits du message clair et P le nombre de bits de la clé. M, N et P sont tels que 2^M, 2^N et 2^P sont de grands nombres par rapport à la puissance de calcul disponible à cet instant. Il est possible de noter différents algorithmes de chiffrement pris dans la littérature sous cette même forme :

– 1977-DES : chiffré$_{64}$ = f($clair_{64}$,clé$_{56}$) où 2^{64} et 2^{56} sont de grands nombres,

– 2001-AES : chiffré$_{128}$ = f($clair_{128}$,clé$_{128}$) où 2^{128} est un grand nombre.

L'intérêt de représenter ces algorithmes de la sorte est de montrer qu'il semble impossible d'utiliser la clé pour calculer, y compris selon une hypothèse quelconque sur le message clair ou le message chiffré qui resterait à définir, une valeur plausible de la clé parce que l'espace des clés représente un grand nombre. De même, tenter d'exploiter un message clair ou chiffré se heurterait au même problème que pour la clé parce que l'espace des messages est aussi un grand nombre. La figure 3.5 illustre cette vision de la force d'un algorithme qui dépend à la fois de l'espace des valeurs de la clé et de l'espace des valeurs des messages.

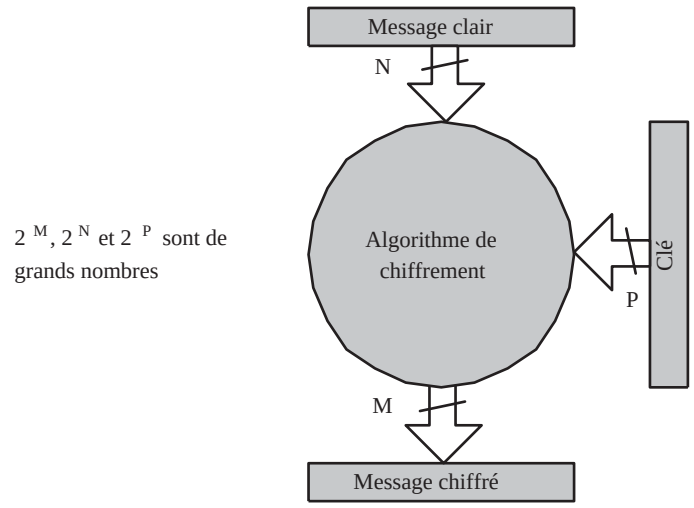


FIG. 3.5 – Vue générale de la force d'un algorithme

3.3.2 Décomposition itérative d'un algorithme cryptographique

Devant ce constat qui semble bloquant, il faut alors analyser dans le détail ces algorithmes. Et dans les faits, ils peuvent tous être réécrits sous une forme itérative :

$$\text{chiffré}_M = f(clair_N, \text{clé}_P) = \left[\begin{array}{l}\text{boucle t}\\ \quad \text{chiffré}_{m(t)}\\ \text{fin boucle t}\end{array}\right. \equiv f(clair_{n(t)}, \text{clé}_{p(t)}) \tag{3.23}$$

où $2^{m(t)} << 2^M$, $2^{n(t)} << 2^N$ et $2^{p(t)} << 2^P$ et $2^{m(t)}$, $2^{n(t)}$ et $2^{p(t)}$ ne sont plus de grands nombres et t représente une étape de l'algorithme.
La figure 3.6 illustre cette nouvelle vision de la "force" d'un algorithme. (a) peut représenter le premier tour d'un algorithme et (b) le dernier tour. De par la construction cryptographique des algorithmes, la DPA pourra être appliquée dans ces deux cas. Une variante de l'attaque sera aussi exposée et (c) peut la représenter. Il ne faut pas oublier que ces schémas sont non limitatifs et que des corrélations entre ces fonctions peuvent aussi servir à mettre en œuvre l'attaque en combinant différents tours de l'algorithme avec différentes fonctions [28]. Il est alors possible de noter à nouveau les différents algorithmes de chiffrement cités précédemment.

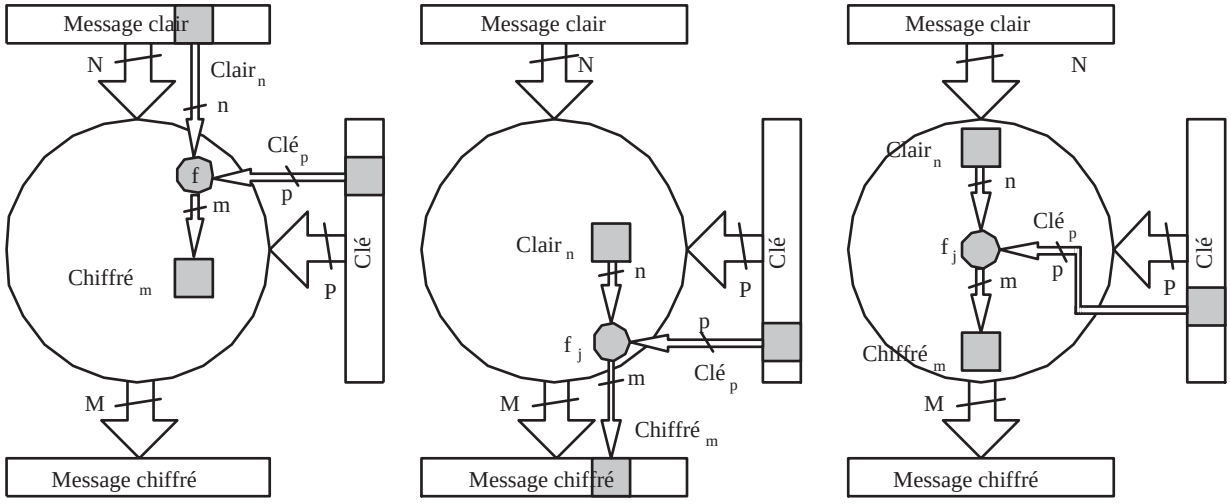


FIG. 3.6 – Partitionnement itératif d'un algorithme cryptographique

3.3.3 1977-DES

L'algorithme DES [33] peut être représenté par les deux formes suivantes. La seconde est une itération de la première et apparaît plus convaincante mais toutes deux illustrent bien le fait que les valeurs intermédiaires ne sont plus de grands nombres.

$$\text{chiffré}_{64} = f(clair_{64}, \text{clé}_{56}) = \left[\begin{array}{l}\text{boucle } t_1 \text{ 1 à 16} \\ \quad \text{chiffré}_{32(t_1)} \\ \text{fin boucle } t_1\end{array}\right. \equiv f(clair_{32(t_1)}, \text{clé}_{48(t_1)}) \tag{3.24}$$

où t_1 représente un tour de l'algorithme.

$$\text{chiffré}_{64} = f(clair_{64}, \text{clé}_{56}) = \left[\begin{array}{l}\text{boucle } t_1 \text{ 1 à 16} \\ \quad \text{boucle } t_2 \text{ 1 à 8} \\ \quad [\,] \ \text{chiffré}_{4(t1,t2)} \\ \quad \text{fin boucle } t_2 \\ \text{fin boucle } t_1\end{array}\right. \equiv f(clair_{6(t1,t2)}, \text{clé}_{6(t1,t2)}) \tag{3.25}$$

où t_1 représente toujours un tour de l'algorithme et t_2 l'exécution séquentielle des 8 fonctions de substitution.

Il est possible de généraliser l'écriture en disant que le message chiffré de 64 bits par le DES est une fonction du message chiffré intermédiaire sur 32 bits par le DES qui est lui même une fonction itérée du message clair intermédiaire sur 6 bits et de la sous-clé sur 6 bits.

$\text{chiffré}_{64} = f(clair_{64}, \text{clé}_{56}) \equiv f_1(clair_{32(t_1)}, \text{clé}_{48(t_1)}) \equiv f2(clair_{6(t_1,t_2)}, \text{clé}_{6(t_1,t_2)})$

3.3.4 2001-AES

L'algorithme AES [34, 37, 11, 12] peut aussi être représenté sous la forme suivante où t_1 représente un tour de l'algorithme.

$$\text{chiffré}_{128} = f(clair_{128}, \text{clé}_{128/192/256}) = \left[\begin{array}{l}\text{boucle } t_1 \text{ 1 à 10/12/14} \\ \quad \text{chiffré}_{8(t_1)} \\ \text{fin boucle } t_1\end{array}\right. \equiv f(clair_{8(t_1)}, \text{clé}_{8(t_1)}) \tag{3.26}$$

La forme généralisée devient alors : $\text{chiffré}_{128} = f(clair_{128}, \text{clé}_{128/192/256}) \equiv f_1(clair_{8(t_1)}, \text{clé}_{8(t_1)})$
L'intérêt de représenter ces algorithmes sous cette nouvelle forme est de montrer qu'il est tout à fait possible de calculer toutes les solutions possibles pour un sous-ensemble restreint de p bits de la clé, de n bits du message clair et de m bits du message chiffré au niveau d'une étape de l'algorithme. Ceci sera exploité pour concevoir les fonctions de sélection DPA.

3.4 Fonction de sélection DPA

La possibilité de calculer l'ensemble des solutions de l'équation $\text{chiffré}_{m(t)} = f(clair_{n(t)}, \text{clé}_{p(t)})$ pour une étape de l'algorithme amène à la notion de fonction discriminante. En effet, sur l'ensemble des valeurs des clés qui peuvent être définies sur 2^p bits, seule une et une seule valeur est la " vraie " valeur égale à celle utilisée lors du chiffrement, toutes les autres sont fausses c'est à dire différentes. Il faut donc analyser ce qui est discriminant du point de vue du calcul lorsque

les p bits de clé sont égaux aux bits de clé. Il est alors nécessaire de regarder les propriétés statistiques des fonctions $f(clair_{n(t)}, \text{clé}_{p(t)})$ de différents algorithmes cryptographiques. Evidemment, ces fonctions ne sont pas choisies au hasard et découlent du fait qu'elles ont pour paramètres n bits du message clair et p bits de la clé pour donner m bits de message chiffré.

3.4.1 Fonction f de l'algorithme DES

En s'appuyant sur l'équation C.1 et la figure C.5 de l'annexe C, qui détaillent respectivement le tour suivant en fonction du tour précédent et la fonction f de l'algorithme DES, il est possible d'exprimer une relation entre un bit de R_{16} et un bit de L_{15}, 6 bits de R_{15} et 6 bits de K_{16}. Par exemple, sur le dernier tour de l'algorithme, le bit $R_{16}(0)$ peut s'écrire sous la forme de l'équation 3.27 dont les détails de construction sont donnés au chapitre 4.

$$R_{16}(0) = P(S_4(E(R_{15}(12:17)) \oplus K_{16}(18:23)))(0) \oplus L_{15}(0) \tag{3.27}$$

En posant $clair_7$ pour $L_{15}(0)$ et $R_{15}(12:17)$, chiffré$_1$ pour $R_{16}(0)$ et clé$_6$ pour $K_{16}(18:23)$, l'équation 3.27 peut être exprimée sous la forme de l'équation 3.28.

$$\text{chiffré}_1 = f(\text{clair}_7, \text{clé}_6) \tag{3.28}$$

Cette relation reste vraie pour tous les tours de l'algorithme entre quelques bits de message et quelques bits de la sous-clé à un instant t. Elle permet le calcul de l'ensemble des solutions en omettant le paramètre t. Il reste à analyser les propriétés statistiques de cette équation qui découlent de la construction des fonctions de substitution. En effet, la probabilité que chacun des 4 bits de sortie pris séparément valent 1 est égale à $\frac{1}{2}$.

Ceci peut se démontrer en montrant que l'effet de la variation d'un bit de clé sur le bit à examiner dépend de la totalité des autres bit de clé du bloc exploré à l'aide de la définition des fonctions de substitution données par le standard et dont un exemple est décrit dans le tableau C.3. Pour cela, il est possible d'analyser, par exemple, le comportement du bit 3 de la fonction de substitution 4, S4(3), lors du tour 1 de l'algorithme. Le standard dit qu'il peut être exprimé ainsi :

$$\begin{aligned} S4(3) = \; & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.\overline{b_2}.b_3.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4)\overline{b_0}.\overline{b_5} + \\ & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & \overline{b_1}.b_2.b_3.b_4 + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4)\overline{b_0}.b_5 + \\ & (\overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \overline{b_1}.b_2.b_3.b_4 + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.b_2.\overline{b_3}.\overline{b_4})b_0.\overline{b_5} + \\ & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.b_4)b_0.b_5 \end{aligned} \tag{3.29}$$

Toujours selon le standard, les bits b_0, b_1, b_2, b_3, b_4 et b_5 sont calculés à partir des valeurs de la sous clé $K_1(18:23)$ et du message clair intermédiaire $R_0(12:17)$. Le bit observé S4(3) ne dépend pas des autres bits de clé au tour 1 de l'algorithme.

$$\begin{aligned} b_0 &= K_1(18) \oplus R_0(12) \\ b_1 &= K_1(19) \oplus R_0(13) \\ b_2 &= K_1(20) \oplus R_0(14) \\ b_3 &= K_1(21) \oplus R_0(15) \\ b_4 &= K_1(22) \oplus R_0(16) \\ b_5 &= K_1(23) \oplus R_0(17) \end{aligned} \tag{3.30}$$

La fonction f_0 représentant le bit observé S4(3) peut désormais être calculée lorsque le bit de clé $K_1(18)$ est égal à 0. Les bits b_0, b_1, b_2, b_3, b_4 et b_5 sont alors définis par les équations suivantes.

$$\begin{aligned} b_0 &= R_0(12) \\ b_1 &= K_1(19) \oplus R_0(13) \\ b_2 &= K_1(20) \oplus R_0(14) \\ b_3 &= K_1(21) \oplus R_0(15) \\ b_4 &= K_1(22) \oplus R_0(16) \\ b_5 &= K_1(23) \oplus R_0(17) \end{aligned} \tag{3.31}$$

f0 est alors définie par l'équation :

$$\begin{aligned} f_0 = \; & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.\overline{b_2}.b_3.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4)\overline{R_0(12)}.\overline{b_5} + \\ & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & \overline{b_1}.b_2.b_3.b_4 + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4)\overline{R_0(12)}.b_5 + \\ & (\overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \overline{b_1}.b_2.b_3.b_4 + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.b_2.\overline{b_3}.\overline{b_4})R_0(12).\overline{b_5} + \\ & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.b_4)R_0(12).b_5 \end{aligned} \tag{3.32}$$

La fonction f_1, qui représente S4(3) lorsque le bit de clé $K_1(18)$ est égal à 1, peut être calculée de la même façon. Les bits b_0, b_1, b_2, b_3, b_4 et b_5 sont alors définis par les équations suivantes :

$$\begin{aligned} b_0 &= \overline{R_0(12)} \\ b_1 &= K_1(19) \oplus R_0(13) \\ b_2 &= K_1(20) \oplus R_0(14) \\ b_3 &= K_1(21) \oplus R_0(15) \\ b_4 &= K_1(22) \oplus R_0(16) \\ b_5 &= K_1(23) \oplus R_0(17) \end{aligned} \tag{3.33}$$

f_1 est donc définie par l'équation suivante :

$$\begin{aligned} f_1 = \; & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.\overline{b_2}.b_3.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4)R_0(12).\overline{b_5} + \\ & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & \overline{b_1}.b_2.b_3.b_4 + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4)R_0(12).b_5 + \\ & (\overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \overline{b_1}.b_2.b_3.b_4 + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.b_2.\overline{b_3}.\overline{b_4})\overline{R_0(12)}.\overline{b_5} + \\ & (\overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.b_4)\overline{R_0(12)}.b_5 \end{aligned} \tag{3.34}$$

Afin de simplier l'écriture, les 4 termes suivants sont ainsi définis :

$$\begin{array}{rl} A = & \overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.\overline{b_2}.b_3.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4 \\ B = & \overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & \overline{b_1}.b_2.b_3.b_4 + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.b_2.\overline{b_3}.\overline{b_4} + b_1.b_2.b_3.b_4 \\ C = & \overline{b_1}.\overline{b_2}.b_3.\overline{b_4} + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \overline{b_1}.b_2.b_3.b_4 + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.\overline{b_3}.b_4 + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.b_2.\overline{b_3}.\overline{b_4} \\ D = & \overline{b_1}.\overline{b_2}.\overline{b_3}.\overline{b_4} + \overline{b_1}.\overline{b_2}.\overline{b_3}.b_4 + \overline{b_1}.b_2.\overline{b_3}.b_4 + \overline{b_1}.b_2.b_3.\overline{b_4} + \\ & b_1.\overline{b_2}.\overline{b_3}.\overline{b_4} + b_1.\overline{b_2}.b_3.\overline{b_4} + b_1.\overline{b_2}.b_3.b_4 + b_1.b_2.\overline{b_3}.b_4 \end{array} \quad (3.35)$$

Ceci permet de réécrire plus simplement les fonctions f_0 et f_1 :

$$\begin{array}{l} f_0 = A.\overline{R_0(12)}.\overline{b_5} + B.\overline{R_0(12)}.b_5 + C.R_0(12).\overline{b_5} + D.R_0(12).b_5 \\ f_1 = A.R_0(12).\overline{b_5} + B.R_0(12).b_5 + C.\overline{R_0(12)}.\overline{b_5} + D.\overline{R_0(12)}.b_5 \end{array} \quad (3.36)$$

Le calcul de la différence booléenne (Équation 3.16) de ces deux fonctions et après simplification donne l'équation suivante :

$$f_1 \triangle f_0 = \overline{b_5}(A \oplus C) + b_5(B \oplus D) \quad (3.37)$$

L'introduction des termes de A, B, C et D donne le résultat :

$$f_1 \triangle f_0 = \overline{b_5}(\overline{b_1}.b_2 + \overline{b_1}.b_4 + \overline{b_2}.b_4 + \overline{b_2}.b_3 + b_3 b_4) + b_5(b_1\overline{b_2} + b_1\overline{b_3} + b_1 b_4 + \overline{b_1} b_3 + \overline{b_2} b_4) \quad (3.38)$$

Et en introduisant les valeurs des paramètres b_1, b_2, b_3, b_4 et b_5 le résultat devient :

$$\begin{array}{rl} f_1 \triangle f_0 = & or(\\ & and(K_1(20), R_0(13), R_0(17), R_0(14), K_1(19), K_1(23)), \\ & and(R_0(13), not(K_1(19)), not(R_0(17)), R_0(15), K_1(21), K_1(23)), \\ & and(R_0(13), R_0(17), not(K_1(19)), not(K_1(23)), R_0(15), K_1(21)), \\ & and(K_1(20), not(K_1(19)), not(R_0(13)), not(K_1(23)), not(R_0(17)), R_0(14)), \\ & and(not(K_1(19)), not(R_0(13)), not(K_1(20)), not(R_0(14)), not(K_1(23)), not(R_0(17))), \\ & and(R_0(13), not(K_1(19)), not(R_0(16)), not(R_0(17)), K_1(22), K_1(23)), \\ & and(R_0(13), R_0(17), not(K_1(19)), not(R_0(16)), not(K_1(23)), K_1(22)), \\ & and(R_0(17), not(R_0(13)), not(R_0(16)), not(K_1(23)), K_1(19), K_1(22)), \\ & and(R_0(13), R_0(16), not(K_1(19)), not(K_1(22)), not(R_0(17)), K_1(23)), \\ & and(R_0(13), R_0(16), R_0(17), not(K_1(19)), not(K_1(22)), not(K_1(23))), \\ & and(R_0(16), not(R_0(13)), not(K_1(22)), not(R_0(17)), K_1(19), K_1(23)), \\ & and(K_1(20), R_0(17), not(K_1(19)), not(R_0(13)), R_0(14), K_1(23)), \\ & and(K_1(20), R_0(13), not(K_1(23)), not(R_0(17)), R_0(14), K_1(19)), \\ & and(R_0(17), not(K_1(19)), not(R_0(13)), not(R_0(16)), K_1(22), K_1(23)), \\ & and(R_0(13), not(K_1(20)), not(R_0(14)), not(K_1(23)), not(R_0(17)), K_1(19)), \end{array}$$

$and(not(R_0(13)), not(R_0(16)), not(R_0(17)), K_1(19), K_1(22), K_1(23)),$
$and(R_0(13), R_0(17), not(K_1(20)), not(R_0(14)), K_1(19), K_1(23)),$
$and(R_0(17), not(K_1(19)), not(R_0(13)), not(K_1(20)), not(R_0(14)), K_1(23)),$
$and(not(R_0(13)), not(R_0(17)), K_1(19), R_0(15), K_1(21), K_1(23)),$
$and(R_0(17), not(R_0(13)), not(K_1(23)), K_1(19), R_0(15), K_1(21)),$
$and(R_0(13), not(K_1(19)), not(K_1(20)), not(R_0(14)), not(R_0(17)), K_1(23)),$
$and(R_0(13), R_0(17), not(K_1(19)), not(K_1(20)), not(R_0(14)), not(K_1(23))),$
$and(K_1(20), not(R_0(13)), not(R_0(17)), R_0(14), K_1(19), K_1(23)),$
$and(K_1(20), R_0(17), not(R_0(13)), not(K_1(23)), R_0(14), K_1(19)),$
$and(K_1(20), R_0(13), not(K_1(19)), not(R_0(17)), R_0(14), K_1(23)),$
$and(K_1(20), R_0(13), R_0(17), not(K_1(19)), not(K_1(23)), R_0(14)),$
$and(not(R_0(13)), not(K_1(20)), not(R_0(14)), not(R_0(17)), K_1(19), K_1(23)),$
$and(R_0(17), not(R_0(13)), not(K_1(20)), not(R_0(14)), not(K_1(23)), K_1(19)),$
$and(R_0(13), not(K_1(21)), not(R_0(17)), K_1(19), R_0(15), K_1(23)),$
$and(R_0(13), R_0(17), not(K_1(21)), not(K_1(23)), K_1(19), R_0(15)),$
$and(not(K_1(19)), not(R_0(13)), not(R_0(16)), not(K_1(23)), not(R_0(17)), K_1(22)),$
$and(R_0(13), R_0(16), not(K_1(22)), not(K_1(23)), not(R_0(17)), K_1(19)),$
$and(R_0(13), not(R_0(15)), not(R_0(17)), K_1(19), K_1(21), K_1(23)),$
$and(R_0(13), R_0(16), R_0(17), not(K_1(22)), K_1(19), K_1(23)),$
$and(R_0(16), not(K_1(19)), not(R_0(13)), not(K_1(22)), not(K_1(23)), not(R_0(17))),$
$and(R_0(16), R_0(17), not(K_1(19)), not(R_0(13)), not(K_1(22)), K_1(23)),$
$and(R_0(13), not(R_0(16)), not(K_1(23)), not(R_0(17)), K_1(19), K_1(22)),$
$and(R_0(13), R_0(17), not(R_0(16)), K_1(19), K_1(22), K_1(23)),$
$and(R_0(13), R_0(17), not(R_0(15)), not(K_1(23)), K_1(19), K_1(21)),$
$and(not(K_1(19)), not(R_0(13)), not(K_1(21)), not(R_0(17)), R_0(15), K_1(23)),$
$and(R_0(17), not(K_1(19)), not(R_0(13)), not(K_1(21)), not(K_1(23)), R_0(15)),$
$and(not(K_1(19)), not(R_0(13)), not(R_0(15)), not(R_0(17)), K_1(21), K_1(23)),$
$and(R_0(17), not(K_1(19)), not(R_0(13)), not(R_0(15)), not(K_1(23)), K_1(21)),$
$and(R_0(13), R_0(17), not(K_1(19)), not(K_1(21)), not(R_0(15)), not(K_1(23))),$
$and(not(R_0(13)), not(K_1(21)), not(R_0(15)), not(R_0(17)), K_1(19), K_1(23)),$
$and(R_0(17), not(R_0(13)), not(K_1(21)), not(R_0(15)), not(K_1(23)), K_1(19)),$
$and(R_0(16), R_0(17), not(R_0(13)), not(K_1(22)), not(K_1(23)), K_1(19)),$
$and(R_0(13), not(K_1(19)), not(K_1(21)), not(R_0(15)), not(R_0(17)), K_1(23)),$
$and(R_0(16), not(K_1(21)), not(K_1(22)), R_0(15)),$
$and(R_0(16), not(R_0(15)), not(K_1(22)), K_1(21)),$
$and(not(K_1(21)), not(R_0(16)), R_0(15), K_1(22)),$
$and(not(R_0(15)), not(R_0(16)), K_1(21), K_1(22)),$
$and(K_1(20), not(K_1(21)), R_0(14), R_0(15)),$
$and(not(K_1(20)), not(R_0(14)), not(K_1(21)), R_0(15)),$

$$
\begin{aligned}
&and(K_1(20), not(R_0(15)), R_0(14), K_1(21)),\\
&and(not(K_1(20)), not(R_0(14)), not(R_0(15)), K_1(21)),\\
&and(K_1(20), R_0(16), not(K_1(22)), R_0(14)),\\
&and(R_0(16), not(K_1(20)), not(R_0(14)), not(K_1(22))),\\
&and(K_1(20), not(R_0(16)), R_0(14), K_1(22)),\\
&and(not(K_1(20)), not(R_0(14)), not(R_0(16)), K_1(22))\\
)
\end{aligned}
\tag{3.39}
$$

S4(3) est la fonction booléenne représentant le bit observé pour la séparation DPA qui dépend de 6 bits de clairs et de 6 bits de clé. f_0 et f_1 sont les fonctions booléennes représentant ce même bit observé lorsque le bit de clé $K_1(18)$ est respectivement égal à 0 et à 1. La différence booléenne des fonctions f_0 et f_1 met particulièrement en évidence l'aspect divergent de la fonction S4(3) en fonction d'un bit de clé. En effet, Le résultat de la différence booléenne est une fonction " ou " entre 60 termes qui sont eux-mêmes une fonction " et " dont 12 ont 4 variables en entrée et 48 ont 6 variables en entrée. Hormis le bit de clé $K_1(18)$ et le bit de clair $R_0(12)$ qui lui est associé, la différence booléenne dépend de tous les autres bits de clé $K_1(19:23)$ et de tous les autres bits de clairs $R_0(13:17)$. De plus, la forme canonique de l'expression précédente n'est pas réductible. Il est possible d'exprimer l'importance des bits des opérandes en fonction de leur profondeur dans l'expression canonique de la fonction. Par exemple, les statistiques d'occurrence des bits permettent d'apprécier cette importance (Tableau 3.1).

$K_1(18)$	0	$R_0(12)$	0
$\overline{K_1(18)}$	0	$\overline{R_0(12)}$	0
$K_1(19)$	24	$R_0(13)$	24
$\overline{K_1(19)}$	24	$\overline{R_0(13)}$	24
$K_1(20)$	12	$R_0(14)$	12
$\overline{K_1(20)}$	12	$\overline{R_0(14)}$	12
$K_1(21)$	12	$R_0(15)$	12
$\overline{K_1(21)}$	12	$\overline{R_0(15)}$	12
$K_1(22)$	12	$R_0(16)$	12
$\overline{K_1(22)}$	12	$\overline{R_0(16)}$	12
$K_1(23)$	24	$R_0(17)$	24
$\overline{K_1(23)}$	24	$\overline{R_0(17)}$	24

TAB. 3.1 – Statistique d'occurrence des bits des opérandes

L'analyse des fonctions " et " à 6 entrées montre qu'elles sont toutes dépendantes de 3 bits de clé différents. Les fonctions " et " à 4 entrées sont toutes dépendantes de 2 bits de clé différents. Cela revient à dire que le changement d'un bit de clé produit une perturbation de la valeur du bit à observer qui dépend de la valeur de tous les autres bits de clé, donc avec une variabilité maximale. Il n'y a aucune continuité ou progressivité entre la bonne valeur des bits de clé et une autre distante d'un ou plusieurs bits. Ceci permet d'écrire les probabilités suivantes :

$$
\left\{
\begin{array}{lcl}
p(\text{chiffré}_1 = 1) & = & p(f(clair_7, \text{clé}_6) = 1) \simeq \frac{1}{2} \quad \forall\, clair_7 \quad \forall\, \text{clé}_6\\
p(\text{chiffré}_1 = 0) & = & p(f(clair_7, \text{clé}_6) = 0) \simeq \frac{1}{2} \quad \forall\, clair_7 \quad \forall\, \text{clé}_6
\end{array}
\right.
\tag{3.40}
$$

Etant donné que les algorithmes de chiffrement sont des bijections, les fonctions f ainsi définies sont aussi bijectives : $clair_7 = g(\text{chiffré}_1, \text{clé}_6)$. Il est aussi possible d'écrire les probabilités suivantes pour chaque bit de $clair_7(i)$:

$$\left\{ \begin{array}{rcccl} p(clair_7(i)=1) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=1) & \simeq & \frac{1}{2} \quad \forall \text{ chiffré}_1 \quad \forall \text{ clé}_6 \\ p(clair_7(i)=0) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=0) & \simeq & \frac{1}{2} \quad \forall \text{ chiffré}_1 \quad \forall \text{ clé}_6 \end{array} \right. \tag{3.41}$$

Pout une valeur v fixée de clé_6, le résultat du calcul du bit $clair_7(i)$ est obtenu avec la probabilité 1.

Pour tout chiffré_1 qui avec $\text{clé}_6 = v$ donne $clair_7(i) = 1$ c'est avec les probabilités suivantes :

$$\left\{ \begin{array}{rcccl} p(clair_7(i)=1) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=1) & = & 1 \quad \forall \text{ chiffré}_1 \quad \text{pour clé}_6 = \text{v} \\ p(clair_7(i)=0) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=0) & = & 0 \quad \forall \text{ chiffré}_1 \quad \text{pour clé}_6 = \text{v} \end{array} \right. \tag{3.42}$$

Et pour tout chiffré_1 qui avec $\text{clé}_6 = v$ donne $clair_7(i) = 0$ c'est avec les probabilités :

$$\left\{ \begin{array}{rcccl} p(clair_7(i)=1) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=1) & = & 0 \quad \forall \text{ chiffré}_1 \quad \text{pour clé}_6 = \text{v} \\ p(clair_7(i)=0) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=0) & = & 1 \quad \forall \text{ chiffré}_1 \quad \text{pour clé}_6 = \text{v} \end{array} \right. \tag{3.43}$$

Pour toute autre valeur $\text{clé}_6 \neq v$ et ce quelle que soit la distance en bit nous avons les probabilités :

$$\left\{ \begin{array}{rcccl} p(clair_7(i)=1) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=1) & \simeq & \frac{1}{2} \quad \forall \text{ chiffré}_1 \quad \text{pour clé}_6 \neq \text{v} \\ p(clair_7(i)=0) & = & p(g(\text{chiffré}_1,\text{clé}_6)(i)=0) & \simeq & \frac{1}{2} \quad \forall \text{ chiffré}_1 \quad \text{pour clé}_6 \neq \text{v} \end{array} \right. \tag{3.44}$$

Sur le dernier tour de l'algorithme DES, chiffré_1 est connu avec certitude. La seule inconnue est clé_6. En parcourant l'ensemble les 64 valeurs possibles de clé_6 et en tenant compte des probabilités définies précédemment, lorsque clé_6 représente la vraie valeur des 6 bits de la sous-clé, alors pour tout chiffré_1 le calcul de $clair_7(i)$ est réalisé avec la probabilité 1. En revanche, dès que clé_6 n'est pas la bonne valeur des 6 bits de la sous-clé alors le calcul de $clair_7(i)$ est réalisé avec une probabilité proche de $\frac{1}{2}$.

En résumé, lorsque clé_6 représente la vraie valeur des 6 bits de la sous-clé, les chiffrés peuvent être classés dans l'ensemble E1 selon le critère $clair_7(i) = 1$ avec la probabilité 1 et dans l'ensemble E0 selon le critère $clair_7(i) = 0$ avec la probabilité 1. Les ensembles E1 et E0 sont bien distincts selon le critère $clair_7(i)$. En revanche, dès que clé_6 représente une fausse valeur des 6 bits de la sous-clé, les chiffrés sont classés dans l'ensemble E1 selon le critère $clair_7(i) = 1$ avec la probabilité $\frac{1}{2}$ et dans l'ensemble E0 selon le critère $clair_7(i) = 0$ avec la probabilité $\frac{1}{2}$. Dans ce cas, la répartition des chiffrés est équiprobable entre les ensembles E1 et E0 qui ne sont plus distincts selon le critère $clair_7(i)$. La fonction f de la relation est donc discriminante.

Un exemple de fonction de sélection discriminante pour l'algorithme DES a été présenté. Il est tout à fait possible d'étendre cette approche à une relation plus générale tant sur le nombre de bits du chiffré, du clair et de la clé que sur des instants différents correspondant à différentes étapes de l'algorithme.

3.4.2 Fonction f de l'algorithme AES

Dans le cas de l'algorithme AES, la fonction f a cette fois-ci pour paramètres 8 bits de message intermédiaire et 8 bits de sous-clé. Les détails de construction de cette relation sont donnés au chapitre 5. Ici aussi, elle inclut la fonction de substitution définie par le standard. La fonction de substitution prend en entrée 1 octet du message clair intermédiaire. Le résultat de la fonction de substitution est combiné bit à bit par un " ou exclusif " avec 1 octet de la sous-clé correspondante du tour en cours. Il est donc possible d'avoir une relation entre un bit de message chiffré avec 1 octet de message intermédiaire et 1 octet de sous-clé comme indiqué par la relation 3.45. Ceci permet à nouveau le calcul de l'ensemble des solutions.

$$\text{chiffré}_{1(t)} = f(clair_{8(t)}, \text{clé}_{8(t)}) \tag{3.45}$$

Comme pour l'algorithme DES, il est possible d'inverser le relation 3.45 et de trouver une équation entre un bit de message intermédiaire $clair_{1(t)}$, un octet de chiffré $\text{chiffré}_{8(t)}$ et un octet de clé $\text{clé}_{8(t)}$ à un instant t de l'algorithme (Équation 3.46). Ce bit dépend toujours de la fonction de substitution de l'algorithme AES.

$$clair_{1(t)} = g(\text{chiffré}_{8(t)}, \text{clé}_{8(t)}) \tag{3.46}$$

A nouveau, il faut analyser l'influence de la modification d'un bit de clé sur la valeur calculée $\text{chiffré}_{1(t)}$. Cette analyse commence par la séparation en ensembles E0 et E1 réalisée pour chaque bit Mi (i , 0 : 7) de l'octet du message intermédiaire. Pour chaque bit Mi, l'octet de clé de chiffrement est tout d'abord fixé à la valeur 0 et les octets de chiffrés sont parcourus de x00 à xFF. Les bits de chiffrés sont combinés bit à bit par un " xor " avec les bits de l'octet de clé afin de retrouver la sortie de la fonction de substitution SubBytes. L'image de la sortie de SubBytes est inversée pour retrouver l'entrée. Selon la valeur du bit Mi, le chiffré est placé soit dans E0 si celui-ci est égal à 0 soit dans E1 s'il est égal à 1. Cette séparation est réitérée pour tous les octets possibles de chiffrement de x00 à xFF afin de prendre en compte toutes les distances en bit entre les octets de clé. ceci définit alors alors toutes les séparations possibles sur la fonction de substitution SubBytes. La figure 3.7 illustre les séparations.

Il est maintenant possible d'analyser l'impact de l'octet de clé considéré lorsqu'il n'est pas l'octet de chiffrement. Par exemple, il est intéressant de compter le nombre de chiffrés communs entre un fichier E0 défini par un bit Mi et une valeur d'octet de chiffrement Kj et tous les autres fichiers E0 définis par le même bit Mi et par toutes les autres valeurs d'octet de chiffrement.

Un premier résultat montre que, quel que soit le bit de séparation Mi et quel que soit l'octet Kj de chiffrement considéré, la moyenne du nombre d'octets communs entre les ensembles E0 définis par Mi et Kj, m(i,j,0), est de 64. Le minimum de chiffrés communs, min(i,j,0), est 56 et le maximum, max(i,j,0), est 72. Seul le bon octet de clé de chiffrement donne un résultat de 128 chiffrés communs. La figure 3.8 montre en exemple le résultat pour l'octet de chiffrement x80. Le résultat pour les ensembles E1 est identique, il découle du précédent.

$$\begin{gathered}\text{Pour Mi (i : 0 à 7) fixé, Kj (j : 0 à 255) et pour Ek (k : 0 à 1) fixé alors} \\ moy(i,j,k) = 64 \\ max(i,j,k) = 72 \\ min(i,j,k) = 56\end{gathered} \tag{3.47}$$

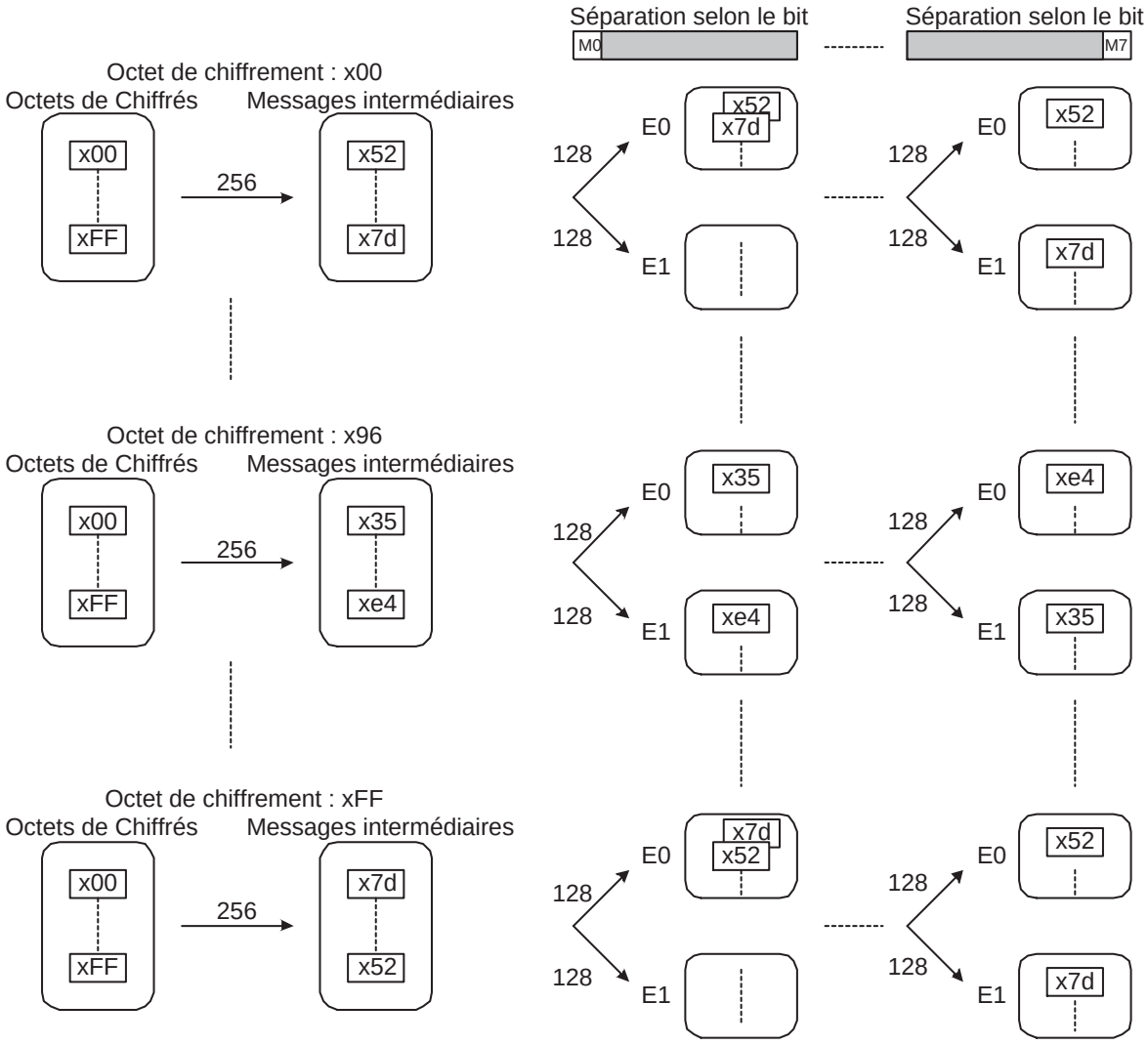


FIG. 3.7 – Analyse statistique d'une fonction f de l'algorithme AES

Chaque valeur de l'octet de chiffrement peut être considérée comme le sommet d'un hyper-cube. Il est possible de définir une distance entre ces sommets comme étant le nombre de bits différents entre eux. Pour un octet de chiffrement considéré " vrai ", tous les autres octets de chiffrement dits " faux " seront classés selon leur distance par rapport à l'octet de chiffrement " vrai ". Une distance de 0 indique que l'octet est l'octet de chiffrement " vrai ", une distance de 1 regroupera tous les octets ayant un seul bit de différence par rapport à l'octet de chiffrement " vrai ". La distance maximale est 8 lorsque tous les bits sont différents de ceux de l'octet de chiffrement " vrai ".

Pour chaque distance, le nombre moyen de chiffrés communs avec l'ensemble Ek de l'octet " vrai " est calculé. Pour la distance 0, celui-ci est égal à 128 chiffrés communs par ensemble Ei puisqu'il correspond à l'octet de chiffrement " vrai ". Pour les autres distances, ce nombre moyen de chiffrés communs varie autour de la moyenne 64. Cette distance est intéressante pour deux raisons. La première raison est la mise en évidence de l'impact d'une différence quelconque entre

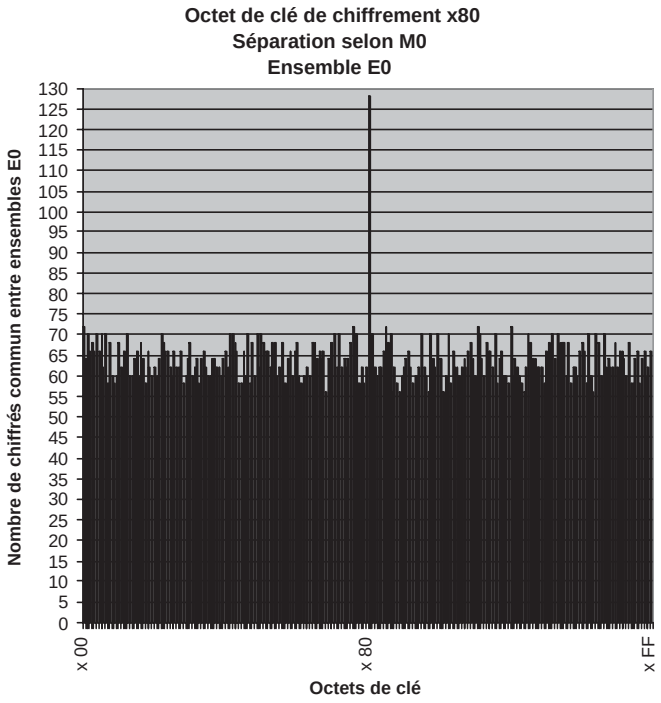


FIG. 3.8 – Nombre de chiffrés communs entre ensembles E0 pour le bit de séparation M0 et l'octet de chiffrement Kx80

l'octet de chiffrement " vrai " et les autres octets " faux ". La seconde raison est la mise en évidence de l'absence d'impact d'une différence quelconque entre octets de chiffrement " faux ". La figure 3.9 résume cette propriété pour tous les ensembles Ek, tous les bits Mi du message intermédiaire et tous les octets de chiffrement Kj.

Le changement d'un bit de clé produit donc une perturbation de la valeur du bit à observer avec une variabilité maximale. A nouveau, il n'y a aucune continuité ou progressivité entre la bonne valeur des bits de clé et une autre distante d'un ou plusieurs bits. Cette analyse nous permet d'écrire les probabilités entre le bit $clair_1$ et les octets chiffré_8 et clé_8 (Équation 3.48).

$$\begin{cases} p(clair_1 = 1) & = & p(g(\text{chiffré}_8, \text{clé}_8)_1 = 1) & \simeq & \frac{1}{2} & \forall\ \text{chiffré}_8 & \forall\ \text{clé}_8 \\ p(clair_1 = 0) & = & p(g(\text{chiffré}_8, \text{clé}_8)_1 = 0) & \simeq & \frac{1}{2} & \forall\ \text{chiffré}_8 & \forall\ \text{clé}_8 \end{cases} \tag{3.48}$$

Les fonctions g ainsi définies sont aussi bijectives. Il est donc possible d'avoir aussi une relation entre un bit de message chiffré avec 1 octet de message clair et 1 octet de sous-clé (Équation 3.49).

$$\begin{cases} p(\text{chiffré}_1 = 1) & = & p(f(clair_8, \text{clé}_8)_1 = 1) & \simeq & \frac{1}{2} & \forall\ \text{clair}_8 & \forall\ \text{clé}_8 \\ p(\text{chiffré}_1 = 0) & = & p(f(clair_8, \text{clé}_8)_1 = 0) & \simeq & \frac{1}{2} & \forall\ \text{clair}_8 & \forall\ \text{clé}_8 \end{cases} \tag{3.49}$$

Il est donc aussi possible d'écrire les probabilités suivantes. Pour une valeur v fixée de clé_8, le résultat du calcul du bit $clair_8(i)$ est obtenu avec la probabilité 1 pour tout chiffré_8. Donc, tout calcul qui avec chiffré_8 qui donne $clair_1 = 1$ avec une valeur v de clé_8 est réalisé avec les probabilités suivantes (Équation 3.50) :

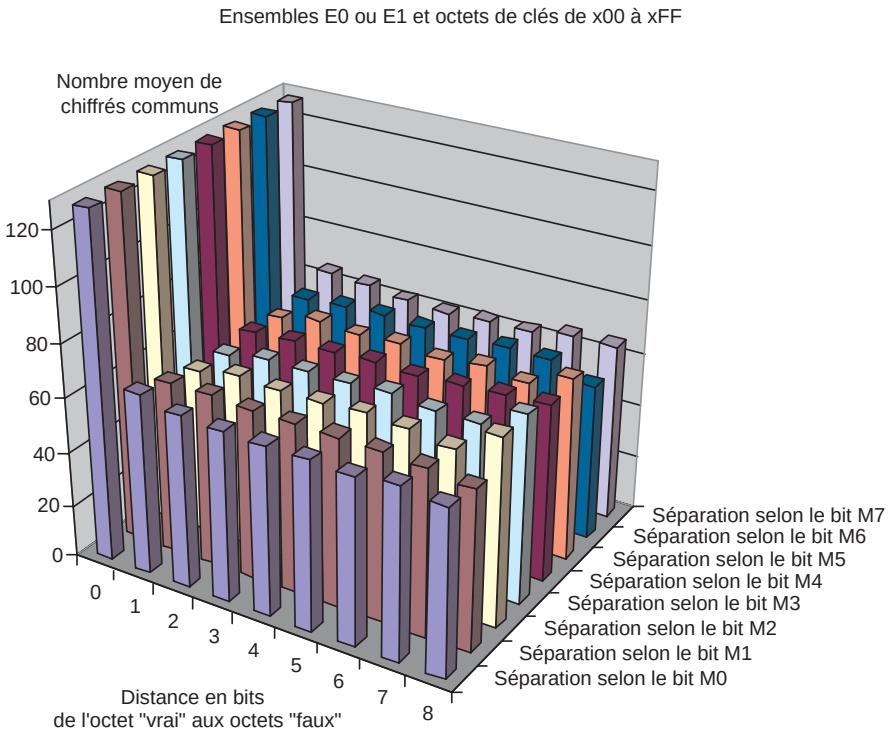


FIG. 3.9 – Impact de la distance entre octets de chiffrement sur les ensembles DPA

$$\left\{\begin{array}{lcccll} p(clair_1=1) & = & p(g(\text{chiffré}_8,\text{clé}_8)_1=1) & = & 1 \quad \forall\ \text{chiffré}_8 & \text{pour clé}_8=\text{v} \\ p(clair_1=0) & = & p(g(\text{chiffré}_8,\text{clé}_8)_1=0) & = & 0 \quad \forall\ \text{chiffré}_8 & \text{pour clé}_8=\text{v} \end{array}\right. \tag{3.50}$$

Et pour tout chiffré$_8$ qui avec clé$_8 = v$ donne $clair_1 = 0$ est réalisé avec les probabilités suivantes (Équation 3.51).

$$\left\{\begin{array}{lcccll} p(clair_1=1) & = & p(g(\text{chiffré}_8,\text{clé}_8)_1=1) & = & 0 \quad \forall\ \text{chiffré}_8 & \text{pour clé}_8=\text{v} \\ p(clair_1=0) & = & p(g(\text{chiffré}_8,\text{clé}_8)_1=0) & = & 1 \quad \forall\ \text{chiffré}_8 & \text{pour clé}_8=\text{v} \end{array}\right. \tag{3.51}$$

Pour toute autre valeur clé$_8 \neq v$ et ce quelle que soit la distance en bit $clair_1$ est calculé avec les probabilités suivantes (Équation 3.52).

$$\left\{\begin{array}{lcccll} p(clair_1=1) & = & p(g(\text{chiffré}_8,\text{clé}_8)_1=1) & \simeq & \frac{1}{2} \quad \forall\ \text{chiffré}_8 & \text{pour clé}_8\neq\text{v} \\ p(clair_1=0) & = & p(g(\text{chiffré}_8,\text{clé}_8)_1=0) & \simeq & \frac{1}{2} \quad \forall\ \text{chiffré}_8 & \text{pour clé}_8\neq\text{v} \end{array}\right. \tag{3.52}$$

Sur le dernier tour de l'algorithme AES, chiffré$_8$ est connu avec certitude. La seule inconnue est clé$_8$. En parcourant l'ensemble des 256 valeurs possibles pour clé$_8$ et en tenant compte des probabilités définies précédemment, lorsque clé$_8$ représente la vraie valeur des 8 bits de la sous-clé, alors le calcul de $clair_1$ est réalisé avec la probabilité 1.

En revanche, lorsque clé$_8$ représente une fausse valeur des 8 bits de la sous-clé, alors le calcul de $clair_1$ est réalisé avec la probabilité proche de $\frac{1}{2}$.

Lorsque clé$_8$ représente la vraie valeur des 8 bits de la sous-clé, les chiffrés sont classés dans l'ensemble E1 selon le critère $clair_1 = 1$ avec la probabilité 1 et dans l'ensemble E0 selon le critère $clair_1 = 0$ avec la probabilité 1. Les ensembles E1 et E0 sont bien distincts selon le critère $clair_1$. En revanche, dès que clé$_8$ représente une fausse valeur des 8 bits de la sous-clé, les chiffrés sont classés dans l'ensemble E1 selon le critère $clair_1 = 1$ avec une probabilité proche de $\frac{1}{2}$ et dans l'ensemble E0 selon le critère $clair_1 = 0$ avec aussi une probabilité proche de $\frac{1}{2}$. Dans ce cas, la répartition des chiffrés est "équiprobable" entre les ensembles E1 et E0 qui ne sont plus distincts selon le critère $clair_1$. La fonction f est discriminante.

Un exemple de fonction de sélection discriminante pour l'algorithme AES a été présenté. Il est encore possible d'étendre cette approche à une relation plus générale tant sur le nombre de bits du chiffré, du clair et de la clé que sur des instants différents correspondants à différentes étapes de l'algorithme.

3.4.3 Définition de la fonction de sélection DPA

De façon générale, une fonction de sélection DPA sera toute fonction bijective discriminante entre m bits de chiffré, n bits de clair et p bits de clé à différents instants t telle que 2^m, 2^n et 2^p soient de " petits " nombres, $\text{chiffré}_m = f(clair_n, \text{clé}_p)$ et ayant les probabilités suivantes.

Tout $clair_n$ qui avec $\text{clé}_p = K$ donne $\text{chiffré}_m = M$ est réalisé avec les probabilités :

$$\left\{ \begin{array}{lcl} p(\text{chiffré}_{m(t)} = M) & = & p(g(clair_{n(t)}, \text{clé}_{p(t)})_m = M) = 1 \quad \forall\, clair_n \quad \text{pour clé}_p = \text{K} \\ p(\text{chiffré}_{m(t)} \neq M) & = & p(g(clair_{n(t)}, \text{clé}_{p(t)})_m \neq M) = 0 \quad \forall\, clair_n \quad \text{pour clé}_p = \text{K} \end{array} \right. \tag{3.53}$$

Pour toute autre valeur $\text{clé}_n \neq K$ et ce quelle que soit la distance en bit le calcul est réalisé les probabilités :

$$\left\{ \begin{array}{lcl} p(\text{chiffré}_{m(t)} = M) & = & p(g(clair_{n(t)}, \text{clé}_{p(t)})_m = M) \simeq \frac{1}{2^m} \quad \forall\, clair_n \quad \text{pour clé}_p \neq K \\ p(\text{chiffré}_{m(t)} \neq M) & = & p(g(clair_{n(t)}, \text{clé}_{p(t)})_m \neq M) \simeq \frac{1}{2^m} \quad \forall\, clair_n \quad \text{pour clé}_p \neq K \end{array} \right. \tag{3.54}$$

Les fonctions f ainsi définies sont bijectives. Il est donc possible d'avoir aussi une relation $clair_n = g(\text{chiffré}_m, \text{clé}_p)$. Il est donc aussi possible d'écrire les probabilités suivantes. Tout chiffré_m qui avec $\text{clé}_p = K$ donne $clair_n = N$ est réalisé avec les probabilités :

$$\left\{ \begin{array}{lcl} p(clair_{n(t)} = N) & = & p(g(\text{chiffré}_{m(t)}, \text{clé}_{p(t)})_n = N) = 1 \quad \forall\, \text{chiffré}_m \quad \text{pour clé}_p = \text{K} \\ p(clair_{n(t)} \neq N) & = & p(g(\text{chiffré}_{m(t)}, \text{clé}_{p(t)})_n \neq N) = 0 \quad \forall\, \text{chiffré}_m \quad \text{pour clé}_p = \text{K} \end{array} \right. \tag{3.55}$$

Pour toute autre valeur $\text{clé}_n \neq K$ et ce quelle que soit la distance en bit le calcul est réalisé avec les probabilités :

$$\left\{ \begin{array}{lcl} p(clair_{n(t)} = N) & = & p(g(\text{chiffré}_{m(t)}, \text{clé}_{p(t)})_n = N) \simeq \frac{1}{2^n} \quad \forall\, \text{chiffré}_m \quad \text{pour clé}_p \neq K \\ p(clair_{n(t)} \neq N) & = & p(g(\text{chiffré}_{m(t)}, \text{clé}_{p(t)})_n \neq N) \simeq \frac{1}{2^n} \quad \forall\, \text{chiffré}_m \quad \text{pour clé}_p \neq K \end{array} \right. \tag{3.56}$$

Chapitre 4

Mise en œuvre de la DPA du DES

Sommaire

4.1 Mesures des consommations

Les conditions de l'attaque DPA sur l'algorithme DES sont les suivantes. La clé de chiffrement est fixée à la valeur aléatoire Ka et N clairs aléatoires sont chiffrés. Ceci permet d'obtenir les chiffrés et leurs consommations électriques respectives. Dans le cas d'une mise en œuvre de la DPA sur le tour 16 d'une architecture itérative du DES, déterminer sur l'appareil de mesure où se situe l'exécution de ce tour revient à compter 16 périodes d'activité du circuit correspondants aux 16 tours. Dans le cas d'une mise en œuvre de la DPA sur le tour 1 du DES, le premier tour de l'activité électrique du composant fournit la consommation du tour 1. La figure 4.1 illustre un exemple possible de mesures pour la mise en œuvre de la DPA sur une architecture itérative du DES.

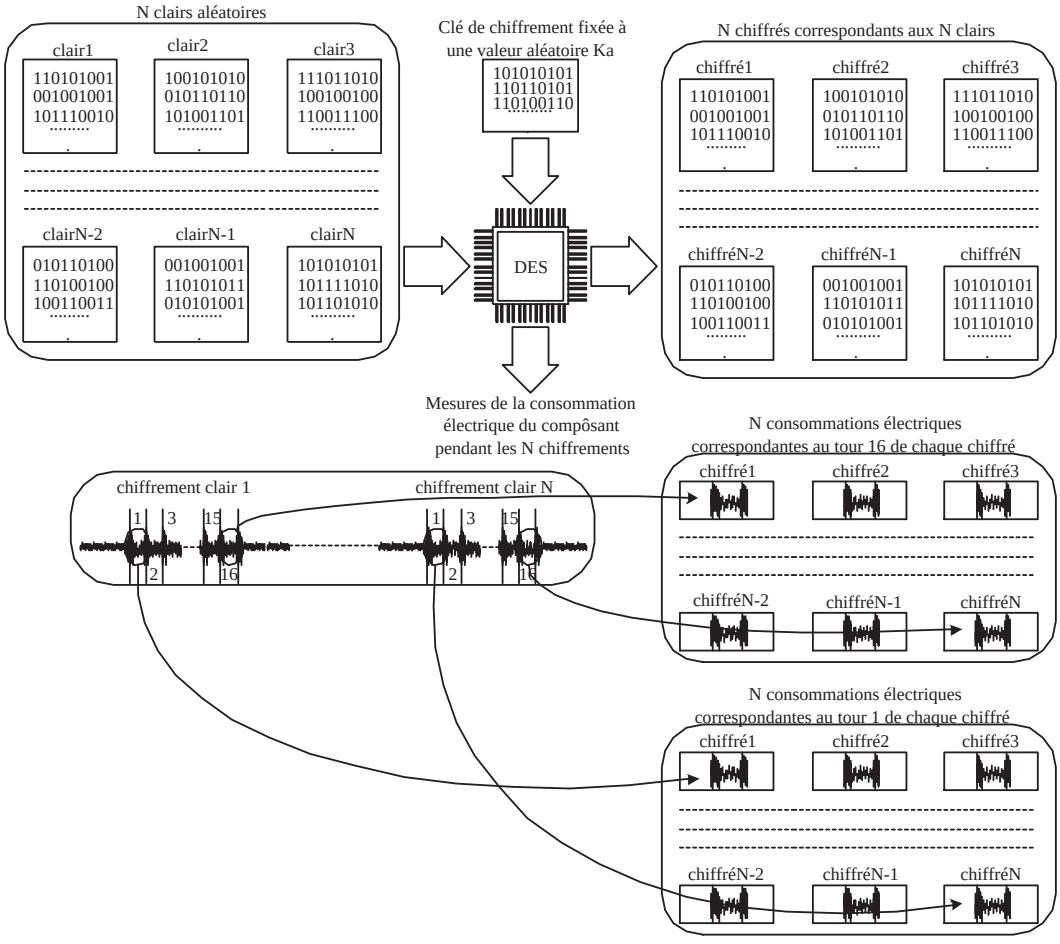


FIG. 4.1 – Exemple de mesures pour la mise en œuvre de la DPA sur le DES

4.2 Construction d'une fonction de sélection DPA pour l'algorithme DES

Pour mettre en œuvre l'attaque DPA sur un composant exécutant le DES, il est nécessaire de trouver où elle peut être théoriquement appliquée. Cela passe par l'analyse de l'algorithme dont le schéma général est illustré par la figure C.4 dans l'annexe C afin de déterminer une fonction de sélection DPA.

Dans l'exemple, l'attaque DPA est mise en œuvre sur le tour 16 du DES dont le schéma est rappelé par la figure 4.2. La sortie de la fonction f est déterminée par la sous-clé K16 et par la donnée R15. Le résultat du " xor " entre cette sortie et la donnée L15 donne R16. L16 est égal à R15.

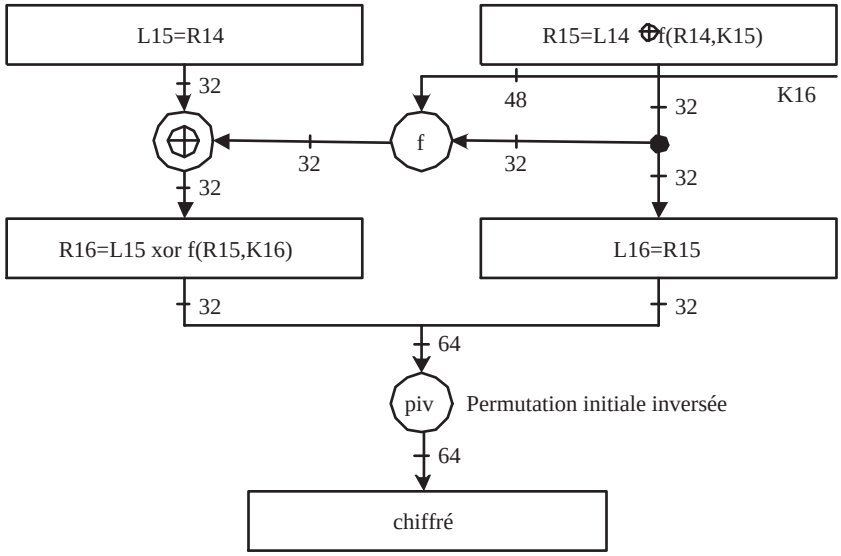


Fig. 4.2 – Tour 16 du DES

Puisque les chiffrés sont connus et que la permutation initiale inverse est donnée par le standard, il est possible de calculer R16 et L16 et l'analyse est réduite au schéma 4.3.

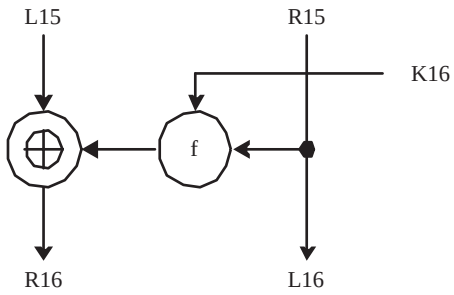


Fig. 4.3 – Tour 16 du DES

Il est maintenant nécessaire d'étudier comment sont calculés les bits de R16. Par exemple, le

schéma de la figure 4.4 met en évidence comment le bit 0 de R16 est calculé à partir de la sous clé K16. Seuls 6 bits de la sous clé sont nécessaires pour calculer le bit R16(0). Si à partir du bit R16(0) il est possible de déterminer la valeur des 6 bits K16(18 :23) il est aussi possible de déterminer les autres bits de K16 en choisissant judicieusement d'autres bits de R16 mettant en œuvre d'autres paquets de 6 bits sur d'autres fonctions de substitution.

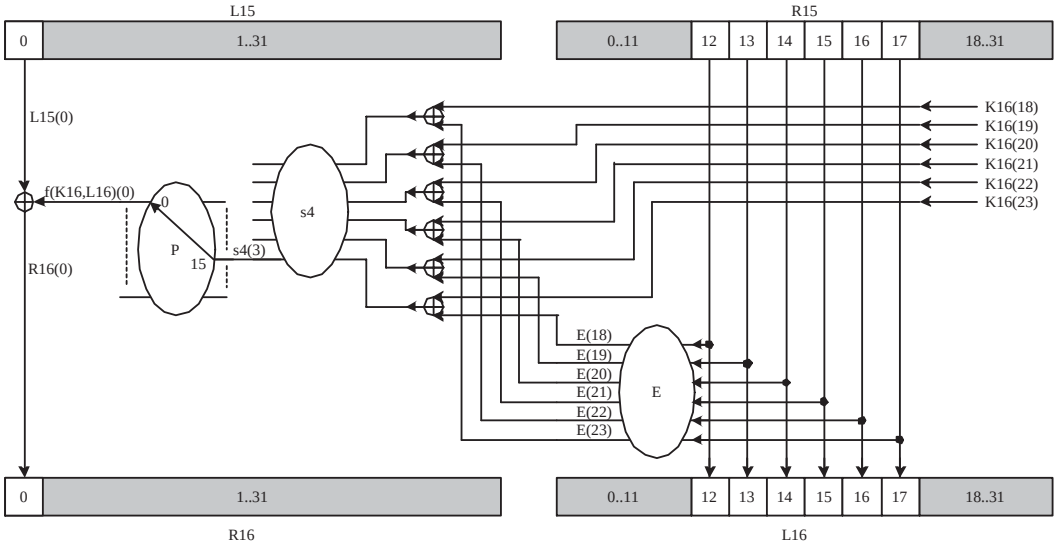


FIG. 4.4 – Calcul du bit R16(0)

Il est ainsi possible de déterminer une fonction de sélection DPA reliant un bit de chiffré à six bits de clair et six bits de sous-clé. Elle peut être écrite sous la forme suivante où E, s4 et P sont respectivement la permutation, une fonction de substitution et la fonction d'expansion décrite au paragraphe de l'annexe A, et respectivement définies par les tableaux C.4, C.3 et C.2. La relation entre le bit observé L15(0), les bits de sous-clé et les bits de message est définie par l'équation suivante :

$$L_{15}(0) = R_{16}(0) \oplus P(S_4(K_{16}(18:23) \oplus R_{15}(12:17))(3))(0) \tag{4.1}$$

En reprenant l'écriture générale d'une fonction de sélection DPA la relation précédente peut être exprimée sous la forme suivante :

$$clair_1 = \text{chiffré}_1 \oplus P(S_4(\text{clé}_6 \oplus \text{chiffré}_6)(3))(0) \tag{4.2}$$

où $L_{15}(0)$, $R_{16}(0)$, $K_{16}(18:23)$ et $R_{15}(12:17)$ sont respectivement $clair_1$, chiffré_1, clé_6 et chiffré_6. Ceci définit bien une fonction de sélection DPA qui a la forme générale suivante :

$$clair_1 = g(\text{chiffré}_7, \text{clé}_6) \tag{4.3}$$

dont le calcul, qui, pour tout chiffré_7 avec $\text{clé}_6 = K$ donne $clair_1 = 1$ est réalisé avec les probabilités :

$$\begin{cases} p(clair_1 = 1) = p(g(\text{chiffré}_7, \text{clé}_6) = 1) = 1 \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 = \text{K} \\ p(clair_1 = 0) = p(g(\text{chiffré}_7, \text{clé}_6) = 0) = 0 \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 = \text{K} \end{cases} \tag{4.4}$$

Tout chiffré$_7$ qui avec clé$_6 = K$ donne $clair_1 = 0$ avec les probabilités :

$$\begin{cases} p(clair_1 = 0) = p(g(\text{chiffré}_7, \text{clé}_6) = 0) = 1 \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 = \text{K} \\ p(clair_1 = 1) = p(g(\text{chiffré}_7, \text{clé}_6) = 1) = 0 \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 = \text{K} \end{cases} \tag{4.5}$$

Pour toute autre valeur clé$_6 \neq K$ et ce quelle que soit la distance en bit le calcul est réalisé avec les probabilités :

$$\begin{cases} p(clair_1 = 1) = p(g(\text{chiffré}_7, \text{clé}_6) = 1) \simeq \frac{1}{2} \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 \neq K \\ p(clair_1 = 0) = p(g(\text{chiffré}_7, \text{clé}_6) = 0) \simeq \frac{1}{2} \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 \neq K \end{cases} \tag{4.6}$$

Les 6 bits $K16(18:23)$ peuvent prendre 64 valeurs. Une seule est exacte et correspond à la clé de chiffrement, les 63 autres sont fausses. Pour chacune des 64 valeurs possibles Ks de $K16(18:23)$, il est possible de calculer pour tous les chiffrés le bit $f(K16, L16)(0)$ à partir des bits et $L16(12:17)$ comme illustré dans la figure 4.5.

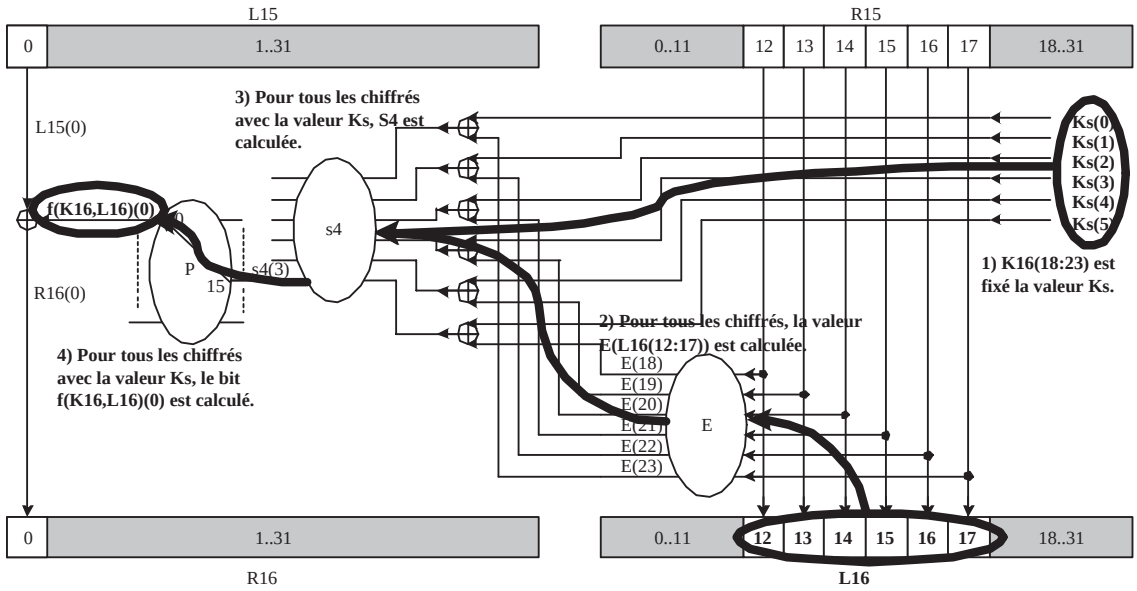
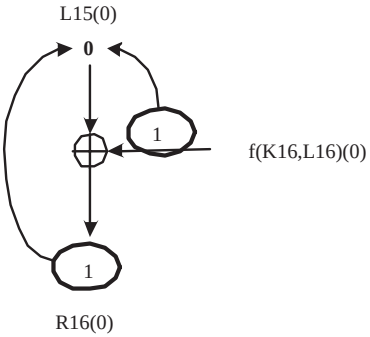


FIG. 4.5 – Calcul du bit f(K16,L16)(0) pour tous les chiffrés avec Ks

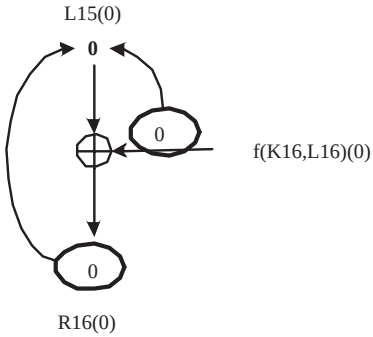
Pour chaque valeur Ks possible de K16(18 :23), les chiffrés ayant le bit R16(0) égal à 1 impliqueront que le bit f(K16,L16)(0) soit égal à 1 ou à 0 selon les valeurs des bits L16(12 :17) correspondants. Il en est de même pour les chiffrés ayant le bit R16(0) égal à 0, ils impliqueront que le bit f(K16,L16)(0) soit égal à 1 ou à 0 selon les valeurs des bits L16(12 :17) correspondants. Pour toute valeur donnée Ks de K16(18 :23), il est possible de calculer pour tous les chiffrés la valeur du bit L15(0) comme illustré dans la figure 4.6.

Étant donné que $L15(0) = R16(0) \oplus f(K16, L16)(0)$, tous les chiffrés vérifiant $L15(0) = 1$ sont

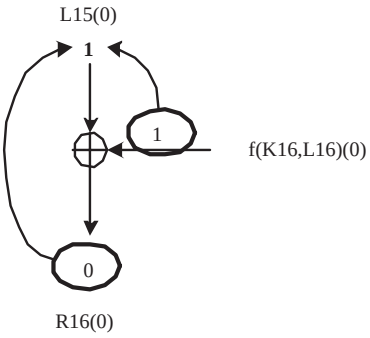
Tous les chiffrés avec la valeur Ks ayant le bit f(K16,L16)(0) égal à 1 et le bit R16(0) égal à 1 ont le bit L15(0) égal à 0.



Tous les chiffrés avec la valeur Ks ayant le bit f(K16,L16)(0) égal à 0 et le bit R16(0) égal à 0 ont le bit L15(0) égal à 0.



Tous les chiffrés avec la valeur Ks ayant le bit f(K16,L16)(0) égal à 1 et le bit R16(0) égal à 0 ont le bit L15(0) égal à 1.



Tous les chiffrés avec la valeur Ks ayant le bit f(K16,L16)(0) égal à 0 et le bit R16(0) égal à 1 ont le bit L15(0) égal à 1.

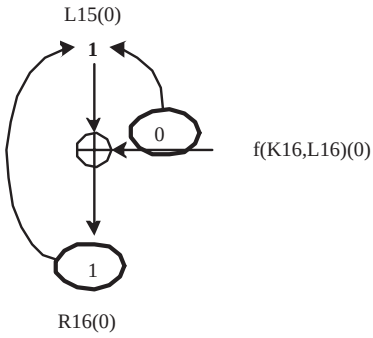


FIG. 4.6 – Calcul du bit L15(0) pour tous les chiffrés avec Ks

regroupés dans l'ensemble E1 et tous les chiffrés vérifiant $L15(0) = 0$ sont regroupés dans l'ensemble E0. La figure 4.7 illustre ce regroupement.

L'ensemble E1 représente les opérations $1 \oplus 0$ et $1 \oplus 1$ alors que l'ensemble E0 représente les opérations $0 \oplus 1$ et $0 \oplus 0$. Le bit L15(0) est toujours égal à 1 dans l'ensemble E1 et il est toujours égal à 0 dans l'ensemble E0. Ils diffèrent toujours de ce bit en entrée de l'opération $\oplus$.

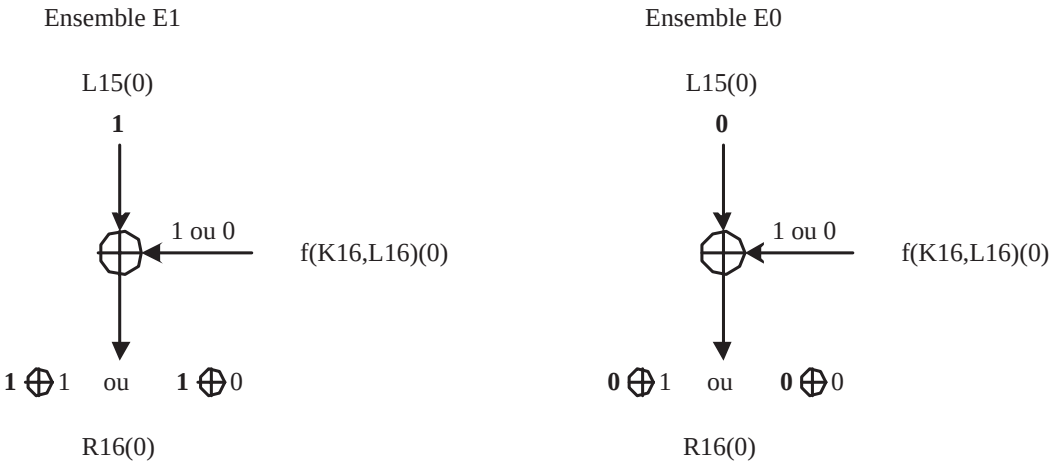


FIG. 4.7 – Regroupement en ensembles E1 et E0 pour une valeur de K16(18 :23)

4.3 Les regroupements

Il est donc possible à partir des chiffrés de les séparer en deux ensembles E1 et E0 suivant la valeur de $L15(0)$ pour une valeur donnée de $K16(18:23)$. Comme $K16(18:23)$ peut prendre 64 valeurs, 64 regroupements E1 et E0 sont réalisés à partir des mêmes chiffrés comme illustré dans la figure 4.8.

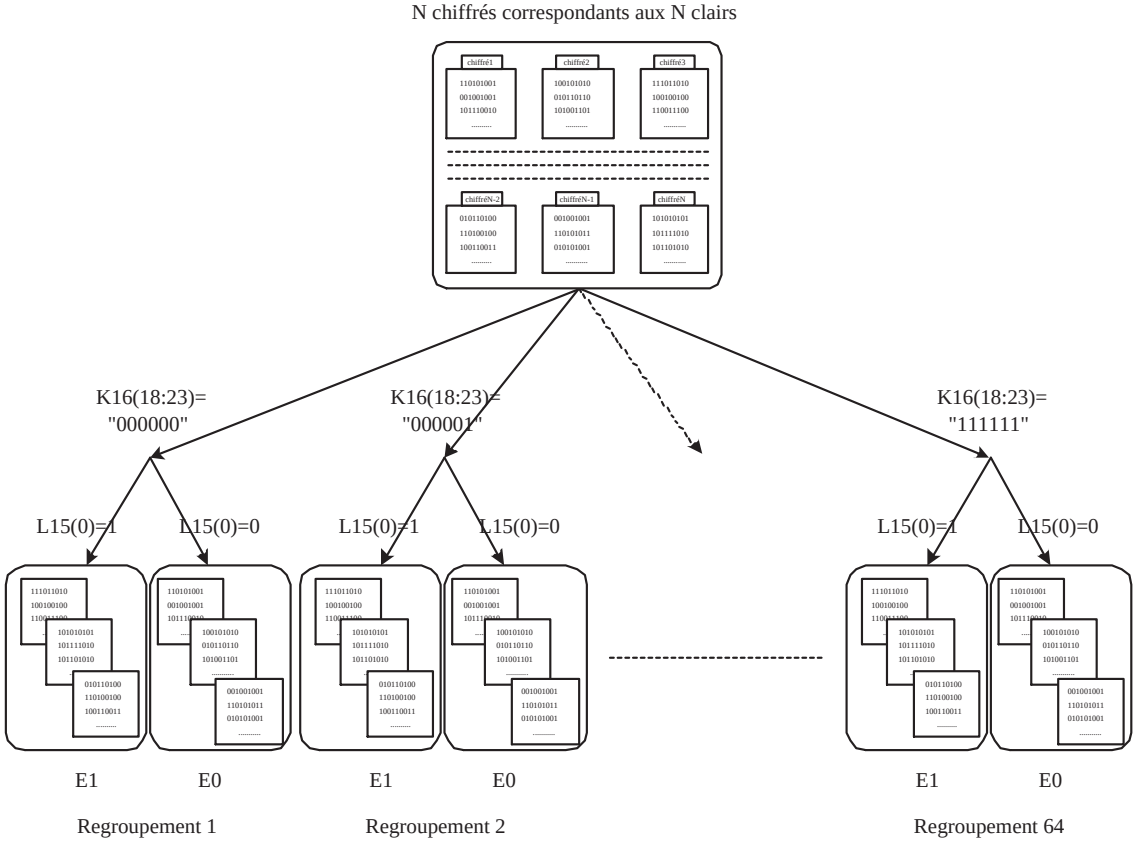


FIG. 4.8 – Les 64 regroupements logiques des chiffrés pour K16(18 :23)

Les traces de consommation associées aux chiffrés sont regroupées suivant les ensembles E1 et E0 comme présenté dans la figure 4.9. Pour chaque ensemble E1 et chaque ensemble E0 de chaque regroupement, les moyennes de consommation sont calculées ainsi que leur différence.

En supposant que que K16(18 :23) soit égal à " 101101 " et que ce soit le bon paquet de bits de clé, tout calcul donnant $clair_1 = 1$ est réalisé avec la probabilité

$$p(clair_1 = 1) \quad = \quad p(g(\text{chiffré}_7, \text{clé}_6) = 1) \quad = \quad 1 \quad \forall\ \text{chiffré}_7 \quad \text{pour clé}_6 = 101101 \qquad (4.7)$$

et tout calcul qui donne $clair_1 = 0$ est réalisé avec la probabilité

$$p(clair_1 = 0) \quad = \quad p(g(\text{chiffré}_7, \text{clé}_6) = 0) \quad = \quad 1 \quad \forall\ \text{chiffré}_7 \quad \text{pour clé}_6 = 101101 \qquad (4.8)$$

Cela signifie que le calcul pour séparer les chiffrés et donc les consommations respectives du

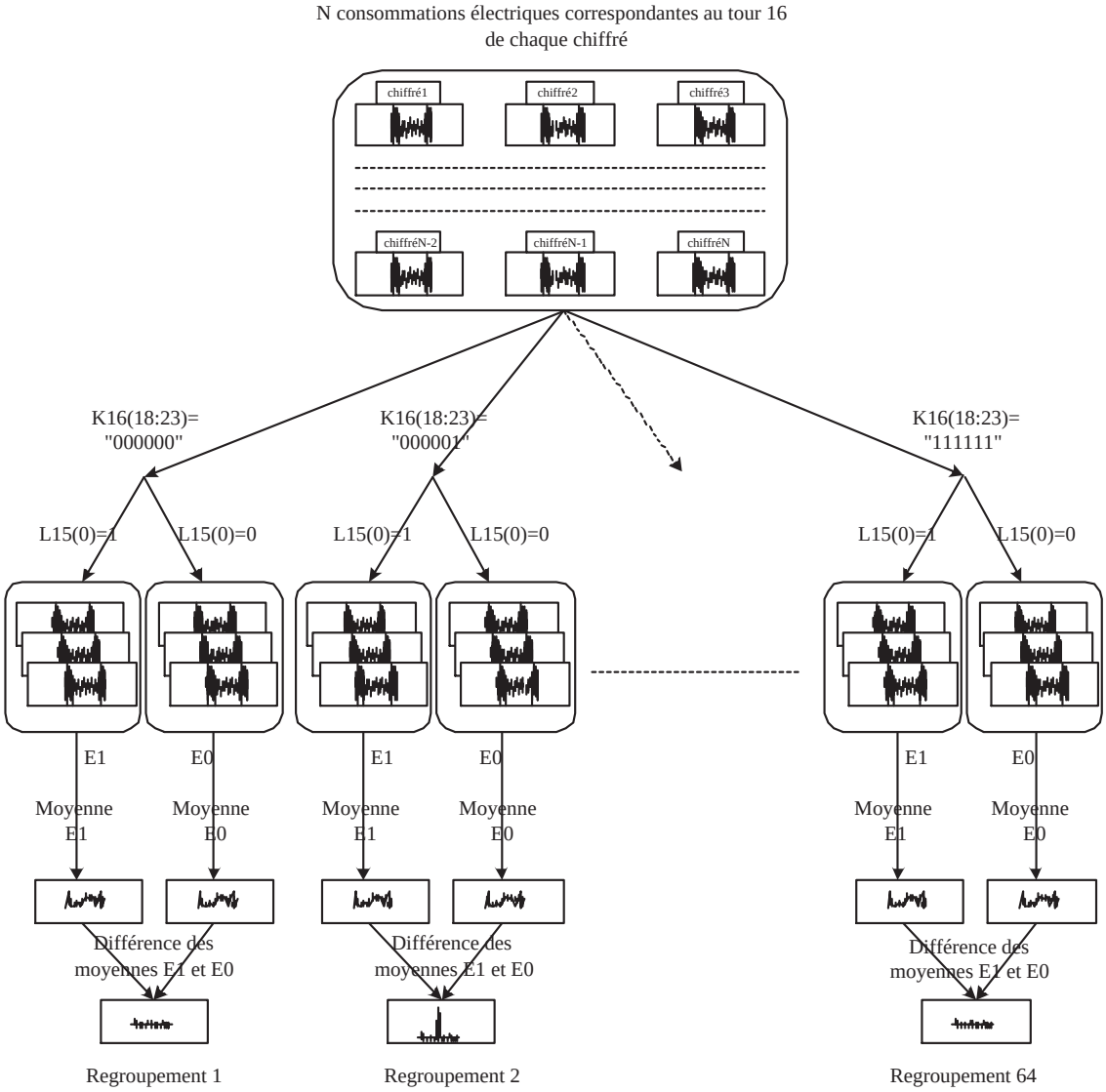


FIG. 4.9 – Les 64 regroupements des consommations correspondantes pour K16(18 :23)

circuit a été correctement réalisé.

Lorsque que $clair_1$ est égal à 1 et que $clé_6$ est la bonne sous-clé, l'ensemble E1 représente bien ce qui s'est réellement passé dans le circuit et l'opération $\oplus$ a toujours été $1 \oplus 1$ et $1 \oplus 0$. De même, lorsque que $clair_1$ est égal à 0 et que $clé_6$ est la bonne sous-clé, l'ensemble E0 représente aussi ce qui s'est réellement passé dans le circuit et l'opération $\oplus$ a toujours été $0 \oplus 1$et $0 \oplus 0$. En supposant que le matériel consomme différemment en fonction des valeurs logiques manipulées, il existe donc une différence de consommation entre E1 et E0 et leurs consommations moyennes tendent vers deux valeurs distinctes visibles par différence : la DPA.

En supposant que K16(18 :23) égal à " 101101 " soit un mauvais sous-groupe de bits de clé, comme dans la figure 4.9, cela signifie que le calcul pour séparer les chiffrés et donc les consommations respectives du circuit a été mal réalisé. En effet, pour toute autre valeur $clé_6 \neq 101101$

et ce quelle que soit la distance en bit le calcul est réalisé avec les probabilités

$$\begin{cases} p(clair_1 = 1) = p(g(\text{chiffré}_7, \text{clé}_6) = 1) \simeq \frac{1}{2} \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 \neq 101101 \\ p(clair_1 = 0) = p(g(\text{chiffré}_7, \text{clé}_6) = 0) \simeq \frac{1}{2} \quad \forall \text{ chiffré}_7 \quad \text{pour clé}_6 \neq 101101 \end{cases} \tag{4.9}$$

E1 ne représente donc plus ce qui s'est réellement passé dans le circuit et l'opération XOR a été de façon "équiprobable" $1 \oplus 1$ ou $1 \oplus 0$ ou $0 \oplus 1$ ou $0 \oplus 0$. E0 aussi ne représente plus ce qui s'est réellement passé dans le circuit et l'opération XOR a été de façon "équiprobable" $1 \oplus 1$ ou $1 \oplus 0$ ou $0 \oplus 1$ ou $0 \oplus 0$. De part la construction du DES, la répartition est statistiquement équilibrée, E1 et E0 sont statistiquement composés pour moitié de bonnes valeurs et pour moitié de mauvaises valeurs. Leurs consommations moyennes tendent vers la même valeur. La différence de consommation tend vers 0.

Le paquet de bits de clé ayant la plus grande différence de consommation est celui qui correspond à la clé enregistrée dans le circuit. Pour les autres paquets de bits de la sous-clé, la même approche est réitérée en choisissant un autre bit de R16 mettant en œuvre une autre fonction de substitution qui met en œuvre un autre paquet de bits de clé.

L'attaque développée sur le dernier tour du DES avec la connaissance des chiffrés et de leurs consommations est réalisable sur le premier tour avec la connaissance des clairs et de leurs consommations sur le tour 1. La figure 4.10 illustre le calcul du bit R1(0) qui servira à la séparation en ensembles E1 et E0 selon sa valeur.

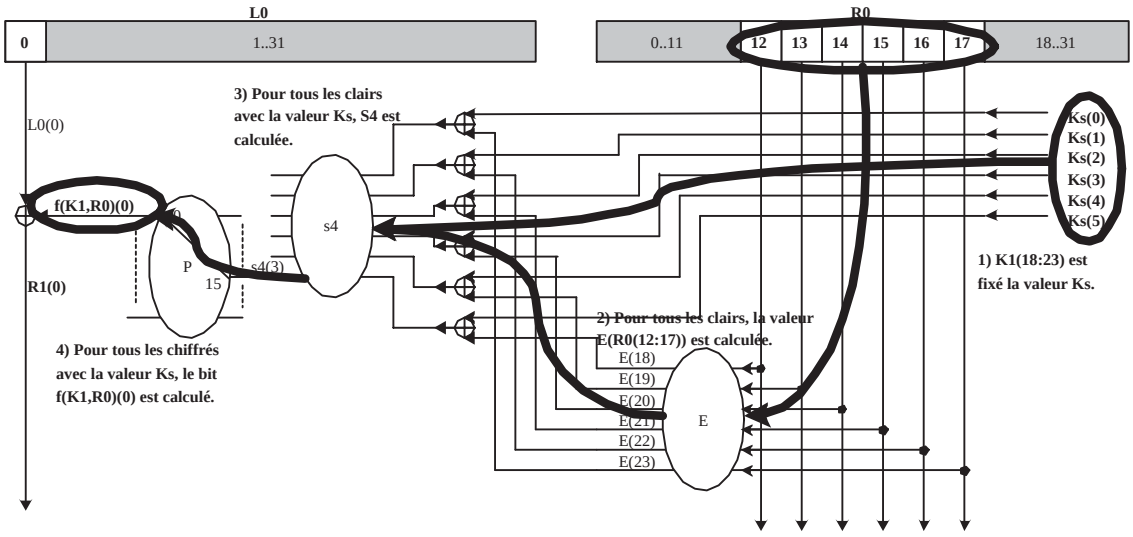


FIG. 4.10 – Mise en œuvre de la DPA sur le tour 1 du DES

4.4 Conclusion

La DPA illustrée précédemment fut l'objet d'une communication en 1999 [25]. D'après les résultats, cette attaque est exploitable sur un circuit intégré, la différence de moyenne obtenue est de l'ordre de 15 à 20 microampères sur moins d'une microseconde environ. En fonction des

technologies, cela peut tout à fait correspondre à une différence de consommation d'une porte logique de type " xor ".

Chapitre 5

Mise en œuvre de la DPA de l'AES

Sommaire

5.1 Mesures des consommations

Les conditions de l'attaque sont quasi identiques à celles du DES si ce n'est que la DPA s'appliquera ici sur le tour 10, 12 ou 14 selon la longueur de la clé de chiffrement. De plus, elle ne s'applique plus à une porte xor comme pour le DES mais à une ou plusieurs portes de la fonction SubBytes. On peut s'attendre à obtenir une différence de moyennes de consommation du même ordre de grandeur que pour un DES dans la même technologie. La clé est explorée par des valeurs successives dont la valeur courante est Ka qui sert à chiffrer N clairs aléatoires. La connaissance des chiffrés et de leurs consommations électriques respectives est obtenue par mesure (Figure 5.1).

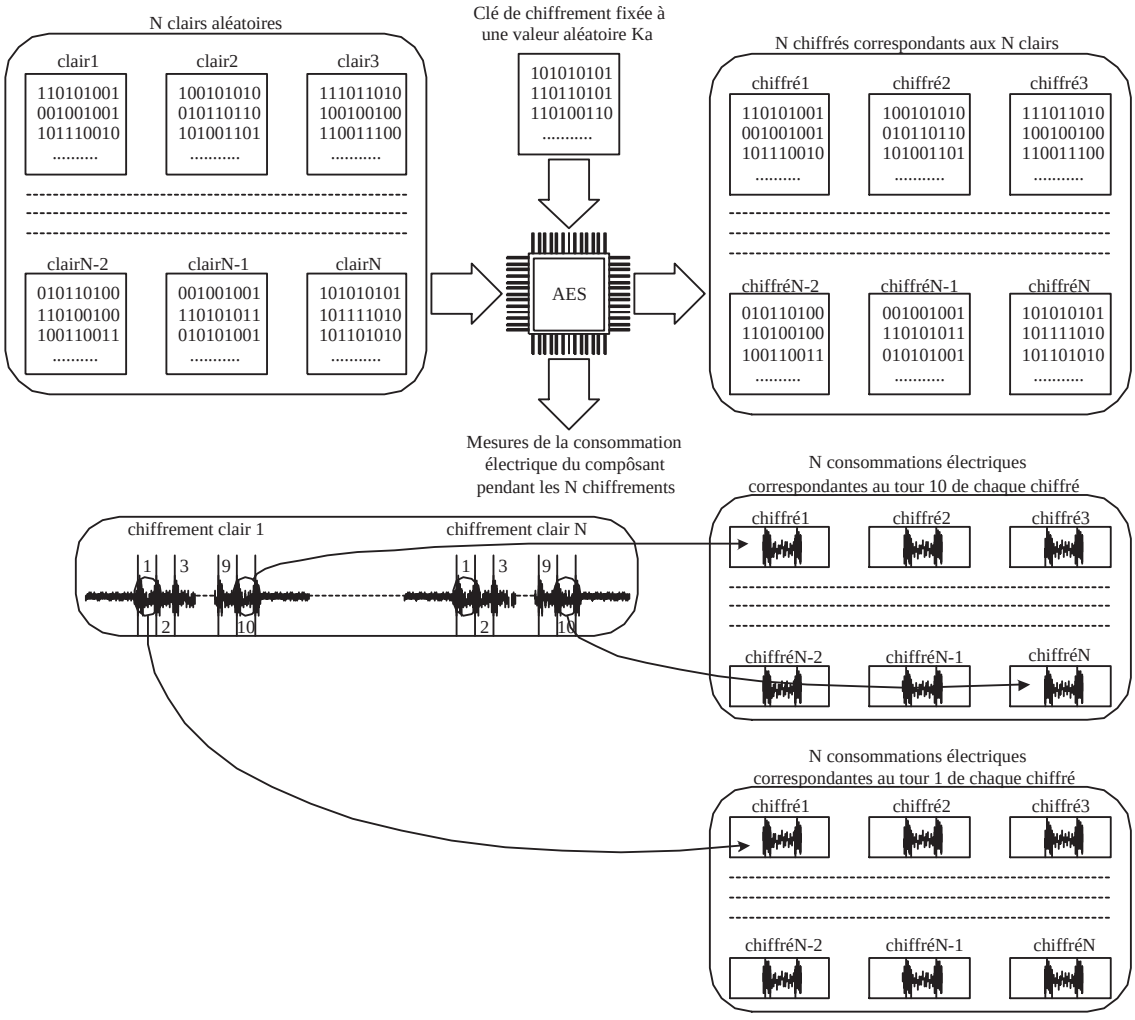


FIG. 5.1 – Exemple de mesures pour la mise en œuvre de la DPA sur l'AES

Dans le cas d'une mise en œuvre de la DPA sur le tour 10 d'une architecture itérative de l'AES, déterminer sur l'appareil de mesure où se situe l'exécution de ce tour revient à compter 10 périodes d'activité du circuit correspondantes aux 10 tours. Dans le cas d'une mise en œuvre de la DPA sur le tour 1 de l'AES, le premier tour de l'activité électrique du composant fournit la consommation du tour 1.

5.2 Analyse de l'algorithme AES : création d'une fonction de sélection DPA

Tout comme pour le DES, il est nécessaire de trouver où la DPA peut être théoriquement appliquée par analyse de l'algorithme. Dans l'exemple, l'attaque DPA est mise en œuvre sur le tour 10 de l'AES en considérant que la clé a une longueur de 128 bits. Le schéma du tour est rappelé dans la figure 5.2.

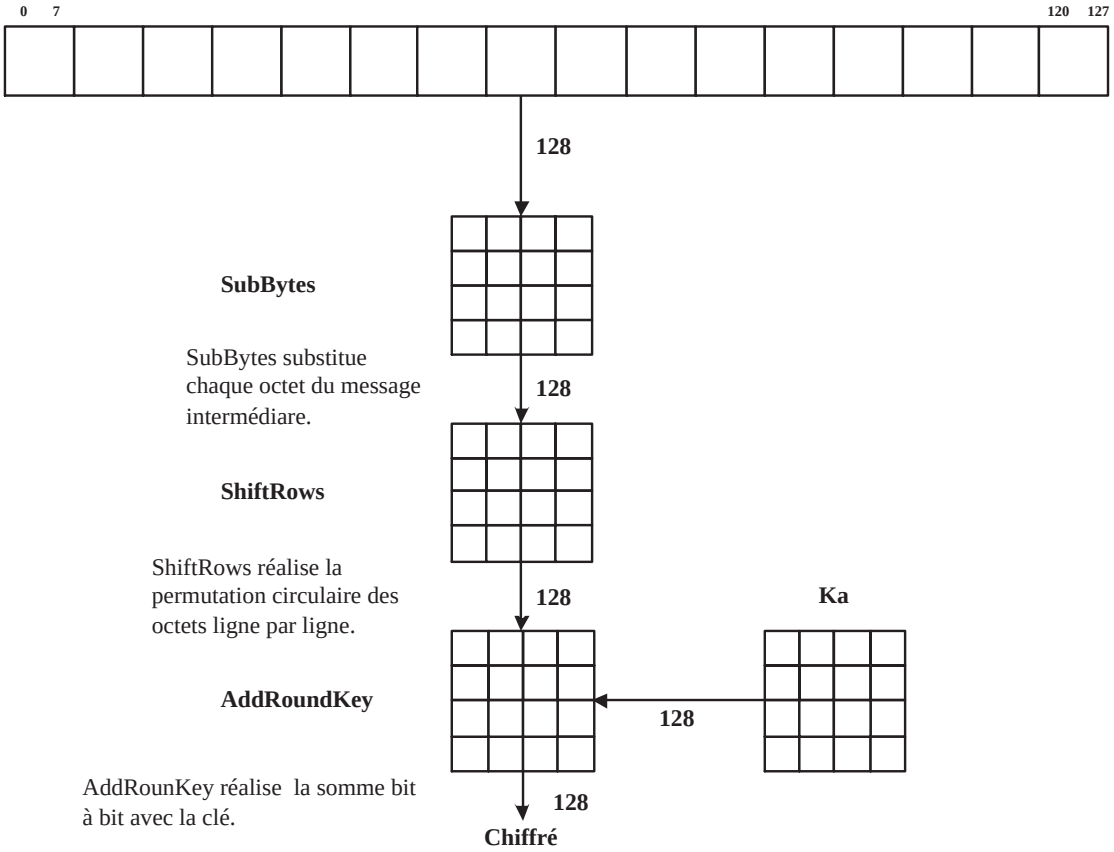


FIG. 5.2 – Tour 10 de l'AES

L'analyse du bit M(0) du message, par exemple, montre qu'il n'agit pas comme un bit isolé dans l'algorithme mais qu'il appartient à un octet qui est l'entrée d'une fonction SubBytes. Celle-ci substitue la valeur de l'octet par celle définie dans le standard. L'octet substitué est alors décalé sur sa ligne par la fonction ShiftRows. Pour le bit M(0), l'octet substitué correspondant ne change pas de place dans le standard. Finalement, l'octet en sortie de ShiftRows est combiné bit à bit par un xor (fonction AddRoundkey) avec un octet de la sous-clé pour donner le chiffré. La figure 5.3 résume comment intervient le bit M(0).

Cette analyse permet de déterminer une fonction de sélection DPA représentable sous la forme suivante :

$$clair_1 = SubBytes^{-1}(ShiftRows^{-1}(AddRoundKey^{-1}(\text{clé}_8, \text{chiffré}_8)))(0) \qquad (5.1)$$

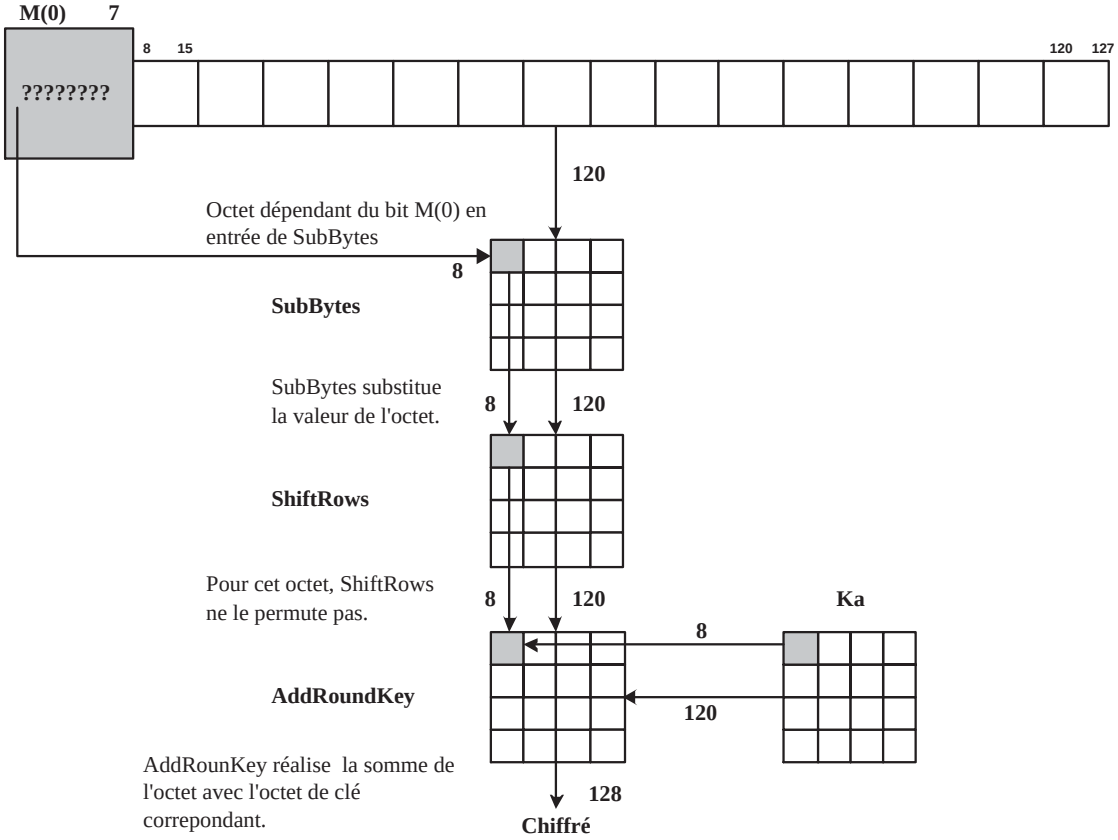


FIG. 5.3 – Analyse de l'intervention du bit M(0)

où $clair_1$ représente le bit M(0) et dont les probabilités vérifient l'équation 3.48 au chapitre 3.

En fixant la valeur à Ks de l'octet de clé intervenant sur le bit M(0), il est possible pour tous les chiffrés d'inverser AddRoundKey sur l'octet correspondant. Après inversion de AddRoundKey il est aussi possible d'inverser ShiftRows sur l'octet. Et finalement, après inversion de ShiftRows, SubBytes peut être inversée sur l'octet. Pour une valeur fixée à Ks, il est donc possible d'obtenir une image de l'octet dépendant du bit M(0) et ce pour chaque chiffré. Certains chiffrés ont M(0) égal à 1 et d'autres ont M(0) égal à 0 avant SubBytes. La figure 5.4 illustre ce processus.

En supposant que Ks soit la vraie valeur, il est alors possible de retrouver l'image exacte de l'octet correspondant de tous les chiffrés en entrée de AddRounKey. Si l'octet obtenu après AddRoundKey est vrai alors l'inverse de ShiftRows est vrai pour tous les chiffrés. Si l'octet obtenu après ShiftRows est vrai alors l'inverse de SubBytes est vrai pour tous les chiffrés. Pour Ks vrai, l'inverse de cet octet est vrai pour tous les chiffrés. Tous les chiffrés ayant M(0) égal à 0 sont regroupés dans l'ensemble E0 et tous les chiffrés ayant M(0) égal à 1 sont regroupés dans l'ensemble E1. Cela repose toujours sur le fait que lorsque K est le bon paquet de bits de clé, tout chiffré_8 qui avec $\text{clé}_8 = K$ donne $clair_1 = 1$ est réalisé avec la probabilité :

$$p(clair_1 = 1) \quad = \quad p(g(\text{chiffré}_8, \text{clé}_8)_1 = 1) \quad = \quad 1 \quad \forall\ \text{chiffré}_8 \quad \text{pour clé}_8 = \text{K} \tag{5.2}$$

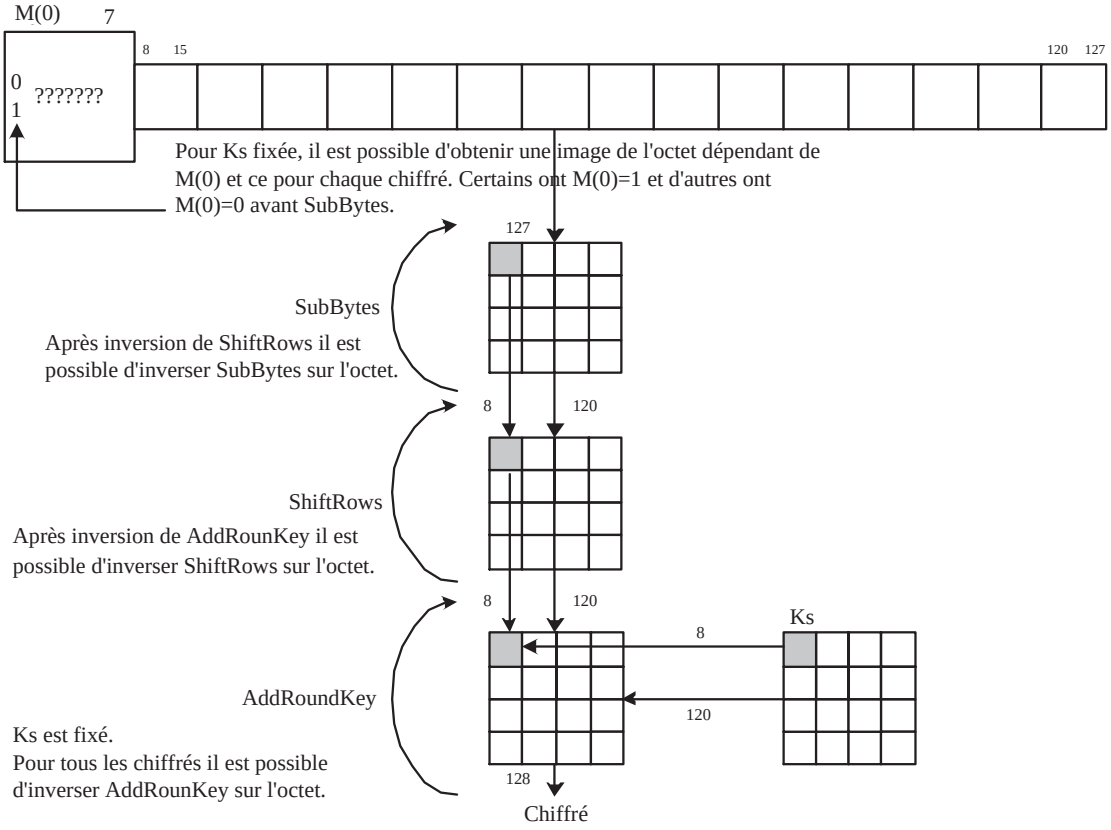


FIG. 5.4 – Calcul de M(0) et séparation des chiffrés selon M(0)=0 et M(0)=1

et que pour tout chiffré$_8$ qui avec clé$_8 = K$ donne $clair_1 = 0$ est réalisé avec la probabilité

$$p(clair_1 = 0) \quad = \quad p(g(\text{chiffré}_8, \text{clé}_8)_1 = 0) \quad = \quad 1 \quad \forall \text{ chiffré}_8 \quad \text{pour clé}_8 = \text{K} \tag{5.3}$$

Ces deux groupes diffèrent toujours de M(0) en entrée de SubBytes et cette différence provoque une différence de consommation entre ces deux ensembles. La porte mise en œuvre n'est sans doute plus une porte " Xor2 " mais on suppose que le phénomène s'applique à tout type de porte. La figure 5.5 illustre le cas où Ks est une vraie valeur.

En supposant que Ks soit un fausse valeur, alors l'inverse de cet octet par AddRoundKey est faux pour tous les chiffrés. Si l'octet obtenu après AddRoundKey est faux alors l'inverse de ShiftRows est faux pour tous les chiffrés. Si l'octet obtenu après ShiftRows est faux alors l'inverse de SubBytes est faux pour tous les chiffrés. Pour Ks faux, l'inverse de cet octet est faux pour tous les chiffrés. Sur 256 valeurs possibles en entrées de SubBytes une moitié donnent M(0) égal à 0 et l'autre moitié donne M(0) égal à 1. Pour la moitié des valeurs donnant M(0) égal à 0, statistiquement une moitié de celles-ci sont de vraies valeurs et une moitié sont de fausses valeurs parce que :

$$\begin{cases} p(clair_1 = 1) \quad = \quad p(g(\text{Chiffré}_8, \text{clé}_8)_1 = 1) \quad \simeq \quad \frac{1}{2} \quad \forall \text{ chiffré}_8 \quad \text{pour clé}_8 \neq K \\ p(clair_1 = 0) \quad = \quad p(g(\text{Chiffré}_8, \text{clé}_8)_1 = 0) \quad \simeq \quad \frac{1}{2} \quad \forall \text{ chiffré}_8 \quad \text{pour clé}_8 \neq K \end{cases} \tag{5.4}$$

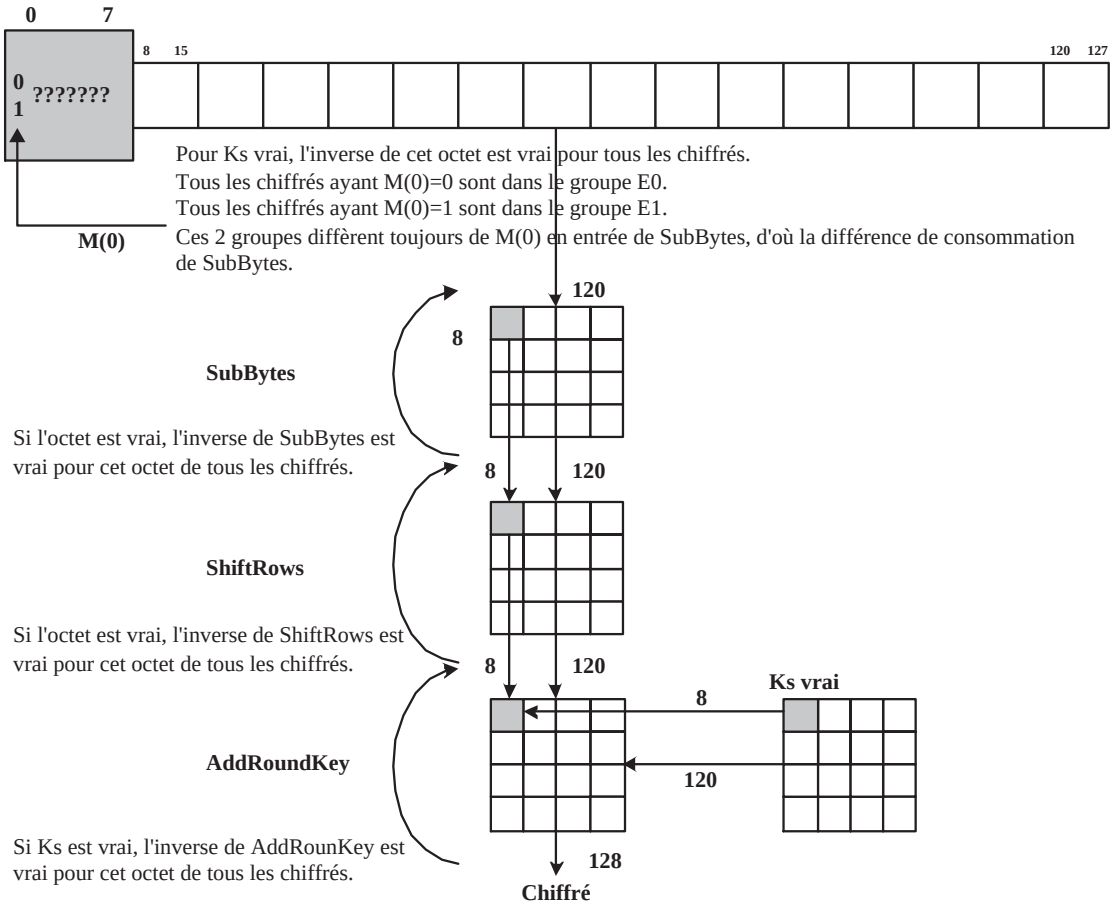


FIG. 5.5 – Cas où Ks est la vraie valeur

E0 est alors composé statistiquement pour moitié de fausses valeurs et pour moitié de bonnes valeurs. E1 est aussi composé pour moitié de fausses valeurs et pour moitié de bonnes valeurs. E0 et E1 ne diffèrent plus de M(0), leurs consommations moyennes tendent vers la même valeur et la différence tend vers 0. La figure 5.6 illustre le cas où Ks est une fausse valeur.

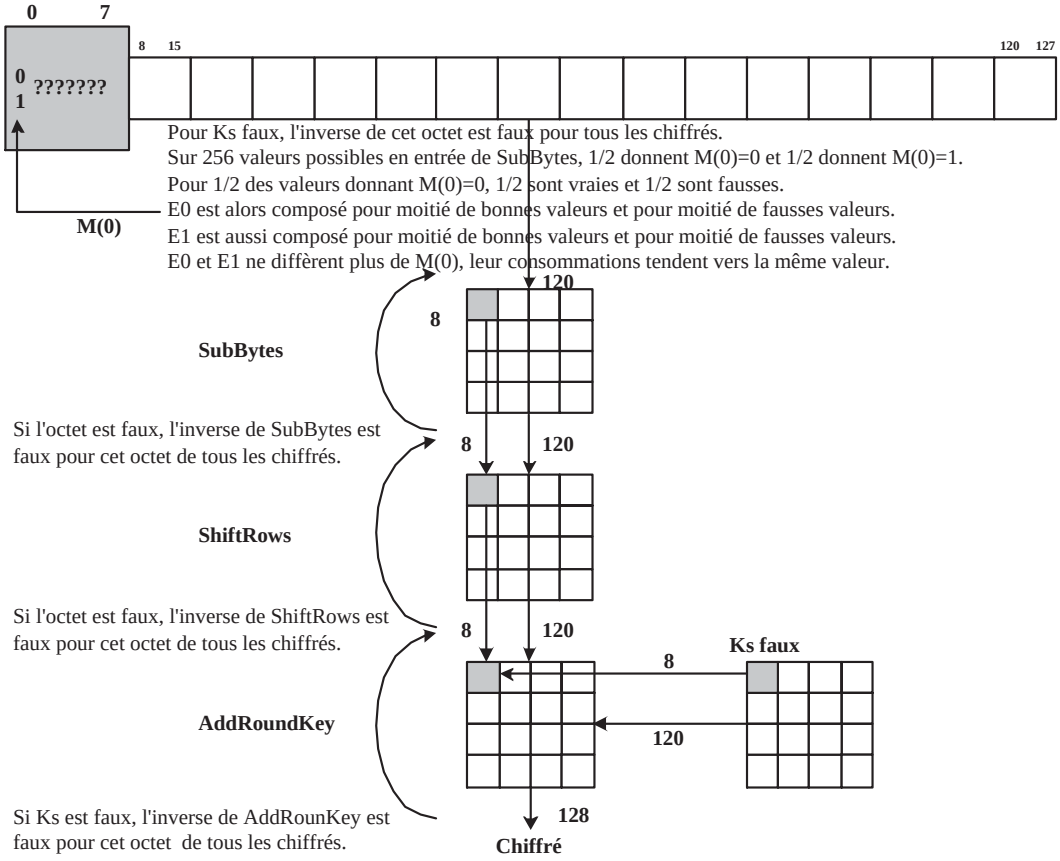


FIG. 5.6 – Cas où Ks est une fausse valeur

5.3 Les regroupements

Les chiffrés peuvent être séparés en deux ensembles E1 et E0 pour une valeur donnée Ks de l'octet de clé correspondant au bit M(0). Ks peut prendre 256 valeurs, on réalise donc 256 regroupements E1 et E0 à partir des mêmes chiffrés comme illustré dans la figure 5.7.

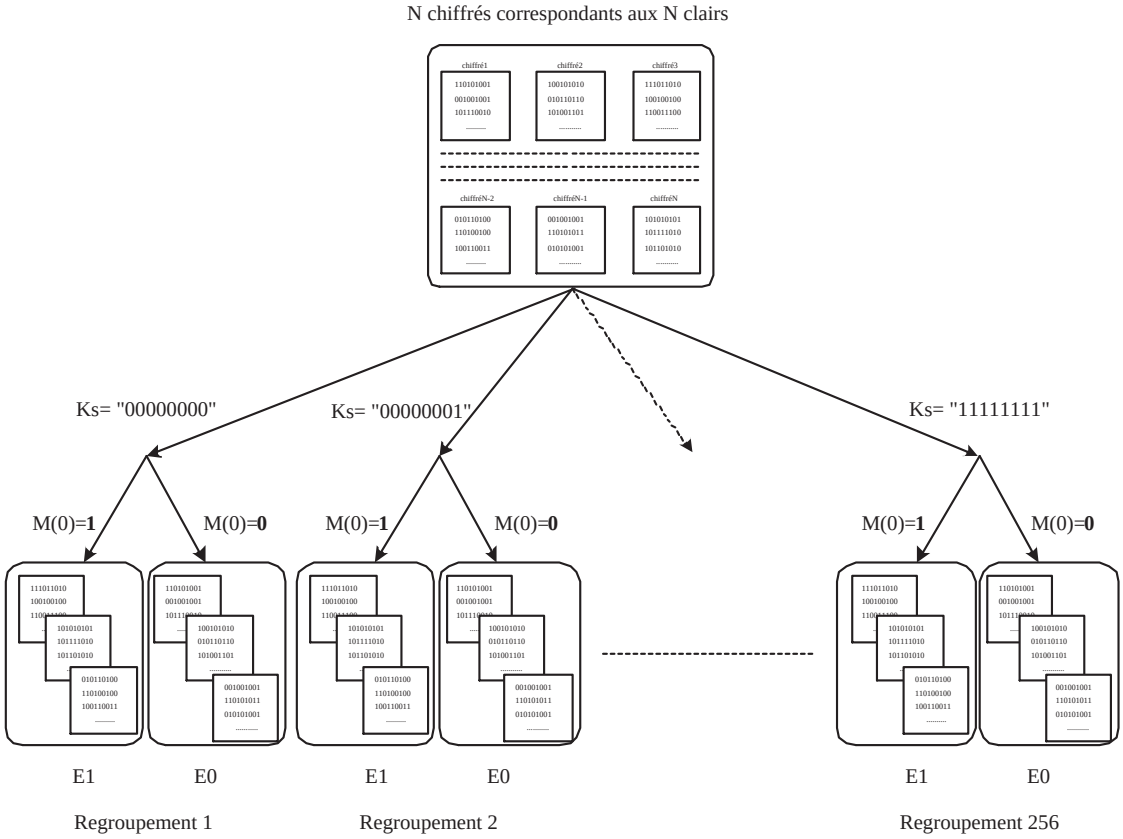


FIG. 5.7 – Les 256 regroupements des chiffrés pour Ks

Comme pour le DES, les regroupements sont réalisés pour les chiffrés et pour les traces de consommations (Figure 5.8). Pour chaque ensemble E1 et chaque ensemble E0 de chaque regroupement les moyennes de consommation sont calculées. La différence de ces deux moyennes est ensuite évaluée.

En supposant que "01001011" soit le vrai octet Ks de sous clé, cela signifie que le calcul pour séparer les chiffrés et donc les consommations respectives du circuit a été correctement réalisé. L'ensemble E1 représente donc bien ce qui s'est réellement passé dans le circuit et l'opération SubBytes a toujours mis en œuvre M(0) égal à 1. L'ensemble E0 représente aussi ce qui s'est réellement passé dans le circuit et l'opération SubBytes a toujours mis en œuvre M(0) égal à 0. Il existe une différence entre les moyennes de consommation de E1 et E0 parce qu'ils ont le bit M(0) différent.

En supposant que "01001011" soit un faux octet Ks de sous clé comme c'est le cas dans la figure 5.6, cela signifie que le calcul pour séparer les chiffrés et donc les consommations respectives du

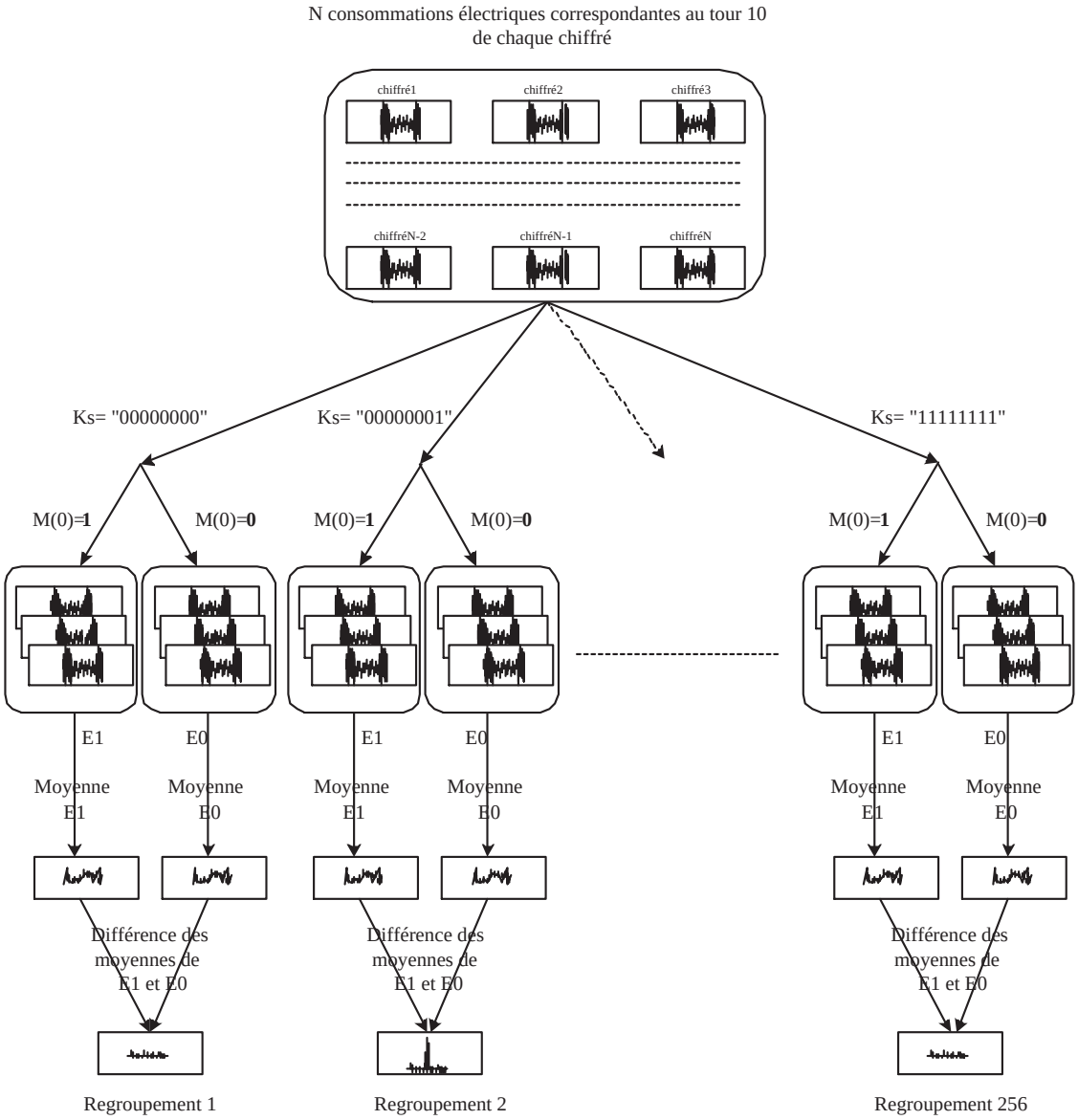


FIG. 5.8 – Les 256 regroupements des consommations correspondantes pour Ks

circuit a été mal réalisé. L'ensemble E1 ne représente plus ce qui s'est réellement passé dans le circuit et l'opération SubBytes a mis en œuvre M(0) égal à 1 et M(0) égal à 0. L'ensemble E0 aussi ne représente plus ce qui s'est réellement passé dans le circuit et l'opération SubBytes a mis en œuvre M(0) égal à 0 et M(0) égal à 1. La répartition est statistiquement équilibrée, E1 et E0 sont composés pour moitié de fausses valeurs et pour moitié de vraies valeurs. Leurs consommations moyennes tendent vers la même valeur. La différence des moyennes de consommation tend vers 0.

Si "01001011" est un faux octet de sous clé alors la séparation des chiffrés est réitérée avec Ks égal à "00000000", ..., " 11111111 ". L'octet de sous clé générant la plus grande différence de consommation après séparation est le vrai octet de sous clé. Pour les autres bits de la sous-clé, la même approche est réalisée avec un autre octet du message.

L'attaque développée sur le dernier tour de l'AES avec la connaissance des chiffrés et de leurs consommations est réalisable sur le premier tour avec la connaissance des clairs et de leurs consommations. Ceci est illustré dans la figure 5.9.

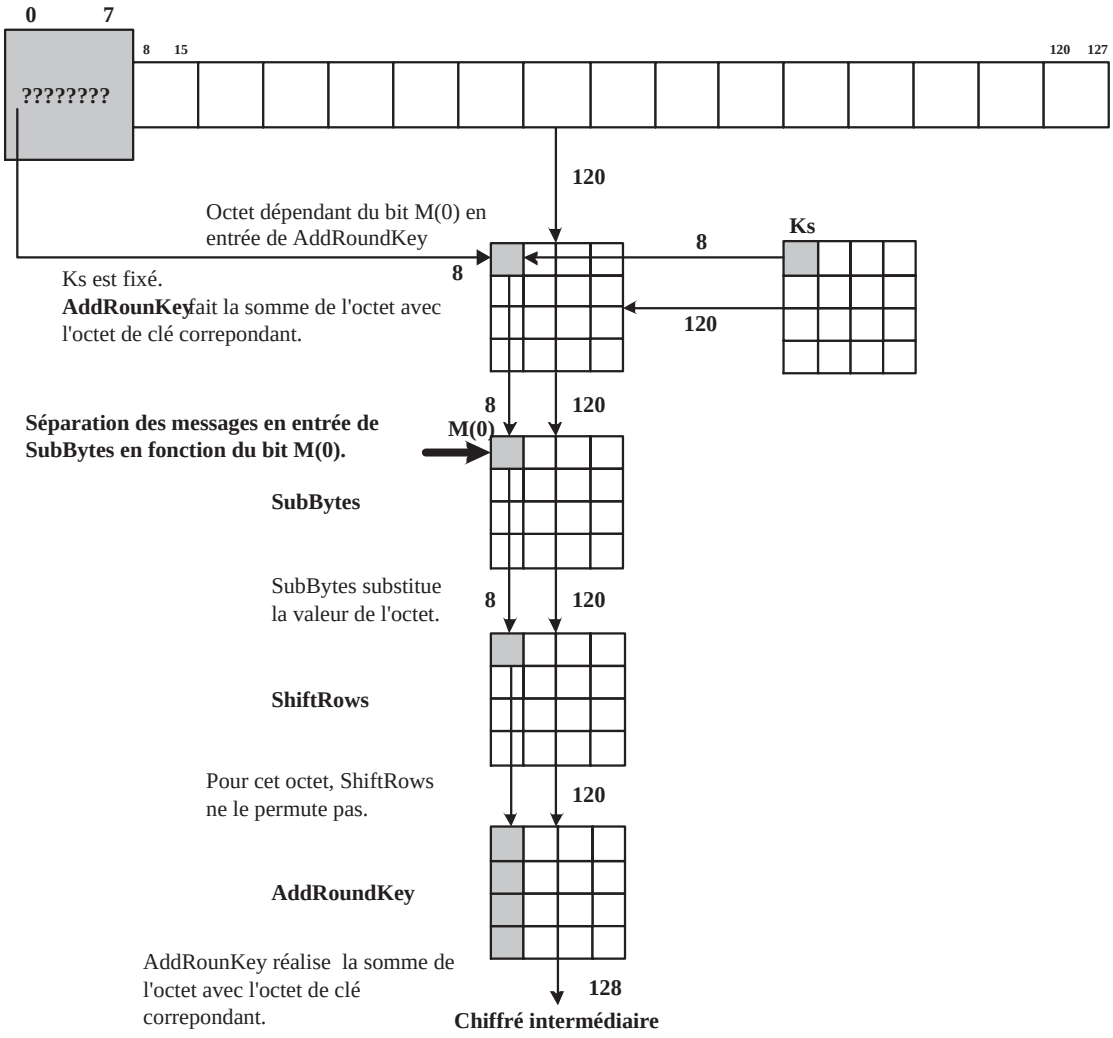


FIG. 5.9 – Mise en œuvre de la DPA sur le tour 1 de l'AES

5.4 Conclusion

Bien que l'AES soit le nouveau standard de chiffrement public, la DPA est toujours applicable et exploitable.

Chapitre 6

Variante de la DPA

Sommaire

6.1 Mesures des consommations

Les conditions de l'attaque et les mesures de consommations sont identiques à celles du DES et de l'AES. Elles peuvent tout à fait être reprises pour cette variante de la DPA sur les deux algorithmes.

6.2 Analyse des algorithmes DES et AES

Jusqu'à présent, une relation entre les chiffrés (ou les clairs), la clé et la consommation du circuit intégré a toujours été exploitée pendant les opérations de chiffrement afin de déterminer exactement la valeur d'un bit ou groupe de bits de clé. Il existe une autre possibilité d'exploitation de la consommation en ne se basant que sur les chiffrés et en ne faisant aucune supposition à priori sur la valeur des bits de clé. Afin d'introduire cette variante de la DPA sur les algorithmes DES et AES, il est possible de se focaliser sur une autre partie plausible d'architecture matérielle de leur implémentation sur un circuit intégré. La figure 4.5 indique que pendant l'exécution du DES, les bits de clé K16(18 :23) sont combinés bit à bit avec les bits E(18 :23) par un Xor. De même, la figure 5.2 montre que pendant l'exécution de l'AES, les bits de chaque octet du message intermédiaire sont aussi combinés bit à bit avec les bits de clé Ks par un Xor. La figure 6.1 résume ces deux remarques.

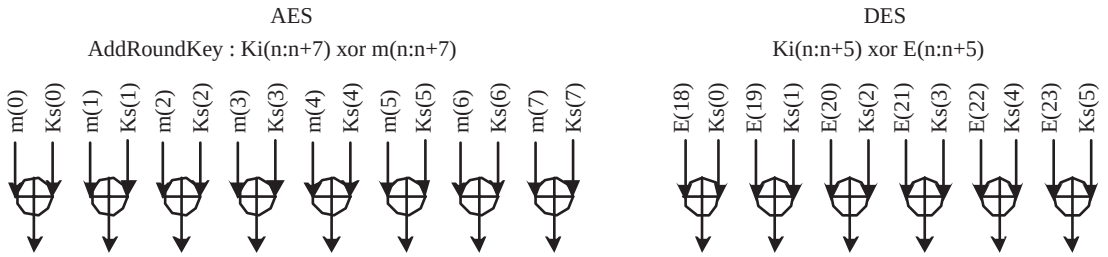


FIG. 6.1 – Analyse des algorithmes DES et AES pour une variante de la DPA

Les hypothèses de travail restent les mêmes que précédemment pour la DPA du DES et de l'AES. Les clairs et les chiffrés sont toujours considérés aléatoires. Que ce soit pour le message intermédiaire m de l'AES ou la sortie E de la fonction d'expansion du DES, ces deux groupes de bits, m(0 :7) et E(18 :23) dans l'exemple, sont donc aléatoires ainsi que leurs sorties. Ceci représente de nouvelles fonctions de sélection DPA ayant les formes suivantes :

$$\begin{aligned} \text{chiffré}_1 &= f(clair_1, \text{clé}_1) \\ clair_1 &= g(\text{chiffré}_1, \text{clé}_1) \end{aligned} \tag{6.1}$$

En revanche, les bits de clé associés sont fixes et cette remarque permet de réaliser une variante de la DPA. En effet, en considérant une porte Xor dont l'une des entrées est fixe et l'autre aléatoire, alors deux hypothèse sur la valeur du bit fixe sont possibles : soit elle vaut 0, soit elle vaut 1. En revanche, le deuxième bit aléatoire prend indifféremment les valeurs 0 et 1 comme illustré dans la figure 6.2.

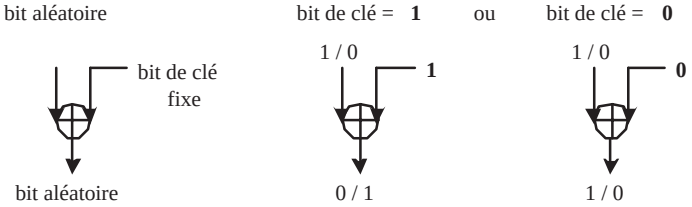


FIG. 6.2 – Les deux cas possibles pour un bit de clé fixe

La sortie des portes Xor est connue dans l'AES parce qu'elle correspond au chiffré dans le dernier tour. La sortie des portes Xor du DES est déductible de la même façon que pour calculer f(K16,L16) à partir des chiffrés. Lorsque le bit i du chiffré C vaut 1 et sachant que $m(i) \oplus k(i) = c(i)$, cela signifie que l'opération a potentiellement été $1 \oplus 0$ ou $0 \oplus 1$. De même, lorsque le bit i du chiffré C vaut 0, cela signifie que l'opération a potentiellement été $1 \oplus 1$ ou $0 \oplus 0$.

Mais le bit de clé est fixe pendant les opérations. En supposant qu'il soit égal à 1, tous les chiffrés ayant le bit C(i) égal à 1 correspondent seulement à la mise en œuvre de l'opération $0 \oplus 1$ dans le circuit intégré. En regroupant tous les chiffrés ayant C(i) égal à 1 et aussi leurs consommations respectives selon ce bit, la moyenne de consommation tend vers une valeur moyenne $M_1 1$. De même, tous les chiffrés ayant le bit C(i) égal à 0 correspondent seulement à la mise en œuvre de l'opération $1 \oplus 1$ dans le circuit intégré avec le bit de clé égal à 1. En regroupant tous les chiffrés ayant C(i) égal à 0 et aussi leurs consommations respectives selon ce bit, la moyenne de consommation tend vers une valeur moyenne $M_1 0$. La différence entre $M_1 1$ et $M_1 0$ sera notée $\triangle_1$ et correspond à la différence de moyennes de consommation du circuit intégré entre les opérations $0 \oplus 1$ et $1 \oplus 1$.

Dans le cas où le bit de clé fixe pendant les opérations serait égal à 0, tous les chiffrés ayant le bit C(i) égal à 1 correspondent cette fois-ci à la mise en œuvre de l'opération $1 \oplus 0$ dans le circuit intégré. En regroupant toujours tous les chiffrés ayant C(i) égal à 1 et aussi leurs consommations respectives selon ce bit, la moyenne de consommation tend cette fois-ci vers une valeur moyenne $M_0 1$. De même, tous les chiffrés ayant le bit C(i) égal à 0 correspondent à la mise en œuvre de l'opération $0 \oplus 0$ dans le circuit intégré avec le bit de clé égal à 0. En regroupant tous les chiffrés ayant C(i) égal à 0 et aussi leurs consommations respectives selon ce bit, la moyenne de consommation tend vers une valeur moyenne $M_0 0$. La différence entre $M_0 1$ et $M_0 0$ sera notée $\triangle_0$ et correspond à la différence de moyennes de consommation du circuit intégré entre les opérations $1 \oplus 0$ et $0 \oplus 0$. La figure 6.3 illustre ces différents cas de moyennes possibles.

6.3 Le regroupement

Comme mentionné précédemment, le bit de clé reste fixe pendant la mise en œuvre de la DPA. Cela signifie qu'une seule différence de moyennes est calculée qui est soit $\triangle_1$, soit $\triangle_0$. Cependant, il est impossible de dire à l'avance laquelle est obtenue. C'est là qu'il faut reprendre les figures 4.8 et 5.7. Dans le cas du DES et de l'AES, les bits de clé ne sont jamais traités seuls mais par groupes. S'il est possible de faire un regroupement selon un bit de clé, il est donc possible d'en faire autant qu'il y a de bits de la clé. Et pour chaque regroupement il est possible de calculer la différence de moyennes de consommation qui est soit $\triangle_1$, soit $\triangle_0$ pour chaque bit. En supposant

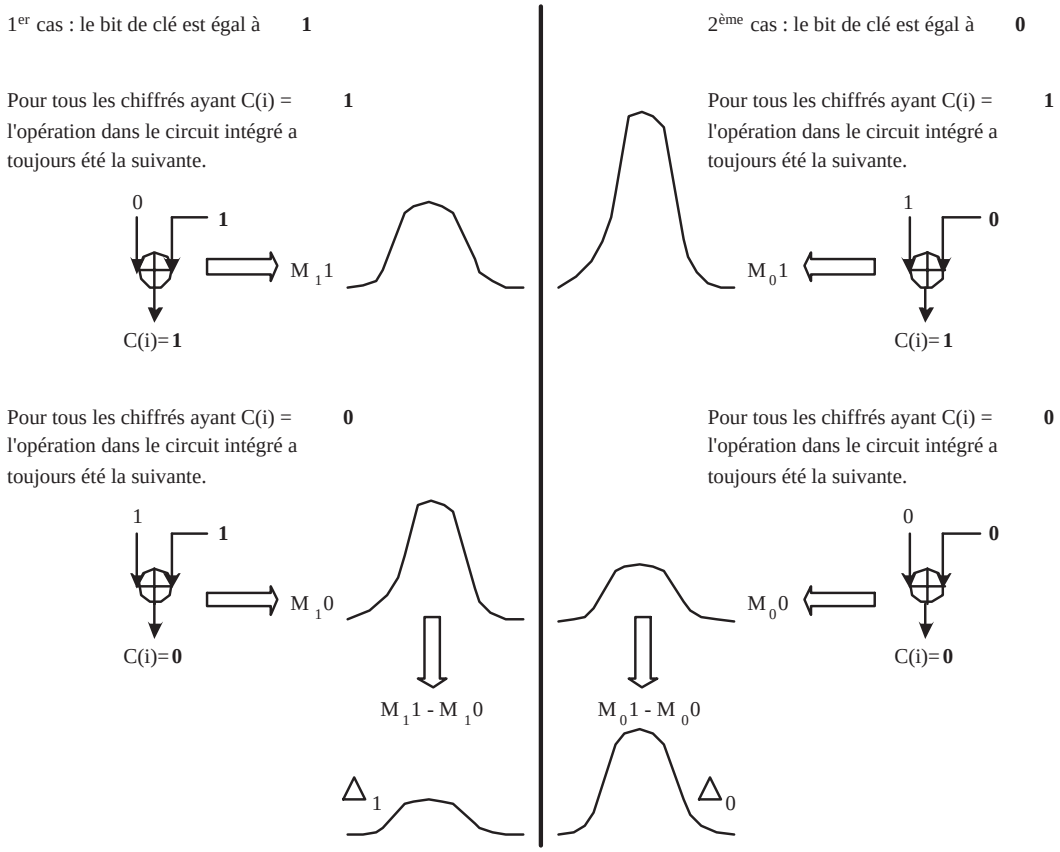


FIG. 6.3 – Différences de moyennes de consommation lorsque le bit de clé fixe est égal à 1 ou à 0

que toutes les différences de moyennes soient identiques pour tous les bits de clé, l'information obtenue indique que tous les bits de clé sont égaux. Alors deux choix sont possibles : soit tous les bits valent 1, soit ils valent tous 0. La décision est faite par déchiffrement avec les deux clés possibles. Dans le cas ou les différences de moyennes de consommation différent d'un bit à l'autre parce qu'elles valent soit $\triangle 1$, soit $\triangle_0$, cela signifie que tous les bits dont la différence de moyennes est égale à $\triangle_1$ sont égaux entre eux et que tous les bits dont la différence de moyennes est égale à $\triangle_0$ sont aussi égaux entre eux. En revanche, ces deux ensembles de bits sont différents. Comme précédemment, deux choix sont possibles : soit que tous les bits dont la différence de moyennes de consommation est $\triangle_1$ sont égaux à 1 et que tous les bits dont la différence de moyennes de consommation est $\triangle_0$ sont égaux à 0, soit que tous les bits dont la différence de moyennes de consommation est $\triangle_1$ sont égaux à 0 et que tous les bits dont la différence de moyennes de consommation est $\triangle_0$ sont égaux à 1. La décision est faite par déchiffrement avec les deux clés possibles. La figure 6.4 résume les cas possibles.

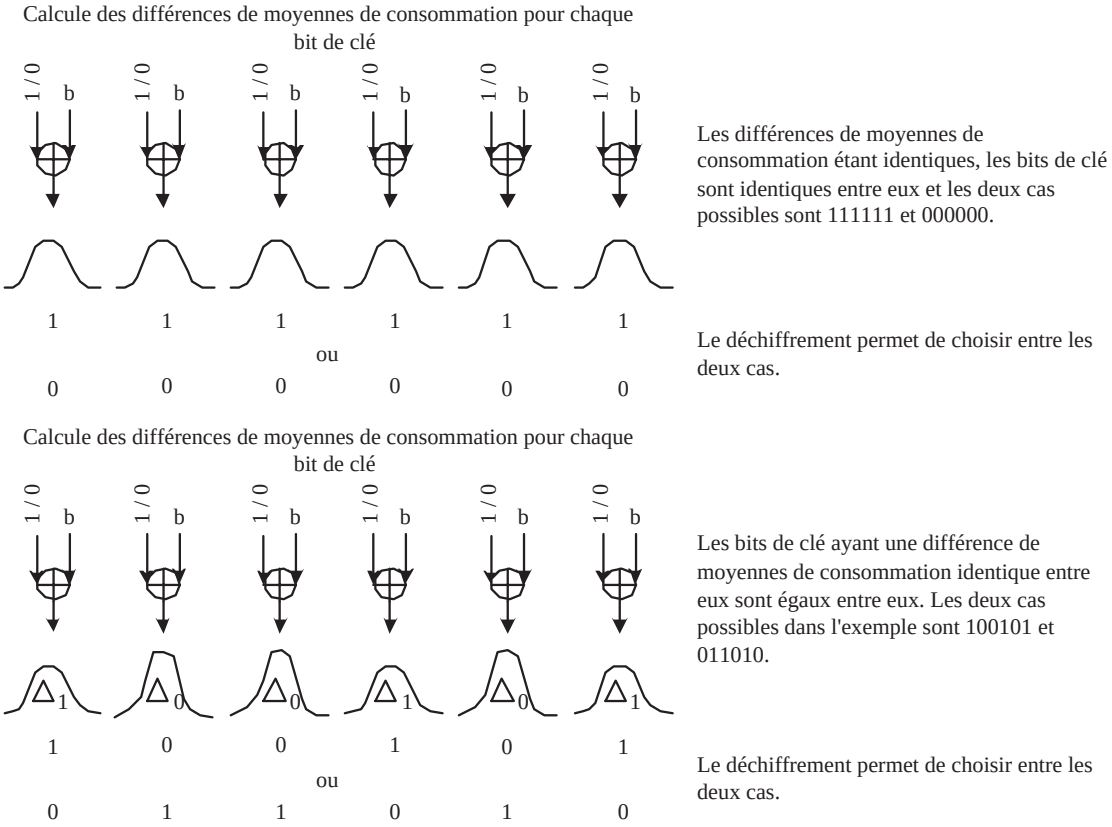


FIG. 6.4 – Calcul des différences de moyennes de consommation pour tous les bit de clé

Dans le cadre de l'AES, le regroupement consiste pour chaque bit C(i) des chiffrés à les séparer en 128 groupes d'ensembles E1 et E0 selon que le bit vaut 1 ou 0. Puis, pour chaque ensemble E1, la moyenne de consommation est calculée pour donner une valeur $M_{1/0}1(i)$, c'est à dire la moyenne de consommation pour le bit i quand il vaut 1 ne sachant pas si le bit de clé vaut 1 ou 0. De même, pour chaque ensemble E0, la moyenne de consommation est calculée pour donner une valeur $M_{1/0}0(i)$, c'est à dire la moyenne de consommation pour le bit i quand il vaut 0 ne sachant pas si le bit de clé vaut 1 ou 0. A partir de ces deux moyennes, la différence est calculée pour donner la différence de moyennes de consommation qui vaut soit $\triangle_1$ soit $\triangle_0$. Les 128 regroupements pour l'AES sont illustrés par la figure 6.5.

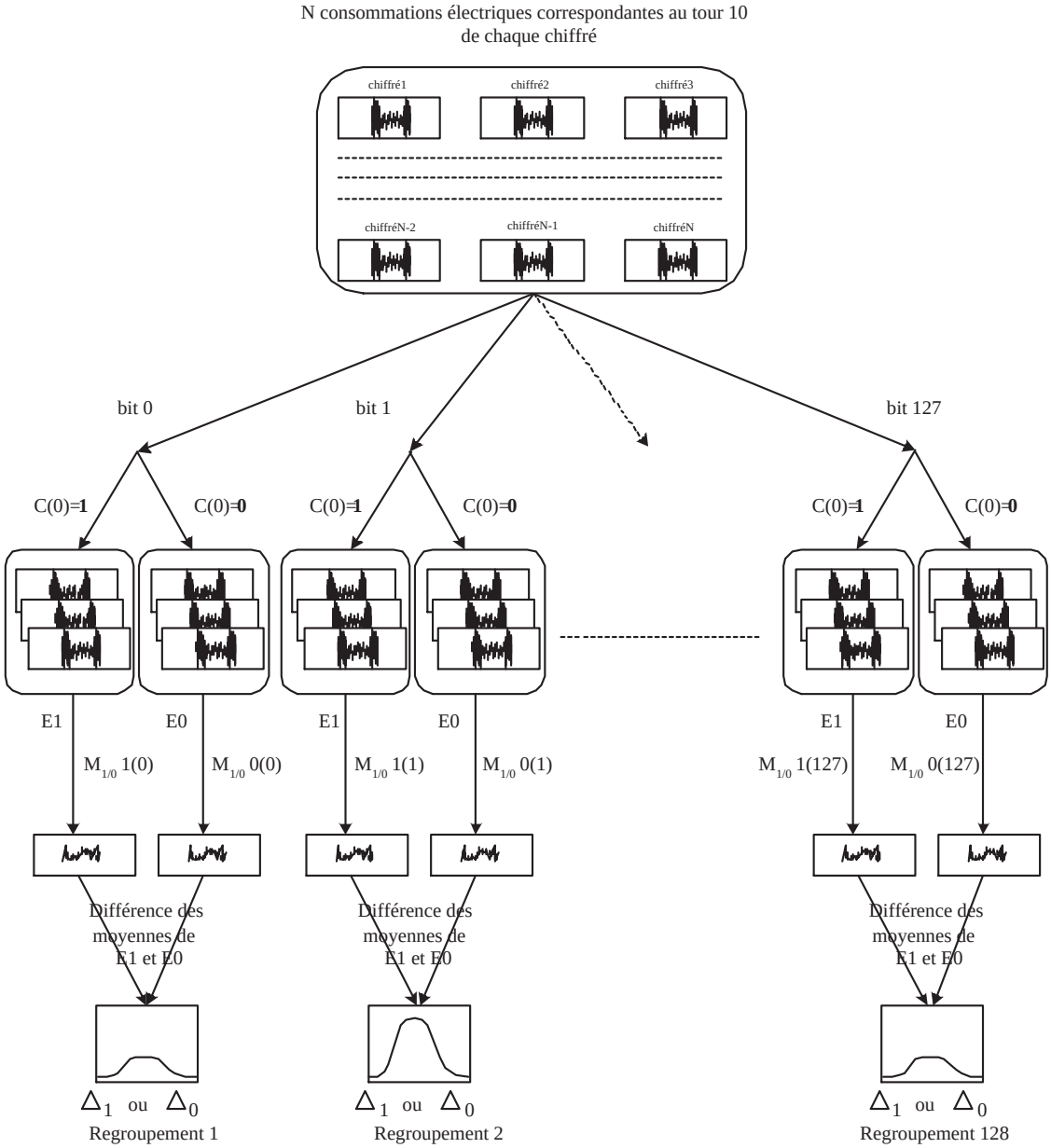


FIG. 6.5 – Les 128 regroupements de la variante DPA pour l'AES

Dans le cadre du DES et en reprenant la figure 4.5, la sortie de la fonction d'expansion E(18 :23) est calculée à partir de de L16(12 :17). Le regroupement consiste pour chaque bit E(i) des chiffrés à les séparer en 6 groupes d'ensembles E1 et E0 selon que le bit vaut 1 ou 0. Puis, pour chaque ensemble E1, la moyenne de consommation est calculée pour donner une valeur $M_{1/0}1(i)$. De même, pour chaque ensemble E0, la moyenne de consommation est calculée pour donner une valeur $M_{1/0}0(i)$. A partir de ces deux moyennes, la différence est calculée pour donner la différence de moyennes de consommation qui vaut soit $\triangle_1$ soit $\triangle_0$. Le regroupement pour le DES pour les bit E(18 :23) est illustré par la figure 6.6. Pour les autres bits, il est nécessaire de recommencer en sélectionnant une autre boîte de substitution, ce qui revient à faire en tout 48 regroupements.

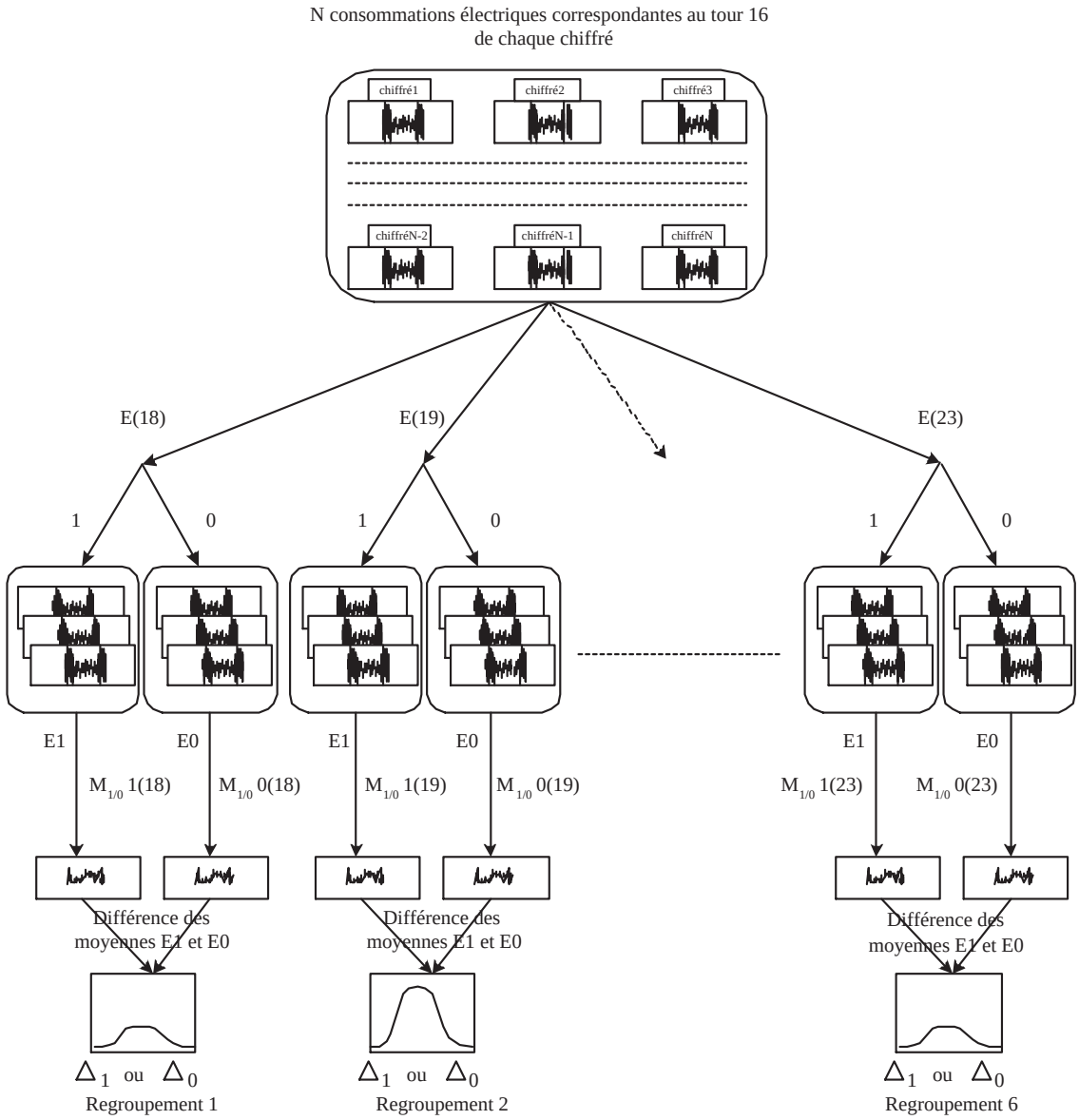


FIG. 6.6 – Les 6 regroupements de la variante DPA pour les bits E(18 :23) du DES

6.4 Conclusion

Cette variante de la DPA permet d'améliorer l'attaque en exploitant toujours le comportement différent de la porte Xor mais dans une autre partie matérielle plausible des algorithmes. En effet, dans le cas de l'AES, le nombre de regroupements total est passé de 256 (par octet) fois 16 (nombre d'octets de sous-clé) c'est à dire de 4096 à 128 pour récupérer la sous-clé. Dans le cas du DES ce nombre est passé de 64 (par fonction de substitution) fois 8 (nombre de fonctions de substitution), c'est à dire de 512 à 48.

Troisième partie

Sources microélectroniques de la DPA

Chapitre 7

Porte logique, réseau de transistors et effet mémoire

Sommaire

7.1 Source architecturale

Jusqu'à présent, la DPA a été vue de façon théorique essentiellement à partir des algorithmes. Seule l'existence d'un circuit intégré réalisant les fonctions DES ou AES a été considérée. Cependant, il existe plusieurs possibilités d'implantation de ces algorithmes et l'objectif de ce chapitre est de montrer que quel que soit le choix d'architecture fait par un concepteur, la ressource matérielle élémentaire, source de la DPA, peut toujours se réduire à un même élément. Plusieurs exemples d'architectures sont illustrés dans les articles [4, 6, 27, 30, 32, 48]. Un rappel sur trois grandes familles d'architectures souvent envisagées lors de la conception d'un système cryptographique est tout d'abord réalisé.

7.1.1 Architecture matérielle de coprocesseur

Le coprocesseur est un circuit intégré ou partie d'un circuit intégré qui a la particularité d'être totalement indépendant fonctionnellement de toute autre ressource matérielle. La seule dépendance fonctionnelle résiduelle est le chargement des données comme les clés, les messages clairs ou les messages chiffrés, le paramétrage éventuel du chiffrement ou du déchiffrement et la lecture du message chiffré ou déchiffré. Le coprocesseur est envisagé lorsque le matériel disponible n'est pas suffisamment performant pour respecter les spécifications. Une architecture classique de coprocesseur cryptographique est illustrée par la figure 7.1.

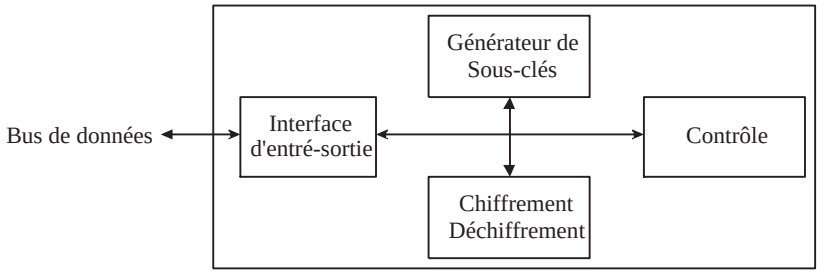


FIG. 7.1 – Architecture génerale d'un coprocesseur cryptographique

De façon générale elle se compose de la façon suivante :

- une interface d'entrée-sortie,
- un bloc réalisant la fonction de chiffrement-déchiffrement,
- un bloc générant les sous-clés pour les différents tours de l'algorithme,
- une machine d'état générant l'ensemble des commandes.

Les algorithmes de chiffrement symétriques sont très souvent conçus avec des fonctions élémentaires itérées sur différentes parties du message :

- une ou plusieurs fonctions de substitution (8 pour le DES, 1 pour l'AES),

– une permutation des bits (schéma de Feistel pour le DES, fonction ShiftRows pour l'AES),

– un mélange des bits de données avec des bits de sous-clés (fonction f pour le DES, fonction AddRoundKey pour l'AES),

– une itération d'un même tour (16 tours pour le DES, 10, 12 ou 14 tours pour l'AES).

Un algorithme de chiffrement symétrique se décompose souvent en un tour itéré n fois. Ce tour est toujours composé des mêmes fonctions réalisant des permutations, des substitutions et des mélanges. Ces fonctions ont, de plus, peu de bits de données et peu de bits de sous-clé en entrée et produisent peu de bits en sortie. Elles représentent le minimum de ressource matérielle à implanter dans un cryptoprocesseur. La vitesse d'exécution de l'algorithme détermine pour beaucoup l'architecture finale du cryptoprocesseur. Lorsque dans un système la vitesse n'est pas importante, les ressources matérielles pourront être les plus petites possibles. Elles seront les fonctions définissant un tour, des registres pour mémoriser les calculs intermédiaires et une machine d'état générant l'ensemble de commandes permettant l'accès séquentiel à chaque ressource matérielle des fonctions dans un tour et l'itération de chaque tour. La figure 7.2 rappelle l'algorithme AES avec une architecture matérielle minimale de cryptoprocesseur. Elle instancie le minimum de ressources matérielles qui sont : une fonction SubBytes, une fonction ShiftRows, une fonction MixColumn et une fonction AddRoundKey. Quelques registres intermédiaires sont nécessaires pour la fonction MixColumn ayant plus de 8 bits en entrée. Une machine d'état commande l'ensemble.

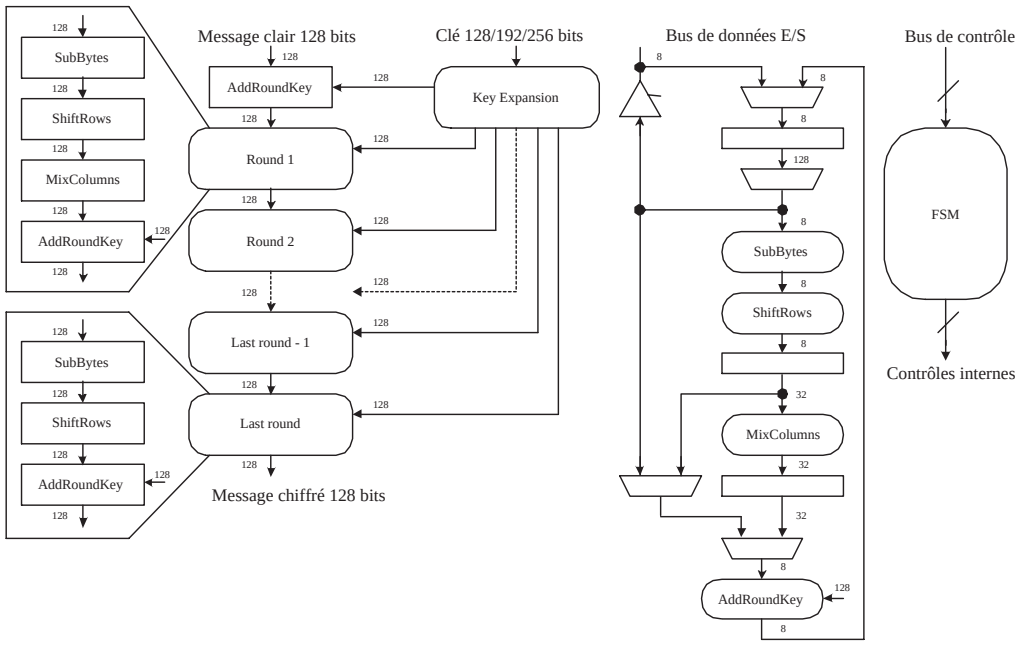


FIG. 7.2 – Algorithme AES et architecture matérielle minimale de cryptoprocesseur

L'architecture présentée dans la figure 7.2 réalise le traitement octet par octet pour chaque tour. Cela nécessite au minimum 4 périodes d'horloge pour obtenir l'entrée de 32 bits de la fonction MixColumn après exécution des fonctions SubBytes et ShiftRows, 1 période d'horloge

pour l'exécution de la fonction MixColumn, 4 périodes d'horloge pour mémoriser ce résultat. Pour finir le tour il faut réaliser encore 4 fois ces opérations sur les autres bits. Il faut au moins 36 périodes d'horloge pour un tour complet si la technologie permet de réaliser toutes les fonctions en une période. L'AES nécessite au minimum 10 tours, ce qui fait au moins 360 périodes d'horloge. Ce cryptoprocesseur est lent et il est tout à fait possible de dériver d'autres architectures plus rapides à partir de celle-ci. Cependant, elles implanteront toutes au minimum les ressources matérielles de l'architecture matérielle minimale. La figure 7.3 illustre une architecture dans laquelle un bus 128 bits est utilisé. Les fonctions minimales de l'AES y sont dupliquées autant de fois que nécessaire pour gérer à chaque étape du tour les 128 bits en parallèle. Dans cet exemple est considéré le fait que la technologie permet l'exécution d'un tour en une période d'horloge. Il faut au minimum 10 périodes d'horloge pour exécuter l'AES. Une architecture intermédiaire entre les deux présentées est disponibles dans [4].

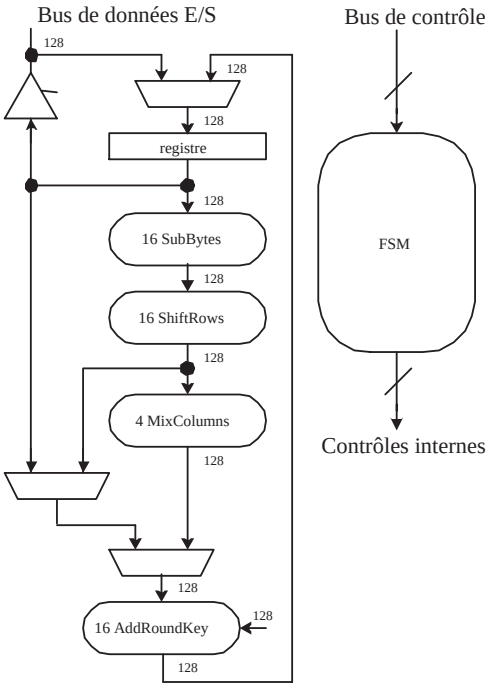


FIG. 7.3 – Architecture plus rapide de cryptoprocesseur AES

7.1.2 Architecture matérielle de microprocesseur avec ressource dédiée

La disponibilité d'un microprocesseur (Figure 7.4-a) peut remettre en cause l'utilisation d'un cryptoprocesseur. Tout dépend des spécifications et il est tout à fait possible qu'un microprocesseur soit suffisamment performant pour prendre à sa charge certaines opérations avec son propre jeu d'instructions mais pas toutes. Les fonctions les plus pénalisantes de l'algorithme sont alors réalisées par des ressources matérielles dédiées qui sont appelées par le microprocesseur. Cela peut s'apparenter à un cryptoprocesseur partiel ou à une modification de l'unité arithmétique et logique par adjonction d'instructions.

Un microprocesseur 8 bits est performant pour exécuter son jeu d'instructions. Mais la fonction MixColumn de l'AES, qui nécessite un opérande de 32 bits, pourrait être pénalisante. L'ad-

jonction d'une nouvelle instruction matérielle dédiée à l'unité arithmétique et logique permet d'améliorer les performances. Un microprocesseur sans instruction de traitement d'un bit pourrait être pénalisé par des fonctions de permutation bit à bit comme la fonction P de l'algorithme DES. Celle-ci aussi pourrait être réalisée par l'adjonction d'une nouvelle instruction. La décision du concepteur est prise en fonction des spécifications et d'un ensemble de contraintes liées à son projet. Il est envisageable de trouver les fonctions des algorithmes soit sous forme d'un cryptoprocesseur partiel soit sous forme d'une nouvelle instruction ajoutée au jeu d'instructions du microprocesseur. La figure 7.4-b illustre une modification d'un microprocesseur avec l'ajout de la fonction *SubBytes* au jeu d'instructions.

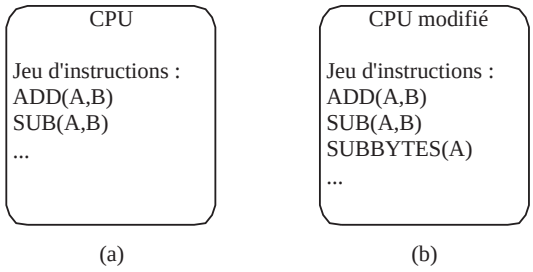


FIG. 7.4 – Microprocesseur et ressource matérielle dédiée

7.1.3 Architecture matérielle de microprocesseur seul

Ce troisième cas est à l'opposé de celui du cryptoprocesseur. Le microprocesseur est suffisamment performant pour prendre à sa charge l'ensemble de l'algorithme tout en respectant les spécifications de performance. L'algorithme est écrit sous la forme d'un programme utilisant le jeu d'instructions du microprocesseur. Chaque fonction de l'algorithme est un sous-programme défini par une suite d'instructions. Par exemple, la fonction *SubBytes* de l'AES pourrait être composée d'instructions ADD, SUB, XOR ... utilisant des registres de travail internes au microprocesseur réalisant ainsi le calcul de substitution d'un octet (Figure 7.5-a). Une mémoire contenant la résolution de la substitution est possible. L'octet à substituer représente l'adresse dans la mémoire qui contient la valeur substituée (Figure 7.5-b).

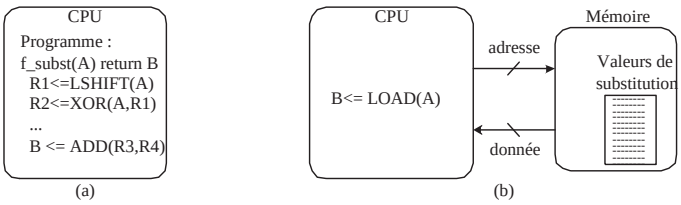


FIG. 7.5 – Microprocesseur seul

Il est maintenant possible de mettre en évidence la ressource matérielle élémentaire source de la DPA à partir des trois grandes familles d'architectures présentées et de l'illustration de la DPA aux chapitres 3, 4, 5 et 6.

7.1.4 Source matérielle élémentaire de la DPA dans une architecture de coprocesseur

La DPA sur l'AES est réalisée au chapitre 5 sur la fonction de substitution *SubBytes*. La conception de cette fonction commence dans un premier temps par sa description fonctionnelle de haut niveau en langages VHDL ou Verilog qui peut prendre de nombreuses formes. Il est possible de l'écrire sous la forme d'équations booléennes en définissant chaque bit de sortie de *SubBytes* en fonction des bits d'entrées, d'affecter le résultat conditionnellement à la valeur de l'octet à substituer, etc. Dans un deuxième temps, la synthèse transforme la description de haut niveau en équations booléennes. Les différentes fonctions booléennes peuvent alors être conçues physiquement en utilisant des portes logiques d'une librairie d'un fondeur. Chaque porte logique de la librairie est un opérateur ayant quelques bits en entrée fournissant un bit de résultat en sortie. La porte logique la plus commune est la porte $NAND2$, c'est à dire une porte réalisant la fonction logique NAND entre deux bits. A partir de cette porte tout peut être construit (Figure 7.6).

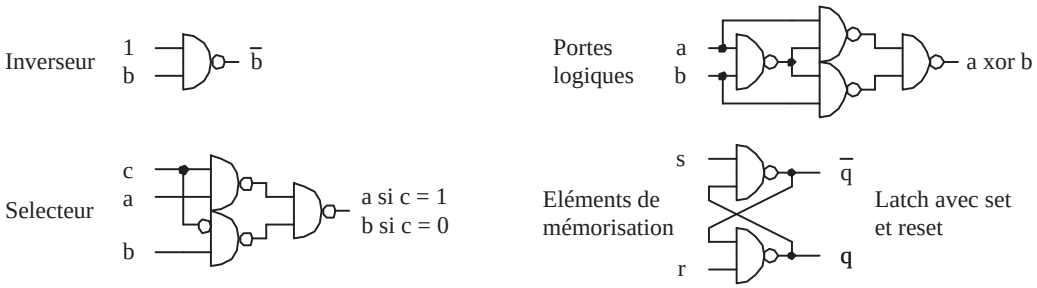


FIG. 7.6 – Conception à partir de la porte NAND2

La fonction *SubBytes* après synthèse est composée d'un ensemble de portes logiques de la librairie du fondeur. Chaque bit d'entrée du premier octet M(0 :7) du message (Chapitre 5) est l'entrée d'une porte logique. Comme montré précédemment (Figure 7.6), il est raisonnable de considérer que le bit M(0), qui a été utilisé pour mettre en œuvre la DPA, soit l'entrée d'une porte NAND2 (Figure 7.7).

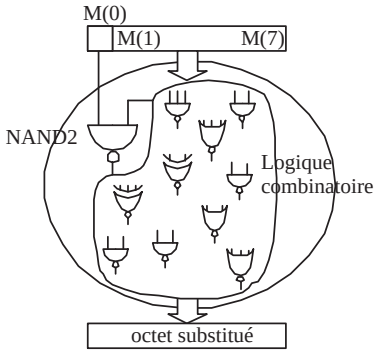


FIG. 7.7 – Exemple d'architecture de la fonction *SubBytes* pour un cryptoprocesseur

La ressource matérielle élémentaire pouvant être la source architecturale de la DPA dans le cas d'un coprocesseur est assimilable à la porte NAND2.

7.1.5 Source matérielle élémentaire de la DPA dans une architecture de microprocesseur avec ressource dédiée

La conception de la ressource dédiée commence aussi par sa description fonctionnelle de haut niveau en langages VHDL ou Verilog. En supposant que la ressource dédiée soit la fonction *SubBytes*, il est à nouveau possible de l'écrire sous les formes décrites précédemment. La synthèse transforme la description de haut niveau en équations booléennes qui peuvent alors être conçues physiquement en utilisant des portes logiques d'une librairie d'un fondeur (Figure 7.8). La porte logique la plus commune reste la porte $NAND2$ citée précédemment. Il est possible de prendre à nouveau comme hypothèse que la ressource matérielle élémentaire pouvant être la source architecturale de la DPA dans le cas d'une ressource dédiée d'un microprocesseur est la porte NAND2.

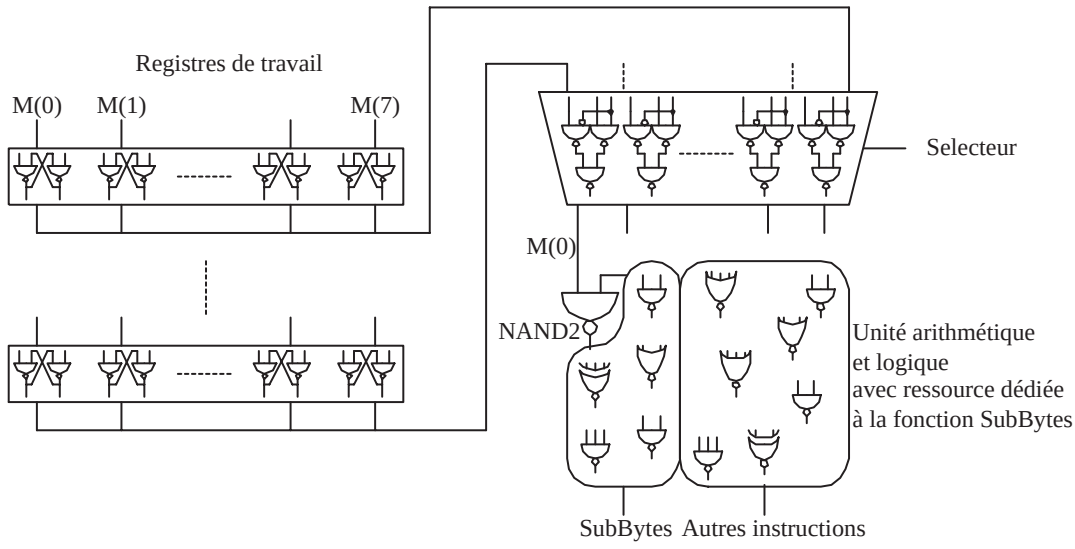


FIG. 7.8 – Exemple d'architecture de ressource dédiée à la fonction *SubBytes*

7.1.6 Source matérielle élémentaire de la DPA dans une architecture de microprocesseur seul

Le processus de conception d'un microprocesseur est identique à celui d'un coprocesseur ou d'une ressource dédiée. Le microprocesseur est décrit fonctionnellement par un langage de haut niveau VHDL ou Verilog. La synthèse transforme la description en équations booléennes et en éléments mémorisants qui peuvent alors être conçus physiquement en utilisant des portes logiques d'une librairie d'un fondeur (Figure 7.9). La porte logique la plus commune reste la porte $NAND2$ citée précédemment. Il est à nouveau possible de prendre comme hypothèse que la ressource matérielle élémentaire pouvant être la source architecturale de la DPA dans le cas d'un microprocesseur est la porte NAND2.

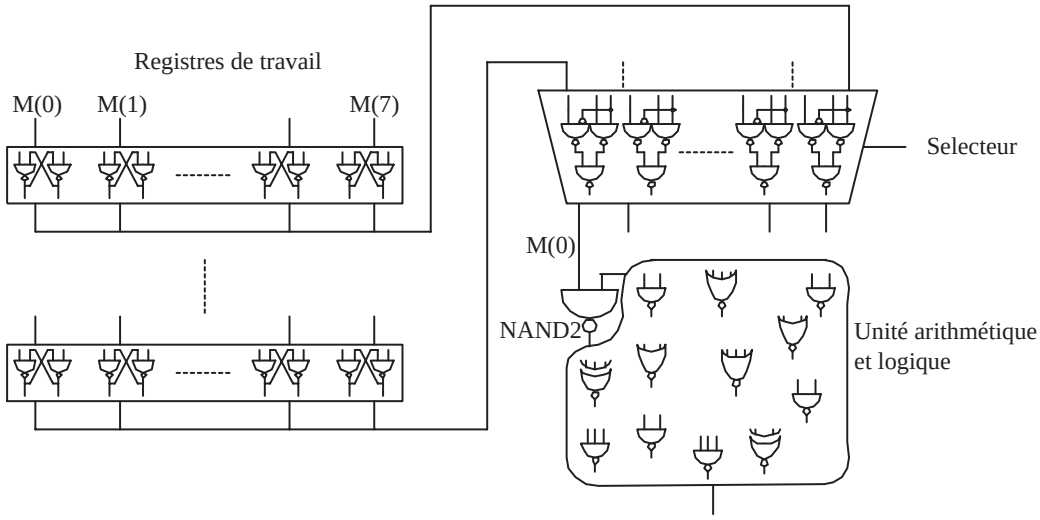


FIG. 7.9 – Exemple d'architecture de ressource dédiée à la fonction $SubBytes$

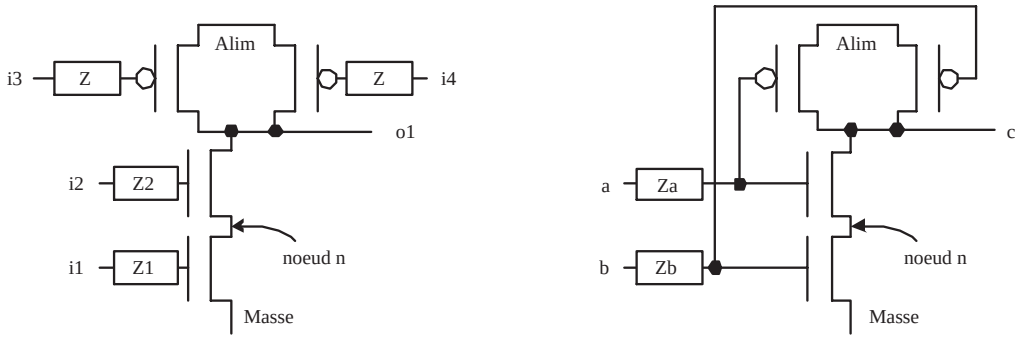
7.1.7 Source architecturale élémentaire de la DPA

Quelle que soit l'architecture envisagée lors de la conception, la source architecturale de la DPA correspond à une porte logique. En effet, que l'architecture du composant soit RISC, CISC, scalaire, avec mémoire cache, avec renommage des registres, avec plusieurs unités arithmétiques et logiques, etc, un bit d'une donnée pertinente sera toujours au minimum l'entrée d'une porte logique à deux entrées. C'est pourquoi dans la suite de la thèse, la porte NAND2 sera utilisée à des fins d'illustration pour les raisonnements. Cependant, ce choix ne modifie en rien tous les résultats obtenus basés sur cette hypothèse. Les conclusions sur les sources microélectroniques de la DPA restent identiques y compris si d'autres portes logiques étaient considérées.

7.2 Les réseaux série de transistors

Une porte CMOS est construite, comme son nom l'indique, à partir de deux réseaux complémentaires de transistors dont l'un est de type NMOS et l'autre de type PMOS. Chaque transistor de la porte peut être dans les différents états d'accumulation, de déplétion ou de forte inversion selon la différence de potentiel appliquée entre la grille et la source (Annexe A, [38] et [31]). A chaque état correspond un modèle électrique différent de transistor. Celui-ci évalue les variations de capacités grille-source, grille-drain et grille-substrat ainsi que les capacités de jonctions source-substrat et drain-substrat. Il calcule aussi les variations de résistance du canal.

La conception d'une fonction logique implique la connexion d'un ensemble de transistors drain-drain, drain-source, etc. Connecter deux ou plusieurs transistors modifie les paramètres électriques de chaque transistor et ce dans chaque état (Chapitre 1). Autrement dit, les paramètres électriques dépendent aussi des tensions appliquées entre le drain et la source qui elles-même dépendent des tensions de grille.



(a) Réseaux série et parallèle de transistors

(b) Porte NAND2 CMOS

Z1 : impédance équivalente vue par le nœud i1
Z2 : impédance équivalente vue par le nœud i2
Z : impédance équivalente vue par les nœuds i3, i4

Za : impédance équivalente vue par le nœud a
Zb : impédance équivalente vue par le nœud b

FIG. 7.10 – Les réseaux de transistors source de la DPA

Tous les transistors d'un réseau série d'une porte CMOS ont une tension de drain différente (Chapitre 1). De fait, ils ont aussi tous une tension de source différente. Ceci se traduit par des impédances équivalentes différentes pour chaque entrée de la porte connectée à une grille d'un transistor en série. En revanche, dès lors que les transistors sont en parallèle, les entrées ont des impédances équivalentes identiques (Chapitre 1). La figure 7.10-a illustre les impédances équivalentes pour les quatre entrées $i1$, $i2$, $i3$ et $i4$. La porte $NAND2$ est construite en connectant les entrées $i1$ à $i4$ et $i2$ à $i3$. Étant donné que les entrées $i1$ et $i2$ ont toujours des impédances équivalentes différentes, les deux entrées a et b ont des impédances équivalentes différentes (Figure 7.10-b). Ceci reste vrai quels que soient les états des transistors (Chapitre 1).

Les évènements 2 et 15 (Figure 7.11-a et figure 7.11-b, chapitre 1) correspondent au même résultat logique pour la porte $NAND2$. Une simulation électrique d'une description SPICE, qui est exactement la représentation schématique de la figure 7.10-a, illustre bien la différence d'impédance entre les deux entrées a et b. Elle est mise en évidence par une différence de consommation en courant entre les deux évènements (Figure 7.11-c). Or lors de la mise œuvre de la DPA au

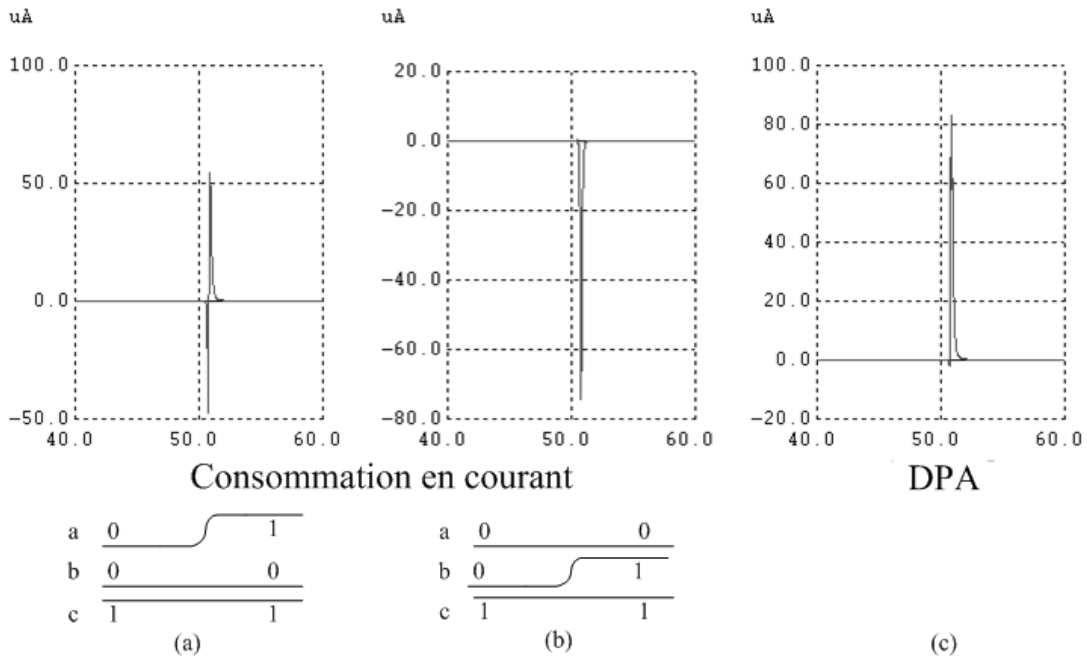


FIG. 7.11 – Simulation des réseaux de transistors source de la DPA

chapitre 5, l'évènement 2 se trouve dans l'ensemble E0 et l'évènement 15 se trouve dans l'ensemble E1 après le regroupement des évènements en ensembles DPA. Les réseaux de transistors des portes sont source de la DPA.

7.3 L'effet mémoire

Dans un réseau de transistors connectés en série, les nœuds d'interconnexion peuvent stocker une charge (Figure 7.10, nœud n). Dans la figure 7.12-a, les entrées a et b sont à l'état logique 00, la capacité C_3 vue par la sortie de la porte est chargée par les transistors PMOS qui sont passants et la capacité C_2 du nœud d'interconnexion est considérée vide. Ceci définit un premier état électrique de la porte $NAND2$.

Dans la figure 7.12-b, les entrées a et b sont à l'état logique 10. Dans ce cas, la capacité C_3 vue par la sortie de la porte est toujours chargée par le transistor PMOS commandé par l'entrée b. Cependant, le transistor NMOS commandé par l'entrée a est devenu passant. La capacité C_2 du nœud d'interconnexion est aussi chargée par le transistor PMOS commandé par l'entrée b.

Lorsque les entrées a et b sont à nouveau dans l'état logique 00 (Figure 7.12-c), le transistor NMOS commandé par l'entrée a est devenu ouvert piégeant ainsi les charges stockées précédemment par la capacité C_2 du nœud d'interconnexion. Les tensions drain-source des transistors sont différentes dans les deux cas ce qui modifie tous les paramètres électriques des transistors. C'est ce qui est appelé ici l'effet mémoire. Bien que l'état logique de la figure 7.12-c corresponde au même état logique que la figure 7.12-a, les deux états correspondent à deux états électriquement différents.

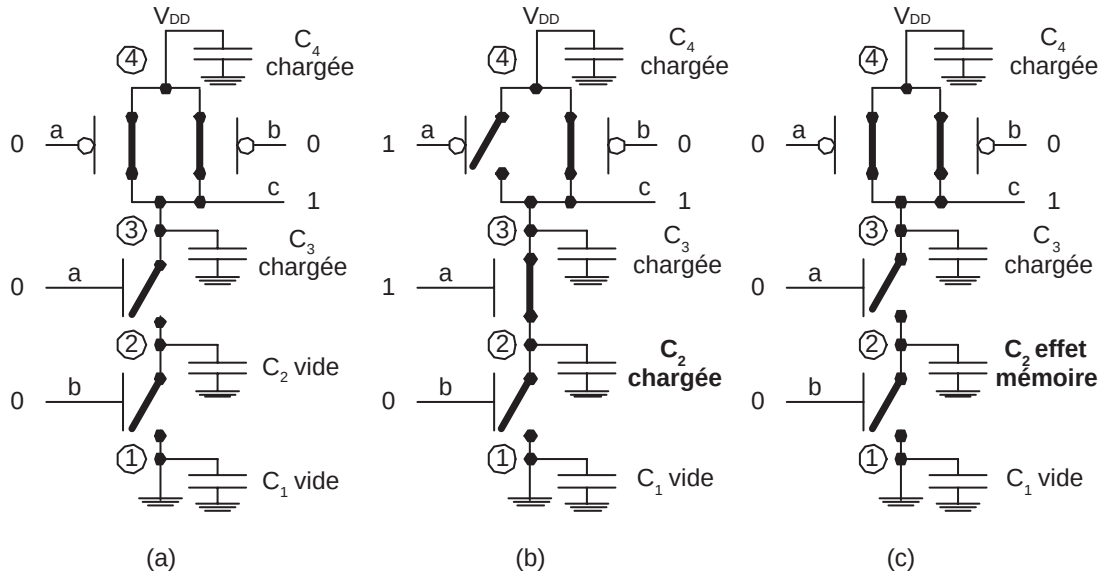


FIG. 7.12 – Effet mémoire

Ceci signifie que le passage de l'état logique de la figure 7.12-a à l'état logique de la figure 7.12-b est électriquement différent du passage de l'état logique de la figure 7.12-c à l'état logique de la figure 7.12-b. Il devient nécessaire de prendre en compte la précédence des signaux qui, à cause de l'effet mémoire, engendre le phénomène DPA (Figure 7.13-a). Les évènements définis au chapitre 1 doivent être finalement complétés. La figure 7.13-b complète la précédence des signaux pour les évènements 2 et 15.

Cette différence s'exprime par une différence de consommation instantanée en courant. Les cas 1

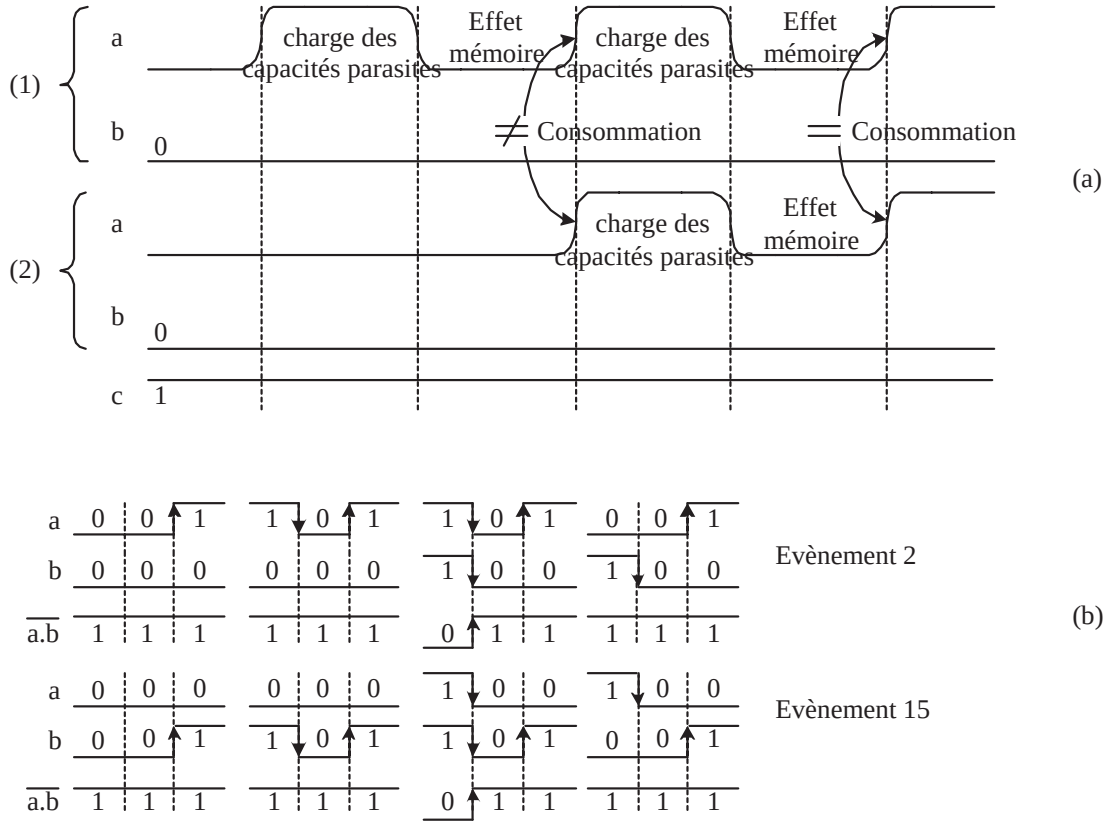


FIG. 7.13 – Précédence des signaux

et 2 de la figure 7.13-a sont simulés afin de comparer leur différence de consommation instantanée. Celle-ci illustre bien l'importance de l'historique du signal a dont la différence de consommation de courant est équivalente à celle provoquée par les réseaux de transistors (Figure 7.14).

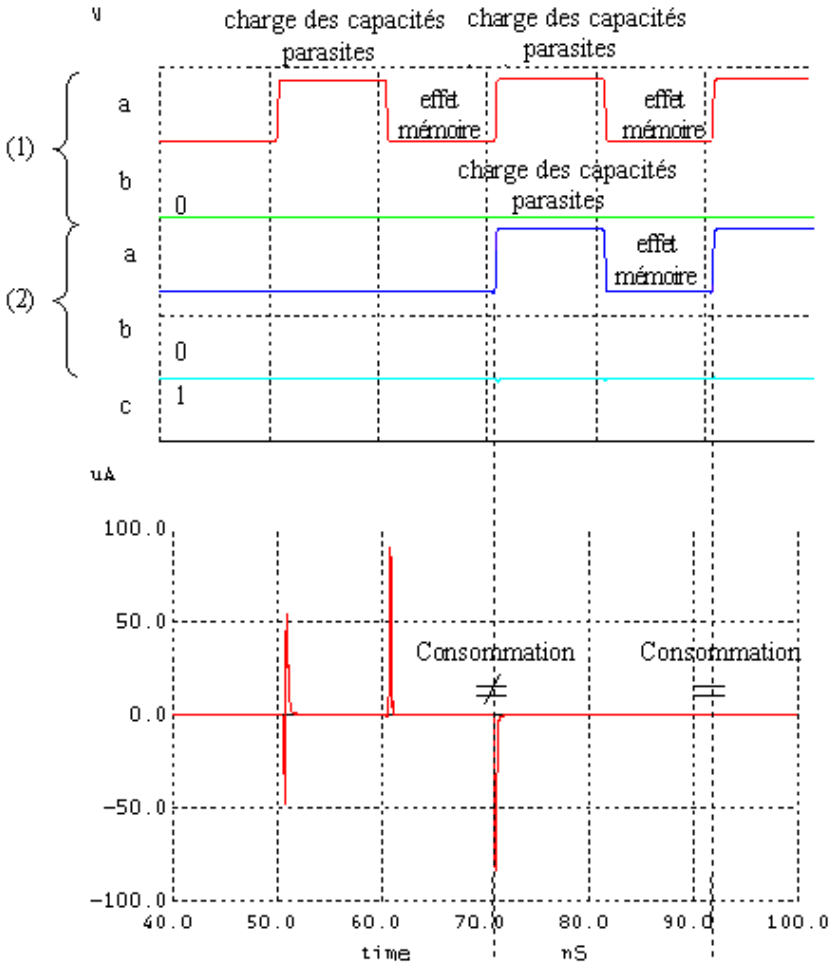


FIG. 7.14 – Simulation de l'effet mémoire

7.4 Les outils de conception

En supposant que chaque porte de la librairie cible ait un comportement électrique identique quelles que soient les entrées et les sorties, les outils de conception peuvent à nouveau les déséquilibrer. Afin d'illustrer ces propos, la synthèse et le placement routage, qui sont deux phases incontournables d'un flot de conception, sont pris en exemple pour montrer comment ces deux étapes peuvent ré-introduire un biais en consommation.

7.4.1 La synthèse

La synthèse consiste dans un premier temps à transformer une description de haut niveau, souvent décrite en VHDL ou Verilog, en équations booléennes pour les parties combinatoires et en éléments mémorisants pour les états du circuit. Cette première étape peut être assimilée à une optimisation booléenne avec la recherche de noyaux communs de fonctions, la minimisation du nombre de couches logiques, etc. Dans un second temps elle remplace les équations et les éléments mémorisants obtenus par les portes disponibles dans la librairie cible. Cette seconde étape peut encore s'accompagner d'optimisations booléennes afin de respecter la fréquence d'horloge, de réduire la surface, etc (Figure 7.15).

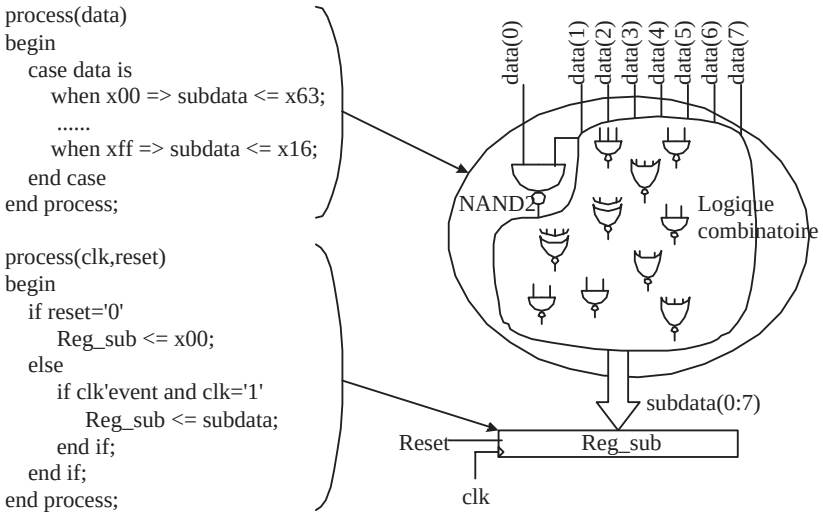


FIG. 7.15 – Transformation d'une description de haut niveau en portes par la synthèse

Les outils réalisant la synthèse ont des priorités. La plus haute priorité est temporelle, c'est à dire le respect absolu de la fréquence d'horloge. Toutes les couches logiques doivent être traversées dans un temps inférieur à la période imposée par le concepteur. Les optimisations booléennes visent au respect absolu de cette priorité soit en diminuant les couches logiques soit en parallèlisant les calculs. Viennent ensuite d'autres priorités inférieures comme la réduction de surface mais sans jamais remettre en cause l'aspect temporel avec, par exemple, la recherche d'une mise en commun de ressources matérielles. En conséquence, les entrées $data(0)$ à $data(7)$ de la fonction $SubBytes$ sur la figure 7.15 peuvent toutes être connectées à un nombre différent de portes. Le fait de connecter une entrée à une porte comme $data(0)$ et une entrée à deux portes comme $data(1)$ déséquilibre électriquement les deux signaux (Figure 7.16). La consommation d'un évènement du bit $data(0)$ est alors différente de la consommation d'un évènement du bit $data(1)$.

Cette différence est visible par différence de la consommation en courant.

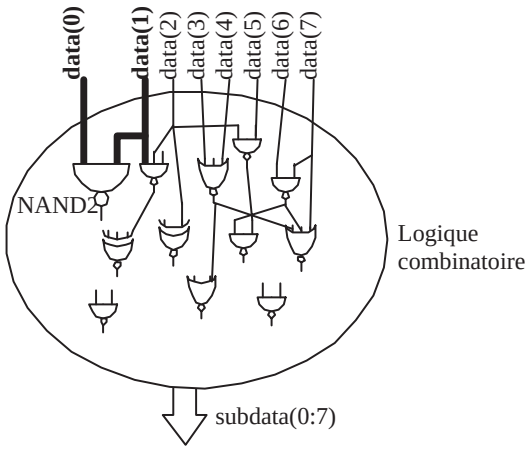


FIG. 7.16 – Introduction théorique d'un biais en consommation entre les bits data(0) et data(1) par la synthèse

Ce cas a été validé concrètement par un projet de recherche [9] [7] [8] consistant à analyser la résistance intrinsèque à la DPA de la technologie asynchrone. La synthèse a provoqué un biais en consommation dans la fonction $XTIME$ entre les entrées 14 et 15 et les sorties 0 et 1 par rapport aux autres entrées et sorties de la fonction (Figure 7.17) en cassant la régularité de la structure.

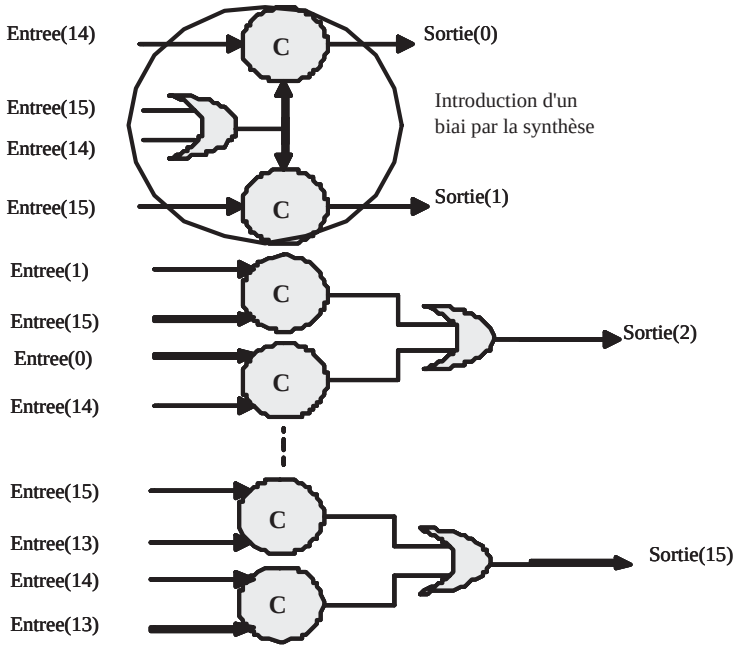


FIG. 7.17 – Introduction observée d'un biais en consommation entre les bits d'entrée et de sortie de la fonction $XTIME$ dans le projet AES asynchrone par la synthèse

7.4.2 Le placement routage

La placement routage consiste à transformer le résultat de la synthèse (souvent elle aussi décrite en VHDL ou Verilog) en une répartition géographique sur une matrice qui est la représentation physique du circuit intégré. Les outils réalisant le placement routage ont aussi des priorités. La plus haute priorité est la même que pour les outils de synthèse, elle est temporelle avec toujours le respect absolu de la fréquence d'horloge. Toutes les couches logiques issues de la synthèses doivent continuer à être traversées dans un temps inférieur à la période imposée par le concepteur après le positionnement géographique (placement) et les connexions des portes (routage).

Les optimisations ne sont plus booléennes mais concernent les paramètres électriques des portes comme le nombre maximal de connexions en entrée comme en sortie. Il arrive que, du fait de la longueur de certaines pistes de signaux, des portes logiques ayant la même fonction soient modifiées électriquement. Les outils adaptent en effet la sortance des portes c'est à dire le nombre de signaux que la porte peut imposer à sa sortie. Dans les technologies actuelles, une porte de sortance 1 peut imposer une valeur à environ 5 autres portes. Les autres tailles définies dans les librairies standards sont 0.5 (suffixe X05), 1 (pas de suffixe), 2 (suffixe P), 4 (suffixe X4), 6 (suffixe X6) ou 8 (suffixe X8). Une porte de taille 6 peut donc imposer environ 30 signaux. Ces valeurs restent indicatives et sont aussi dépendantes des capacités parasites des pistes sur lesquelles transitent les signaux en sortie de la porte. Toujours dans le cas du projet de composant de chiffrement AES asynchrone [36] [35] [10] [39], l'analyse des rails 12 et 13 dans la fonction $XTIME$ après placement routage illustre bien les modifications réalisées par l'outil de placement routage. Bien que la fonction soit identique du point de vue logique, le rail At est l'entrée de deux buffers dont l'un a le suffixe P et le second le suffixe X6. En revanche, le rail Af est l'entrée de deux buffers dont l'un a le suffixe P et le second le suffixe X4. Ces deux rails sont déséquilibrés entre eux du fait du placement routage. Une différence en consommation instantanée apparaît entre le 0 logique véhiculé par le rail Af et le 1 logique véhiculé par le rail At permettant l'attaque DPA (Figure 7.18).

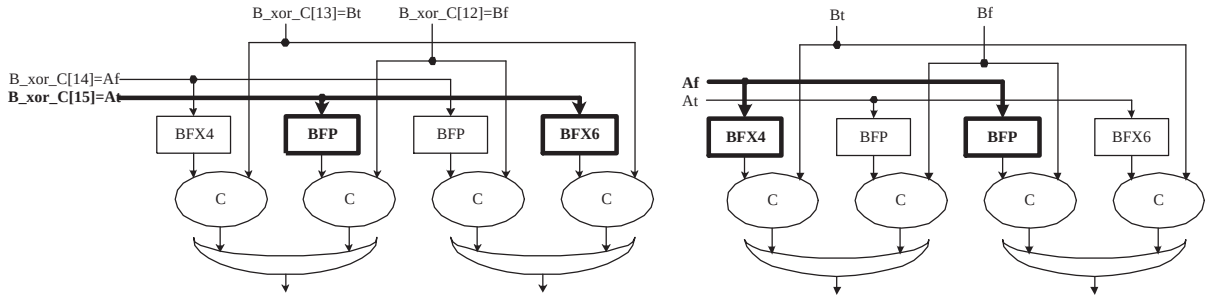


FIG. 7.18 – Introduction observée d'un biais en consommation entre deux bits d'entrée de la fonction $XTIME$ dans le projet AES asynchrone par le placement routage

7.5 Conclusion

La source architecturale élémentaire de la DPA est définie par une porte CMOS quelle que soit l'architecture envisagée. Une porte CMOS est construite à partir de réseaux complémen-

taires de transistors connectés en série et en parallèle. Les réseaux ainsi constitués forment une deuxième source de la DPA. Des nœuds d'interconnexion de transistors mémorisent sous certaines conditions des charges dans des capacités parasites créant un effet mémoire. Deux transitions logiques identiques peuvent devenir deux transitions électriques différentes. L'effet mémoire est une troisième source de la DPA. En supposant que les portes soient intrinsèquement résistantes à la DPA, les outils de conception peuvent les déséquilibrer. Le point commun à toutes ces sources ainsi définies est la dispersion instantanée des caractéristiques électriques du circuit. La DPA ne fait que les mettre en évidence par différence de grandeurs physiques mesurables. Dans la partie présentant une solution anti-DPA, des portes logiques seront construites avec la propriété intrinsèque de résistance à la DPA. Elles seront les briques de base pour construire un circuit intégré résistant à la DPA mais il faudra modifier les outils de conception pour éviter d'introduire un biais physique entre les portes.

Chapitre 8

Application à la DPA

Sommaire

8.1 Application à la DPA du DES

8.1.1 Séparation des chiffrés et des évènements microélectroniques associés

La séparation des chiffrés est faite en supposant une certaine valeur sur un bit de donnée et sur six bits de clé en entrée d'un fonction de substitution (4). Lorsque les 6 bits $K16(18:23)$ sont le bon sous-groupe de bits de clé utilisé lors du chiffrement, l'ensemble E1 représente ce qui s'est réellement passé dans le circuit et l'opération $\oplus$ a toujours été $1 \oplus 0$ ou $1 \oplus 1$. Dire que l'opération $\oplus$ a été $1 \oplus 0$ ou $1 \oplus 1$ signifie que le résultat d'une transition est une configuration d'entrée (1,0) ou (1,1) réalisée par les huit évènements 1 à 8 définis au chapitre 1 (Figure 8.1).

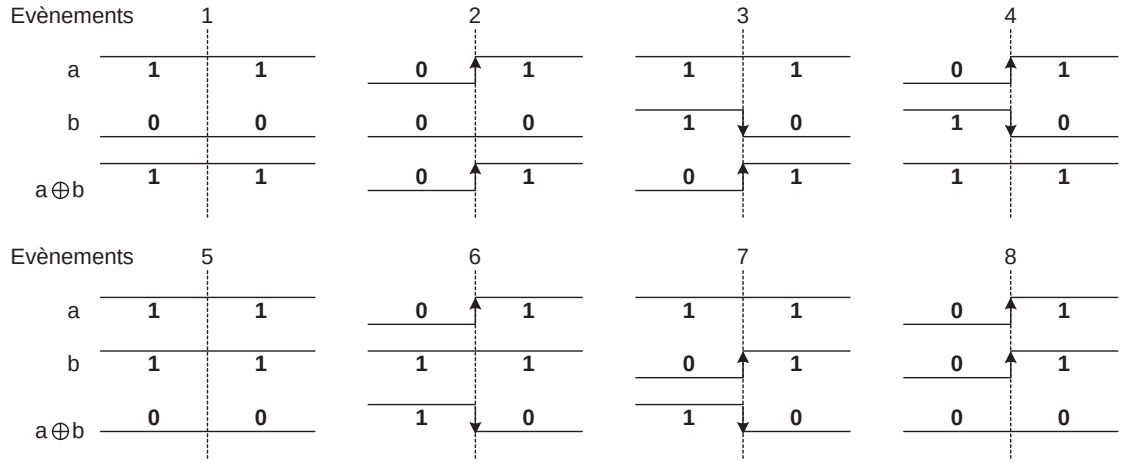


FIG. 8.1 – Ensemble E1 et évènements microélectroniques associés

L'ensemble E0 représente aussi ce qui s'est réellement passé dans le circuit et l'opération $\oplus$ a toujours été $0 \oplus 0$ ou $0 \oplus 1$. Dire que l'opération $\oplus$ fut $0 \oplus 0$ ou $0 \oplus 1$ signifie que le résultat d'une transition est une configuration d'entrée (0,0) ou (0,1), ce que réalisent les huit évènements 9 à 16 définis au chapitre 1 (Figure 8.2).

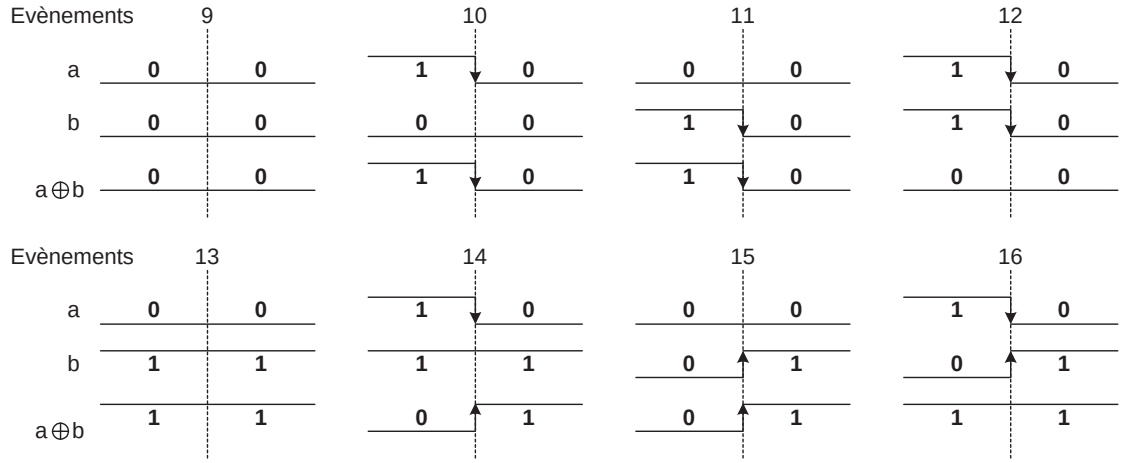


FIG. 8.2 – Ensemble E0 et évènements microélectroniques associés

8.1.2 Schémas possibles de la fonction ⊕ du DES

Le calcul ⊕ dans le DES peut se faire de multiples façons. Il peut être fait par un microprocesseur qui appelle une unité arithmétique et logique pour réaliser cette fonction ou bien il peut être exécuté dans un coprocesseur cryptographique. Dans tous les cas, l'opération ⊕ mettra en œuvre un groupe de transistors pendant une ou plusieurs périodes d'horloge lorsque l'opération est pipelinée (Chapitre 7).

Pour l'exemple, trois types de circuiterie sont retenus. Le premier utilise comme élément de construction de base la porte Nand2. Celle-ci est utilisable par un microprocesseur qui appellerait répétitivement la porte Nand2 pour le calcul. Le calcul est considéré exécutable en une période d'horloge. Le deuxième est une construction CMOS de la porte ⊕ avec comme élément de construction de base le transistor. Le troisième utilise des transistors dits " de passage " (Figure 8.3).

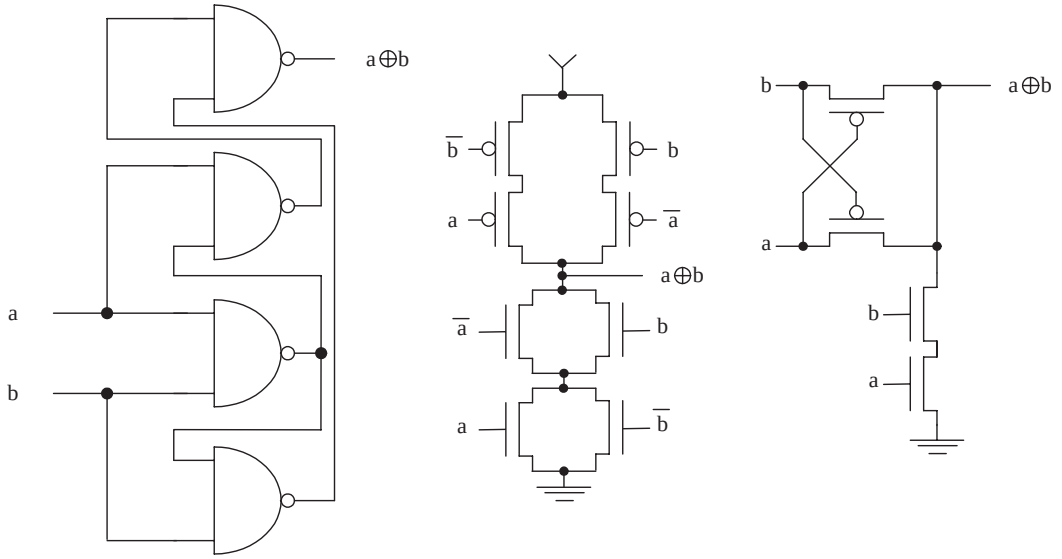


FIG. 8.3 – Schémas possibles de la porte Xor2

L'architecture à base de portes Nand2 vérifie bien la fonction " xor " (Équation 8.1).

$$\begin{aligned} \overline{\overline{(a.\overline{(a.b)})}.\overline{(\overline{(a.b)}.b)}} &= \overline{(\overline{a}+a.b).(a.b+\overline{b})} = (a.\overline{a.b}) + (\overline{a.b}.b) = a.(\overline{a}+\overline{b}) + (\overline{a}+\overline{b}).b = a.\overline{b} + \overline{a}.b \\ &= a \oplus b \end{aligned} \tag{8.1}$$

L'architecture CMOS de la porte Xor2 a été construite à partir de l'équation 8.2, sachant que la construction CMOS de celle-ci vérifie $\overline{\overline{a \oplus b}} = a \oplus b$.

$$\overline{a \oplus b} = \overline{a.\overline{b} + \overline{a}.b} = (a+\overline{b}).(\overline{a}+b) \tag{8.2}$$

8.1.3 Simulation de la DPA sur la porte Xor2 du DES

La DPA fonctionne avec un nombre quelconque de chiffrés. Ceux-ci peuvent être considérés comme aléatoires puisque aucune hypothèse les concernant n'a été faite. Statistiquement, l'ensemble $E1$ est composé de chiffrés qui génèrent de façon équiprobable les événement 1 à 8. Pour calculer la moyenne de consommation de cet ensemble, il suffit de faire la moyenne de consommation des 8 évènements tout en sachant que les évènements 1 et 5 ne consomment pas. De même, l'ensemble E0 est statistiquement composé de chiffrés qui génèrent de façon équiprobable les événements 9 à 16. Pour calculer la moyenne de consommation de cet ensemble, il suffit de faire la moyenne de consommation des 8 évènements tout en sachant que les évènements 9 et 13 ne consomment pas. La différence de ces deux moyennes de consommation est présentée dans la figure 8.4 pour les trois types de circuiterie.

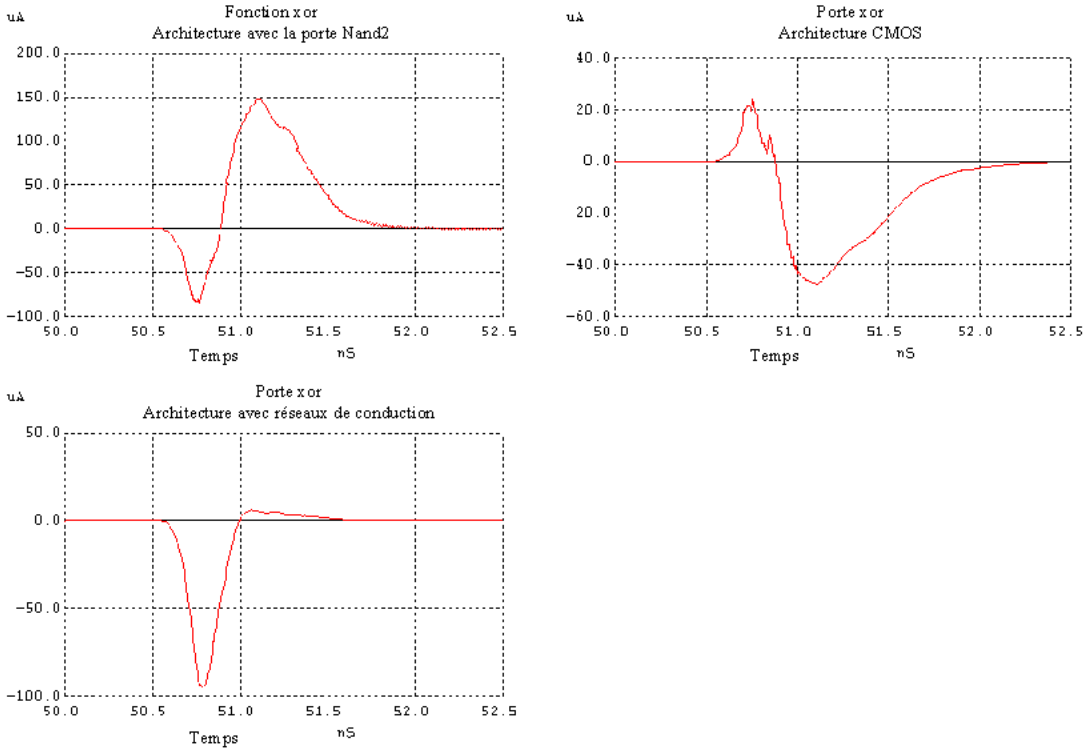


FIG. 8.4 – Simulation de la DPA du DES

Lorsque les 6 bits $K16(18:23)$ sont un sous-groupe de bits de clé différent de celui utilisé lors du chiffrement, les ensembles $E1$ et $E0$ ne représentent plus ce qui s'est réellement passé dans le circuit parce que les chiffrés et leurs consommations respectives ont été distribués de façon équiprobable. Pour chaque ensemble, l'opération XOR regroupe toutes les possibilités $0 \oplus 0$, $0 \oplus 1$, $1 \oplus 0$ et $1 \oplus 1$. Leurs moyennes de consommation tendent vers la même valeur et leur différence tend vers 0.

La DPA illustrée par la publication [25] indiquait une différence de moyenne de l'ordre de 20 microampères sur moins d'une microseconde environ. Il s'agit du même ordre de grandeur que la simulation de la porte CMOS *xor* ce qui permet de justifier cette approche de la simulation

de la DPA.

8.2 Application à la DPA de l'AES

8.2.1 Séparation des chiffrés et des évènements microélectroniques associés

SubBytes est composée de plusieurs portes logiques et le bit M(0) peut être l'entrée d'une porte Nand, d'une porte Nor ou d'une autre porte logique à 2 ou plus d'entrées (Chapitre 7). La séparation des chiffrés a été faite sur un plan mathématique en supposant une valeur sur un octet de sous-clé qui donne la valeur à un bit du message intermédiaire.

Lorsque l'octet de sous-clé correspond à celui utilisé lors du chiffrement, l'ensemble $E1$ représente ce qui s'est réellement passé dans le circuit et l'octet en entrée de la fonction de substitution SubBytes a toujours été sous la forme $1xxxxxx$. Dire que l'entrée de SubBytes a toujours été sous la forme $1xxxxxx$ signifie que la valeur 1 a toujours été l'entrée de la porte logique de SubBytes concernant ce bit. De même, l'ensemble $E0$ représente aussi ce qui s'est réellement passé dans le circuit et l'entrée de la fonction de substitution SubBytes a toujours été sous la forme $0xxxxxx$. Dire que l'entrée de SubBytes a toujours été sous la forme $0xxxxxx$ signifie que la valeur 0 a toujours été l'entrée de la porte logique de SubBytes concernant le même bit. Par exemple, si la porte logique rencontrée par le bit $M(0)$ est une porte $Nand2$, $E1$ représente alors les opérations $\overline{1.1}$ et $\overline{1.0}$ et $E0$ représente les opérations $\overline{0.1}$ et $\overline{0.0}$ (Figure 8.5).

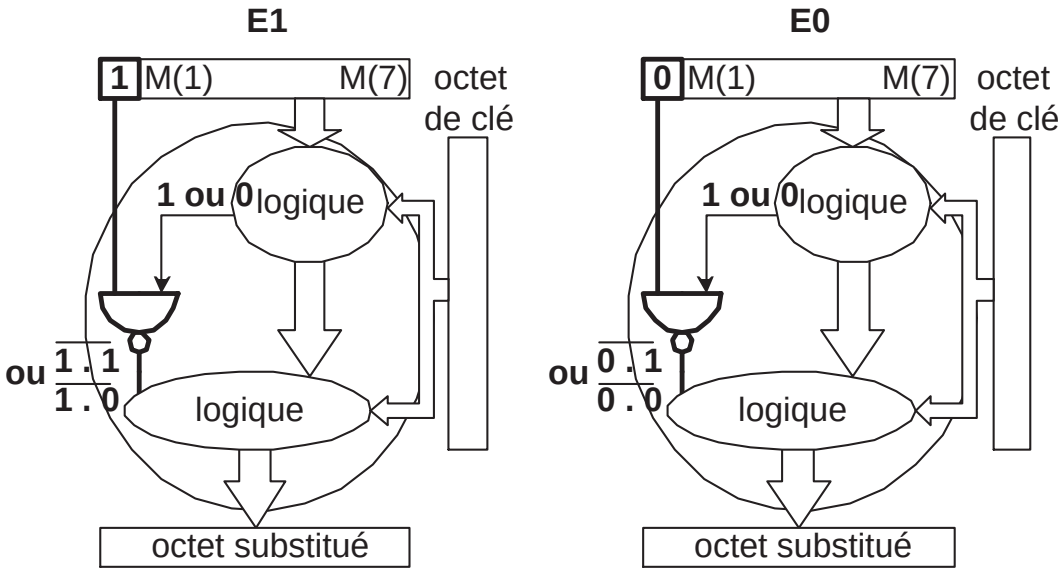


FIG. 8.5 – Opérations dans les ensembles E1 et E0

Dire que l'opération $Nand2$ a été $\overline{1.1}$ ou $\overline{1.0}$ signifie que le résultat d'une transition est une configuration d'entrée (1,1) ou (1,0) qui est réalisée par les huit évènements 1 à 8 définis au chapitre 1 (Figure 8.6).

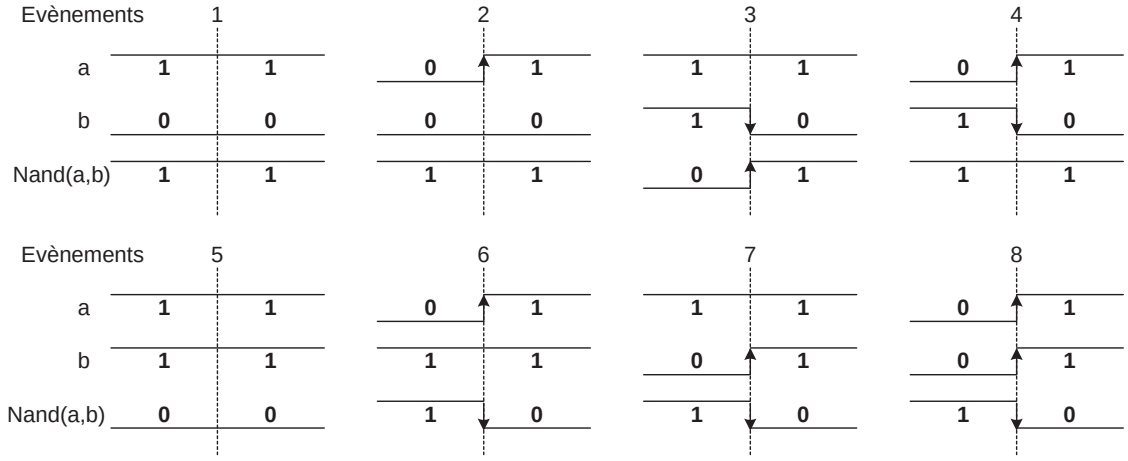


FIG. 8.6 – Ensemble E1 et évènements microélectroniques associés

Dire que l'opération $Nand2$ a été $\overline{0.1}$ ou $\overline{0.0}$ signifie que le résultat d'une transition est une configuration d'entrée (0,1) ou (0,0) réalisée par les huit évènements 9 à 16 définis au chapitre 1 (Figure 8.7).

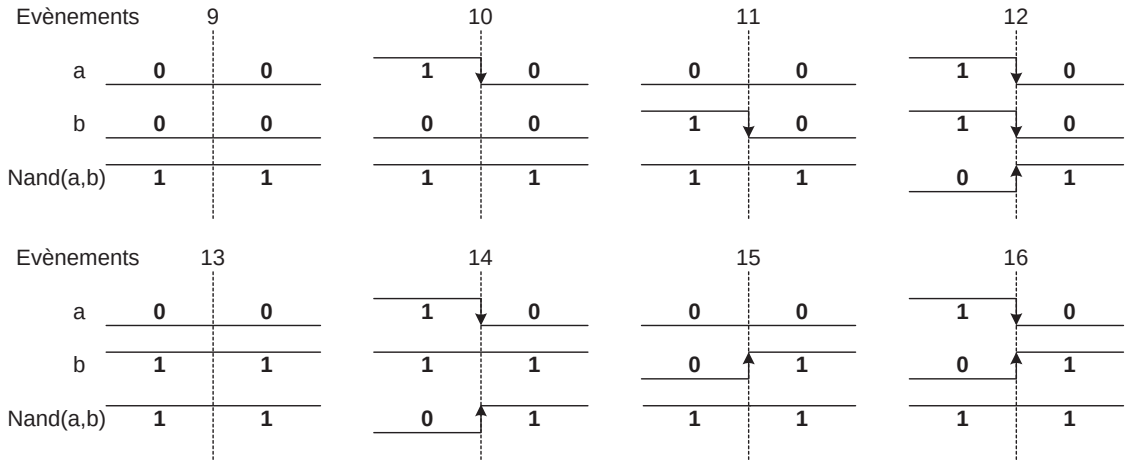


FIG. 8.7 – Ensemble E0 et évènements microélectroniques associés

8.2.2 Schéma de la fonction $Nand2$

L'architecture retenue est une construction CMOS de la porte $Nand2$ avec, comme élément de construction de base, le transistor (Annexe A).

8.2.3 Simulation de la DPA sur la porte $Nand2$

Dans cet exemple aussi, DPA fonctionne un avec nombre quelconque de chiffrés. Ceux-ci sont aussi considérés comme aléatoires puisque aucune hypothèse les concernant n'a été faite.

Statistiquement, l'ensemble $E1$ est composé de chiffrés qui génèrent de façon équiprobable les événements 1 à 8. Pour calculer la moyenne de consommation de cet ensemble, il suffit de faire la moyenne de consommation des 8 évènements tout en sachant que les évènements 1 et 5 ne consomment pas. De même, l'ensemble E0 est statistiquement composé de chiffrés qui génèrent de façon équiprobable les événements 9 à 16. Pour calculer la moyenne de consommation de cet ensemble, il suffit de faire la moyenne de consommation des 8 évènements tout en sachant que les évènements 9 et 13 ne consomment pas. La différence de ces deux moyennes de consommation est présentée dans la figure 8.8.

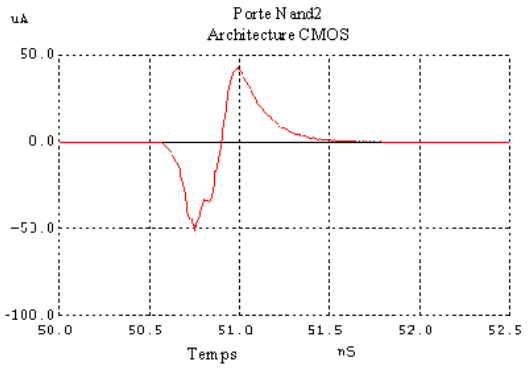


FIG. 8.8 – Simulation de la DPA de l'AES

Dans le cas où l'octet supposé est un octet différent de celui utilisé pendant le chiffrement, les ensembles $E1$ et $E0$ ne représentent plus ce qui s'est réellement passé dans le circuit parce que les chiffrés et leurs consommations respectives ont été distribués de façon équiprobable. Pour chaque ensemble, l'opération $Nand2$ regroupe toutes les possibilités 0.0, 0.1, 1.0 et 1.1. Leurs moyennes de consommation tendent vers la même valeur et leur différence tend vers 0.

8.3 Application à la variante de la DPA du DES et de l'AES

8.3.1 Séparation des chiffrés et des évènements microélectroniques associés

La variante de la DPA a été introduite au chapitre 6. Le bit de clé n'est jamais connu et les deux cas où il vaut 1 et 0 doivent être considérés pour les calculs afin de le déterminer en final. La consommation de la technologie du circuit intégré n'est pas non plus connue à priori. En supposant que le bit de clé i ait pour valeur 1, cela signifie que l'opération $Xor2$ a été $0 \oplus 1$ lorsque le chiffré a pour résultat 1 et $1 \oplus 1$ lorsque le chiffré a pour résultat 0. Le résultat d'un chiffré est le résultat d'une transition qui est une configuration d'entrée (0,1) ou (1,1) réalisée par les huit évènements 1 à 8 définis au chapitre 1 (Figure 8.9).

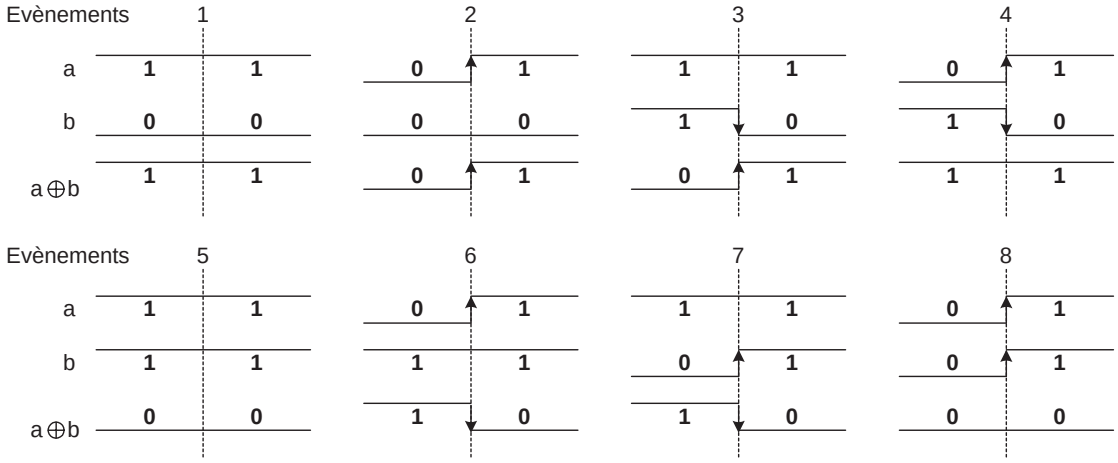


FIG. 8.9 – $0 \oplus 1$ et $1 \oplus 1$ et évènements microélectroniques associés

Les évènements 1 à 4 permettent de calculer la moyenne de consommation $M_1 1$ et les évènements 5 à 8 permettent de calculer la moyenne de consommation $M_1 0$. La différence de ces deux moyennes donne $\Delta 1$.

Il se peut que le bit de clé i considéré ait pour valeur 0. Cela signifie que l'opération Xor2 a été $0 \oplus 0$ lorsque le chiffré a pour résultat 0 et $1 \oplus 0$ lorsque le chiffré a pour résultat 1. Le résultat d'un chiffré est le résultat d'une transition qui est une configuration d'entrée (0,0) ou (1,0) réalisée par les huit évènements 9 à 16 définis au chapitre 1 (Figure 8.10).

Les évènements 9 à 12 permettent de calculer la moyenne de consommation $M_0 0$ et les évènements 13 à 16 permettent de calculer la moyenne de consommation $M_0 1$. La différence de ces deux moyennes donne $\Delta 0$.

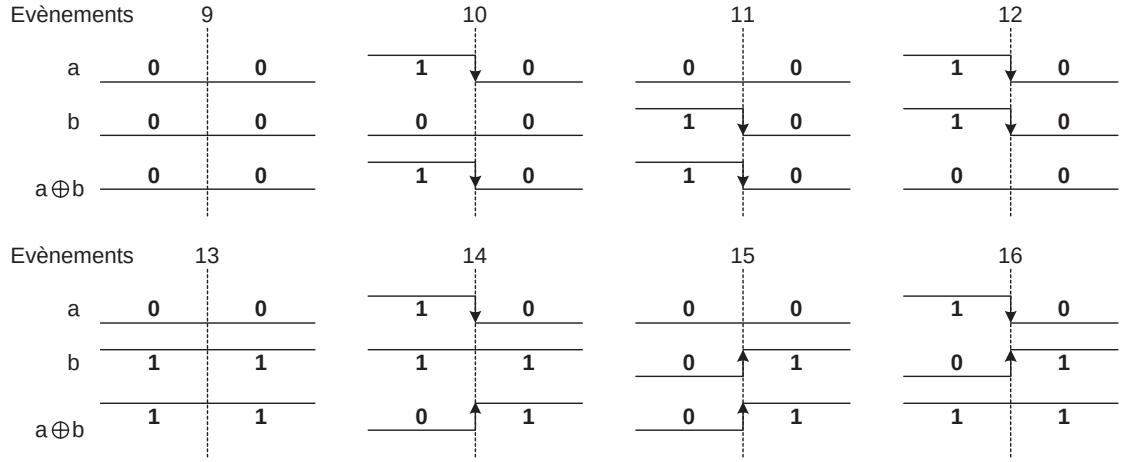


FIG. 8.10 – $0 \oplus 0$ et $1 \oplus 0$ et évènements microélectroniques associés

8.3.2 Schéma de la porte Xor2

Le schéma retenu est la construction CMOS de la porte Xor de la figure 8.3.

8.3.3 Simulation de la variante de la DPA sur la porte Xor2

Les hypothèses restent les mêmes que précédemment pour la simulation. Statistiquement, les chiffrés permettant le calcul de $M_1 1$ sont considérés comme équiprobables. Pour calculer cette moyenne de consommation, il suffit de faire la moyenne de consommation des évènements 1 à 4 tout en sachant que l'événement 1 ne consomme pas. Les moyennes de consommation $M_1 0$, $M_0 0$ et $M_0 1$ sont obtenues respectivement à partir des évènements 5 à 8, 9 à 12 et 13 à 16 en simulation. La figure 8.11 illustre ces quatre moyennes et leurs différences $Delta1$ et $Delta0$ permettent de déterminer la valeur du bit de clé.

Tous les bits donnant comme résultat de consommation $Delta1$ lors de la séparation sont égaux entre eux et tous les bits donnant comme résultat $Delta0$ sont aussi égaux entre eux.

$Delta1$ ne peut représenter que deux possibilités pour les bits menant à ce résultat. Soit ils correspondent à 1, soit ils correspondent à 0. Afin de déterminer avec certitude la correspondance entre $Delta1$ et une valeur logique 1 ou 0 des bits, deux déchiffrements sont à réaliser. Un premier déchiffrement pose que tous les bits ayant pour résultat $Delta1$ correspondent au 0 logique et un second qui pose que tous ces mêmes bits correspondent au 1 logique. Un seul des deux déchiffrement donne un résultat intelligible et détermine la clé.

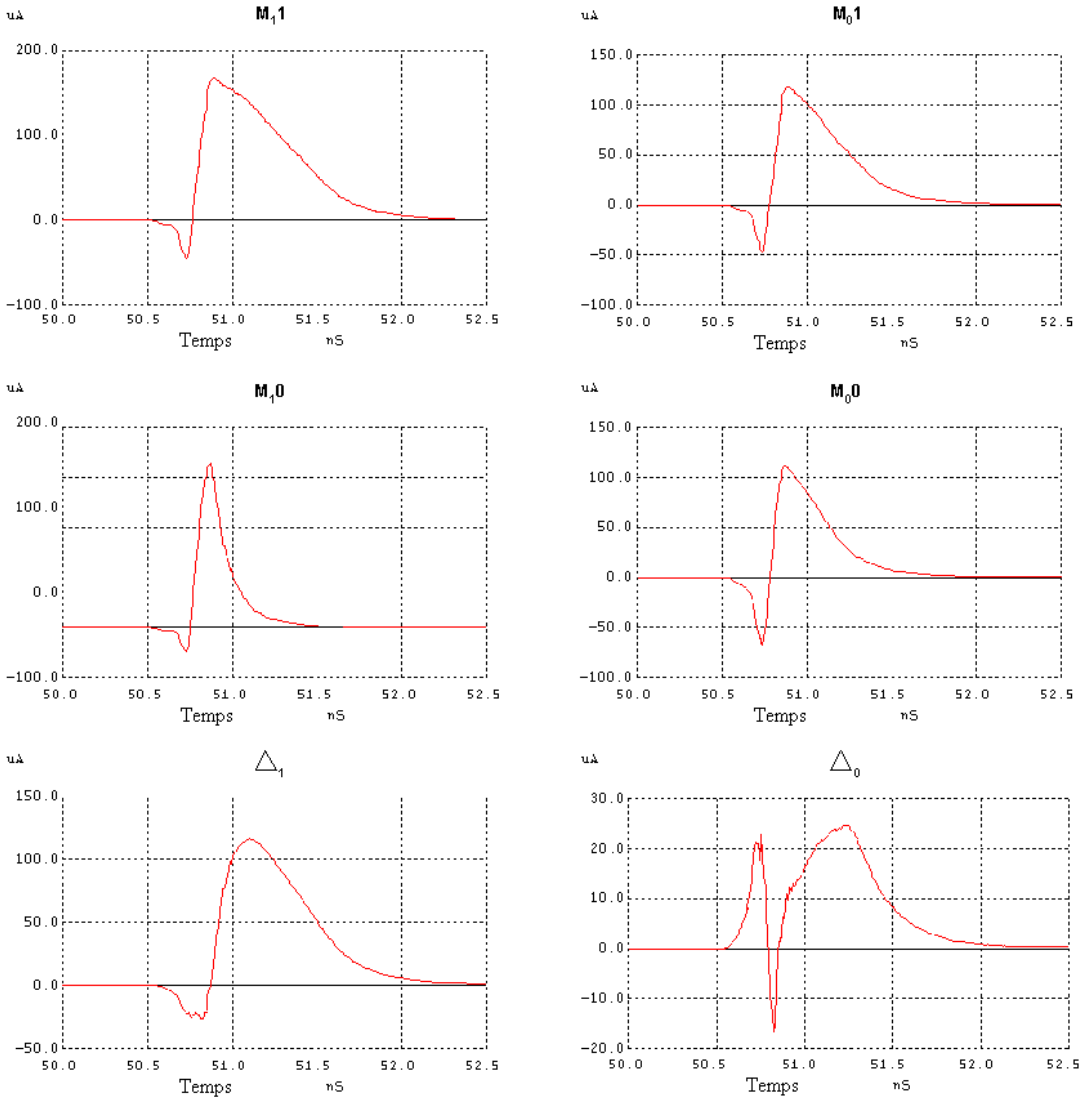


FIG. 8.11 – Simulation de la variante de la DPA

8.4 Application à la DPA de l'AES en technologie asynchrone

8.4.1 Introduction

Jusqu'à présent, seuls les circuits synchrones furent pris en considération pour la mise œuvre de la DPA. Ceux-ci peuvent être représentés de façon générale par un pipeline de registres entre lesquels existe ou non une partie logique (Figure 8.12). On retrouve cette structure aux chapitres 4, 5 et 7 lors de la présentation des architectures pour le DES et l'AES.

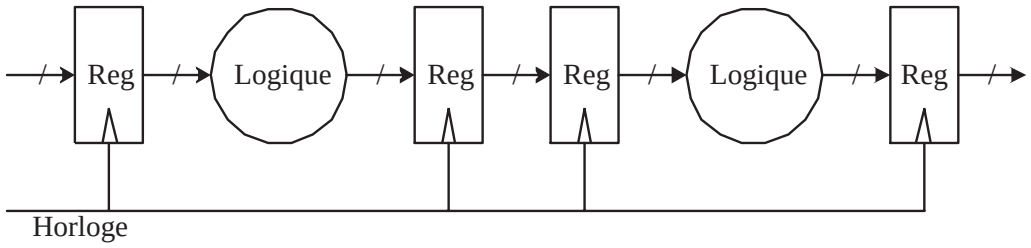


FIG. 8.12 – Représentation générale d'un circuit synchrone

Cette représentation est en fait celle d'un concepteur qui se focalise sur la fonction logique qu'il doit réaliser en elle-même et qui considère une horloge globale. Mais dans la réalité, les circuits sont bien plus complexes parce que les technologies actuelles permettent une très grande intégration de fonctions. On ne parlera plus d'une horloge globale mais d'un arbre d'horloge qui répond aux besoins de chaque fonction. Il permet soit de garantir les délais des signaux dans le cas le plus classique entre deux registres, soit de s'arrêter pour diminuer la consommation dans le cas d'une gestion d'énergie, voire d'être utilisé comme signal de donnée dans le cas d'un contrôle de l'horloge comme le font certaines fonctions de sécurité. On obtient donc un schéma plus complexe présenté par la figure 8.13.

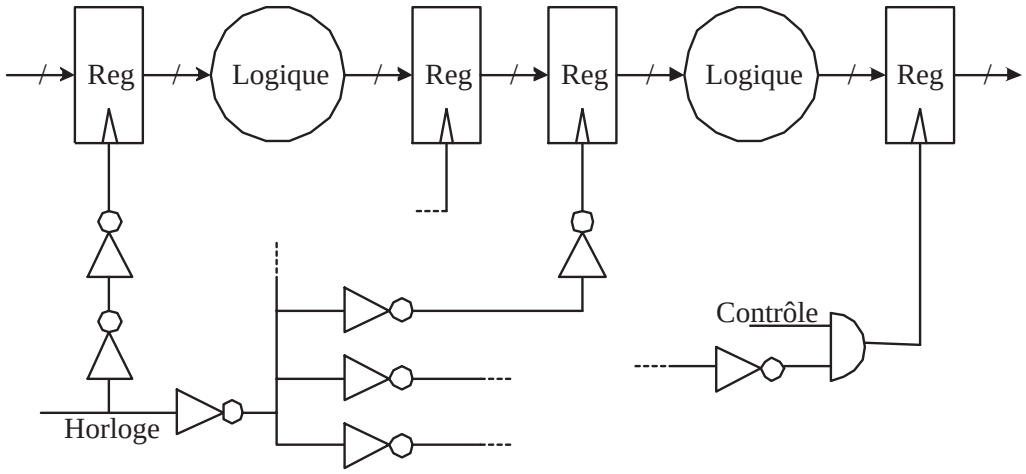


FIG. 8.13 – Représentation générale d'un circuit synchrone avec son arbre d'horloge

Les circuits asynchrones sont une alternative à cette structure en remplaçant le signal d'horloge par un mécanisme de poignée de mains entre registres voisins. Ce mécanisme peut prendre la

forme d'une requête et d'un acquittement comme présenté dans la figure 8.14.

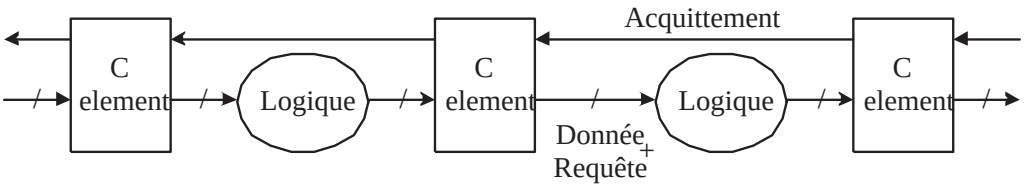
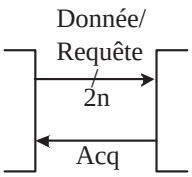


FIG. 8.14 – Représentation générale d'un circuit asynchrone

Il existe différents protocoles pour réaliser un circuit asynchrone. Dans l'exemple de mise en œuvre de la DPA sur un tel circuit, seul le protocole double rail 4 phases sera pris en considération. Ce dernier encode le signal de requête dans les signaux de données en utilisant deux rails par bit d'information. Il est possible de considérer ces deux rails comme deux signaux de requête, le premier transmettant un 1 (vrai) et le second transmettant un 0 (faux). On notera alors un bit sous la forme (b.v, b.f) et il pourra prendre les valeurs (0,0) qui signifient l'absence de donnée ou donnée non valide, (0,1) pour un 0 valide et (1,0) pour un 1 valide. La valeur (1,1) est interdite par le protocole mais peut servir dans le cas de détection d'erreur [42]. Le protocole fonctionne de la manière suivante. L'état de départ est une donnée non valide (0,0) et l'acquittement est à 0. La donnée passe dans l'état donnée valide correspondant à un 0 (0,1) ou à un 1 (1,0). Lorsque celle-ci a été prise en compte par la logique, l'acquittement passe à 1 pour l'indiquer. Avec la réception de cet acquittement la donnée est remise à l'état donnée non valide (0,0). La logique remet alors l'acquittement à 0. La figure 8.15 illustre ce fonctionnement.

	b.v	b.f
Absente	0	0
0 valide	0	1
1 valide	1	0
Interdit	1	1



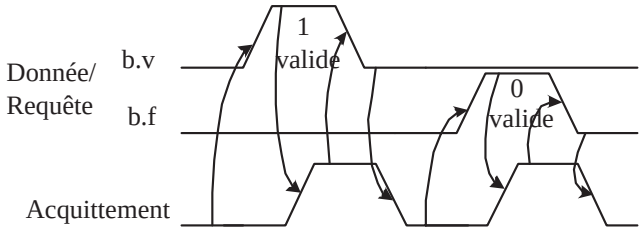


FIG. 8.15 – Protocole double rail 4 phases

Le protocole a été illustré sur un bit. Dans le cas d'une logique ayant plus d'un bit en entrée, celle-ci attend que tous soient dans l'état valide avant d'effectuer son exécution. Cette synchronisation est réalisée par l'élément C de Muller. Cette fonction permet de mémoriser un état comme le fait un latch avec set et reset. Lorsque les deux entrées sont égales à 0, la sortie prend la valeur 0 et lorsqu'elles sont égales à 1 la sortie prend la valeur 1. Dans le cas où les deux entrées sont différentes, la sortie est mémorisée. Avec cette procédure de mémorisation, l'élément C a la particularité d'indiquer en plus que toutes les entrées sont à 1 ou à 0. La figure 8.16 illustre cette porte avec le comportement logique, la représentation schématique et deux représentations avec transistors.

En utilisant les propriétés de l'élément C il est possible de construire un pipeline respectant le protocole double rail 4 phases. Un bit est encodé (b.v, b.f). Lorsqu'il prend une valeur valide, seul

a	b	c
0	0	0
0	1	Mémorisation
1	0	Mémorisation
1	1	1

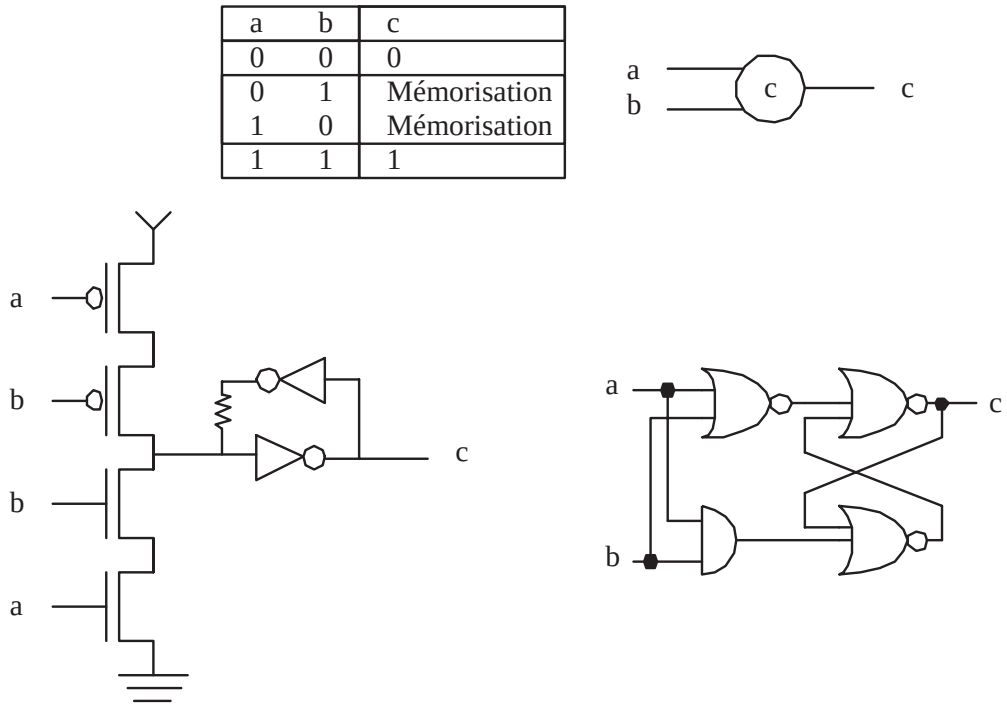


FIG. 8.16 – Élément C de Muller

l'un des deux rails b.v ou b.f prend la valeur 1. Du fait de l'inverseur sur le signal d'acquittement, les deux entrées de l'élément C sont égales à 1 et la sortie de celui-ci prend la même valeur. La donnée étant prise en compte, l'acquittement est transmis. Il correspond en fait à l'un des deux rails b.v ou b.f. Comme ils ont le même rôle dans la génération du signal d'acquittement, le *ou* des deux constitue ce signal. La figure 8.17 présente ce pipeline.

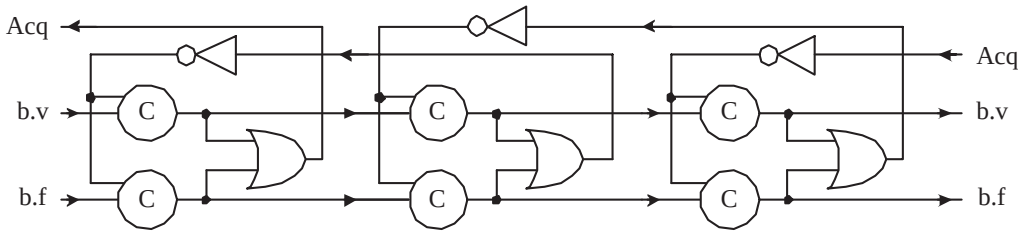


FIG. 8.17 – Pipeline asynchrone avec protocole double rail 4 phases

Lors de la présentation du protocole double rail 4 phases la logique attend que tous les bits soient dans l'état valide avant d'effectuer son opération. Cette synchronisation est réalisée par l'élément C. La sortie de la fonction est encodée en double rail (c.v, c.f) et chacun prend une valeur logique qui dépend de la combinaison des rails des entrées (a.v, a.f) et (b.v, b.f). L'architecture asynchrone de la fonction And2 est présentée dans la figure 8.18.

a	b	c.f	c.v
Absente	Absente	0	0
0 valide	0 valide	1	0
0 valide	1 valide	1	0
1 valide	0 valide	1	0
1 valide	1 valide	0	1

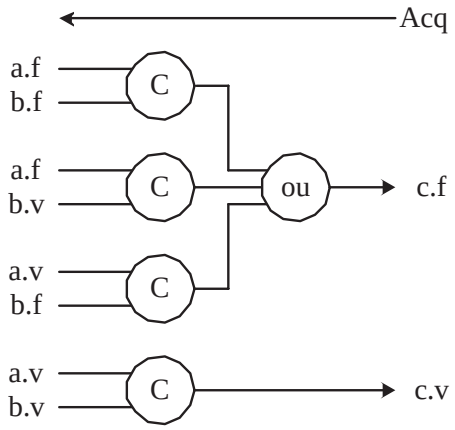


FIG. 8.18 – Architecture asynchrone de la porte And2

8.4.2 Séparation des chiffrés et des évènements microélectroniques associés

Au chapitre 5 de mise en application de la DPA sur l'AES la porte Nand2 fut prise en exemple. Ici, la DPA sera appliquée sur une porte And2 asynchrone en supposant que le bit M(0) est une entrée de celle-ci.

Dire que l'opération And2 a été 1.1 ou 1.0 signifie que l'un des deux évènements microélectroniques 1 ou 2 de la figure 8.19 s'est produit. Dire que l'opération And2 a été 0.1 ou 0.0 signifie que l'un des deux évènements microélectroniques 3 ou 4 de la figure 8.19 s'est produit.

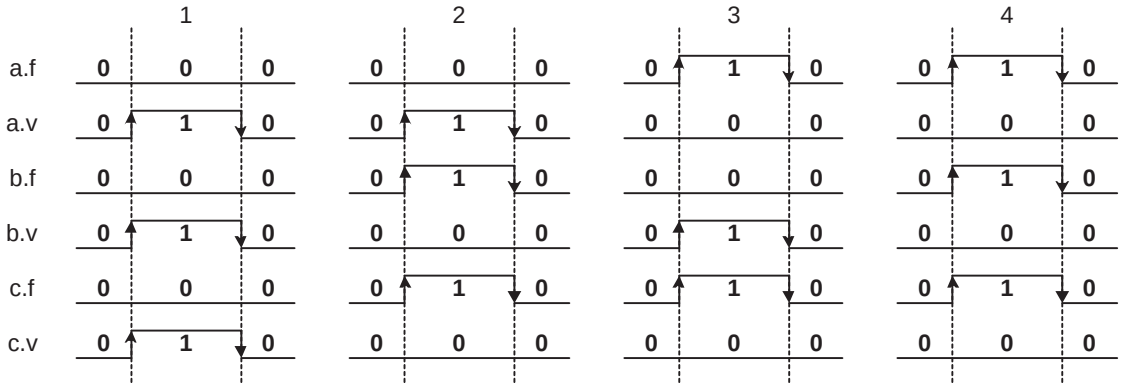


FIG. 8.19 – 1.1, 1.0, 0.1 et 0.0 et évènements microélectroniques associés

8.4.3 Simulation de la DPA sur la porte And asynchrone

Statistiquement, l'ensemble E1 est composé de chiffrés qui correspondent de façon équiprobable aux évènements 1 et 2. La moyenne de consommation de cet ensemble est calculée à partir des consommations des évènements 1 et 2. De même, l'ensemble E0 est statistiquement composé de chiffrés qui correspondent aussi de façon équiprobable aux événements 3 et 4. La moyenne de consommation de cet ensemble est calculée à partir des consommations des évènements 3 et 4.

La différence de ces deux moyennes représente la DPA (Figure 8.20).

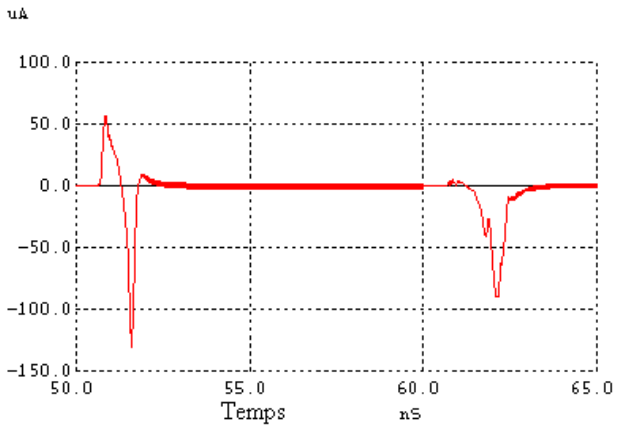


FIG. 8.20 – Simulation de la DPA de l'AES asynchrone

Les éléments C sont une première source de la DPA. En effet, chaque rail de chaque bit voit une impédance différente des autres. Une deuxième source est la connexion de trois éléments C à la porte *ou* trois entrées qui modifie l'impédance du nœud de leurs sorties alors que la sortie de quatrième est une sortie directe. Une troisième source de DPA est la mémorisation. Selon les données, sur les trois éléments C générant c.f, un ou deux mémorisent pendant que deux ou un évaluent la valeur logique du rail.

Bien que la technologie asynchrone présente des propriétés intéressantes pour lutter contre la DPA (double rail et symétrie), elle doit quand même être équilibrée.

8.5 Application à la DPA en technologie SABL

La technologie SABL fonctionne avec une phase de précharge des capacités parasites suivie d'une phase d'évaluation de la fonction logique. Elle se focalise sur la quantité de charge consommée qu'elle rend constante entre différents évènements. Mais ce concept de chargement n'est pas instantanément identique pour deux évènements différents.

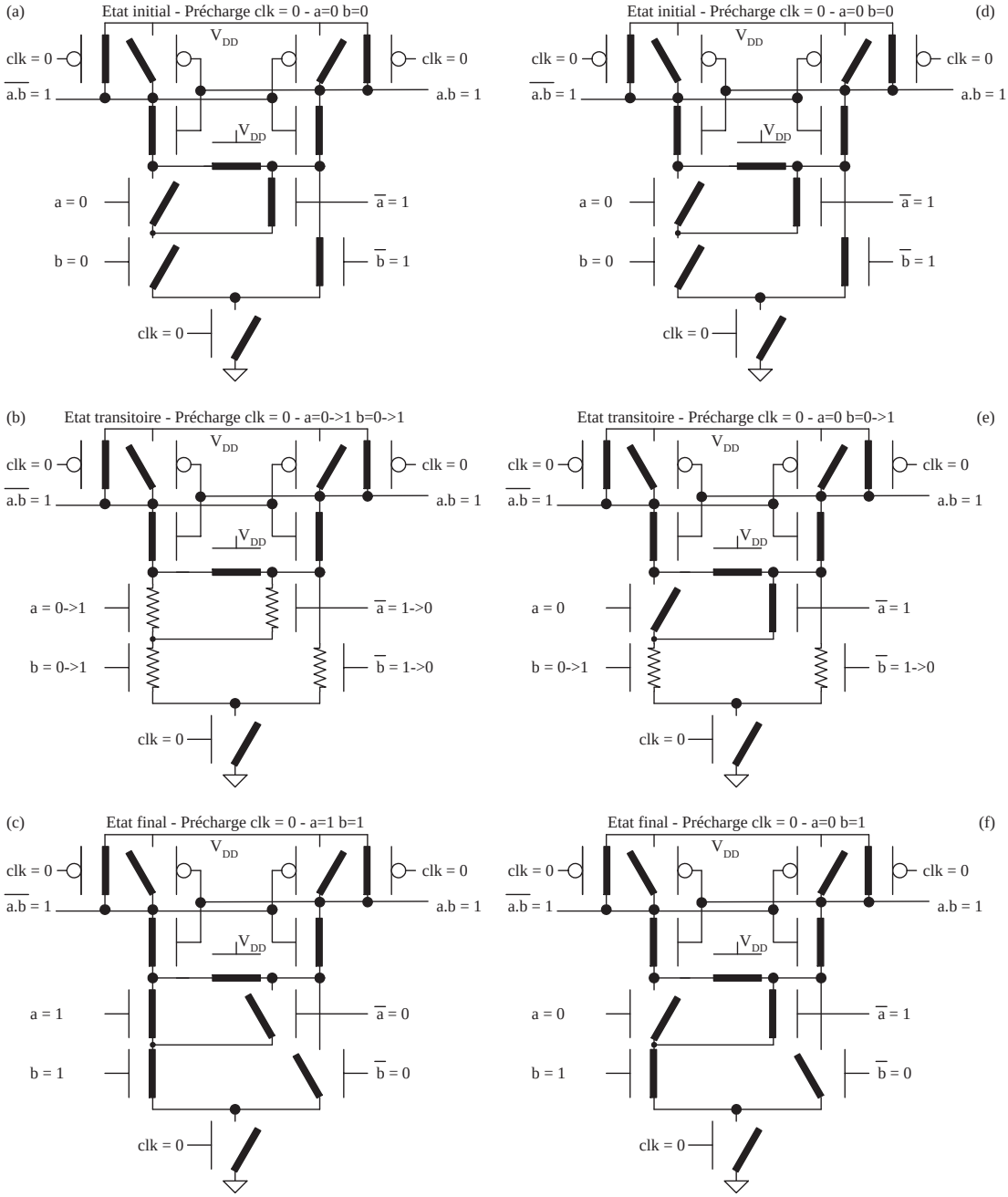


FIG. 8.21 – Phase de précharge SABL et caractéristiques électriques des réseaux de transistors

Par exemple, les évènements 8 et 15 définis au chapitre 1 appartiennent respectivement aux

ensembles DPA $E1$ et $E0$. Dans les deux cas, les entrées a et b ont la même valeur initiale 0 et la porte est en phase de précharge (Figures 8.21-a et 8.21-d). Lors de cette même phase, les signaux a et b sont prépositionnés à leur valeurs logiques suivantes, 11 pour l'évènement 8 et 01 pour l'évènement 15 (Figures 8.21-b et 8.21-e). Les réseaux de transistors entre ces deux figures sont différents et ont des caractéristiques électriques différentes. Leurs comportements instantanés sont donc différents lors de la phase de précharge lorsque les signaux a et b ont des valeurs transitoires.

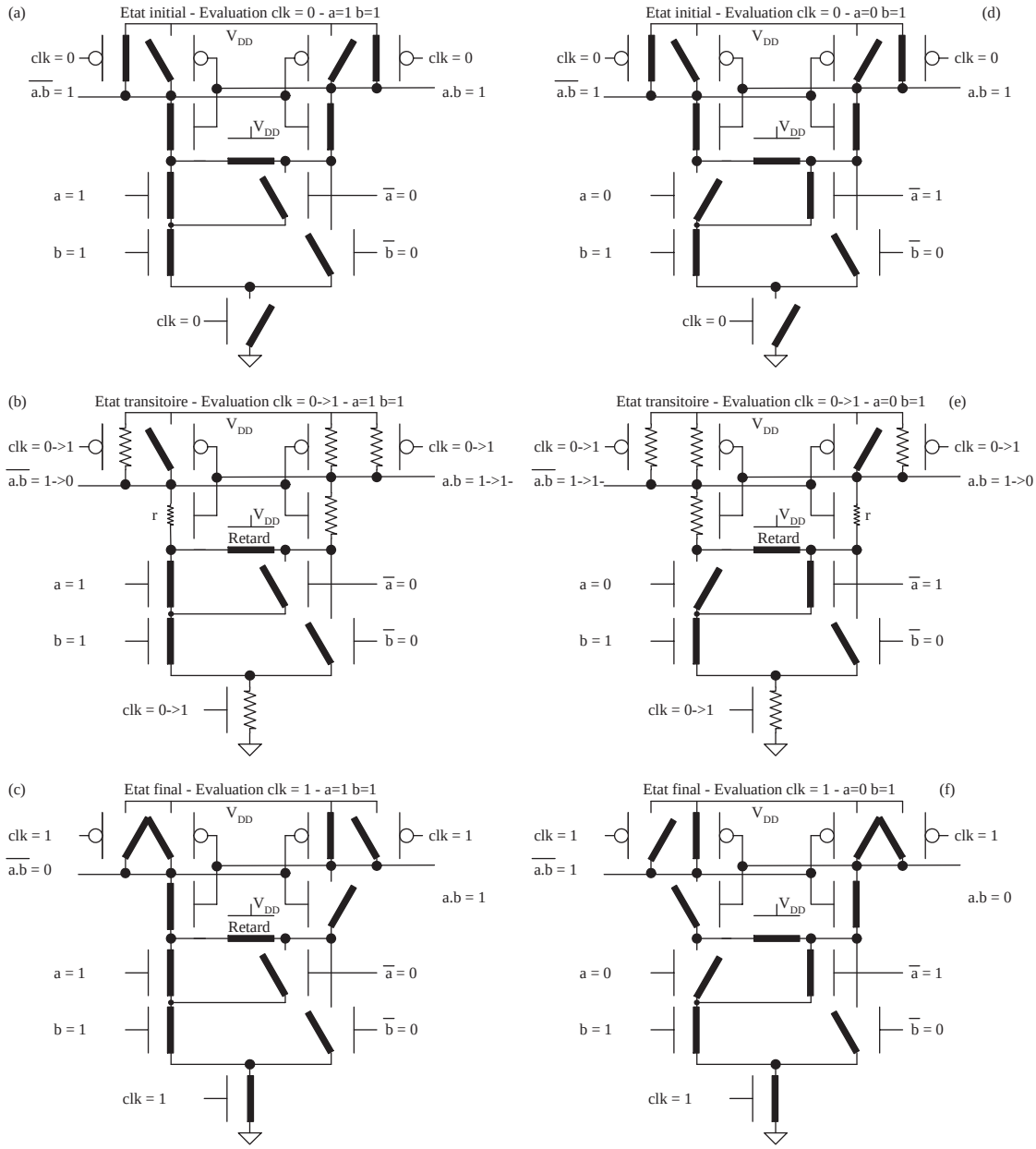


FIG. 8.22 – Phase d'évaluation SABL et caractéristiques électriques des réseaux de transistors

Dans l'état final, les signaux a et b prennent respectivement les valeurs 11 pour l'évènement 8 et 01 pour l'évènement 15 (Figures 8.21-c et 8.21-f). Ceci représente aussi les états initiaux de la

phase d'évaluation de la technique SABL (Figures 8.22-a et 8.22-d).

Lors de la phase d'évaluation (Figures 8.22-b et 8.22-e), le signal *clk* passe par une valeur transitoire. De fait, les réseaux de transistors entre ces deux figures sont différents et ont des caractéristiques électriques différentes. Leurs comportements instantanés sont donc différents lors de la phase d'évaluation.

Puisque les réseaux de transistors ont des caractéristiques électriques instantanées différentes lors des phases de précharge et d'évaluation pour des évènements différents (Exemple des évènements 8 et 15, figures 8.23-a et 8.23-b), la DPA peut être observée à ces deux instants pour la technique SABL (Figure 8.23-c).

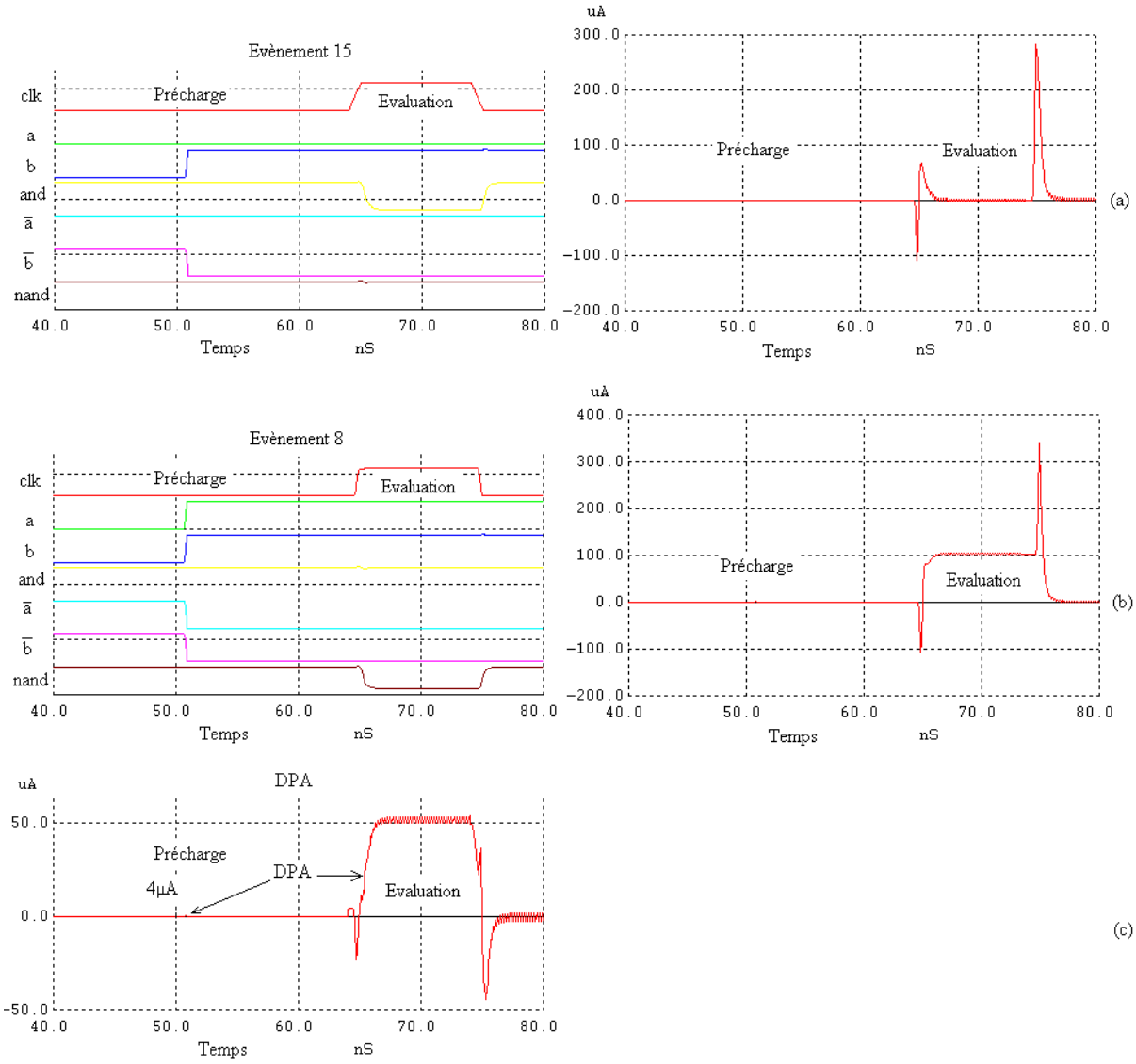


FIG. 8.23 – Simulation de la DPA en technique SABL

Quatrième partie

Une solution au problème

Chapitre 9

Solution technique versus sources microélectroniques de la DPA

Sommaire

9.1 Introduction

La solution de contre-mesure DPA proposée dans cette dernière partie est construite en deux étapes. La première étape décrite dans le présent chapitre fait abstraction des signaux de commande et concerne la circuiterie avale de la solution. Les entrées qui commandent cette circuiterie sont supposées être générées sans dispersion instantanée des caractéristiques électriques. Ce chapitre montre comment construire une fonction logique insensible à la DPA à partir de propriétés des transistors en technologie CMOS. La seconde étape décrite dans le chapitre suivant est la mise en application de la solution. Elle explique comment générer les entrées insensibles à la DPA et montre la circuiterie amont de la solution. L'ensemble des deux circuiteries forme une solution permettant la construction d'une porte logique intrinsèquement résistante à la DPA.

9.2 Propriété d'isolation du transistor MOS dans l'état d'accumulation

Le transistor en état d'accumulation, c'est à dire lorsqu'il est ouvert du point de vue logique, présente une capacité de grille correspondant à la totalité de la surface de grille (Équation 1.10) et une impédance infinie entre le drain et la source (Équation 1.13). Dans ces conditions, la polarisation du drain et de la source permet de fixer la tension de seuil à une valeur V_{THacc}. Trois cas se présentent alors pour le transistor NMOS ou PMOS. Le premier est défini par des tensions de source et de drain toutes deux égales au 0 logique (Figure 9.1-a), le deuxième est défini par une tension de source égale au 1 logique et une tension de drain égale au 0 logique et inversement (Figure 9.1-b) et le troisième est défini par des tensions de source et de drain toutes deux égales au 1 logique (Figure 9.1-c). Un transistor NMOS ou PMOS en état d'accumulation a toujours la même valeur de capacité de grille et toujours la même impédance infinie entre le drain et la source quelles que soient les tensions de drain et de source. C'est en ce sens que le transistor peut être considéré comme ayant une propriété de forte isolation du drain et de la source en état d'accumulation. Cependant, cette propriété est tout de même paramétrée par les tensions de drain et de source du transistor. En effet, les trois cas présentés dans la figure 9.1 ont tous une tension de seuil différente parce qu'elle dépend justement de ces deux tensions. Le comportement électrique instantané du transistor varie donc d'un cas à l'autre et plus particulièrement lorsque la tension de grille le fait passer de l'état d'accumulation aux états de déplétion puis de forte inversion. Ce déséquilibre se retrouve systématiquement dans les portes logiques dans les réseaux série de transistors et permet l'attaque DPA.

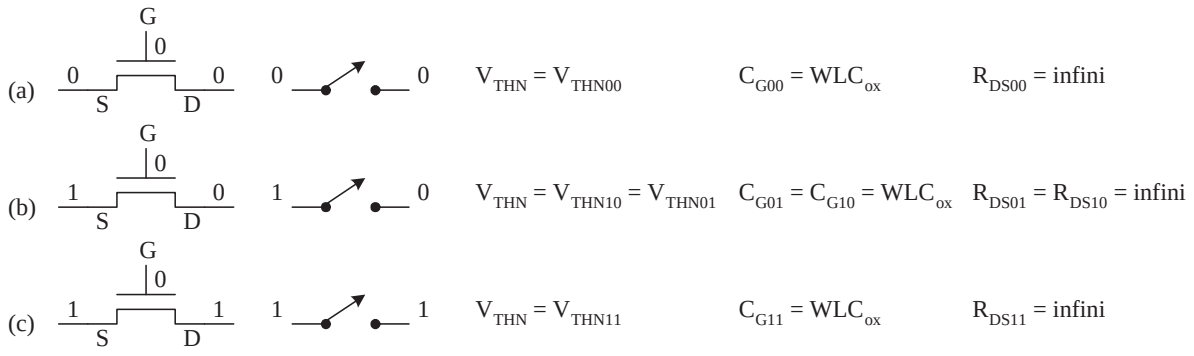


FIG. 9.1 – Paramètres électriques du transistor NMOS en état d'accumulation

9.3 Propriété de déséquilibre des portes CMOS

Une porte CMOS est dite déséquilibrée parce que la transmission électrique du résultat logique 0 est toujours différente de la transmission électrique du résultat logique 1. C'est le cas pour toutes les portes CMOS qu'elles soient de type $NAND$, NOR, XOR, etc.
Dans le cas d'une porte à deux entrées comme la porte $NAND2$, trois combinaisons d'entrée donnent le résultat logique 1 et une seule donne le résultat logique 0. Il est possible de conceptualiser le déséquilibre en considérant quatre commandes représentant les quatre combinaisons des entrées 00, 01, 10 et 11 chacune connectée à la grille d'un transistor. Trois transistors ont leurs sources connectées au 1 logique et un seul a sa source connectée au 0 logique pour créer la fonction $NAND2$ (Figure 9.2). Lorsque les entrées ont pour valeur logique 00, la commande C_{00} prend la valeur logique 1 et le transistor dont la grille est commandée par cette commande transmet la valeur logique 1. Il en est de même pour les valeurs logiques 01 et 10 des entrées. Lorsque les entrées ont pour valeur logique 11, la commande C_{11} prend la valeur logique 1 et le transistor dont la grille est commandée par cette commande transmet le 0 logique.

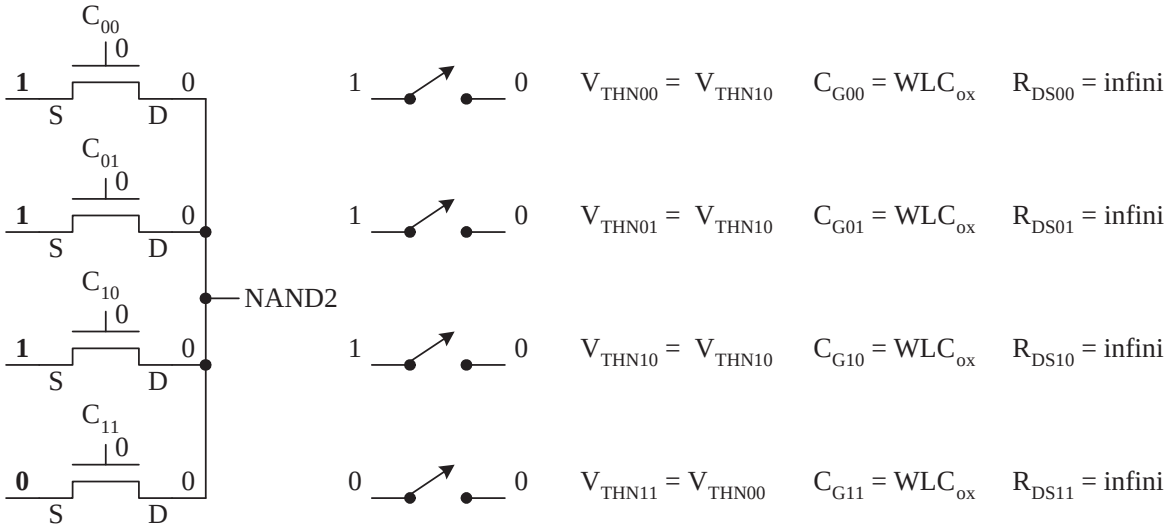


FIG. 9.2 – Conceptualisation du déséquilibre électrique instantané d'une porte CMOS

Les quatre transistors ont obligatoirement leurs drains à la même référence correspondant à un état initial, 0 par exemple. Une seule commande est active à la fois et transmet soit 1 soit 0. Les transistors transmettant le 1 logique ont tous les mêmes tensions de drain et de source. Ils ont tous les trois la même tension de seuil V_{THN10}. Or, le transistor transmettant le 0 logique a obligatoirement une tension de source différente de celle des trois autres transistors. Sa tension de seuil est donc différente et égale à V_{THN00}. Dans une porte CMOS, certains transistors ont obligatoirement une polarisation drain-source différente entre la transmission du 1 logique et la transmission du 0 logique. Ceci a pour conséquence des comportements électriques instantanés toujours différents entre les combinaisons des entrées et permet la DPA.

9.4 La nécessité d'un nouveau codage de données

L'objectif est donc maintenant de supprimer les déséquilibres observés. Ceci va se traduire par la nécessité théorique d'introduire un nouveau codage de donnée. La première approche consiste à tirer parti des propriétés d'isolation du transistor tandis que la seconde approche exploite la répartition en ensembles DPA afin de supprimer les déséquilibres.

9.4.1 Équilibrage par extension des réseaux de transistors

Dans la figure 9.2, les transistors commandés par C_{00}, C_{01} et C_{10} ne se différencient du transistor commandé par C_{11} que par la tension de source. Pour que le transistor commandé par C_{00} ne se différencie plus du transistor commandé par C_{11}, il est nécessaire de faire une extension de circuiterie afin que C_{00} commande aussi un transistor dont la tension de source soit identique à celle du transistor commandé par C_{11}. Il faut réitérer cette démarche aux transistors commandés par C_{01} et C_{10}. Cependant, le transistor commandé par C_{11} n'a jamais de tension de source égale à la tension du 1 logique. Il est aussi nécessaire de faire une extension de circuiterie afin que C_{11} commande aussi un transistor dont la tension de source soit celle du 1 logique (Figure 9.3).

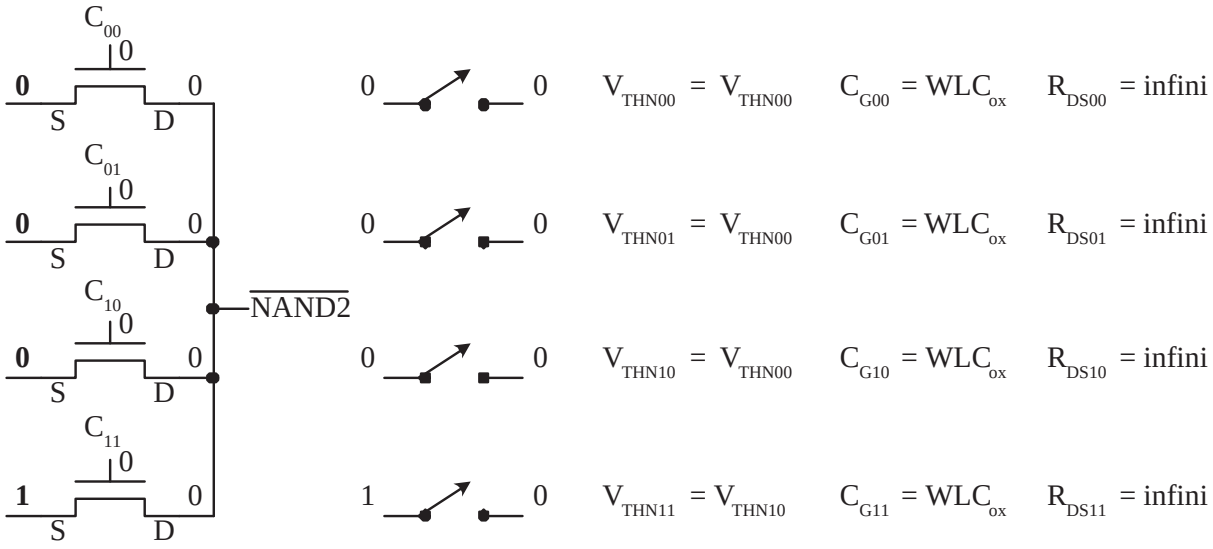


FIG. 9.3 – Extension de circuiterie des transistors

Cette extension de circuiterie permet d'avoir en permanence deux transistors commandés par chaque commande dont l'un a le drain et la source polarisés 10 et le second a le drain et la source polarisés 00. Ceci équilibre de façon globale l'ensemble des transistors en faisant en sorte que le groupe de transistors présente en permanence des caractéristiques électriques instantanées identiques (Figure 9.4). L'extension de circuiterie ainsi obtenue est la fonction $\overline{NAND2}$ qui impose donc un nouveau codage de la sortie sur 2 rails (c1,c0) codant le 0 logique 01 et le 1 logique 10. Ce même codage est applicable aux bits d'entrée en posant pour le bit a le codage (a1,a0) et pour le bit b le codage (b1,b0). La commande C_{00}, qui signifie que les bits a et b sont égaux à 0, peut alors être réécrite $C_{a0,b0}$. Les commandes C_{01}, C_{10} et C_{11} deviennent respectivement $C_{a0,b1}$, $C_{a1,b0}$ et $C_{a1,b1}$.

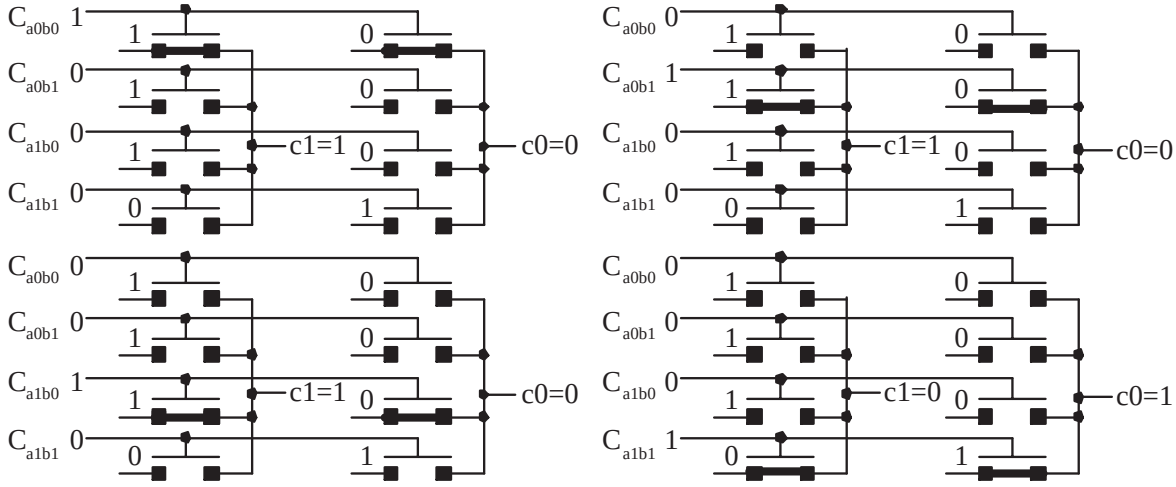
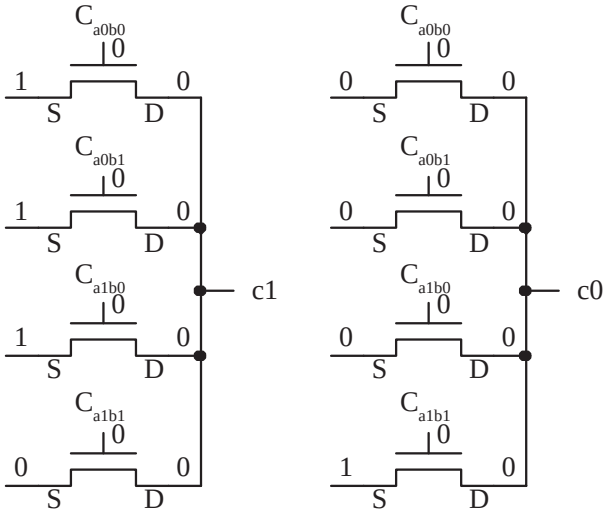


FIG. 9.4 – Principe de fonctionnement et équilibrage électrique instantané global

L'état initial, arbitrairement choisi dans l'exemple imposant le drain à 0 pour tous les transistors, est obtenu en fixant toutes les commandes à 0. Cet état ne représente aucune donnée valide et est appelé état non valide NV. Une donnée est dite valide lorsqu'un seul de ces deux rails prend la valeur logique 1 imposée par le nouveau codage. Lorsque les deux données sont valides, une seule des quatre commandes est alors active à la fois avec ce codage et l'état est dit valide V. Une combinaison de signaux pose problème lorsque les deux rails des données sont égaux à 1. Dans ce cas, un court-circuit est créé entre les transistors conduisant le 1 et le 0 simultanément. Cet état est dit interdit I (Figure 9.5).



	c = NAND(a,b)						
	a		b		c		
	a1	a0	b1	b0	c1	c0	état
	0	0	0	0	0	0	NV
C_{a0b0}	0	1	0	1	1	0	V
C_{a0b1}	0	1	1	0	1	0	V
C_{a1b0}	1	0	0	1	1	0	V
C_{a1b1}	1	0	1	0	0	1	V
	1	1	1	1	X	X	I

FIG. 9.5 – Extension de circuiterie et nouveau codage de donnée

9.4.2 Équilibrage par extension des ensembles DPA

Dans la partie III chapitre 8 les évènements microélectroniques ont été séparés en deux ensembles DPA appelés E0 et E1. Le critère de séparation des évènements microélectroniques était la valeur supposée 0 ou 1 d'un bit. Cependant, il est tout à fait possible d'utiliser un groupe de bits définissant une autre fonction de sélection DPA pour réaliser cette séparation. Ceci entraîne de fait une toute autre séparation des évènements microélectroniques. L'objectif est donc de s'affranchir de toutes les séparations possibles en s'appuyant sur les ensembles DPA d'évènements précédemment obtenus.

Les 16 évènements microélectroniques possibles pour les signaux d'entrée d'une porte logique CMOS à deux entrées ont été définis au chapitre 1 (Figure 9.6).

E9	E10	E11	E12
E1	E2	E3	E4
E5	E6	E7	E8
E13	E14	E15	E16

FIG. 9.6 – Les 16 évènements microélectroniques applicables à une porte CMOS à deux entrées

La DPA consiste à séparer les 16 évènements en deux ensembles selon un critère. Dans la partie II, la DPA sur 1 bit sépare les évènements en deux ensembles $E0$ et $E1$ selon la valeur 0 ou 1 calculée du bit (Figure 9.7).

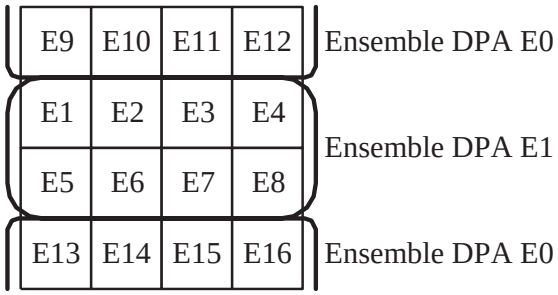


FIG. 9.7 – Séparation en ensembles DPA $E0$ et $E1$ des 16 évènements microélectroniques

Les évènements 3 et 14 diffèrent l'un de l'autre par la simple permutation des rôles des signaux d'entrée a et b. L'évènement 3 est défini par un signal constant sur a et une transition descendante sur b alors que l'évènement 14 est défini par une transition descendante sur a et un signal constant sur b. Ces évènements sont appelés évènements permutés (Figure 9.8). Les évènements 8 et 10 diffèrent totalement l'un de l'autre avec des signaux d'entrée a et b ayant des transitions différentes. Ils sont appelés exclusifs. La DPA 1 bit sépare donc les évènements en deux ensembles donc chacun est composé d'évènements permutés et exclusifs (Figure 9.8).

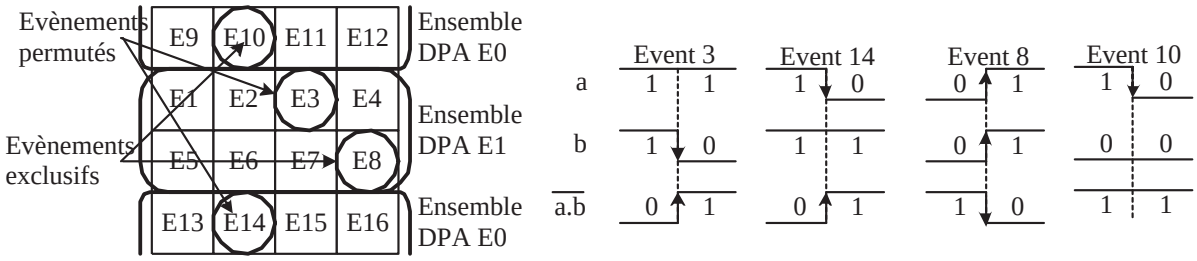


FIG. 9.8 – Ensembles DPA $E0$ et $E1$ et évènements microélectroniques permutés et exclusifs

La DPA 1 bit n'est qu'un exemple de DPA reposant sur une fonction de sélection ne calculant qu'un bit pour la séparation des ensembles d'évènements. Une solution de contre-mesure DPA doit s'affranchir de toute séparation possible des évènements et aussi de toute statistique d'occurrence des évènements. Ceci est réalisé par une extension du tableau des évènements (Figure 9.9). Cette extension du tableau permet d'équilibrer les ensembles DPA mais introduit la nécessité d'un nouveau codage de donnée correspondant à une extension de circuiterie prenant en compte les évènements complémentaires (Figure 9.9).

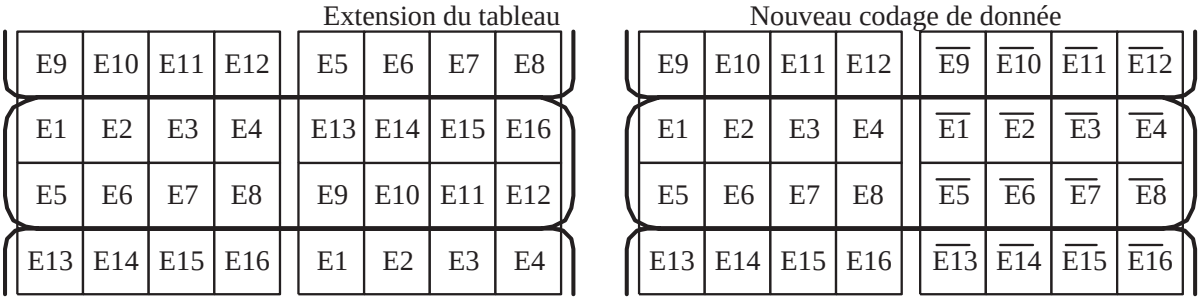


FIG. 9.9 – Extension du tableau des évènements et nouveau codage de donnée

La prise en compte des signaux complémentaires encode les signaux d'entrée sur deux rails par bit. Afin de garder une cohérence globale, la sortie de la porte doit aussi être codée par un double rail par bit. En posant que le 1 logique est codé 10 et que le 0 logique est codé 01 le nouveau codage aboutit au même résultat que précédemment (Figures 9.5 et 9.10).

a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1

FIG. 9.10 – Extension du tableau des évènements et nouveau codage de donnée pour une porte $Nand2$

9.5 La nécessité d'un nouveau protocole de données

9.5.1 Nécessité d'un état électrique initial connu par extension des réseaux de transistors

L'approche transistor a initialement défini la nécessité d'un nouveau codage de donnée (Figure 9.5). Dans l'exemple de la porte $Nand2$, un état initial imposant le drain à 0 pour tous les transistors est nécessaire. Il garantit un état électrique initial connu pour toute donnée valide et permet l'équilibrage électrique instantané global. La porte est dans cet état lorsque toutes les commandes sont à 0, c'est à dire lorsque tous les doubles rails ont pour valeur 00. Ceci correspond à la nécessité d'un état de donnée non valide. La porte ne peut évaluer un calcul logique que lorsque les données sont dites valides. Un état de donnée valide est nécessaire et correspond à une donnée dont un seul de ces deux rails prend la valeur logique 1 imposée par le nouveau codage. Une combinaison électrique de signaux pose problème lorsque les deux rails des données sont égaux à 1. Dans ce cas, un court-circuit est créé entre les transistors conduisant le 1 et le 0 simultanément. Cet état est dit interdit et les données ne doivent jamais l'atteindre.

Un nouveau protocole de donnée est nécessaire comprenant les états non valide et valide avec deux transitions d'état allant de non valide à valide et de valide à non valide. Passer dans l'état non valide vient de la nécessité de conserver un état électrique connu initial pour toute donnée valide. Dans le cas contraire, des passages logiques s'effectueraient à partir d'états éléctriques différents permettant à nouveau la DPA. Une extension de circuiterie est nécessaire afin de mettre en état non valide la sortie de la porte. Cela peut s'effectuer avec des transistors NMOS dont la source est reliée à la masse (Figure 9.11).

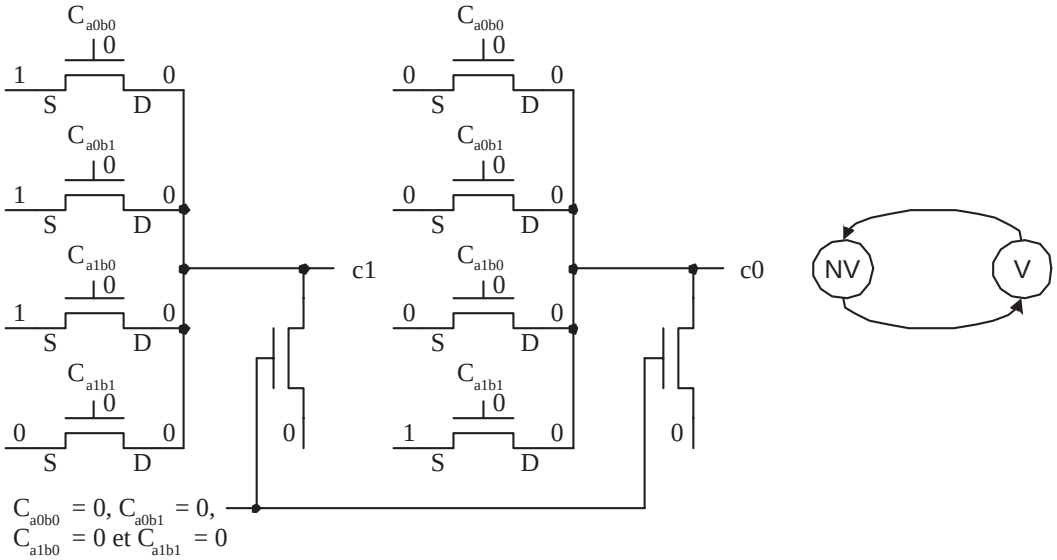


FIG. 9.11 – Extension de circuiterie et nouveau protocole de donnée

9.5.2 Nécessité d'un état électrique initial connu par suppression de l'effet mémoire

L'effet mémoire a été défini dans la partie III chapitre 7. L'extension de circuiterie qui découle de la nécessité d'un nouveau codage de donnée (Figure 9.3) est sensible à l'effet mémoire. Un premier état électrique est défini en supposant que l'état non valide initial ait une capacité parasite de sortie non chargée (Figure 9.12-a). Le passage dans l'état valide charge la capacité parasite C de sortie (Figure 9.12-b). Lors du retour à l'état non valide la capacité parasite de sortie conserve des charges générant l'effet mémoire (Figure 9.12-c). En conséquence, l'état non valide (Figure 9.12-a) et l'état non valide (Figure 9.12-c) sont bien deux états logiquement identiques mais électriquement différents.

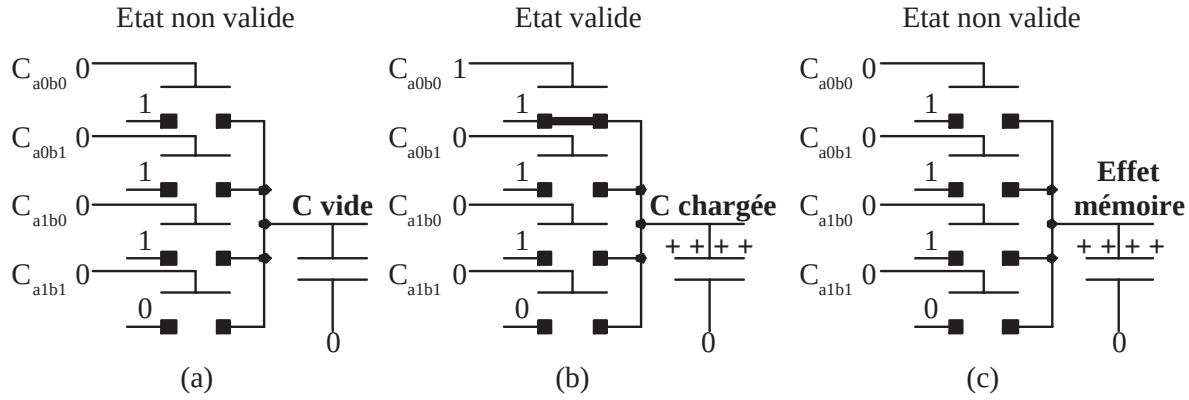


FIG. 9.12 – Extension de circuiterie et effet mémoire

Le passage de l'état non valide (Figure 9.12-a) à l'état valide (Figure 9.12-b) se fait donc à partir d'un état électrique dont la tension de drain est définie par l'absence de charge parasite dans la capacité C. Ce passage est différent du passage de l'état non valide (Figure 9.12-c) à l'état valide (Figure 9.12-b) qui se fait à partir d'un état électrique dont la tension de drain est définie par la présence de charges parasites dans la capacité C. Dans le premier cas, la tension de seuil des transistors est définie par une polarisation V_{THN10} ou V_{THN00} et dans le second cas la tension de seuil des transistors est définie par une polarisation V_{THN11} ou V_{THN01}. Ils ont des caractéristiques électriques instantanées différentes permettant la DPA.

L'effet mémoire impose donc un nouveau protocole de donnée en plus du nouveau codage afin de garantir le même état électrique pour toute transition de l'état non valide à l'état valide. Il utilise l'état non valide afin de décharger les capacités parasites de tous les effets mémoire possibles au sein des réseaux de transistors. Les données passent alors de l'état non valide à l'état valide comme précédemment. Toujours sur l'exemple de la porte $Nand2$, le nouveau protocole de donnée qui résulte de la gestion de l'effet mémoire a pour conséquence une extension de la circuiterie identique à celle de la figure 9.11. Des transistors NMOS permettent de décharger les capacités parasites et garantissent un état non valide connu et identique pour toute transition vers l'état valide. Ceci peut aussi être vu comme un système polyphasé avec une phase de décharge correspondant à l'état non valide et une phase d'évaluation correspondant à l'état valide.

9.6 Propriétés de la solution

9.6.1 Propriétés électriques

La solution proposée, qui permet un équilibrage électrique instantané global d'une fonction logique à deux entrées (Figure 9.11), repose sur un ensemble de propriétés électriques (Figure 9.13). L'état non valide, tout d'abord, place la porte dans un état électrique connu toujours identique. Cela supprime l'effet mémoire. Dans cet état, tous les transistors connectés à $c1$ et tous les transistors connectés à $c0$ sont dans l'état d'accumulation sauf les transistors supprimant l'effet mémoire. En conséquence, toute transition vers l'état valide se fait toujours à partir du même état logique et électrique. Les nœuds $c1$ et $c0$ sont isolés des tensions de sources des transistors auxquels ils sont connectés en exploitant la propriété d'isolation du transistor en état d'accumulation.

Lorsque les données sont valides, une seule commande est active et place la porte dans l'état valide. Une commande contrôle un transistor conduisant $c1$ et un transistor conduisant $c0$. Toutes les autres commandes sont inactives. Sur les deux transistors commandés par une commande active, l'un a sa source à 0 et l'autre a sa source à 1. Tous les autres transistors sont dans l'état d'accumulation. En conséquence, lors de la transition de l'état non valide vers l'état valide, un transistor commandé a une polarisation source-drain 10 et l'autre transistor commandé a une polarisation source-drain 00. Toute transition de l'état non valide à l'état valide met en jeu un couple de transistors dont les caractéristiques électriques initiales peuvent être globalement représentées par V_{THN10} et V_{THN00}. Les nœuds $c1$ et $c0$ sont isolés des sources des autres transistors en utilisant la propriété d'isolation du transistor en état d'accumulation. La sortie est valide lorsque $c1$ ou $c0$ passe à 1. Le 0 n'est jamais conduit, l'état non valide l'ayant déjà fixé. Seul le 1 est conduit. Dans l'état valide, un transistor commandé a une polarisation source-drain 11 et un transistor a une polarisation source-drain 00. Les caractéristiques électriques de l'état valide peuvent être globalement représentées par V_{THN11} et V_{THN00}. Cet état est le même quelle que soit la transition logique sur les données d'entrée.

Quel que soit le résultat logique de l'état valide, il est toujours électriquement le même et peut être représenté par V_{THN11} et V_{THN00}. Lors du retour de l'état valide à l'état non valide, la transition se fait toujours à partir du même état électrique. Une seule commande est désactivée. En conclusion, cette solution permet de construire toutes les fonctions logiques à deux entrées avec un comportement électrique instantané identique.

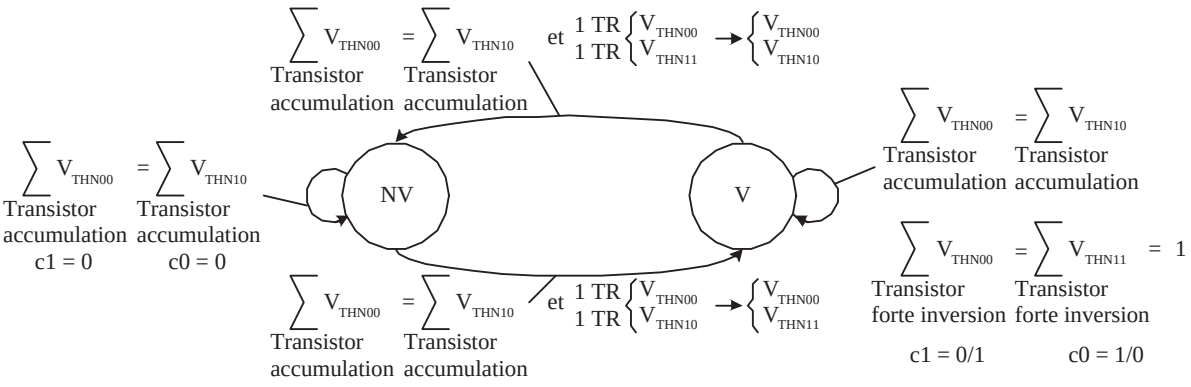


FIG. 9.13 – Propriétés électriques de la solution par état et par transition

9.6.2 Effets parasites

La solution exploite la propriété d'isolation du transistor en état d'accumulation. Cela signifie, dans une première approche, que la tension appliquée à la source des transistors restés en état d'accumulation n'a aucune influence sur la tension du drain lorsqu'un des transistors conduit. Cependant, dans l'exemple de la fonction $Nand2$ (Figure 9.11), le nœud $c1$ est connecté à trois transistors dont la polarité source-drain est 10, c'est à dire avec des caractéristiques électriques résumées par V_{THN10}, et un transistor dont la polarité source-drain est 00 avec des caractéristiques électriques résumées par V_{THN00}. Le nœud $c0$ est connecté à trois transistors dont la polarité source-drain est 00, c'est à dire avec des caractéristiques électriques résumées par V_{THN00}, et un transistor dont la polarité source-drain est 10 avec des caractéristiques électriques résumées par V_{THN10}. Le nombre de transistors connectés à $c1$ caractérisés par V_{THN10} est différent du nombre de transistors connectés à $c0$ caractérisés par V_{THN10}. Il en est de même pour les transistors caractérisés par V_{THN00} (Figure 9.14).

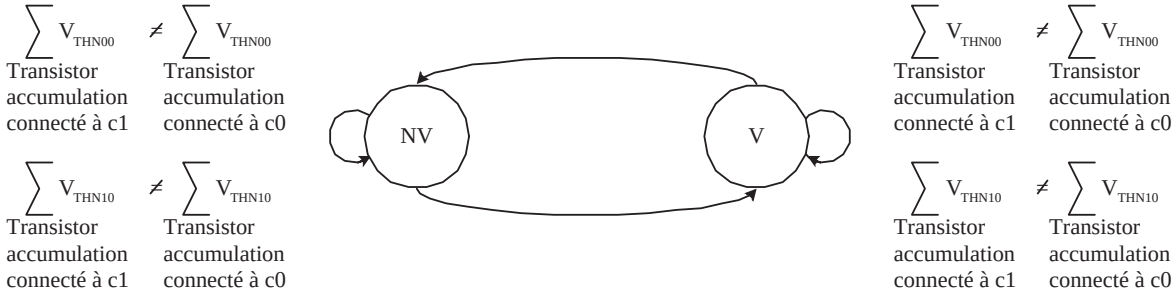


FIG. 9.14 – Effet parasite des tensions de source sur les nœuds $c1$ et $c2$

L'influence directe source-drain pour un transistor en état d'accumulation est bien négligeable pour la solution qui exploite cette propriété. Mais l'influence indirecte source-drain par la grille l'est moins. Cet effet parasite peut rompre l'équilibre électrique instantané recherché entre toutes les transitions logiques. En conséquence, les caractéristiques électriques du nœud $c1$ et du nœud $c0$ sont, dans une seconde approche, dépendantes des tensions de source des transistors malgré l'état d'accumulation qui est censé isoler les nœuds de ces tensions (Figure 9.15).

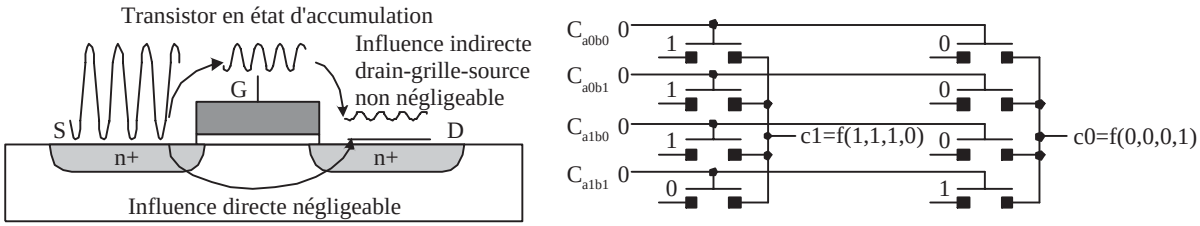


FIG. 9.15 – Influence directe et indirecte de la tension de source sur la tension de drain pour un transistor en état d'accumulation

Il existe plusieurs possibilités permettant de réduire cette influence indirecte par la grille. Il est possible d'augmenter la surface de grille en utilisant des transistors ayant une longueur de canal plus grande ou ayant une largeur plus grande. Une mise en série de transistors réalise le même effet. La couche d'isolant sous la grille peut aussi être augmentée. Dans la suite, une mise en

série de plusieurs transistors sera utilisée (Figure 9.16).

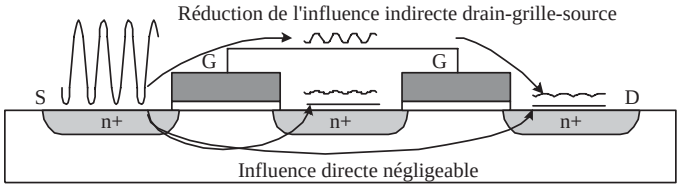


FIG. 9.16 – Réduction de l'influence indirecte source-grille-drain

9.6.3 Conception de la solution

Jusqu'à présent, la solution a été abordée sur le plan d'une fonctionnalité ayant des caractéristiques électriques instantanées constantes. Dans l'exemple de la fonction $Nand2$, la solution doit dans certains cas conduire un 0 logique et dans d'autres cas conduire un 1 logique. Le 0 logique est représenté par la masse Gnd et le 1 logique est représenté par la tension nominale de fonctionnement de la technologie employée V_{DD}. La conduction du 0 logique doit donc être faite par un transistor de type NMOS qui garantit un 0 logique fort, c'est à dire une tension égale à celle de la masse et non supérieure (Annexe A). La conduction du 1 logique doit être faite par un transistor de type PMOS qui garantit un 1 logique fort, c'est à dire une tension égale à la celle de la tension d'alimentation et non inférieure (Annexe A). Il existe une porte en technologie CMOS qui réalise parfaitement cette fonction, il s'agit de l'interrupteur CMOS. La mise en série d'interrupteurs permet de construire la solution tout en supprimant l'influence indirecte de la tension de source sur la tension de drain dans l'état d'accumulation. Cependant, l'effet mémoire est introduit par les réseaux série de transistors. Il est nécessaire d'utiliser des transistors NMOS afin de décharger les nœuds d'interconnexion de charges parasites lors de l'état non valide (Figure 9.17). Le nombre d'interrupteurs mis en série est fonction de la technologie. Une caractérisation est préalablement nécessaire afin de le déterminer.

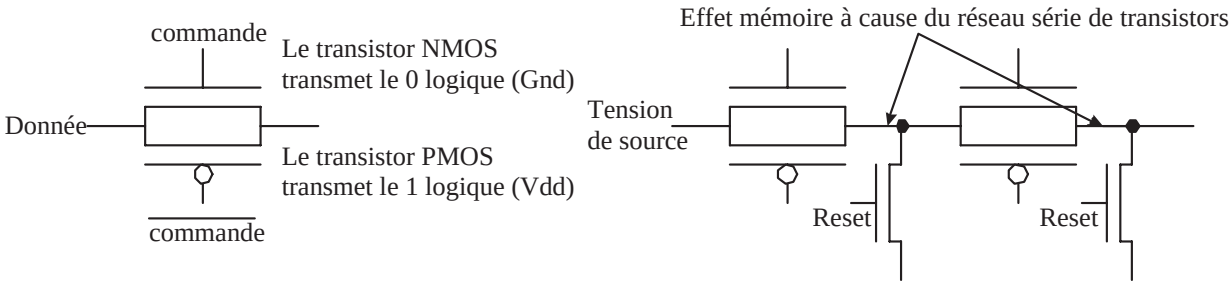


FIG. 9.17 – Interrupteur CMOS et circuiterie de la solution

9.6.4 Optimisation de la solution

Les nœuds $c1$ et $c0$ sont remis à 0 lors de l'état non valide. Lorsque les données sont valides, un seul des deux nœuds passe à 1 tandis que l'autre reste à 0. Cette dernière remarque signifie que le 0 logique n'est jamais conduit par la solution. Puisque le 0 logique n'est jamais conduit, il est possible de supprimer le transistor NMOS de l'interrupteur et de ne conserver qu'un demi-interrupteur construit avec un transistor PMOS. Ceci permet de réduire de façon importante le nombre de transistors de la solution tout en conservant les propriétés électriques instantanées constantes (Figure 9.18).

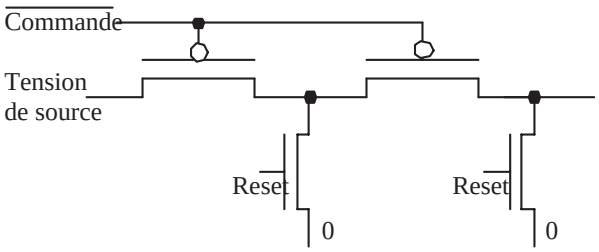
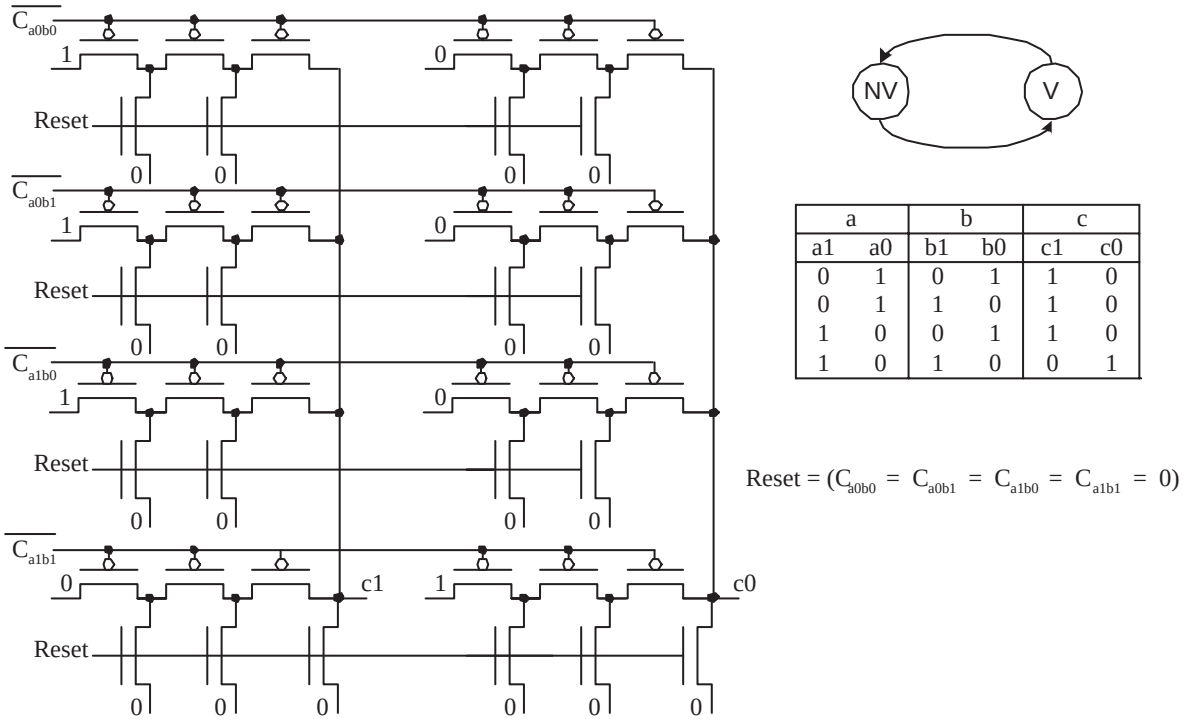


FIG. 9.18 – Optimisation de la circuiterie de la solution

La circuiterie de la fonction $Nand2$, prise en exemple jusqu'à présent, est représentée de façon optimisée avec interrupteurs par la figure 9.19



a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1

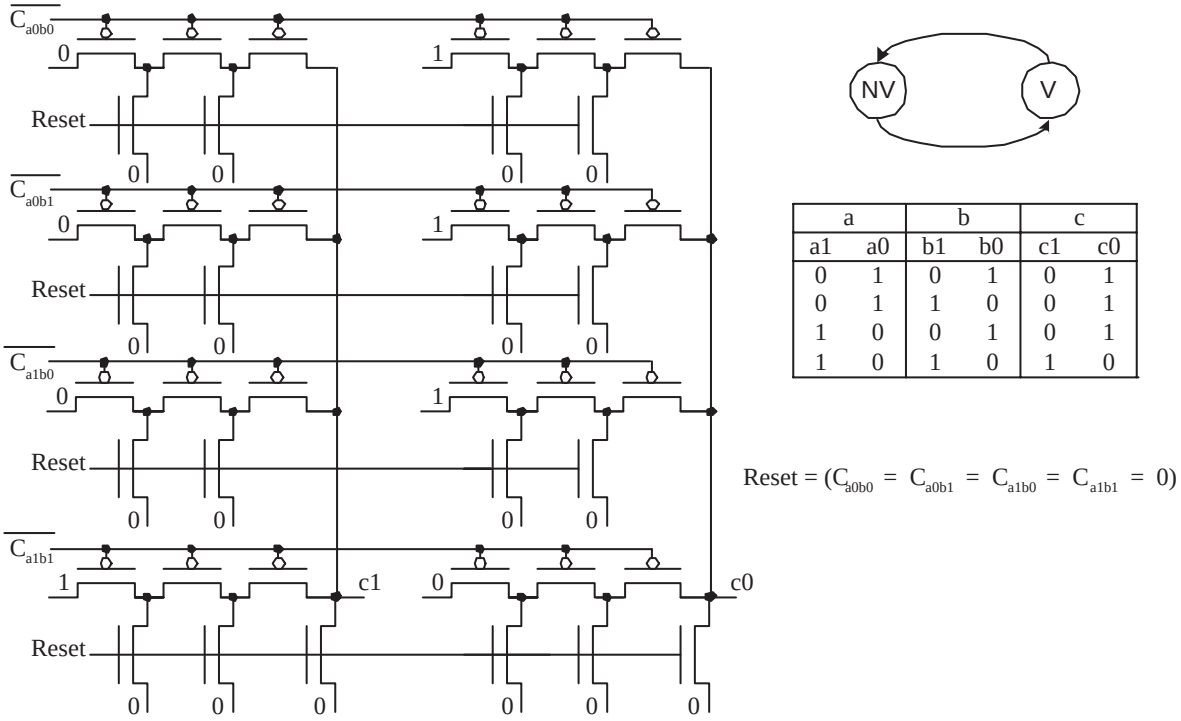
FIG. 9.19 – Conception de la fonction logique $Nand2$

9.7 Conception d'autres fonctions logiques

Les paragraphes 9.4 et 9.5 ont mis en évidence la nécessité d'un nouveau codage et d'un nouveau protocole de donnée. La fonction $Nand2$ a été prise en exemple afin d'illustrer la faisabilité de l'équilibrage électrique instantané d'une porte logique contrant les sources microélectroniques de la DPA (Figure 9.19). Á partir du même principe et du schéma de la figure 9.19, il est tout à fait possible de construire d'autres fonctions logiques.

9.7.1 La fonction logique And2

La table de vérité de la fonction $And2$ est rappelée dans la figure 9.20. La sortie prend la valeur logique 1, c'est à dire $c1 = 1$ et $c0 = 0$, lorsque les deux entrées sont logiquement égales à 1, c'est à dire $C_{a1b1} = 1$. Pour tous les autres cas lorsque $C_{a0b0} = 1$ ou $C_{a0b1} = 1$ ou $C_{a1b0} = 1$, la sortie est logiquement égale à 0, c'est à dire $c1 = 0$ et $c0 = 1$. La fonction est construite en connectant les transistors commandés par C_{a1b1} respectivement à 1 pour $c1$ et à 0 pour $c0$. Les transistors commandés par C_{a0b0}, C_{a0b1} et C_{a1b0} sont respectivement connectés à 0 pour $c1$ et à 1 pour $c0$ (Figure 9.20).

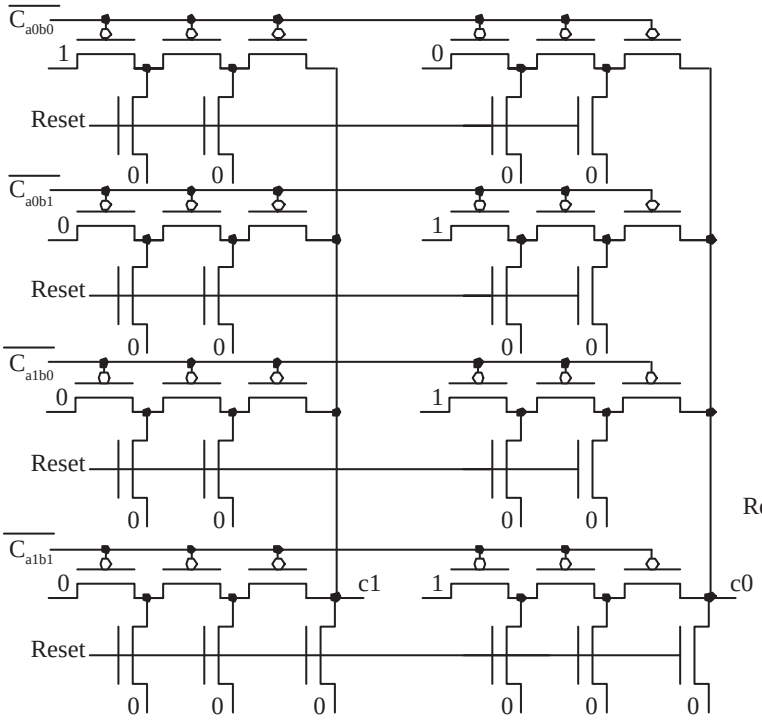


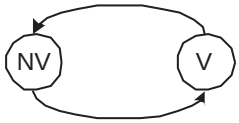
a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0

FIG. 9.20 – Conception de la fonction logique $And2$

9.7.2 La fonction logique Nor2

La table de vérité de la fonction $Nor2$ est rappelée dans la figure 9.21. La sortie prend la valeur logique 1, c'est à dire $c1 = 1$ et $c0 = 0$, lorsque les deux entrées sont logiquement égales à 0, c'est à dire $C_{a0b0} = 1$. Pour tous les autres cas lorsque $C_{a1b1} = 1$ ou $C_{a0b1} = 1$ ou $C_{a1b0} = 1$, la sortie est logiquement égale à 0, c'est à dire $c1 = 0$ et $c0 = 1$. La fonction est construite en connectant les transistors commandés par C_{a0b0} respectivement à 1 pour $c1$ et à 0 pour $c0$. Les transistors commandés par C_{a1b1}, C_{a0b1} et C_{a1b0} sont respectivement connectés à 0 pour $c1$ et à 1 pour $c0$ (Figure 9.21).





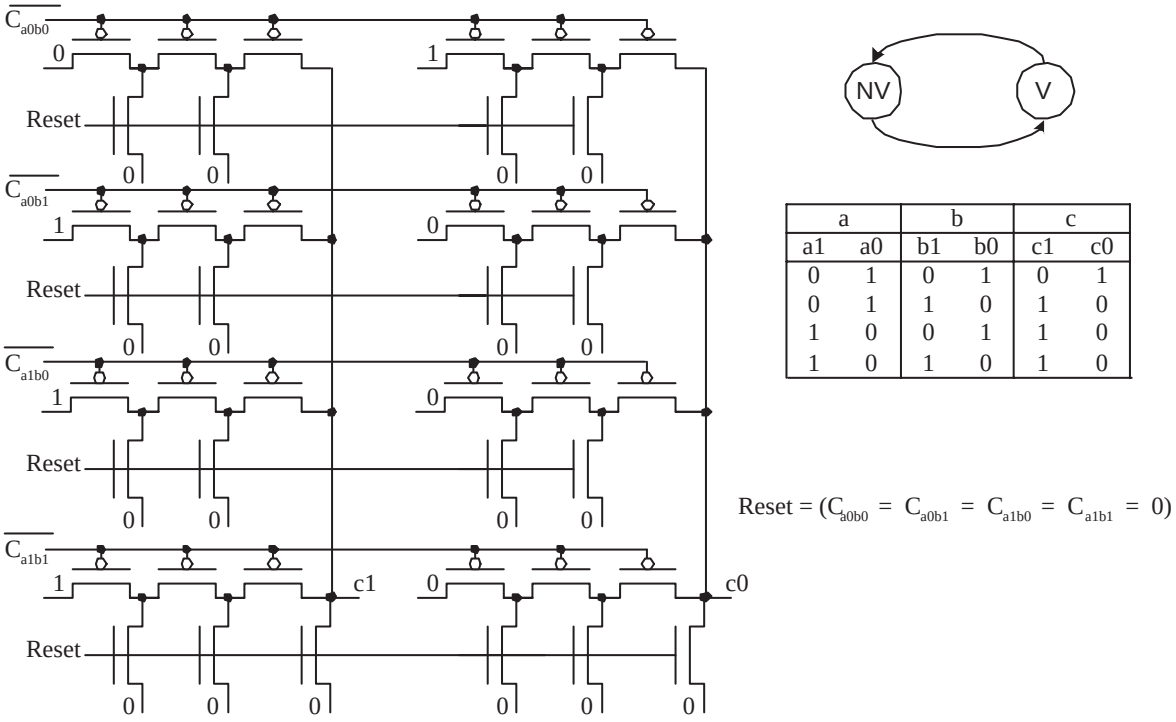
a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1

Reset = (C_{a0b0} = C_{a0b1} = C_{a1b0} = C_{a1b1} = 0)

FIG. 9.21 – Conception de la fonction logique $Nor2$

9.7.3 La fonction logique Or2

La table de vérité de la fonction $Or2$ est rappelée dans la figure 9.22. La sortie prend la valeur logique 1, c'est à dire $c1 = 1$ et $c0 = 0$, lorsque l'une des deux entrées est logiquement égale à 1, c'est à dire $C_{a1b1} = 1$ ou $C_{a0b1} = 1$ ou $C_{a1b0} = 1$. Lorsque $C_{a0b0} = 1$, la sortie est logiquement égale à 0, c'est à dire $c1 = 0$ et $c0 = 1$. La fonction est construite en connectant les transistors commandés par $C_{a1b1} = 1$, $C_{a0b1} = 1$ et $C_{a1b0} = 1$ respectivement à 1 pour $c1$ et à 0 pour $c0$. Les transistors commandés par C_{a0b0} sont connectés à 0 pour $c1$ et à 1 pour $c0$ (Figure 9.22).

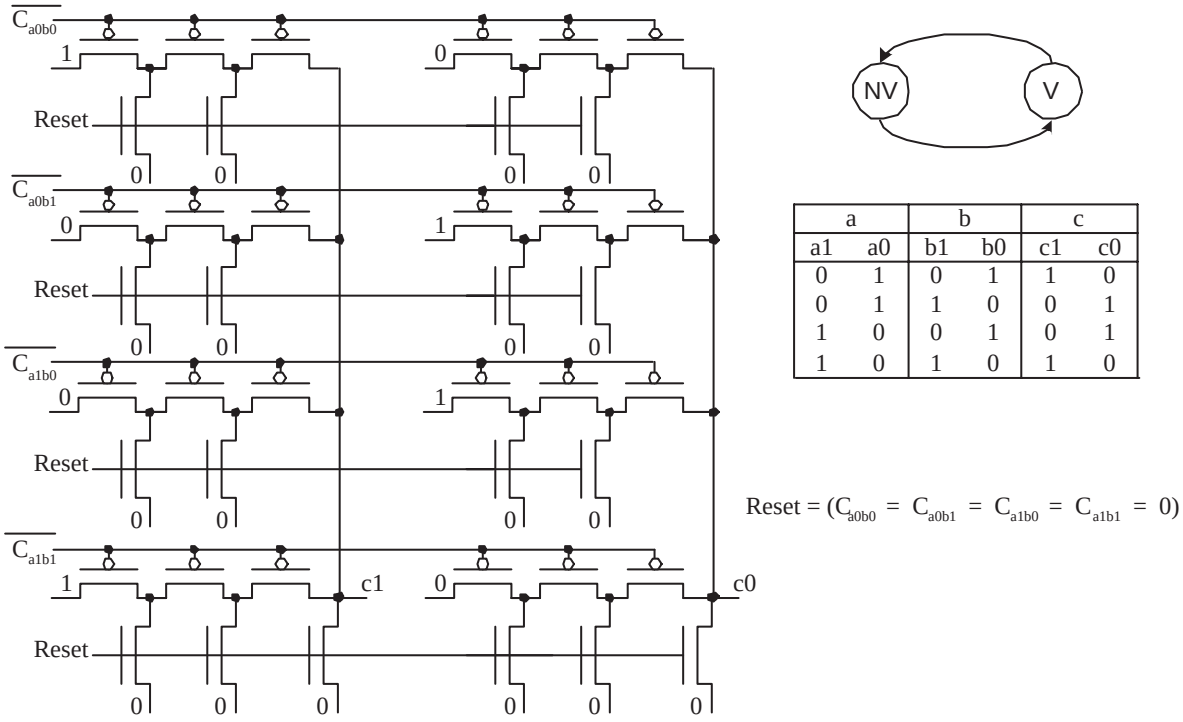


a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0

FIG. 9.22 – Conception de la fonction logique $Or2$

9.7.4 La fonction logique Xnor2

La table de vérité de la fonction $Xnor2$ est rappelée dans la figure 9.23. La sortie prend la valeur logique 1, c'est à dire $c1 = 1$ et $c0 = 0$, lorsque les deux entrées sont logiquement égales entre elles, c'est à dire $C_{a1b1} = 1$ ou $C_{a0b0} = 1$. Lorsque les deux entrées sont logiquement différentes entre elles, c'est à dire $C_{a1b0} = 1$ ou $C_{a0b1} = 1$, la sortie est logiquement égale à 0, c'est à dire $c1 = 0$ et $c0 = 1$. La fonction est construite en connectant les transistors commandés par C_{a1b1} et C_{a0b0} respectivement à 1 pour $c1$ et à 0 pour $c0$. Les transistors commandés par C_{a1b0} et C_{a0b1} sont connectés à 0 pour $c1$ et à 1 pour $c0$ (Figure 9.23).

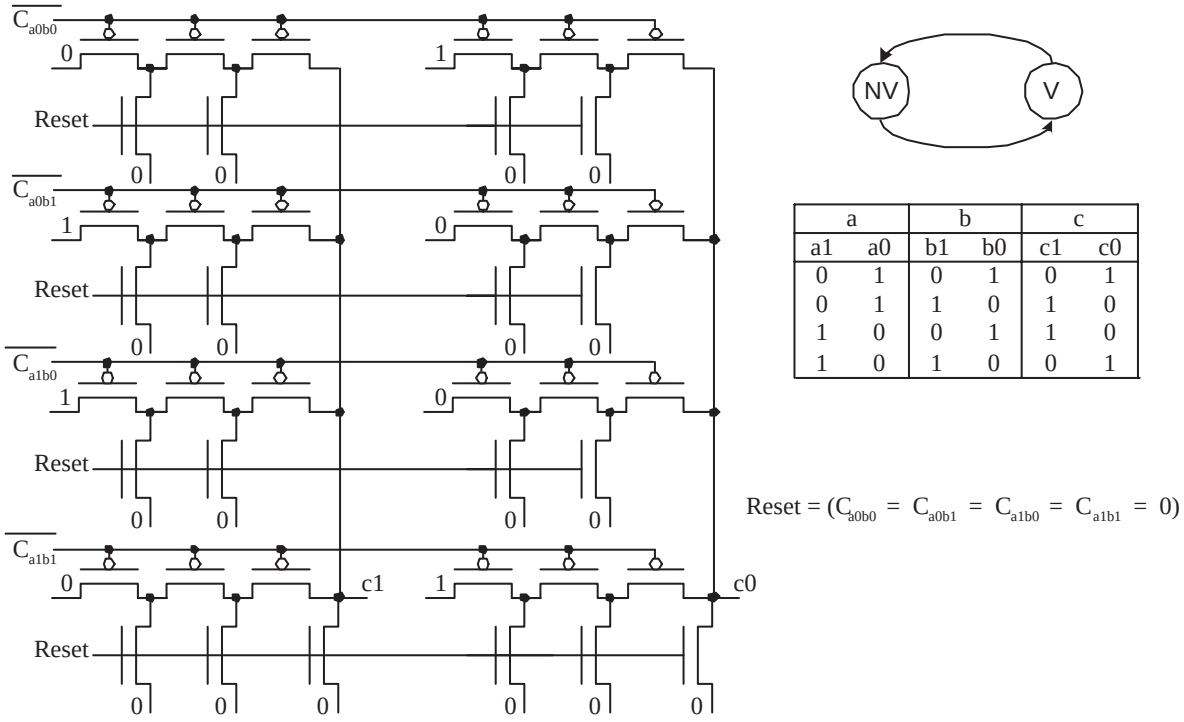


a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0

FIG. 9.23 – Conception de la fonction logique $Xnor2$

9.7.5 La fonction logique Xor2

La table de vérité de la fonction $Xor2$ est rappelée dans la figure 9.24. La sortie prend la valeur logique 1, c'est à dire $c1 = 1$ et $c0 = 0$, lorsque les deux entrées sont logiquement différentes entre elles, c'est à dire $C_{a1b0} = 1$ ou $C_{a0b1} = 1$. Lorsque les deux entrées sont logiquement différentes entre elles, c'est à dire $C_{a1b1} = 1$ ou $C_{a0b0} = 1$, la sortie est logiquement égale à 0, c'est à dire $c1 = 0$ et $c0 = 1$. La fonction est construite en connectant les transistors commandés par C_{a1b0} et C_{a0b1} respectivement à 1 pour $c1$ et à 0 pour $c0$. Les transistors commandés par C_{a1b1} et C_{a0b0} sont connectés à 0 pour $c1$ et à 1 pour $c0$ (Figure 9.24).



a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1

FIG. 9.24 – Conception de la fonction logique $Xor2$

9.8 Testabilité de la solution

Un facteur critique très important lors de toute conception est la prise en compte du besoin de test en production. La conception doit se faire en intégrant les méthodes de test des circuits dès le début des spécifications. Elles sont en effet dimensionnantes en terme de performance comme le temps d'exécution et en terme de valeur économique avec l'accroissement de la surface.
Un circuit intégré à n entrées binaires est exhaustivement testé avec 2^n vecteurs de test. Un algorithme cryptographique tel que l'AES a 128 bits de message et jusqu'à 256 bits de clé. Théoriquement, il faut 2^{256} vecteurs pour le tester exhaustivement. Ceci est irréalisable avec les technologies actuelles. Dans le cas contraire cela signifierait que l'AES pourraît être cassé en force brute. Afin de déterminer si un circuit est fonctionnellement bon ou mauvais, il est nécessaire de définir des modèles de faute. Un modèle consiste à dire comment une faute est créée et quel est son impact sur le circuit. Une faute doit être idéalement observable. L'observabilité d'une faute dépend de son environnement qui, dans certains cas, ne permet pas toujours de transmettre une faute sur la sortie du circuit. L'observabilité des fautes est améliorée par les méthodes de test. Certaines fautes sont impossibles à observer par des vecteurs de test et d'autres nécessiteraient un si grand nombre de vecteurs que le coût de test deviendrait irraisonnable. Il est alors nécessaire de les contrôler par d'autres méthodes. La contrôlabilité d'une faute est la possibilité d'imposer une valeur à un nœud du circuit. Dans ce cas, un concepteur peut être amené à ajouter des ressources matérielles spécifiques pour le test comme des plots d'accès direct.
Dans le cas d'un algorithme cryptographique, les clés sont mémorisées dans un registre de n bits. Elles ne sont jamais accessibles sur un bus interne en lecture. Un concepteur est confronté au problème du test du collage à 0 ou à 1 des n bascules du registre. Autrement dit, il doit être en mesure de trouver une relation entre la sortie chiffrée observable par lecture et les n bits de clé. Elle doit être efficace pour faire le test avec peu de vecteurs. En ajoutant un mode de test avec la possibilité d'un accès en lecture au registre de clé, le concepteur peut tester tous les collages à 0 et à 1 pour tous les bits de clé avec seulement 2 vecteurs. Il est donc très important de prendre en compte le test dès les spécifications et d'autant plus pour un système de sécurité. En effet, celui-ci peut être la source d'une faille de sécurité majeure si un attaquant peut remettre le mode de test en fonctionnement et lire les clés.
L'objectif du test est d'avoir une couverture de fautes suffisante pour garantir la confiance dans le bon fonctionnement du circuit intégré. La couverture de fautes est le nombre de fautes observées lors du test sur le nombre total de fautes déterminées dans le circuit. L'industrie vise souvent l'objectif de 95% de couverture de fautes en éliminant les fautes des ressources matérielles destinées au test. En effet, le test n'est pas testé et seule la partie fonctionnelle a une importance. Si une erreur survient à cause des ressources matérielles dédiées au test, le circuit sera mis au rebus.
La couverture de fautes ne doit pas être confondue avec la couverture de code VHDL ou Verilog. Les tests fonctionnels du circuit doivent systématiquement aboutir à une couverture de code de 100%. Cet objectif est toujours possible. Dans le cas contraire, il existe un manque dans le test fonctionnel du composant qui sera très probablement la source d'un disfonctionnement logique après la production. Ceci pourra à nouveau entraîner une faille de sécurité majeure.

9.8.1 Modèle de faute avec collage à 0 et collage à 1

Ce modèle de faute est le plus connu. Il consiste à modéliser une entrée erronée de la porte avec un collage à la masse, en anglais Stuck-At-0 ou SA0, ou avec un collage à la tension d'alimentation, Stuck-At-1 ou SA1. Ce type de faute apparaît fréquemment pour les oxydes minces

qui peuvent être court-circuités lors d'un défaut de production (Figure 9.25).

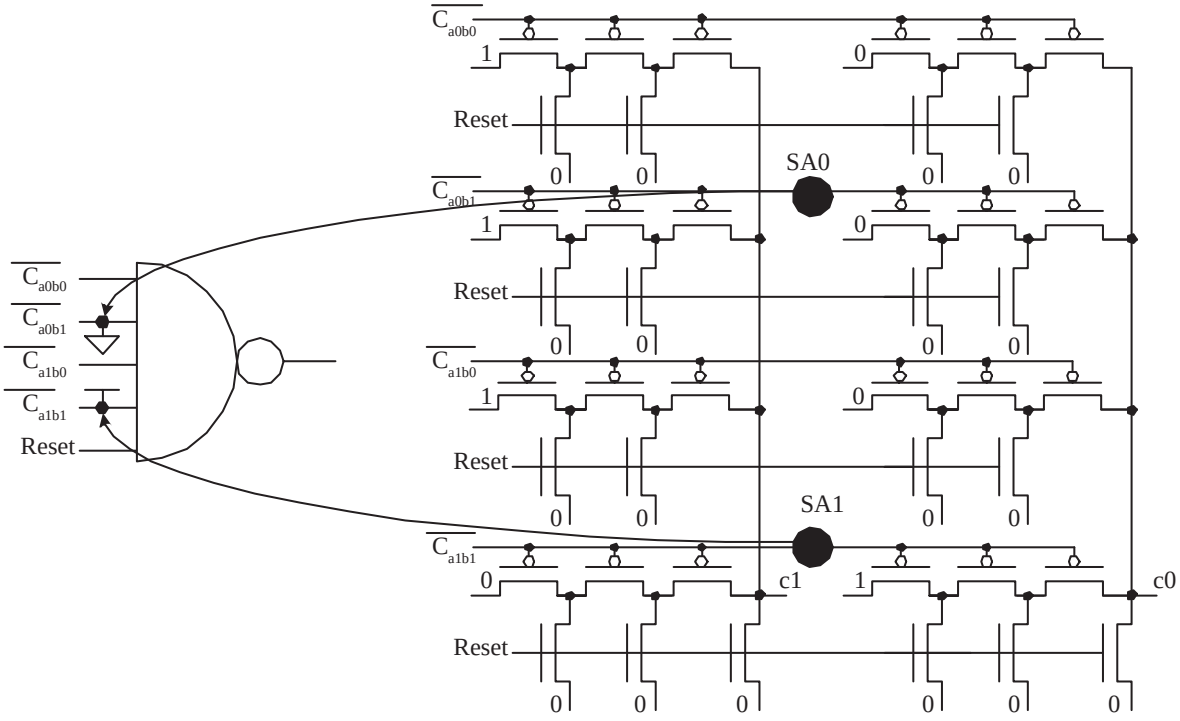


FIG. 9.25 – Modèle de faute SA1 et SA0

9.8.2 Modèle de faute avec court-circuit et circuit ouvert

D'autres modèles de faute utilisent des court-circuits et des circuit ouverts. Dans certains cas ils sont équivalents aux collage à 0 et à 1 définis précédemment et dans d'autres cas ils modifient la fonction logique. Dans la solution proposée, un court-circuit entre les nœuds $\overline{C_{a0b0}}$, $\overline{C_{a1b0}}$ et $\overline{C_{a0b1}}$ ne modifie pas la fonction logique. Seul un court-circuit entre le nœud $\overline{C_{a1b1}}$ et un des nœuds $\overline{C_{a0b0}}$, $\overline{C_{a1b0}}$ ou $\overline{C_{a0b1}}$ modifie la fonction (Figure 9.26). Un conflit apparaît sur les deux sorties $c1$ et $c0$ qui sont simultanément conduites à la masse et à la tension d'alimentation. La valeur de sortie a une forte probabilité d'entraîner la métastabilité.

9.8.3 Vecteurs de test

Un collage à 0 sur les nœuds $\overline{C_{a0b0}}$, $\overline{C_{a1b0}}$, $\overline{C_{a0b1}}$ et $\overline{C_{a1b1}}$ supprime l'état non valide et les sorties $c1$ et $c0$ transmettent l'erreur pendant l'état non valide. La porte en aval prend en compte cette donnée valide mais elle attend que l'autre donnée d'entrée soit valide. La porte en aval bloque l'erreur. Pour les commandes $\overline{C_{a0b0}}$, $\overline{C_{a1b0}}$ et $\overline{C_{a0b1}}$, la détection de l'erreur ne peut se faire que si le vecteur d'entrée de la porte est 1110, c'est à dire $\overline{C_{a1b1}} = 0$. La sortie est alors 01 qui entre en conflit avec une faute SA0 dont la sortie serait 10. Un court-circuit apparaît entraînant avec une forte probabilité la métastabilité qui a la particularité de se transmettre très

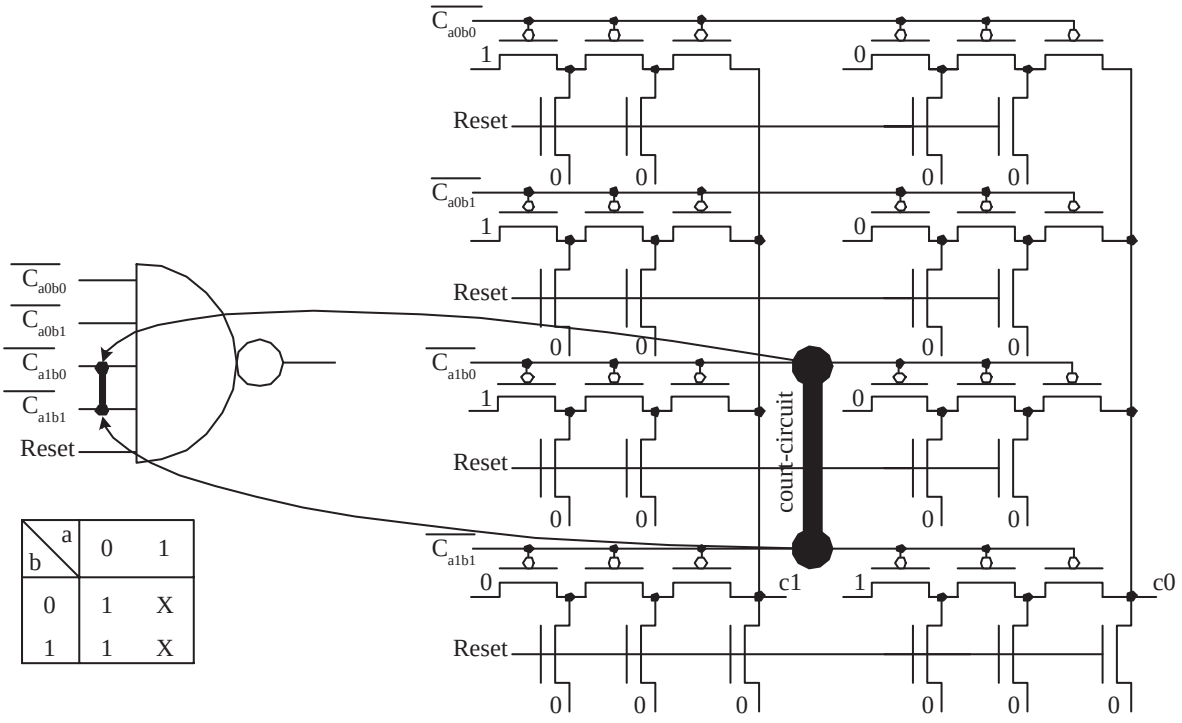


FIG. 9.26 – Modèle avec court-circuit

rapidement à l'ensemble du circuit. Pour la commande $\overline{C_{a1b1}}$, la détection de l'erreur ne peut se faire que si le vecteur d'entrée de la porte est 0111 ou 1011 ou 1101, c'est à dire $\overline{C_{a0b0}} = 0$ ou $\overline{C_{a1b0}} = 1 = 0$ et $\overline{C_{a0b1}} = 0$. La sortie est alors 10 qui entre en conflit avec une faute SA0 dont la sortie serait 01. De nouveau un court-circuit apparaît entraînant une forte probabilité de métastabilité. Cette faute est observable.

Un collage à 1 sur les nœuds $\overline{C_{a0b0}}$, $\overline{C_{a1b0}}$, $\overline{C_{a0b1}}$ et $\overline{C_{a1b1}}$ supprime l'état valide et les sorties $c1$ et $c0$ transmettent l'erreur. La porte en aval attend que cette donnée devienne valide alors que l'autre donnée d'entrée est valide. La porte en aval bloque à nouveau le processus de calcul. La faute est observable parce que le résultat de sortie correspond à la valeur d'une donnée non valide.

Un collage à 0 sur le nœud *Reset* ne permet plus de mettre les deux rails de sortie $c1$ et $c0$ à 0 dans l'état non valide. La donnée reste valide avec la valeur logique précédente. Les sorties $c1$ et $c0$ transmettent l'erreur pendant l'état non valide mais cette donnée valide est bloquée par la porte en aval qui a sa seconde entrée non valide. Lorsque la donnée devient à nouveau valide, les sorties $c1$ et $c0$ transmettent la nouvelle donnée valide. L'erreur peut ne pas être détectée si le protocole de donnée n'inclue pas une détection d'absence d'état non valide entre deux données valides. La faute est potentiellement non observable. Elle est d'autant plus gênante qu'elle remet en cause la résistance intrinsèque à la DPA, la suppression de l'effet mémoire n'étant plus garantie.

Un collage à 1 sur le nœud *Reset* laisse en permanence les deux rails de sortie $c1$ et $c0$ à 0 dans l'état non valide y compris lorsque des données valides sont en entrée. Les sorties $c1$ et $c0$ transmettent l'erreur pendant l'état valide ce qui bloque la porte en aval. L'erreur est observable et correspond à une donnée non valide.

9.9 Dérive technologique

La solution proposée repose sur l'exploitation de la propriété d'isolation drain-source du transistor en état d'accumulation. Le but est de supprimer une dispersion instantanée des caractéristiques électriques de la porte quelles que soient les valeurs passées, présentes et futures des bits d'entrée et de sortie. Mais pour que ceci fonctionne, il faut que les transistors de la porte ne présentent pas, eux aussi, de dispersion instantanée de leurs caractéristiques électriques et qu'ils soient identiques.

Les caractéristiques électriques du transistor sont très influencées par la longueur de canal L et la largeur du transistor W. Autrement dit, si deux transistors ont une même longueur de canal L et une largeur W différente alors ils ont des caractéristiques électriques différentes. Si deux transistors ont une longueur de canal L différente et une largeur W identique, ils ont aussi des caractéristiques électriques différentes (Équation 9.1).

$$\begin{cases} L1 = L2 \\ W1 \neq W2 \end{cases} \begin{cases} L1 \neq L2 \\ W1 = W2 \end{cases} \Rightarrow \text{carac. élec. Transistor 1} \neq \text{carac. élec. Transistor 2} \tag{9.1}$$

Il existe aussi une dispersion due à la dérive technologique. Deux transistors ayant une même longueur de canal et une même largeur peuvent aussi avoir des caractéristiques électriques différentes sur un même circuit intégré (Équation 9.2).

$$\begin{cases} L1 = L2 \\ W1 = W2 \end{cases} \Rightarrow \text{carac. élec. Transistor 1} \neq \text{carac. élec. Transistor 2} \tag{9.2}$$

Cette difference est mesurée par un paramètre appelé "matching" qui est proportionnel à l'inverse de la surface de grille, c'est à dire proportionnel à l'inverse du produit de la longueur de canal L et de la largeur de transistor W (Équation 9.3). Plus le produit $W.L$ est grand plus le matching est petit et plus les transistors sont électriquement identiques. Ceci explique pourquoi en conception analogique les transistors utilisés sont larges pour garantir les propriétés analogiques du circuit avec un minimum de dispersion des propriétés électriques de la fonction.

$$Matching = \frac{1}{\text{surface de grille}} \tag{9.3}$$

L'objectif des technologues est de garantir le matching. Il s'agit de produire des transistors qui, avec une même longueur de canal et une même largeur, ont des caractéristiques électriques identiques. Cet objectif est de plus en plus difficile à atteindre avec des technologies de plus en plus fines. Un problème particulièrement difficile à contrôler est le dopage du substrat du wafer qui doit être rigoureusement uniforme. Hors il est très difficile de garantir un nombre identique d'impuretés pour tous les transistors sur un même circuit. La diminution du matching permet un résultat statistiquement correct. Mais l'utilisation de transistors de taille minimale dans une technologie fine augmente le risque de dispersion instantanée des caractéristiques électriques des transistors. Quelles que soient les contre-mesures proposées exigeant une parfaite maîtrise des caractéristiques électrique d'un circuit, il est nécessaire de maîtriser le matching. Ce n'est pas applicable qu'à cette solution.

Chapitre 10

Application de la solution

Sommaire

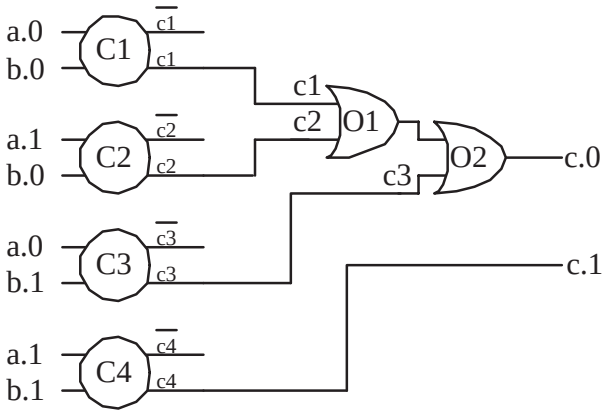
La solution proposée permet de garantir que les caractéristiques électriques instantanées sont identiques quelles que soient les transitions logiques sur les entrées, quel que soit le résultat logique en sortie et quelle que soit la fonction logique elle-même. Cependant, elle est tributaire des commandes contrôlant les séries d'interrupteurs qui ne doivent pas introduire un dispersion instantanée des caractéristiques électriques. La génération des commandes répondant à ce critère est explicitée dans les applications asynchrone et synchrone où des portes de chaque type sont construites avec les entrées a et b codées en double rail.

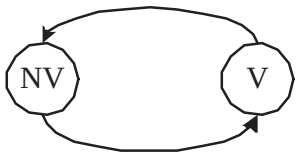
10.1 Application asynchrone

La mise en application de la solution proposée en technologie asynchrone est réalisée à partir de la technologie asynchrone QDI (Quasi Delay Insensitive [21] [7] [9] [8]) utilisant un codage de donnée double rail et un protocole 4-phase. Cette technologie asynchrone présente donc l'avantage d'intégrer intrinsèquement deux conditions nécessaires afin de contrer la DPA (Paragraphes 9.4 et 9.5). Cependant, elle ne la contre pas parce que les sources microélectroniques de la DPA ne sont pas prises en compte (Chapitre 8 paragraphe 8.4 [36] [10]).

10.1.1 Conception de la porte *And*2 asynchrone

La porte *And*2 classique en technologie asynchrone QDI est construite à l'aide de 4 portes C-Muller et d'une partie logique combinatoire. Les C-muller permettent la synchronisation des données valides. La logique combinatoire réalise un *OU* logique entre trois C-Muller générant le rail $c.0$. Le rail $c.1$ est la sortie du quatrième C-Muller dont les entrées sont $a.1$ et $b.1$. La sortie vaut 1 si a et b sont égaux à 1 lorsque les rails $a.1$ et $b.1$ sont à 1. Dans les autres cas, a ou b vaut 0 si l'un des rails $a.0$ ou $b.0$ est égal à 1. La sortie vaut 0, c'est à dire $c.0$ vaut 1 (Figure 10.1).





a		b		c	
a1	a0	b1	b0	c1	c0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0

FIG. 10.1 – Circuiterie de la porte *And*2 en technologie asynchrone QDI

Les portes C-Muller sont construites avec des réseaux série de transistors (Figure 10.2-a). Les transistors des réseaux série sont sensibles à l'effet de substrat et à l'effet mémoire générant la DPA. Afin de supprimer les réseaux comme source de la DPA, il est nécessaire de réaliser une extension de circuiterie en dupliquant les réseaux série et en permutant les entrées sur les grilles par rapport aux réseaux initiaux (Figure 10.2-b). Une autre architecture de la porte C-Muller avec la même propriété est disponible dans l'article [19] mais avec un nombre de transistors

supérieur.

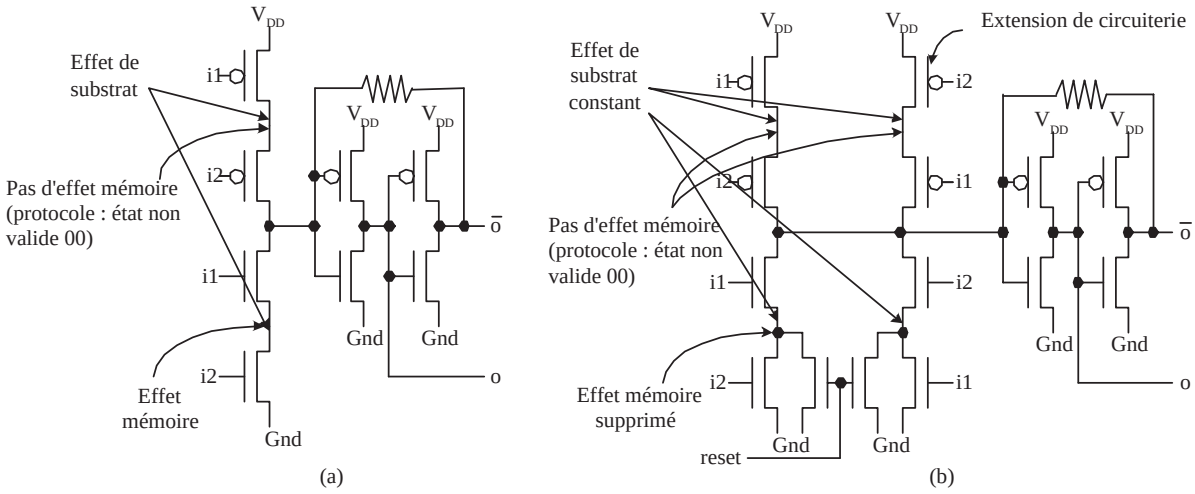


FIG. 10.2 – Circuiterie de la porte C-Muller avec DPA (a) et sans DPA (b)

La solution a pour entrées 4 commandes contrôlant les séries de demi-interrupteurs à base de transistors PMOS (Figure 9.20). La commande $\overline{C_{a0b0}}$ prend la valeur 0 quand les données valides sont toutes deux égales à 0 lorsque $a.0$ et $b.0$ valent 1. Dans ce cas, le nœud $c0$ prend la valeur 1 en étant conduit par la série d'interrupteurs contrôlés par la commande $\overline{C_{a0b0}}$. Cette commande est justement générée par la porte C-Muller 1 (Figure 10.1) sans DPA avec la circuiterie de la figure 10.2-b. Il est donc possible de générer toutes les commandes sans DPA avec des caractéristiques électriques instantanées identiques en utilisant pour chacune d'elle la circuiterie de la figure 10.2-b (Figure 10.3).

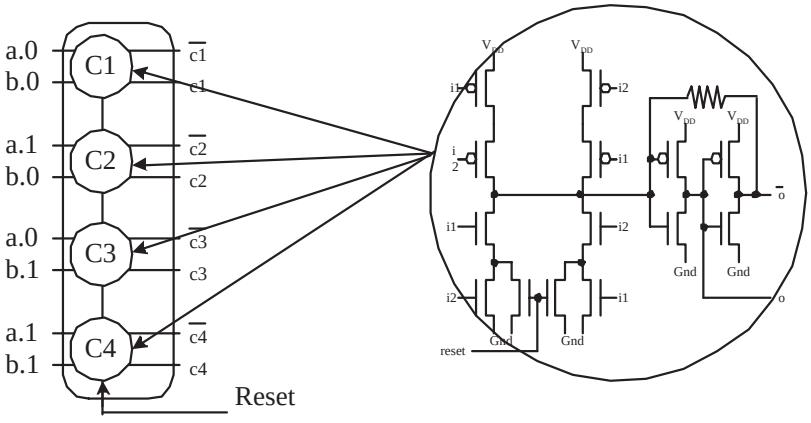


FIG. 10.3 – Conception des commandes de la porte $And2$ sans DPA

La logique combinatoire de la porte $And2$ asynchrone classique (Portes $Or2$ et $Nor2$ de la figure 10.1) doit être remplacée par la solution garantissant les propriétés électriques recherchées pour la fonction de la figure 9.20 (Figure 10.4).

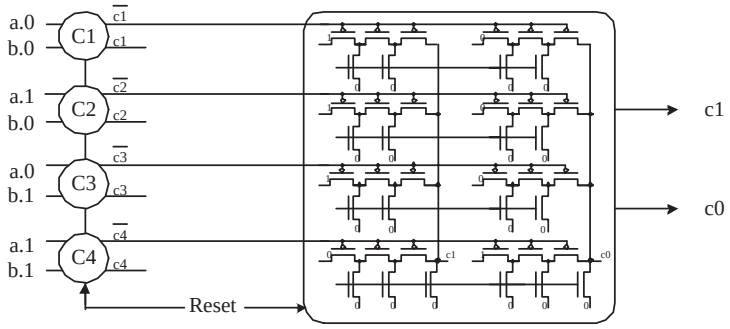


FIG. 10.4 – Conception de la logique de sortie de la porte *And*2 sans DPA

Le signal *Reset* est utilisé dans l'état non valide pour décharger les nœuds d'interconnexion afin de supprimer l'effet mémoire. Dans l'état non valide, tous les rails des données ont pour valeur 0 lorsque toutes les commandes C_{a0b0}, C_{a1b0}, C_{a0b1} et C_{a1b1} ont pour valeur 0 (Équation 10.1).

$$Reset \quad = \quad \overline{C_{a0b0}}.\overline{C_{a1b0}}.\overline{C_{a0b1}}.\overline{C_{a1b1}} = \overline{(C_{a1b0} + C_{a0b1}) + (C_{a0b1} + C_{a1b1})} \tag{10.1}$$

Le signal de Reset peut être généré à partir des commandes C_{a0b0}, C_{a1b0}, C_{a0b1} et C_{a1b1} en utilisant deux portes $Or2$ et une porte $Nor2$. Elles sont construites avec des réseaux série de transistors sensibles à l'effet de substrat et l'effet mémoire qui permettent la DPA. Afin de supprimer ces sources de la DPA, il est nécessaire de réaliser une extension de circuiterie en dupliquant les réseaux série et en permutant les entrées sur les grilles par rapport aux réseaux initiaux (Figure 10.5).

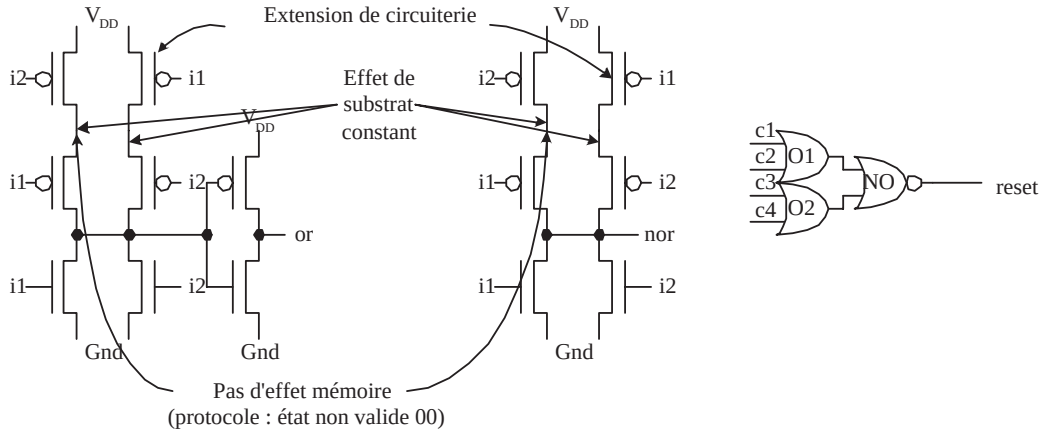


FIG. 10.5 – Conception du signal de *Reset* sans DPA et circuiterie des portes $Or2$ et $Nor2$

En posant $INT0$ l'ensemble des demi-interrupteurs générant la sortie $c.0$ et $INT1$ l'ensembles des demi-interrupteurs générant la sortie $c.1$, le fonctionnement global de la porte *And*2 avec le passage de l'état non valide à l'un des quatre états valides puis le retour à l'état non valide est

illustré par la figure 10.6.

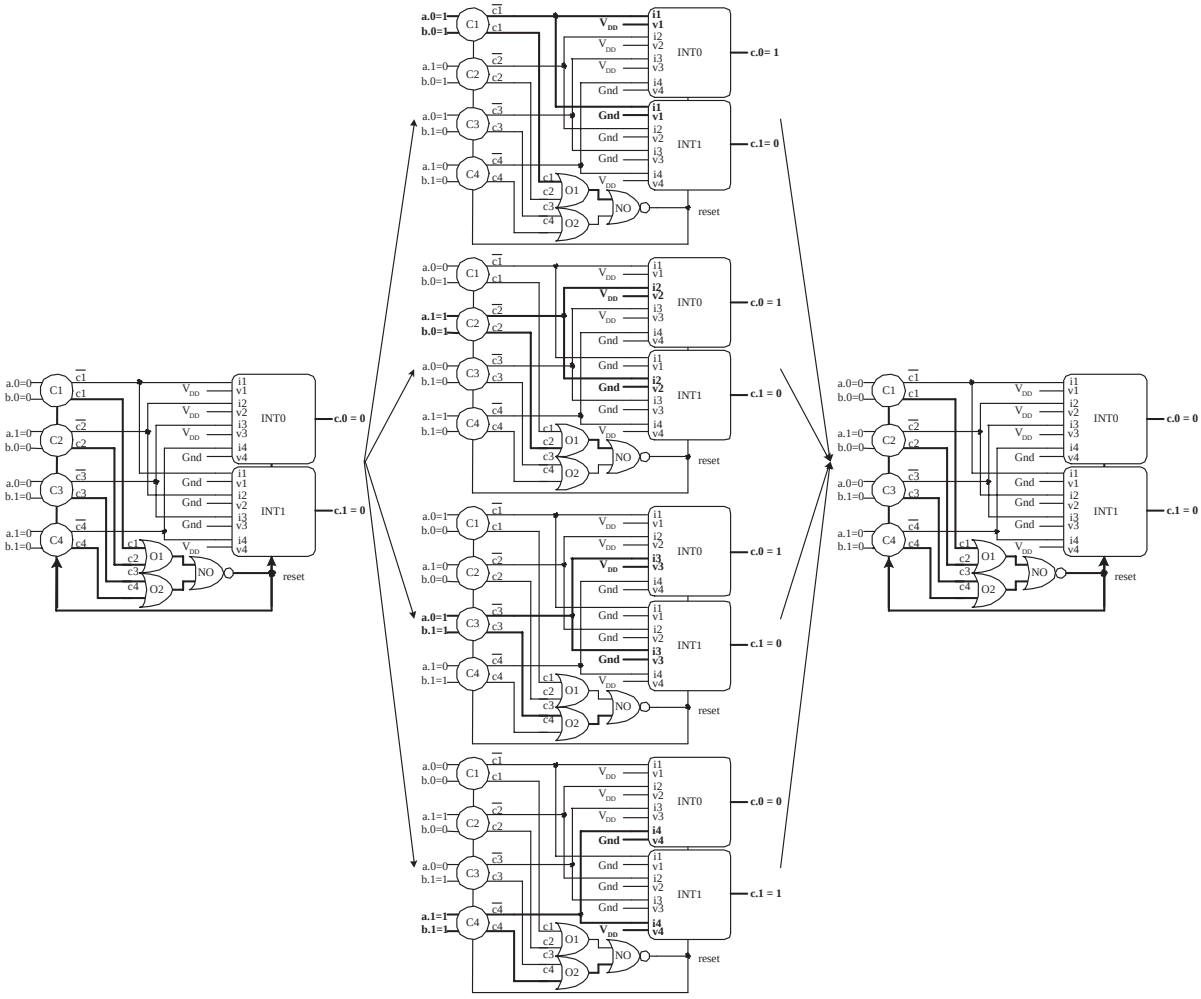


FIG. 10.6 – Fonctionnement global de la porte *And*2 asynchrone sans DPA

10.1.2 Conception d'autres portes logiques asynchrones

La solution proposée permet de concevoir toutes les fonctions logiques avec deux entrées tout en garantissant un comportement électrique instantané identique. La différence entre deux fonctions logiques ne repose que sur les tensions appliquées aux sources en entrées des séries de demi-interrupteurs générant les sorties $c1$ et $c0$ (Figure 10.7).

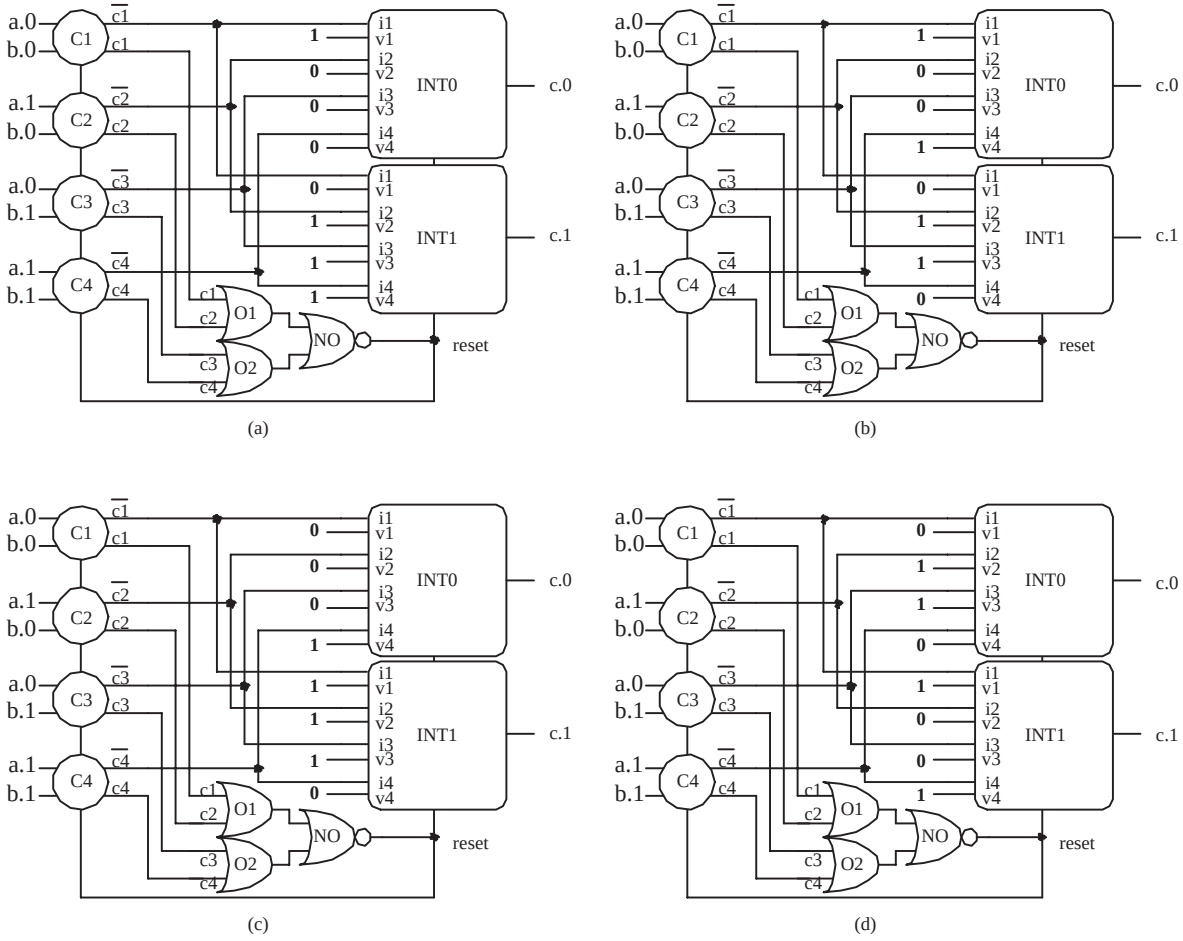


FIG. 10.7 – Portes asynchrones sans DPA $Or2$ (a), $Xor2$ (b), $Nand2$ (c) et $Xnor2$ (d)

10.1.3 Résultats de simulations électriques

Les 4 vecteurs de tests pour le porte $Nand2$ de la figure 10.8 permettent de prendre en compte toutes les transitions possibles y compris la précédence des signaux pour l'effet mémoire. Les résultats croisés de simulation donnent une différence de comportement électrique instantané en femto ampère. Les grandeurs en elles-mêmes n'ont pas de sens puisque les limites du simulateur ont été dépassées et dépendent des méthodes d'intégration. Le seul résultat pertinent est l'absence de dispersion des caractéristiques électriques instantanées, objectif recherché et atteint pour contrer le DPA (Figure 10.8).

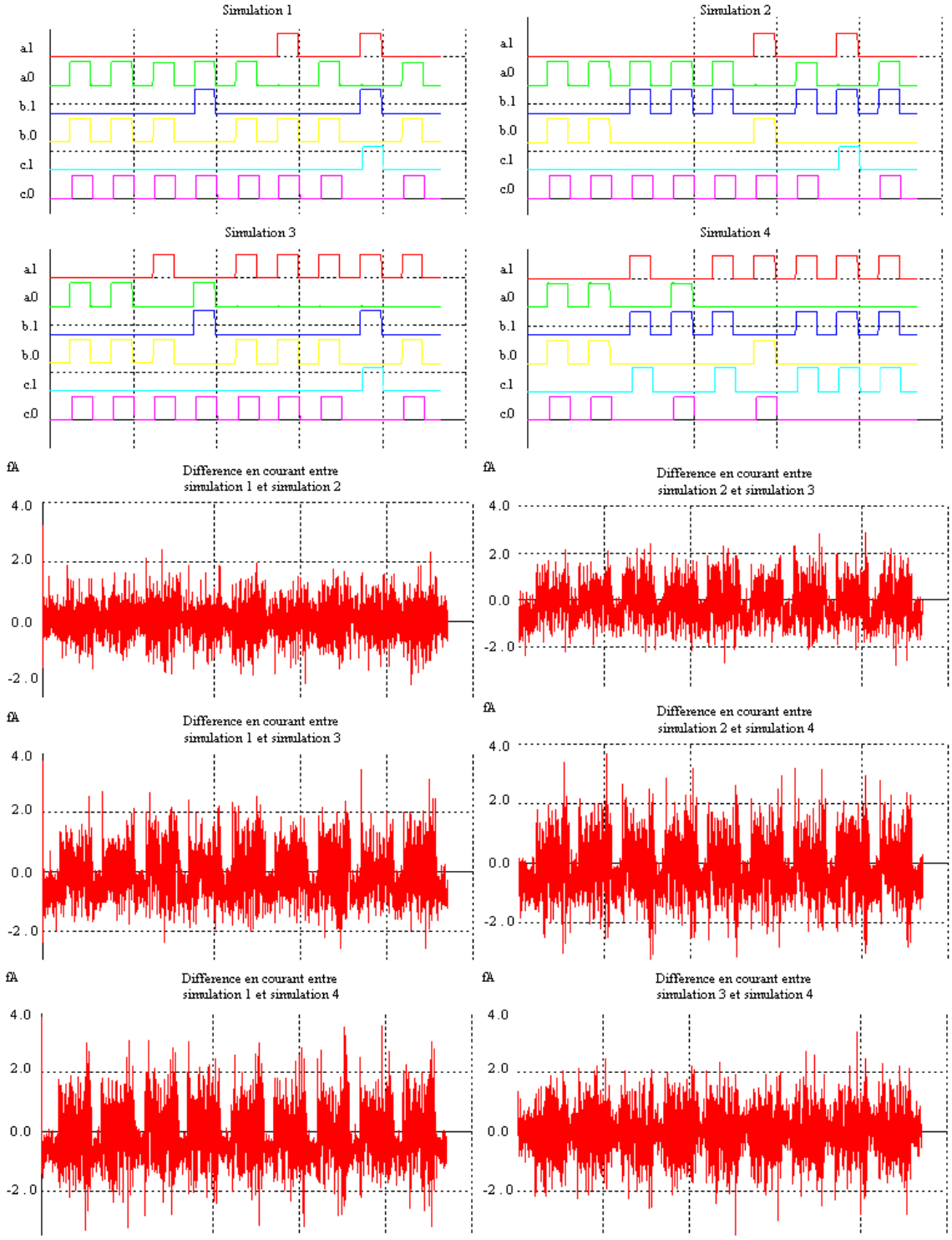


FIG. 10.8 – Absence de dispersion des caractéristiques électriques instantanées de la porte $Nand2$

L'avantage de cette solution est la possibilité d'utiliser indifféremment toute porte logique pour

une fonction. En effet, que ce soit une porte $And2$, $Or2$, $Xor2$, etc, la DPA devient impossible à mettre en œuvre (Figure 10.9). Les résultats croisés de simulation donnent à nouveau une différence de comportement électrique instantané en femto ampère montrant l'absence de dispersion des caractéristiques électriques instantanées.

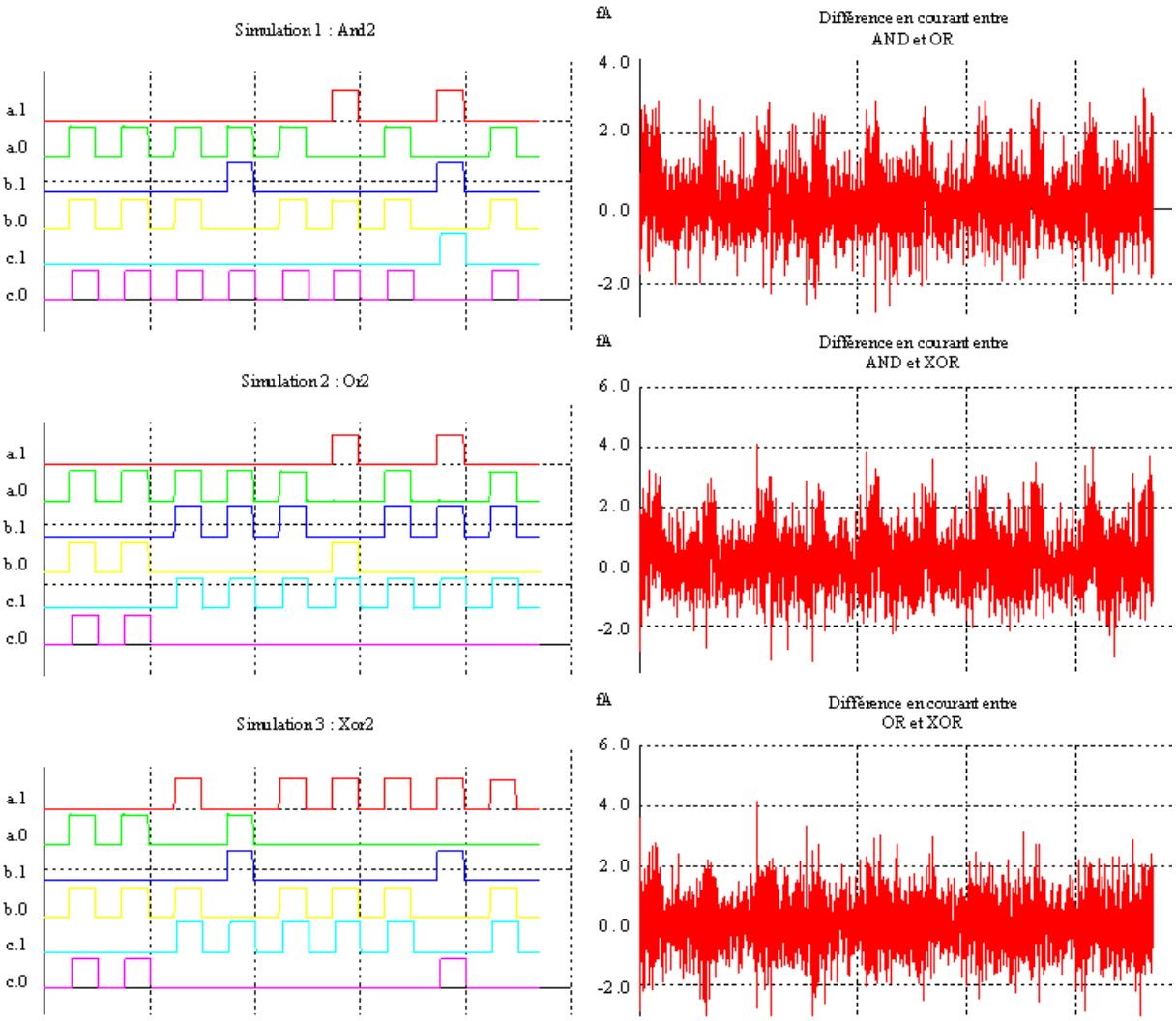


FIG. 10.9 – Absence de dispersion des caractéristiques électriques instantanées entre différentes portes asynchrones

10.1.4 Conception de fonctions logiques asynchrones

Les portes ainsi construites ont toutes le même comportement électrique instantané. Il est possible de construire des étages de logiques contrant de façon intrinsèque la DPA (Figure 10.10). Les étages de logiques $E1$, $E2$ et $E3$ sont résistants à la DPA. Cependant, cette solution peut facilement être déséquilibrée par des outils de conception par exemple lors de la synthèse ou du placement-routage (Chapitre 7).

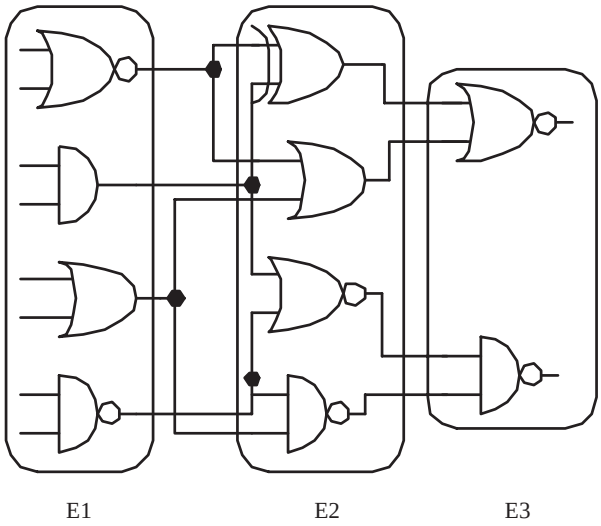


FIG. 10.10 – Possibilité de combiner différentes portes logiques dans différents étages de fonctions asynchrones résistantes à la DPA

10.1.5 Testabilité

L'étude de la testabilité de la solution proposée au chapitre 9 a montré qu'un collage à 0 sur le nœud *Reset* peut ne pas être détecté. Cela a pour conséquence de remettre en cause la résistance intrinsèque à la DPA parce que la suppression de l'effet mémoire n'est plus garantie. Ceci est particulièrement vrai si le protocole de donnée n'inclue pas une détection d'absence d'état non valide entre deux états valides. Le protocole 4-phase fonctionne avec un principe de requête acquittement. Une porte asynchrone remet sa sortie valide dans l'état non valide après avoir reçu l'acquittement de la porte en aval. Tant que la sortie n'est pas dans l'état non valide, la porte en aval est bloquée avec activation de son acquittement. Ce protocole est donc bloqué si l'état non valide est impossible. La faute devient observable en sortie et correspond à une valeur de donnée non valide. La suppression de l'effet mémoire peut donc être garantie par le test.

10.2 Application synchronisée

10.2.1 Conception de la porte $Nand2$ synchrone

La nécessité d'avoir un nouveau codage de donnée en double rail introduit les entrées complémentées. Une porte synchrone à deux entrées doit être maintenant considérée avec quatre entrées a, b, $\overline{a}$ et $\overline{b}$. Une porte $Nand2$ CMOS classiquement utilisée en technologie synchrone n'est pas conçue pour prendre en compte ces deux nouvelles entrées. Les entrées peuvent être combinées de quatre manières, (a,b), (a,$\overline{b}$), ($\overline{a}$,b) et ($\overline{a}$,$\overline{b}$). Elles seront notées $a.1$ pour a, $a.0$ pour $\overline{a}$, $b.1$ pour b et $b.0$ pour $\overline{b}$. Les combinaisons (a,$\overline{a}$) et (b,$\overline{b}$) respectivement notées ($a.1$,$a.0$) et ($b.1$,$b.0$) sont pour l'instant écartées parce que leur résultat est toujours égal à 1. Une extension de circuiterie s'impose et consiste à dupliquer par quatre la porte $Nand2$ pour prendre en compte ces combinaisons. Leurs quatre sorties logiques s'apparentent aux commandes des séries de demi-interrupteurs qui générent les sorties c et $\overline{c}$, notées $c.1$ et $c.0$, de la fonction $Nand2$ synchrone en codage double rail. Cette circuiterie est équivalente en taille à la porte asynchrone utilisant quatre portes C-Muller (Figure 10.7-c).

La fonction logique $Nand2$ doit être générée en utilisant la solution proposée afin de garantir l'absence de dispersion instantanée des caractéristiques électriques. Cette partie est commune aux deux portes asynchrone et synchrone et est équivalente en taille (Figure 10.7-c).

La nécessité d'avoir un nouveau protocole de donnée afin de supprimer l'effet mémoire impose la génération d'un signal *Reset* à partir de combinaisons (a,$\overline{a}$) et (b,$\overline{b}$) notées ($a.1$,$a.0$) et ($b.1$,$b.0$). L'égalité entre une entrée et son complément indique l'état non valide et peut être réalisée à partir de portes $Nand2$ et $And2$ synchrones. Tout comme la porte asynchrone qui utilise deux portes $Or2$ et une porte $Nor2$, le *Reset* de la porte synchrone peut être réalisé avec deux portes $Nand2$ et une porte $And2$. Ceci est à nouveau équivalent en taille à la porte asynchrone de la figure 10.7-c.

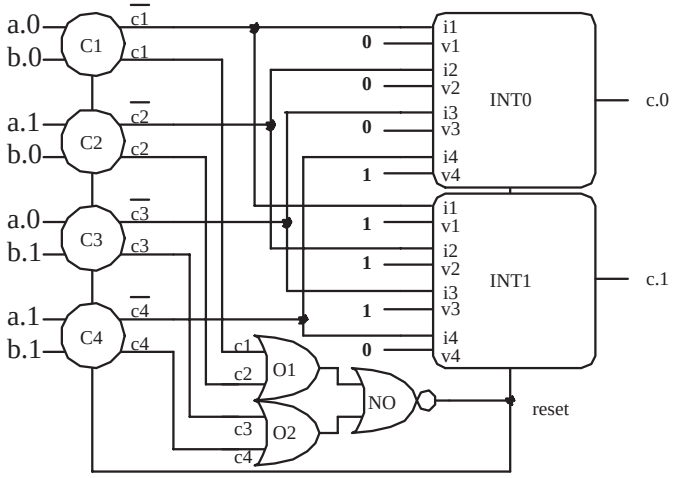
La porte $Nand2$ est conçue à partir de réseaux série et parallèle de transistors. Elle est donc sensible à la DPA. Une extension de circuiterie des réseaux série est nécessaire tout comme cela est le cas pour les portes C-Muller (Figure 10.2-b).

En conclusion, la conception d'une porte synchrone résistante intrinsèquement à la DPA à partir de la solution proposée est équivalente en taille à la porte asynchrone. La porte asynchrone peut donc être réutilisée en technologie synchrone, il s'agit d'asynchrone synchronisé (Figure 10.7).

La différence entre les deux portes n'est pas architecturale puisqu'elle est la même. Elle réside seulement dans la nécessité de caractériser les temps de traversée de la cellule afin de pouvoir les fournir aux outils de conception qui doivent prendre en compte des périodes d'horloge. En asynchrone le circuit intégré fonctionne à la vitesse de la technologie alors qu'en synchrone le circuit intégré fonctionne à la fréquence d'horloge imposée (Figure 10.11).

10.2.2 Conception d'autres portes logiques synchrones

La solution proposée permet de concevoir toutes les fonctions logiques synchrones avec deux entrées tout en garantissant un comportement électrique instantané identique. La différence entre



Temps de montée	Temps de descente
a.0 -> c.0	a.0 -> c.0
a.0 -> c.1	a.0 -> c.1
b.0 -> c.0	b.0 -> c.0
b.0 -> c.1	b.0 -> c.1
a.1 -> c.0	a.1 -> c.0
a.1 -> c.1	a.1 -> c.1
b.1 -> c.0	b.1 -> c.0
b.1 -> c.1	b.1 -> c.1

FIG. 10.11 – Conception de la porte Nand2 synchrone et caractéristiques temporelles

deux fonctions logiques ne repose toujours que sur les tensions appliquées aux sources en entrées des séries de demi-interrupteurs générant les sorties $c1$ et $c0$. Seule la caractérisation temporelle des portes est en plus nécessaire (Figure 10.7).

10.2.3 Résultats de simulations électriques

Les résultats de simulations électriques des portes asynchrones (Figure 10.8) sont applicables aux portes synchrones.

10.2.4 Conception de fonctions logiques synchrones

Les portes synchrones ont aussi toutes le même comportement électrique instantané. Il est possible de construire des étages de logiques contrant de façon intrinsèque la DPA comme présenté dans la figure 10.10. Á nouveau, cette solution peut facilement être déséquilibrée par des outils de conception (Chapitre 7). La seule différence réside dans l'utilisation d'éléments de mémorisation cadencés par une horloge et la gestion des états non valide et valide.

Le concepteur peut utiliser des éléments de mémorisation à deux étages. L'état initial est l'état non valide après reset avec des données non valides. Un jeton blanc symbolise une donnée non valide ayant ses deux rails à 0 (Figure 10.12-a).

Lorsque une donnée valide est présentée en entrée, celle-ci est mémorisée dans le premier étage sur le front d'horloge suivant. Un jeton noir symbolise une donnée valide ayant un seul de ses deux rails à 1 (Figure 10.12-b).

Elle est transmise au second étage sur le front d'horloge suivant alors qu'une donnée non valide doit être présentée en entrée (Figure 10.12-c).

Le résultat de la fonction logique est mémorisée dans le premier étage de sortie sur le front d'horloge suivant alors que le premier étage de l'entrée est libre d'accepter une nouvelle donnée valide (Figure 10.12-d).

La sortie de la fonction logique est valide sur le second étage de la sortie pendant que la fonction logique est libre d'évaluer un nouveau résultat logique à partir de la nouvelle donnée valide (Figure 10.12-e).

Il s'agit d'un mécanisme pipeliné transférant des jetons de données valides et non valides. Seules les entrées et les sorties doivent être contrôlées par une machine d'état gérant le protocole synchrone à deux états.

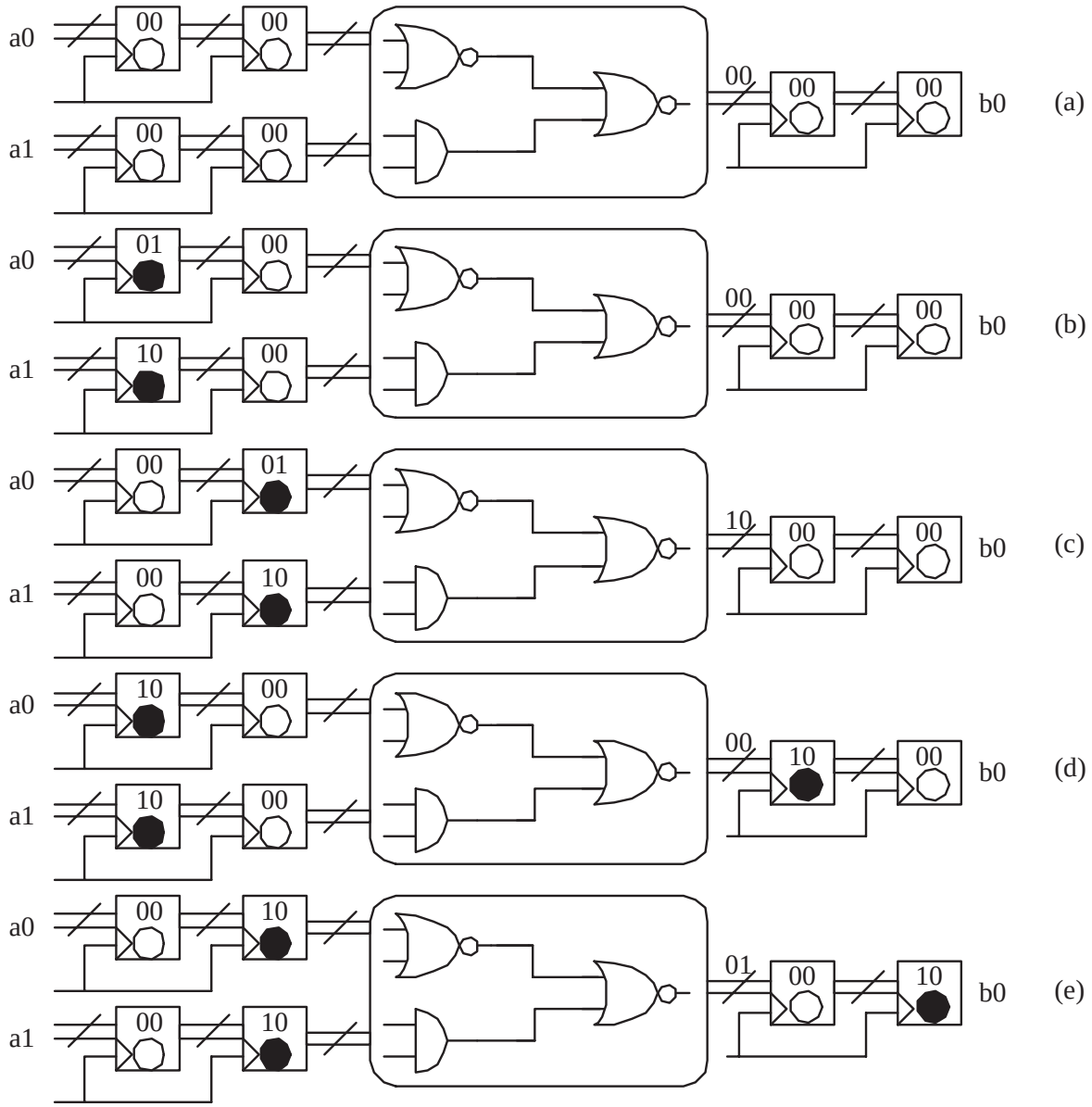


FIG. 10.12 – Système synchrone avec états non valide et valide

10.2.5 Testabilité

La testabilité de la version asynchrone est applicable pour la version synchrone.

10.3 Application cryptographique

L'objectif de la solution est de supprimer toute corrélation entre le comportement électrique instantané d'un circuit intégré cryptographique et les données manipulées comme les clés secrètes et les messages. A partir des portes présentées, il est tout à fait possible de concevoir une fonction de substitution de bits de message insensible à la DPA suivie d'une fonction d'addition de bits de clé insensible à la DPA, l'ensemble étant toujours insensible à la DPA. La figure 10.13 représente un étage élémentaire d'un algorithme cryptographique à clé secrète du type de l'AES.

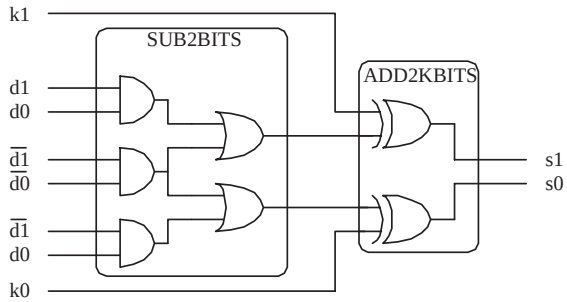


FIG. 10.13 – Étage élémentaire d'un algorithme de cryptographie à clé secrète

L'utilisation de portes conçues à partir de la solution proposée permet de concevoir cet étage cryptographique élementaire avec un comportement électrique identique quels que soient les bits de données et quels que soient les bits de clés manipulés. La différence de comportement électrique instantané est à nouveau exprimée en femto ampère qui n'a pas de sens en tant que valeur mais indique l'infaisabilité de la DPA avec cette technique de circuiterie (Figure 10.14).

Cette illustration n'est pas limitative. Toute architecture utilisant cette solution est insensible à la DPA. La DPA peut être renommée plus exactement comme une attaque temporelle des caractéristiques électriques du circuit. Elle est en fait une sous-classe de l'attaque temporelle d'un circuit intégré. Le nombre variable de cycles d'horloge par bit de la clé est aussi une caractéristique électrique du circuit.

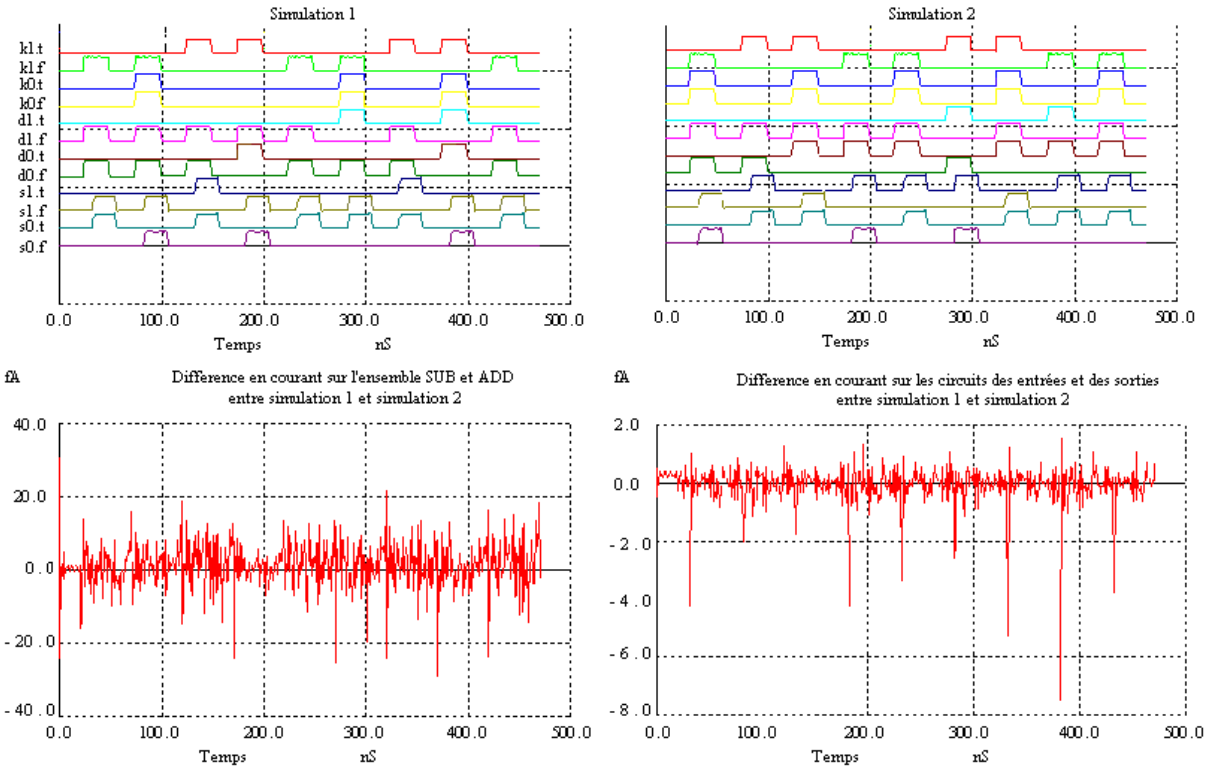


FIG. 10.14 – Infaisabilité de la DPA avec la solution proposée sur l'étage élémentaire d'un algorithme de cryptographie à clé secrète

Conclusion

Dans cette thèse, j'ai montré que la source physique de la DPA est la dispersion instantanée des caractéristiques électriques d'un circuit lors de différents évènements microélectroniques. Un processus DPA ne fait que la mettre en évidence par la différence temporelle de grandeurs physiques mesurables.

J'ai proposé une solution de portes logiques intrinsèquement résistantes à l'attaque DPA. Toutes les fonctions logiques à deux entrées sont réalisables avec cette solution qui garantit un comportement électrique instantané identique. Ceci permet de les combiner pour créer une fonction logique quelconque et rend donc théoriquement possible la conception d'un circuit intégré résistant à la DPA.

Cette solution pourra naturellement s'inscrire dans un flot de conception classique. L'étape suivante de la thèse est le dessin des portes elles-mêmes sur une technologie précise. La régularité de la solution permettra de capitaliser rapidement le placement routage de chaque cellule et d'envisager la conception d'une librairie standard. Cette librairie de cellules résistantes à la DPA sera utilisable pour des circuits synchrones ou asynchrones. Cependant, les outils de conception seront eux-aussi à adapter afin d'éviter l'introduction de biais permettant à nouveau la faisabilité physique de la DPA.

La solution proposée a fait l'objet de l'obtention d'un brevet au nom de l'État français représenté par le Secrétariat général de la défense nationale [15].

Cinquième partie

Annexe

Annexe A

Théorie du transistor MOS

Sommaire

A.1 Le transistor MOS logique

Le silicium est la matière première pour un grand nombre de circuits intégrés. La structure MOS (Metal-Oxide-Silicon) est fabriquée à partir d'une superposition de différentes couches de matériaux conducteurs, semi-conducteurs et isolants. Elles sont créées selon un processus de fabrication précis dont les principales phases sont l'oxydation, la diffusion, l'implantation et le dépôt. L'ensemble est réalisé sur un wafer qui est un disque fin et plan de silicium dopé qui est produit avec du silicium en fusion mélangé avec des impuretés et à partir d'une maille cristalline parfaite de silicium. Selon le type de dopage du silicium, il est possible d'obtenir deux sortes de transistor MOS, NMOS et PMOS. Le transistor NMOS est fabriqué à partir de silicium dopé négativement, c'est à dire avec des impuretés dont le nombre d'électrons de valence est en nombre supérieur au nombre d'électrons de valence de l'élément silicium. Ceci a pour effet de libérer un ou plusieurs électrons. Le transistor PMOS est quant à lui fabriqué à partir de silicium dopé positivement. Les impuretés ont un nombre d'électrons de valence de valeur inférieure au nombre d'électrons de valence de l'élément silicium. Ceci a pour effet de libérer un ou plusieurs trous, c'est à dire de libérer des absences d'électron. Les structures physiques des deux types de transistor sont présentées dans la figure A.1. Le transistor NMOS est réalisé sur du silicium dopé p appelé génériquement substrat. Deux zones distinctes de silicium dopé n sont implantées formant deux contacts appelés *drain* et *source*. Elles sont séparées par une zone de substrat sur laquelle est déposée une couche d'isolant d'oxyde de silicium SiO_2 qui est elle même recouverte par une couche conductrice de polysilicium appelée *grille* et formant un troisième contact. De façon similaire, le transistor PMOS est réalisé sur un substrat cette fois-ci dopé n. Le *drain* et la *source* sont formés par deux zones distinctes de silicium dopé p. La grille est fabriquée de la même façon que pour un transistor NMOS.

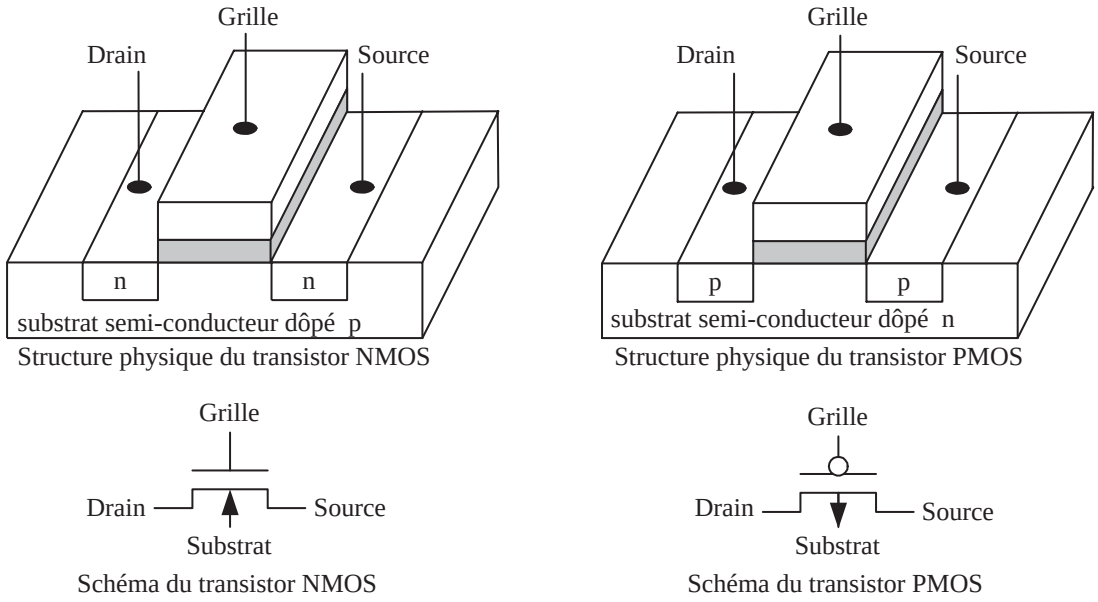


FIG. A.1 – Structure physique des transistors MOS et schémas associés

La structure MOS permet de contrôler le passage d'un courant entre le drain et la source par l'intermédiaire de la grille. Il est possible d'assimiler un transistor à sa plus simple expression qui

est un interrupteur dont la commande est la grille ouvrant et fermant le passage entre le drain et la source. La tension appliquée aux entrées du transistor peut varier du niveau de l'alimentation noté V_{DD} au niveau de la masse noté V_{SS}. Lorsqu'un signal est au niveau V_{DD} il est appelé du point de vue logique 1 ou 1 fort. Si le niveau du signal est proche mais inférieur à V_{DD}, il sera dit 1 faible. De même, lorsqu'un signal est au niveau V_{SS}, il est appelé 0 ou 0 fort. Si le niveau du signal est proche mais supérieur à V_{SS}, il est dit 0 faible. Cela sous-entend qu'il existe une plage entre 1 faible et 0 faible pour laquelle le signal n'a pas une valeur logique déterminable. Cette plage dépend de la technologie.

Le transistor NMOS (Figure A.2-a) est fermé (Figure A.2-b) dès lors qu'un 1 est appliqué à la grille. En effet, la différence de potentiel crée un champ électrique entre la grille et le substrat générant une force électrique attirant les électrons dans le substrat sous la grille. Il se forme un canal d'électrons entre le drain et la source qui, de ce fait, se trouvent connectés. Un courant peut alors passer. En revanche, le transistor NMOS est ouvert (Figure A.2-b) dès lors qu'un 0 est appliqué à la grille. Les électrons ne sont plus attirés, ce sont les trous qui reviennent dans le substrat du canal. Le drain et la source ne sont plus connectés et le courant ne peut plus passer. Le transistor NMOS conduit mal le 1 (Figure A.2-c). Le signal obtenu en sortie est toujours un 1 faible (Paragraphe A.4). Cependant, il conduit bien le 0 (Figure A.2-c).

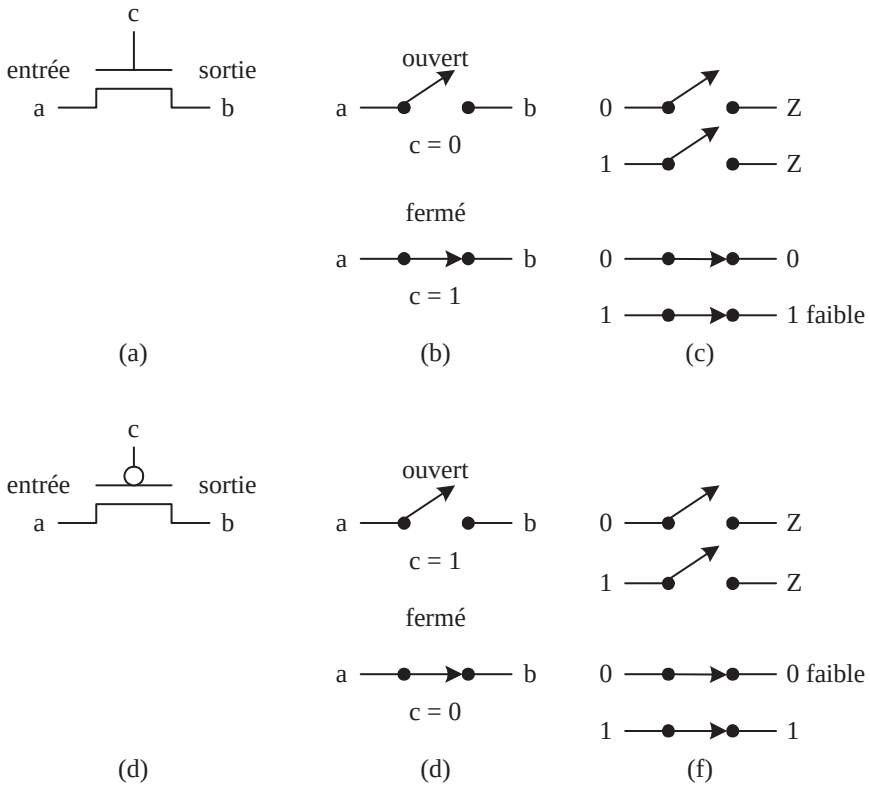


FIG. A.2 – Interrupteurs NMOS et PMOS

Le transistor PMOS (Figure A.2-d) est fermé (Figure A.2-e) dès lors qu'un 0 est appliqué à la grille. Un canal de trous se forme entre le drain et la source qui se trouvent connectés. Un courant peut alors passer. En revanche, le transistor PMOS est ouvert (Figure A.2-e) dès lors qu'un 1

est appliqué à la grille, les électrons revenant dans le substrat du canal. Le transistor PMOS conduit mal le 0 (Figure A.2-f). Le signal obtenu en sortie est toujours un 0 faible (Paragraphe A.4). Cependant, il conduit bien le 1 (Figure A.2-f).

Lorsque les transistors NMOS et PMOS sont ouverts, ils ne conduisent pas. Cela signifie qu'ils n'imposent aucune tension en sortie et ce quelle que soit la variation de tension appliquée en entrée. Ce niveau est dit haute impédance noté Z (Figures A.2-c et A.2-f).

A.2 La logique CMOS

A.2.1 L'inverseur

La table de vérité de l'inverseur est donnée par la tableau A.1. Lorsque l'entrée est 0, la sortie est 1. Un interrupteur PMOS connectant V_{DD} à la sortie et commandé par l'entrée reliée à la grille réalise cette fonction dite pull-up. Cela permet de conduire un 1 fort. Lorsque l'entrée est 1, la sortie est 0. Un interrupteur NMOS connectant V_{SS} à la sortie et commandé par l'entrée reliée à la grille réalise cette fonction pull-down. Elle permet de conduire un 0 fort.

entrée	sortie
0	1
1	0

TAB. A.1 – Table de vérité de l'inverseur

L'inverseur dont la table de résolution est donnée par le tableau A.2 est créé en connectant un transistor PMOS et un transistor NMOS. Lorsque l'entrée est 0, la fonction pull-up (transistor PMOS) conduit un 1 alors que la fonction pull-down (transistor NMOS) génère l'état Z. Comme l'état haute impédance n'impose aucune tension, c'est la tension imposée par le transistor PMOS qui est en sortie. De même, lorsque l'entrée est 1, la fonction pull-down conduit un 0 alors que la fonction pull-up génère l'état Z. La tension en sortie est celle imposée par le transistor NMOS. Si les deux transistors NMOS et PMOS sont ouverts simultanément, la sortie prendra donc l'état Z puisque aucune tension n'est imposée. Mais si les deux transistors sont fermés simultanément, l'alimentation V_{DD} et la masse V_{SS} sont reliées et un court-circuit est provoqué. Cet état peut survenir lorsque l'entrée change d'état de 0 vers 1 ou de 1 vers 0.

sortie NMOS	sortie PMOS	sortie INVERSEUR
0	Z	0
Z	1	1
Z	Z	Z
0	1	court-circuit

TAB. A.2 – Table de résolution de l'inverseur

A.2.2 Combinaison de transistors

Si deux transistors NMOSa et NMOSb sont connectés en série (Figure A.3-a), l'interrupteur ainsi produit est fermé lorsque le transistor NMOSa et le transistor NMOSb sont simultanément fermés. Ceci correspond à une fonction logique AND. Si deux transistors NMOSa et NMOSb sont connectés en parallèle (Figure A.3-b), l'interrupteur ainsi produit est fermé lorsque le transistor NMOSa est fermé ou lorsque le transistor NMOSb est fermé. Ceci correspond à une fonction logique OR. Il en est de même pour deux transistors PMOS connectés en série (Figure A.3-c) ou en parallèle (Figure A.3-d). A partir de ces résultats il est alors possible de construire des fonctions logiques en combinant des réseaux de transistors NMOS à des réseaux de transistors PMOS.

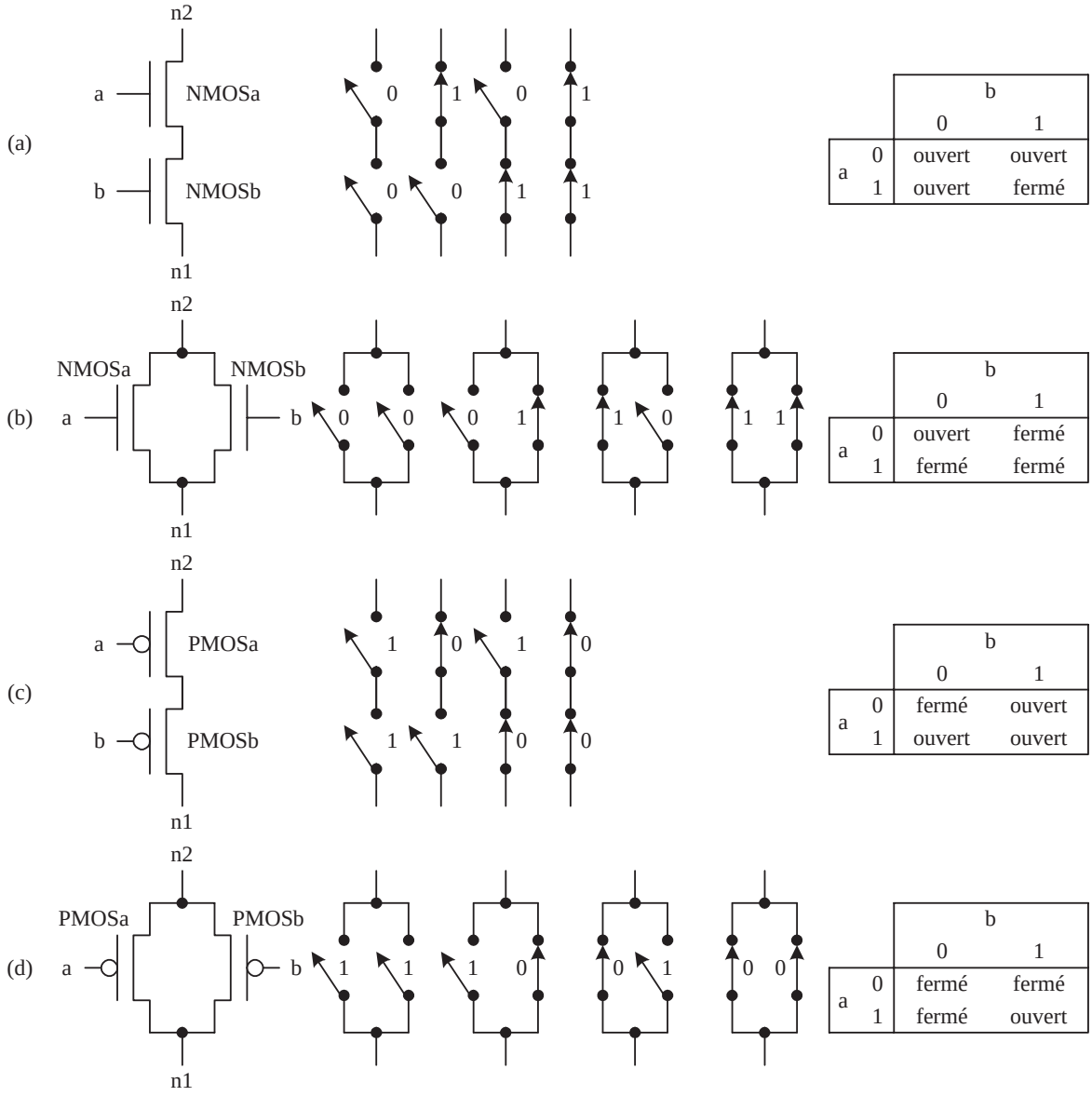


FIG. A.3 – Réseaux série et parallèle d'interrupteurs NMOS et PMOS

A.2.3 La porte NAND

La fonction NAND est décrite dans la figure A.4 avec son circuit (Figure A.4-a), sa table de vérité (Figure A.4-b) et son symbole (Figure A.4-c). Elle consiste à sortir 0 lorsque les deux entrées a et b sont à 1. C'est tout à fait ce que réalise un interrupteur construit avec deux transistors NMOS en série (Figure A.3-a) connectant le nœud $n2$ à la sortie et le nœud $n1$ à V_{SS}. Elle doit aussi sortir 1 lorsque l'entrée a ou l'entrée b est à 0. C'est ce que réalise un interrupteur construit avec deux transistors PMOS en parallèle (Figure A.3-d) connectant le nœud $n1$ à la sortie et le nœud $n2$ à V_{DD}.

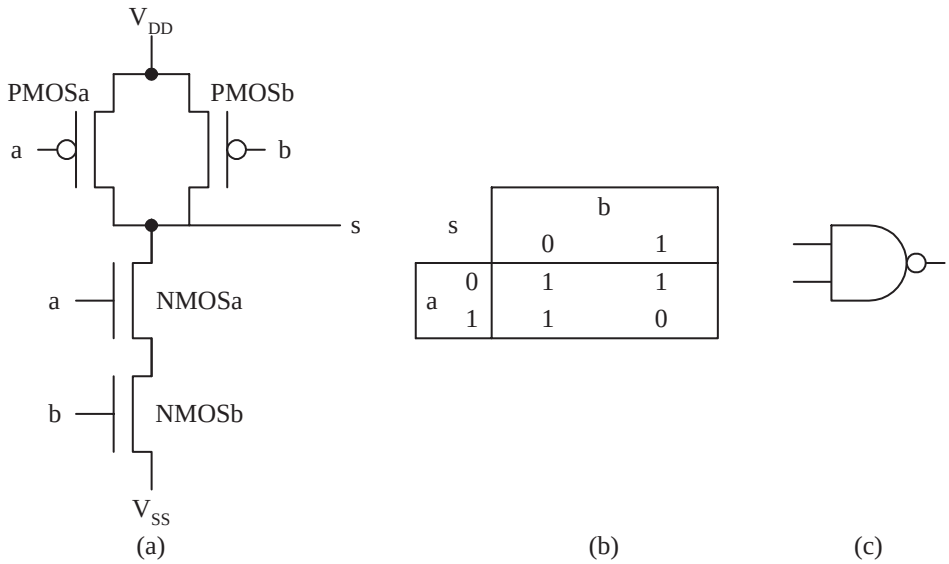


FIG. A.4 – Porte NAND CMOS

Deux transistors NMOS série connectés à V_{SS} forment une fonction pull-down dont la table de vérité est décrite par le tableau A.3-a. Les deux transistors PMOS parallèles connectés à V_{DD} forment une fonction pull-up dont la table de vérité est décrite par le tableau A.3-b. Lorsque les sorties des fonctions pull-down et pull-up sont connectées, la fonction résultante est la fonction $NAND$ dont la table de vérité (Tableau A.3-c) est la superposition des deux tables de vérités précédentes.

pull-down		
a	b	s
0	0	Z
0	1	Z
1	0	Z
1	1	0

(a)

pull-up		
a	b	s
0	0	1
0	1	1
1	0	1
1	1	Z

(b)

Nand			
a	b	c	
0	0	(Z-1)	1
0	1	(Z-1)	1
1	0	(Z-1)	1
1	1	(0-Z)	0

(c)

TAB. A.3 – Table de vérité des fonctions pull-down et pull-up de la porte NAND

A.2.4 La porte NOR

La fonction NOR est décrite avec son circuit (Figure A.5-a), sa table de vérité (Figure A.5-b) et son symbole (Figure A.5-c). Elle consiste à sortir 0 lorsque l'entrée a ou l'entrée b est à 1. Ceci est réalisé par un interrupteur construit avec deux transistors NMOS en parallèle (Figure A.3-b) connectant le nœud $n2$ à la sortie et le nœud $n1$ à V_{SS}. Elle doit aussi sortir 1 lorsque l'entrée a et l'entrée b sont à 0. C'est ce que réalise un interrupteur construit avec deux transistors PMOS en série (Figure A.3-c) connectant le nœud $n1$ à la sortie et le nœud $n2$ à V_{DD}.

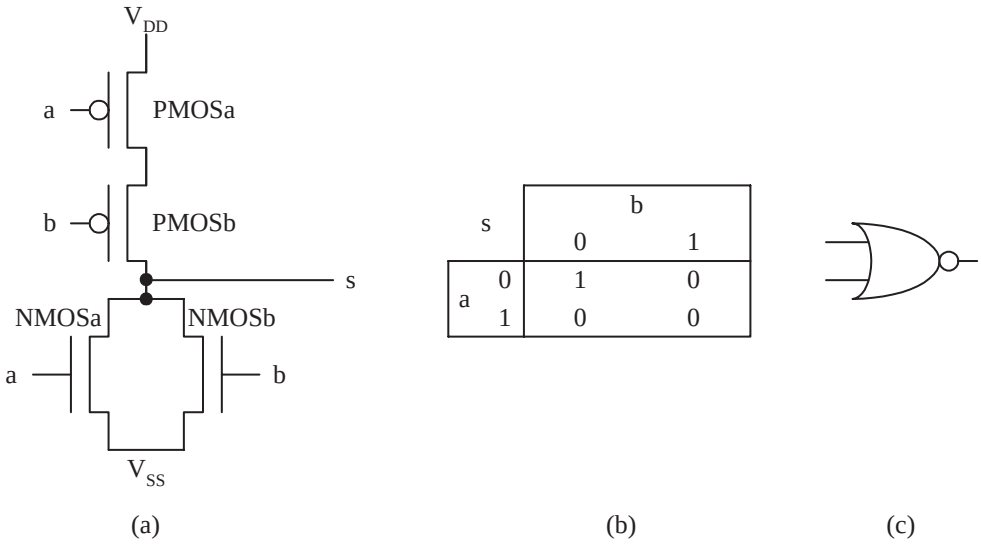


FIG. A.5 – Porte NAND CMOS

Deux transistors NMOS parallèles connectés à V_{SS} forment une fonction pull-down dont la table de vérité est décrite par le tableau A.4-a. Les deux transistors PMOS série connectés à V_{DD} forment une fonction pull-up dont la table de vérité est décrite par le tableau A.4-b. Lorsque les sorties des fonctions pull-down et pull-up sont connectées, la fonction résultante est la fonction NOR dont la table de vérité (A.4-c) est la superposition des deux tables de vérité précédentes.

pull-down		
a	b	s
0	0	Z
0	1	0
1	0	0
1	1	0

(a)

pull-up		
a	b	s
0	0	1
0	1	Z
1	0	Z
1	1	Z

(b)

Nor		
a	b	s
0	0	(Z-1) 1
0	1	(0-Z) 0
1	0	(0-Z) 0
1	1	(0-Z) 0

(c)

TAB. A.4 – Table de vérité des fonctions pull-down et pull-up de la porte NOR

A.2.5 Les portes complexes

Une porte complexe correspond à une équation booléenne combinant plusieurs fonctions NAND, NOR, etc. La fonction XOR en est un exemple (Équation A.1).

$$a \oplus b = \overline{a}.b + a\overline{b} \tag{A.1}$$

De telles portes nécessitent de combiner plusieurs réseaux en série et/ou parallèles de transistors NMOS pour réaliser la fonction pull-down associée. Il en est de même pour les transistors PMOS. Deux possibilités de construction de la fonction XOR sont illustrées. La première utilise une combinaison de portes NAND (Figure A.6-a) et vérifie l'équation suivante :

$$\begin{aligned} \text{s} &= \text{NAND(NAND(a,NAND(a,b)),NAND(b,NAND(a,b)))} \\ &= \overline{\overline{\overline{a.\overline{a.b}}.\overline{b.\overline{a.b}}}} \\ &= \overline{(\overline{a}+a.b).(\overline{b}+a.b)} \\ &= \overline{(\overline{a}.\overline{b})+(a.b)} \\ &= (a+b).(\overline{a}+\overline{b}) \\ &= a.\overline{b}+\overline{a}.b \\ &= a \oplus b \end{aligned} \tag{A.2}$$

La seconde construction (Figure A.6-b) considère les signaux $\overline{a}$ et $\overline{b}$ disponibles et reprend les principes évoqués pour les portes NAND et OR. Le symbole associé à une porte XOR est illustré par la figure A.6-c.

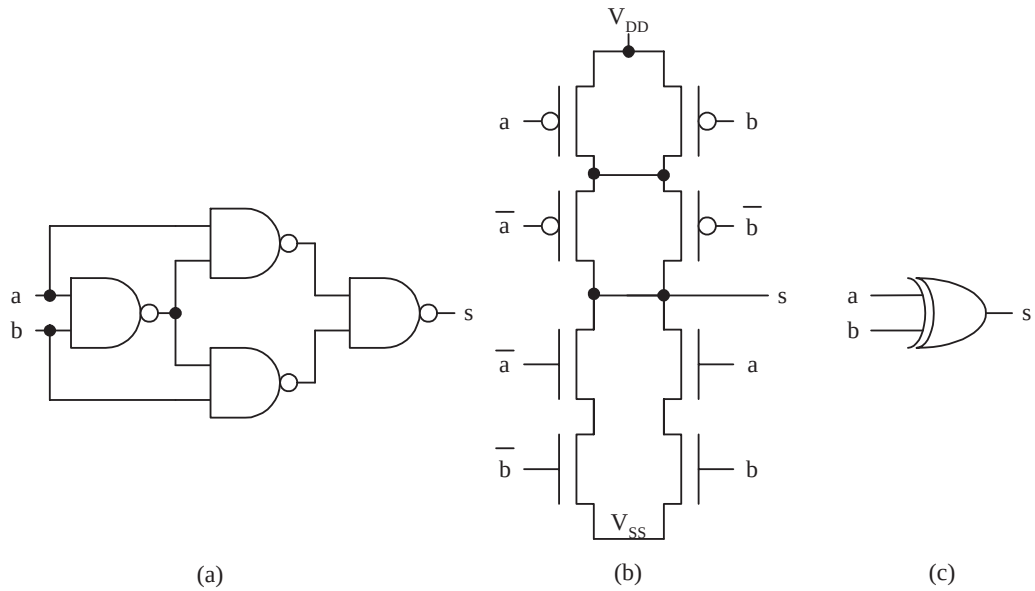


FIG. A.6 – Exemple d'une porte complexe CMOS : la porte XOR CMOS

A.3 Le transistor MOS analogique

Jusqu'à présent, le transistor MOS fut considéré comme un interrupteur qui représente dans les faits le plus simple modéle comportemental de celui-ci. Cette première approche est cependant suffisante pour la conception de fonctions numériques. Mais lorsque le transistor MOS considéré comme un interrupteur passe d'un état ouvert à un état fermé, il passe en fait de l'état d'accumulation à l'état de déplétion pour arriver à l'état de forte inversion. Le passage de l'état ouvert à l'état fermé n'est pas instantané et un modèle comportemental analogique est associé au transistor MOS permettant de décrire cette évolution.

A.3.1 Capacités du transistors NMOS

Le transistor NMOS peut se trouver dans l'état accumulation (Figure A.7) lorsque la tension grille-source V_{GS} est négative. Cette polarisation attire les absences d'électrons du substrat appelés trous sous l'oxyde de grille d'où le terme accumulation. Dans cet état il apparaît une capacité grille-substrat C_{GB}. La capacité par unité de surface de l'oxyde de silicium S_iO_2 est donnée par l'équation A.3 où e_{ox} est l'épaisseur d'oxyde.

$$C_{ox} = \frac{\varepsilon_0.\varepsilon_{S_iO_2}}{e_{ox}} = \frac{\varepsilon_{ox}}{e_{ox}} \tag{A.3}$$

La surface dessinée de l'oxyde de grille est égale à $W * L$. Mais lors de la fabrication physique du transistor il existe un phénomène parasite qui consiste à la diffusion des zones $n+$ de la source et du drain sous la grille d'une longueur de diffusion L_{diff} chacune. La longueur de canal effective est donnée par $L_{eff} = L - 2 * L_{diff}$. En revanche, la largeur de transistor W est considérée égale à celle dessinée parce qu'aucune zone de diffusion ne la limite à ses extrémités. La surface effective de grille créant la capacité grille-substrat est égale à $W * L_{eff}$ et la valeur de la capacité est donnée par l'équation A.4.

$$C_{GB} = C_{ox} * W * L_{eff} \tag{A.4}$$

Ce phénomène parasite de diffusion de la source et du drain sous l'oxyde de grille crée une capacité grille-source C_{GS} et une capacité grille-drain C_{GD}. La surface concernée pour chaque capacité est égale à $L_{diff} * W$. Les capacités C_{GS} et C_{GD} sont données par l'équation A.5.

$$C_{GS} = C_{ox} * W * L_{diff} = C_{GD} \tag{A.5}$$

La capacité totale entre la grille et la masse est égale à la somme de C_{GB}, C_{GS} et C_{GD} (Équation A.6).

$$C_{total} = C_{ox} * W * L \tag{A.6}$$

L'augmentation de la tension grille-source V_{GS}, tout en la maintenant inférieure à la tension de seuil du transistor V_{THN}, fait passer le transistor NMOS de l'état d'accumulation à l'état de déplétion (Équation A.8). La polarisation V_{GS} n'est pas suffisamment négative pour attirer les trous sous l'oxyde de grille et leur absence crée une charge négative sans porteur libre et un canal n induit dans le silicium S_i. La tension grille-source V_{GS} n'est pas aussi suffisamment positive pour y attirer des électrons. Il en résulte une zone libre de charge mobile sous l'oxyde de grille d'où le terme de déplétion qui crée une capacité C_{dep} (Équation A.7).

$$C_{dep} = \frac{\varepsilon_0.\varepsilon_{S_i}}{d} * W * L_{eff} \tag{A.7}$$

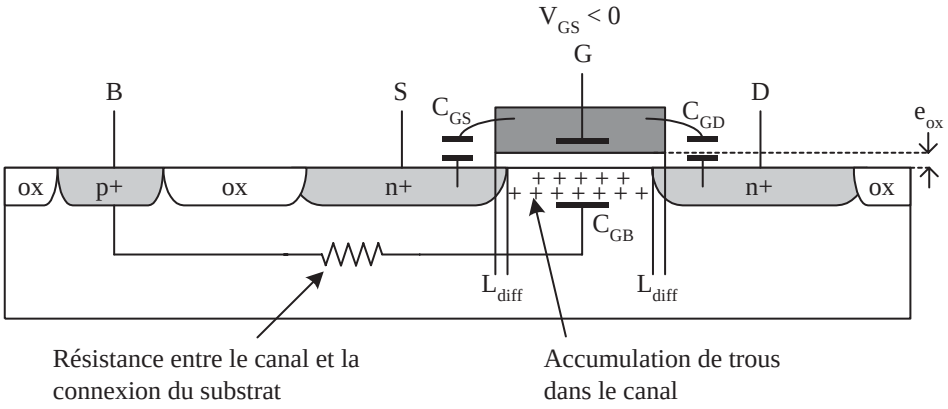


FIG. A.7 – Transistor NMOS en état d'accumulation

La capacité grille-substrat C_{GB} est alors égale à la capacité d'oxyde C_{ox} de la grille en série avec la capacité de déplétion C_{dep} (Équation A.8).

$$C_{GB} = \frac{C_{ox}.C_{dep}}{C_{ox} + C_{dep}} \tag{A.8}$$

L'apparition du canal entre la source et le drain fait que les capacités grille-source V_{GS} et grille-drain V_{GD} deviennent significatives et ne concernent plus seulement les parties de recouvrement dues aux diffusions parasites L_{diff}. A ce stade de polarisation du transistor NMOS, la grille joue un rôle équivalent pour la source et le drain et il est alors considéré que la moitié de la surface dessinée de la grille crée la capacité grille-source et que l'autre moitié crée la capacité grille-drain (Équation A.9).

$$C_{GS} = C_{ox}.\frac{W * L}{2} = C_{GD} \tag{A.9}$$

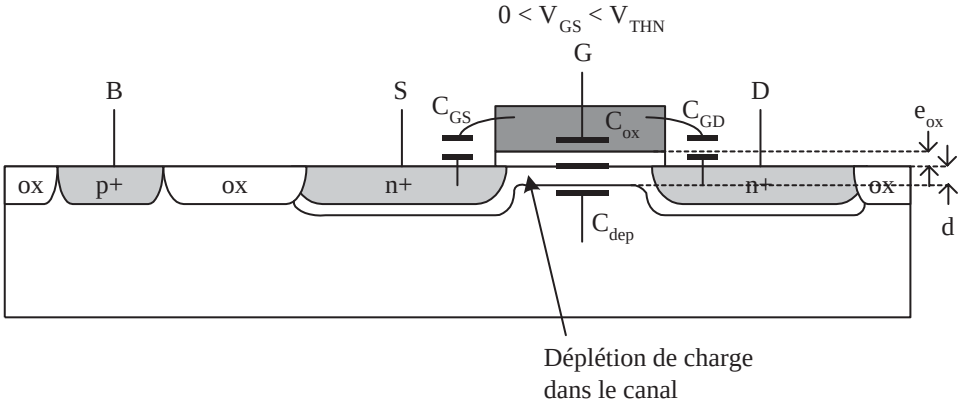


FIG. A.8 – Transistor NMOS en état de déplétion

En augmentant encore plus la tension grille-source V_{GS} bien au-delà de la tension de seuil du transistor V_{THN}, celui-ci passe de l'état de déplétion à l'état de forte inversion (Figure A.9). La polarisation V_{GS} est suffisamment positive pour attirer un grand nombre des électrons sous

l'oxyde de grille d'où le terme d'inversion (de charge) puisque le substrat était initialement dopé p. Il est alors considéré que le canal écrante le substrat de la grille et que donc la capacité grille-substrat C_{GB} est nulle (Équation A.10).

$$C_{GB} = 0 \tag{A.10}$$

Dans l'état de forte inversion avec $V_{GS} - V_{THN} < V_{DS}$, le canal formé est pincé du côté du drain. La grille crée une capacité grille-source C_{GS} pour $\frac{2}{3}$ de sa surface (Équation A.11) et du fait du pincement la capacité grille-drain C_{GD} est nulle (Équation A.12).

$$C_{GS} = C_{ox}.\frac{2.W.L}{3} \tag{A.11}$$

$$C_{GD} = 0 \tag{A.12}$$

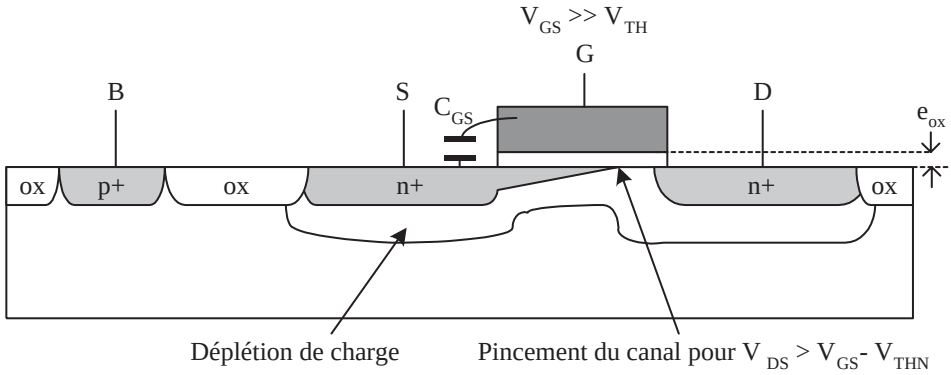


FIG. A.9 – Transistor NMOS en état de forte inversion

Pour être complet, il faut aussi mentionner que les zones de diffusion $n+$ de la source et du drain dans le substrat dopé p entraînent des jonctions NP. Elles créent une capacité de jonction entre la source et le substrat C_{JS} et une capacité de jonction entre le drain et le substrat C_{JD}. Elles sont toujours présentes quel que soit l'état du transistor. Le tableau A.5 récapitule l'ensemble des capacités en fonction de l'état du transistor NMOS et la figure A.10 les schématise.

Capacités	Accumulation	Déplétion	Inversion
C_{GB}	$C_{ox} * W * L_{eff}$	$\frac{C_{ox}.C_{dep}}{C_{ox}+C_{dep}}$	0
C_{GS}	$C_{ox} * W * L_{diff}$	$C_{ox}.\frac{W*L}{2}$	$C_{ox}.\frac{2.W.L}{3}$
C_{GD}	$C_{ox} * W * L_{diff}$	$C_{ox}.\frac{W*L}{2}$	0

TAB. A.5 – Capacités du transistor NMOS

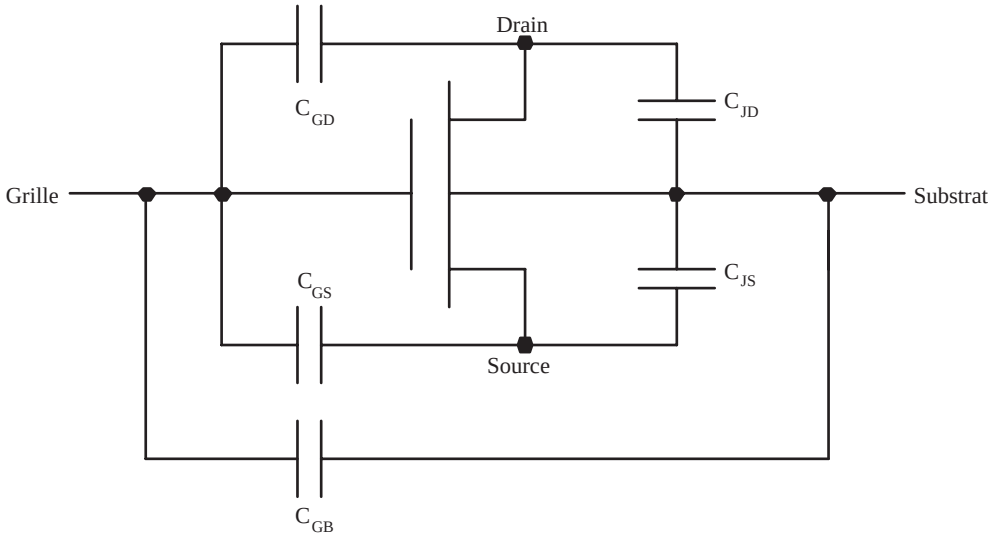


FIG. A.10 – Schéma de transistor avec capacités

A.3.2 Fonctionnement du transistor NMOS en mode non saturé

Le fonctionnement du transistor en mode non saturé correspond à un état transitoire du transistor lorsque celui-ci passe de l'état d'accumulation à l'état de forte inversion et inversement. L'état de déplétion fait parti de ce mode de fonctionnement. La figure A.11 illustre les conditions requises de cette analyse.

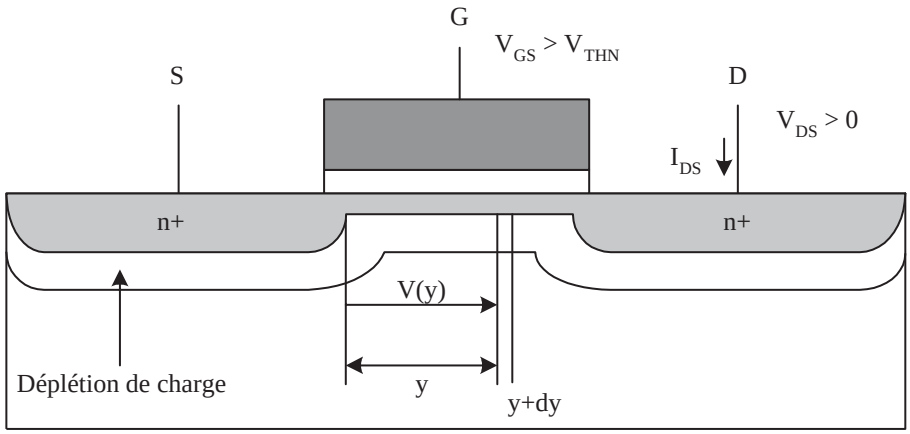


FIG. A.11 – Transistor NMOS en mode non saturé

La tension grille-source V_{GS} est supérieure à la tension de seuil du transistor V_{THN} ce qui inverse la surface sous l'oxyde grille. La tension drain-source V_{DS} est positive ce qui entraîne le passage d'un courant I_{DS} du drain vers la source.

La différence de potentiel entre la grille et l'abscisse y est égale à $V_{GS} - V(y)$ en prenant la tension de source V_S comme référence. La charge stockée par la capacité de grille et par unité de surface est égale à la capacité par unité de surface d'oxyde de grille C_{ox} (Équation A.3) multipliée par la différence de potentiel en y (Équation A.13).

$$Q = C_{ox}.(V_{GS} - V(y)) \tag{A.13}$$

Le transistor ne conduit qu'à partir de l'instant où la tension grille-source V_{GS} est égale à la tension de seuil V_{THN}. Cette différence de potentiel entraîne une charge Q_{THN} par unité de surface dans le canal (Équation A.14).

$$Q_{THN} = C_{ox}.V_{THN} \tag{A.14}$$

La charge par unité de surface finalement disponible en y est la charge Q moins la charge nécessaire à la conduction du transistor Q_{THN} (Équation A.15).

$$Q(y) = C_{ox}.(V_{GS} - V(y) - V_{THN}) \tag{A.15}$$

La mobilité des électrons μ_n dans le canal à l'abscisse y est définie par le rapport entre la vitesse moyenne de ceux-ci et le champ appliqué $V(y)$ qui les contraint. La résistivité du canal en y est inversement proportionnelle au nombre de charges libres $Q(y)$ présentes et à la mobilité des porteurs (Équation A.16). Plus le nombre de charges est important par cm^3 ou plus les porteurs

sont mobiles et moins le matériau est résistant.

$$\rho_{canal} = \frac{1}{\mu_n.Q(y)} \tag{A.16}$$

Le nombre de charges traversant le transistor de largeur W est égal à $\mu_n.Q(y).W$. La résistivité du canal en y est donc égale à $\frac{1}{\mu_n.Q(y).W}$. La variation de résistance du canal dR pour passer de y à $y + dy$ avec dy petit est alors égale à :

$$dR = \frac{1}{\mu_n.Q(y).W}.dy \tag{A.17}$$

La variation de la différence de potentiel entre y et $y + dy$ est alors égale à :

$$dV(y) = I_{DS}.dR = \frac{I_{DS}}{\mu_n.Q(y).W}.dy = \frac{I_{DS}}{\mu_n.C_{ox}.(V_{GS} - V(y) - V_{THN}).W}.dy \tag{A.18}$$

Le courant traversant dy est égal à :

$$I_{DS}.dy = \mu_n.C_{ox}.(V_{GS} - V(y) - V_{THN}).W.dV(y) \tag{A.19}$$

Le courant traversant le canal de longueur effective L_{eff} est égal à :

$$I_{DS} \int_0^{L_{eff}} dy = \mu_n.C_{ox}.W. \int_0^{V_{DS}} (V_{GS} - V(y) - V_{THN}).dV(y) \tag{A.20}$$

Ce qui donne :

$$I_{DS} = \mu_n.C_{ox}.\frac{W}{L_{eff}}.\left((V_{GS} - V_{THN}).V_{DS} - \frac{{V_{DS}}^2}{2}\right) \tag{A.21}$$

Pour un transistor PMOS, le courant drain-source se calcule de façon identique et est égal à :

$$I_{DS} = \mu_p.C_{ox}.\frac{W}{L_{eff}}.\left(V_{SG} - V_{THP}).V_{SD} - \frac{{V_{SD}}^2}{2}\right) \tag{A.22}$$

En conclusion, lorsqu'un transistor fonctionne en mode non saturé, le courant drain-source est fonction de la différence de potentiel grille-source V_{GS} et de la différence de potentiel drain-source V_{DS} (Équation A.23). C'est vrai pour les deux types de transistor NMOS et PMOS.

$$I_{DS} = f(V_{GS}, V_{DS}) \tag{A.23}$$

A.3.3 Fonctionnement du transistor NMOS en mode saturé

Le fonctionnement du transistor en mode saturé correspond à l'état de forte inversion de celui-ci. La figure A.12 illustre les conditions requises de ce mode de fonctionnement.

La tension appliquée en l'abscisse $y = L$, $V(L)$, est égale à la tension drain-source V_{DS}. Jusqu'à présent, cette dernière a toujours été considérée inférieure à la tension grille-source moins la tension de seuil du transistor $V_{GS} - V_{THN}$. Cette contrainte permet de garantir que l'inversion de charge libre dans le canal n'est jamais nulle. La tension $V_{GS} - V_{THN}$ est appelée tension de saturation $V_{DS,sat}$ parce que, lorsque la tension V_{DS} est égale à $V_{GS} - V_{THN}$, le canal est pincé en l'abscisse $y = L$, c'est à dire que la charge libre présente dans le canal est égale à 0. En effet,

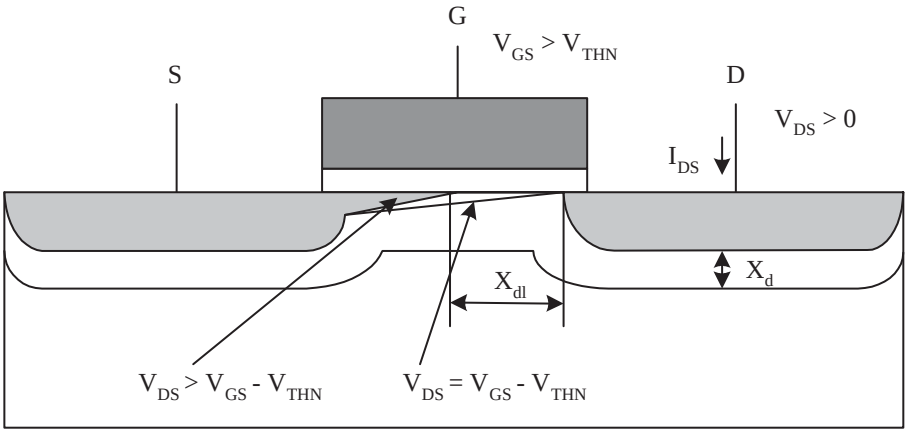


FIG. A.12 – Transistor NMOS en mode saturé

la charge libre en l'abscisse $y = L$ est égale à :

$$Q = C_{ox}.(V_{GS} - V(L) - V_{THN}) \tag{A.24}$$

où $V(L)$ est égal à V_{DS}. Si $V_{DS} = V_{GS} - V_{THN}$ alors la charge libre présente dans le canal devient :

$$Q = C_{ox}.(V_{GS} - V_{DS}) = C_{ox}.(V_{GS} - V_{GS} + V_{THN} - V_{THN}) = 0 \tag{A.25}$$

Une zone de déplétion de charge commence à apparaître dans le canal à partir du drain en l'abscisse $y = L$. L'augmentation de la tension drain-source V_{DS} au-delà de la tension de saturation $V_{DS,sat}$ entraîne un pincement du canal en l'abscisse y où la tension $V(y)$ est égale à la tension de saturation $V_{DS,sat}$. En effet, si $V_{DS} > V_{GS} - V_{THN}$ en $y = L$, alors il existe une abscisse $y < L$ tel que la tension $V(y)$ soit égale à $V_{DS,sat} = V_{GS} - V_{THN}$. Ceci a pour effet de pincer le canal en y et d'augmenter la zone de déplétion de charge du côté du drain (X_{dl} sur la figure A.12).

Lorsque le transistor opère en mode saturé, c'est à dire lorsque $V_{DS} > V_{GS} - V_{THN}$, le transistor est dans l'état de forte inversion. Le courant est alors égal au courant défini par l'équation A.21 dans laquelle V_{DS} est remplacée par $V_{GS} - V_{THN}$ (Équation A.26).

$$I_{DS} = \mu_n.C_{ox}.\frac{W}{L_{eff}}.\frac{(V_{GS} - V_{THN})^2}{2} \tag{A.26}$$

La tension drain-source V_{DS} impose le pincement du canal en l'abscisse y si celle-ci est supérieure à la tension de saturation $V_{DS,sat}$. La longueur effective du canal L_{eff} est alors égale à la longueur dessinée L moins la longueur de diffusion du drain L_{diff} moins la longueur de diffusion de la source L_{diff} et moins la longueur de déplétion X_{dl} due à la tension V_{DS} (Équation A.27).

$$L_{eff} = L - 2 * L_{diff} - X_{dl} \tag{A.27}$$

Le courant drain-source I_{DS} devient alors :

$$I_{DS} = \mu_n.C_{ox}.\frac{W}{L - 2.L_{diff} - X_{dl}}.\frac{(V_{GS} - V_{THN})^2}{2} \tag{A.28}$$

Ceci signifie que lorsque la tension drain-source V_{DS} augmente, la longueur effective du canal diminue parce que la zone de déplétion augmente, et que donc le courant drain-source I_{DS}

augmente aussi. L'influence de la tension drain-source V_{DS} sur le courant drain-source I_{DS} est déterminée par :

$$\frac{\partial I_{DS}}{\partial V_{DS}} = -\mu_n.C_{ox}.\frac{W}{(L-2.L_{diff}-X_{dl})^2}.\frac{(V_{GS}-V_{THN})^2}{2}.\frac{d(L-2.L_{diff}-X_{dl})}{dV_{DS}} \tag{A.29}$$

Ce qui donne :

$$\frac{\partial I_{DS}}{\partial V_{DS}} = I_{DS}.\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{DS}} \tag{A.30}$$

D'où l'expression du courant drain-source I_{DS} en fonction aussi de la tension drain-source V_{DS} :

$$I_{DS} = \frac{\mu_n.C_{ox}.W}{L-2.L_{diff}-X_{dl}}.\frac{(V_{GS}-V_{THN})^2}{2}.\left(1+\left(\frac{1}{L-2.L_{diff}-X_{dl}}.\frac{dX_{dl}}{dV_{DS}}\right).(V_{DS}-V_{DS,sat})\right) \tag{A.31}$$

En conclusion, lorsqu'un transistor fonctionne en mode saturé, le courant drain-source I_{DS} est aussi une fonction de la tension grille-source V_{GS} et de la tension drain-source V_{DS} (Équation A.32). C'est vrai pour les deux types de transistor NMOS et PMOS.

$$I_{DS} = f(V_{GS}, V_{DS}) \tag{A.32}$$

A.4 1 faible, 1 fort, 0 faible et 0 fort

D'après le paragraphe A.3.3 et l'équation A.24, la charge libre présente dans le canal entre le drain et la source est égale à :

$$\begin{aligned} Q &= C_{ox}.(V_{GS} - V_{DS} - V_{THN}) \\ &\quad \text{avec } V_{DS} < V_{GS} - V_{THN} \end{aligned} \tag{A.33}$$

Dans le cas d'un transistor NMOS en mode saturé et ayant sa tension de drain égale à V_{DD}, le courant drain-source I_{DS} est non nul tant que la tension drain-source V_{DS} est strictement inférieure à $V_{GS} - V_{THN}$ (Équation A.25), c'est à dire strictement inférieure à $V_{DD} - V_{THN}$ dans le cas présent (Équation A.34).

$$\begin{aligned} V_{DS} &< V_{DD} - V_{THN} \\ I_{DS} &\neq 0 \end{aligned} \tag{A.34}$$

Dès lors que le transistor NMOS a une tension drain-source V_{DS} supérieure ou égale à la tension $V_{DD} - V_{THN}$, le courant drain-source I_{DS} devient nul. Le transistor est alors dans l'impossibilité de véhiculer plus de charge sur sa source. Une capacité C_{sortie} connectée à sa source ne peut donc pas recevoir plus de charge que $C_{sortie}.(V_{DD} - V_{THN}) = C_{sortie}.V_S$. La tension maximale de sortie sur la source V_S est donc égale à $V_{DD} - V_{THN}$. Un transistor NMOS saturé ayant un 1 fort sur son drain représenté par la tension V_{DD} ne peut transmettre qu'un 1 faible sur sa source représenté par $V_{DD} - V_{THN}$.

Dans le cas d'un transistor NMOS en mode saturé et ayant sa tension de source égale à $0V$, le courant drain-source I_{DS} est non nul tant que la tension drain-source V_{DS} est strictement inférieure à $V_{DD} - V_{THN}$ (Équation A.34). Une capacité C_{sortie} connectée au drain et contenant une charge $Q = C_{sortie}.V_D$ peut donc toujours être entièrement vidée jusqu'à ce que la tension de drain V_D soit nulle et égale à la tension de source V_S parce que la tension drain-source V_{DS} reste toujours strictement inférieure à la tension grille-source V_{GS}. La tension minimale de sortie sur le drain V_D est donc égale à $0V$. Un transistor NMOS saturé ayant un 0 fort sur sa source représenté par la tension $0V$ transmet un 0 fort sur son drain.

Un raisonnement identique démontre qu'un transistor PMOS ne peut transmettre qu'un 0 faible représenté par V_{THP} et qu'il transmet un 1 fort représenté par V_{DD}.

A.5 L'effet de substrat

Ce phénomène apparaît dès lors que la source du transistor, utilisée comme référence dans toutes les expressions jusqu'à présent, et le substrat du transistor présentent une différence de potentiel. Cette différence de potentiel modifie les caractéristiques électriques du transistor dont la tension de seuil est notée V_{TH}. La tension de seuil du transistor correspond à la tension grille-source V_{GS} minimale à partir de laquelle il devient passant et permet le passage d'un courant drain-source I_{DS}. La figure A.13 illustre un circuit permettant d'évaluer la tension de seuil d'un transistor NMOS en fonction de la tension source-substrat. Lorsque la différence de potentiel source-substrat V_{SB} est nulle, un courant drain-source I_{DS} apparaît dès que la tension grille-source V_{GS} atteint la tension de seuil V_{THN0} d'environ 0.7V. Dès que la tension source-substrat V_{SB} varie, la tension de seuil du transistor varie. La figure A.13 montre cinq valeurs de tensions de seuil différentes correspondant à cinq différences de potentiel source-substrat différentes.

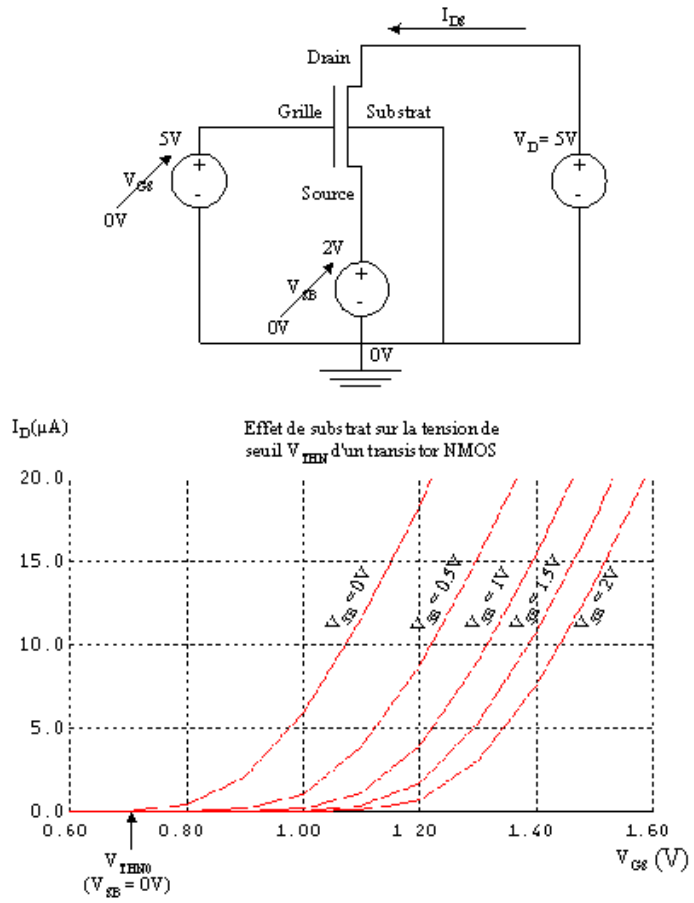


FIG. A.13 – Effet de substrat sur la tension de seuil

Lorsque la différence de potentiel grille-source V_{GS} est égale à la tension de seuil V_{THN}, la force électrique $\overrightarrow{F_{GS}elec}$ ainsi générée est suffisamment intense pour attirer les électrons sous la grille et rendre le transistor NMOS passant.

Lorsque la différence de potentiel source-substrat V_{SB} est nulle (Figure A.14-a), les électrons sous le canal ne sont soumis qu'à la force électrique $\overrightarrow{F_{GS}}$. Les électrons dans la source et le drain

ne sont soumis à aucune autre force électrique puisque la différence de potentiel drain-source V_{DS} est ici nulle. Dans ces conditions, le canal est créé à partir d'une tension minimale dite tension de seuil V_{THN0}.

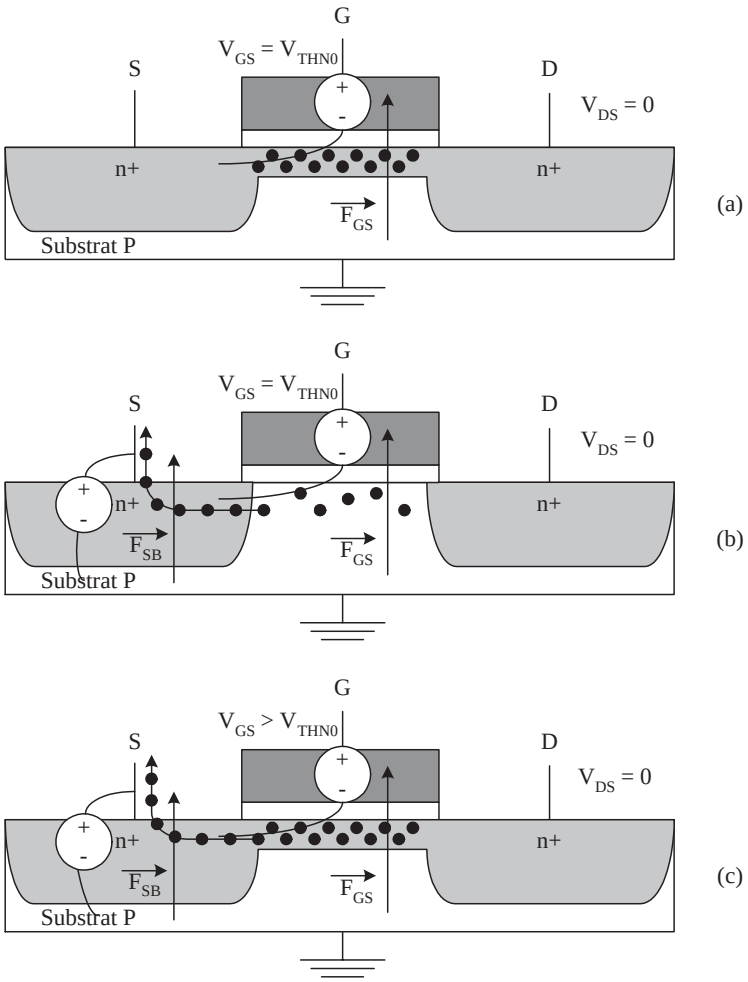


FIG. A.14 – Tension de seuil et différence de potentiel source-substrat

Lorsqu'une différence de potentiel positive V_{SB} apparaît entre la source et le substrat, les électrons de la source sont soumis à une force électrique $\overrightarrow{F_{SB}}$ créant un courant qui implique aussi les électrons du canal. Les charges diminuant dans le canal, le transistor est à nouveau bloqué (Figure A.14-b).

Il faut alors augmenter la différence de potentiel grille-source V_{GS} pour attirer plus d'électrons sous la grille et maintenir le canal conducteur et le transistor passant. Le canal est créé à partir d'une tension minimale V_{THN} qui est supérieure à V_{THN0} (Figure A.14-c).

Comme mentionné précédemment, l'effet de substrat apparaît dès lors que la tension source devient différente de la tension de substrat. Ce cas se présente systématiquement lorsque la source d'un transistor est isolée d'une tension de référence. La tension de la source devient flottante alors que le substrat est toujours référencé, à la masse, pour un transistor NMOS ou à l'alimentation pour un transistor PMOS.

Ce phénomène est classique d'un réseau série de transistors (Figure A.15-a). Le transistor NMOSa a sa source connectée en permanence à la masse. Il est impossible de créer une différence de potentiel source-substrat non nulle et donc impossible d'avoir un effet de substrat. En revanche, le transistor NMOSb peut avoir sa source non référencée à la masse lorsque la tension appliquée en a est nulle. La tension au nœud n est flottante et une différence de potentiel non nulle apparaît entre la source et le substrat du transistor NMOSb. L'effet de substrat a pour conséquence de modifier les caractéristique électriques du transistor NMOSb dont la tension de seuil.

Dans le cas d'un réseau parallèle de transistors (Figure A.15-b) les transistors NMOSa et NMOSb ont leurs sources connectées en permanence à la masse. Il est impossible de créer une différence de potentiel source-substrat non nulle et donc impossible d'avoir un effet de substrat pour les deux transistors.

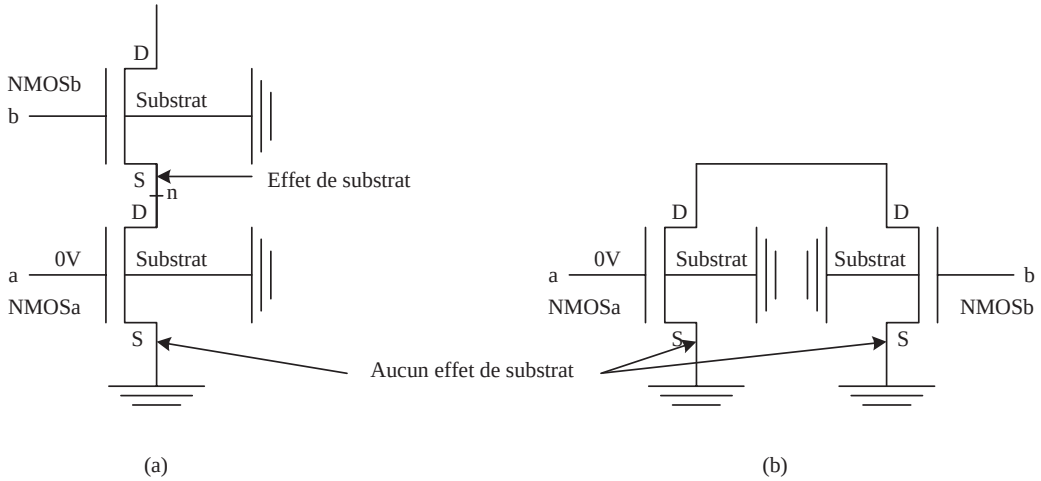


Fig. A.15 – Réseau de transistor et effet de substrat

Dans le cas d'un réseau série de transistors, les transistors ont un modèle de transistor identique mais paramétré par des caractéristiques électriques différentes qui leur sont spécifiques. Dans le cas d'un réseau parallèle de transistors, les transistors ont non seulement un modèle identique mais aussi des caractéristiques électriques identiques.

- Réseau série de transistors : caractéristiques électriques différentes
- Réseau parallèle de transistors : caractéristiques électriques identiques

Annexe B

Caractérisation de la technologie

Sommaire

B.1 Processus de fabrication des transistors NMOS et PMOS et modèles spice

Avant de dessiner un transistor il convient de faire quelques rappels rapides sur le processus de fabrication permettant de comprendre les règles de schéma. La figure B.1 présente les différentes étapes de fabrication d'un transistor NMOS en coupe transversale. La première étape (Figure B.1-a) commence avec un substrat dopé P sur lequel on a fait croître une épaisse couche d'oxyde de silicium Si02. A l'aide du masque NDIFF (Figure B.1-b), la zone d'implantation du transistor est ouverte par élimination du Si02. Il est alors nécessaire de faire croître l'oxyde mince de grille (Figure B.1-c). A l'aide du masque POLY1 (Figure B.1-d) le poly-silicium de la grille est déposé. Il faut ensuite supprimer l'oxyde mince (Figure B.1-e) définissant les zones de diffusion fortement dopées N. Puis, avec le masque NPLUS (Figure B.1-f), l'implantation (ou la diffusion) des zones fortement dopées N est réalisée.

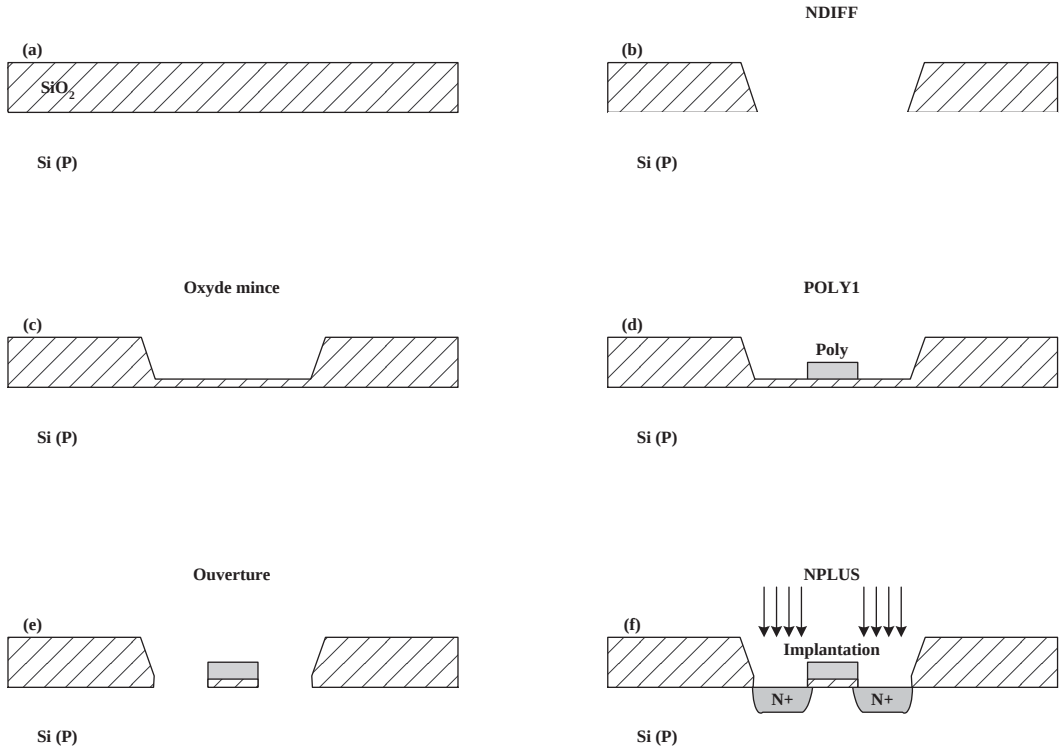


FIG. B.1 – Processus de diffusion d'un transistor NMOS

Les règles minimales de conception définissent la plus petite taille possible du transistor NMOS. La largeur du transistor NMOS sera fixée à 0,8 μm et celle du transistor PMOS à 1,6 μm. En doublant celle-ci, les temps de transition et de propagation deviennent équivalents. La figure B.2 présente les dimensions de ces deux types de transistors.

SPICE est un outil capable de simuler le comportement électrique des transistors MOS à partir de leur géométrie et de paramètres physiques. Une présentation efficace de ces paramètres est donnée dans les modèles BSIM (Berkeley Short-Channel Insulated Gate Field Effect Transistor

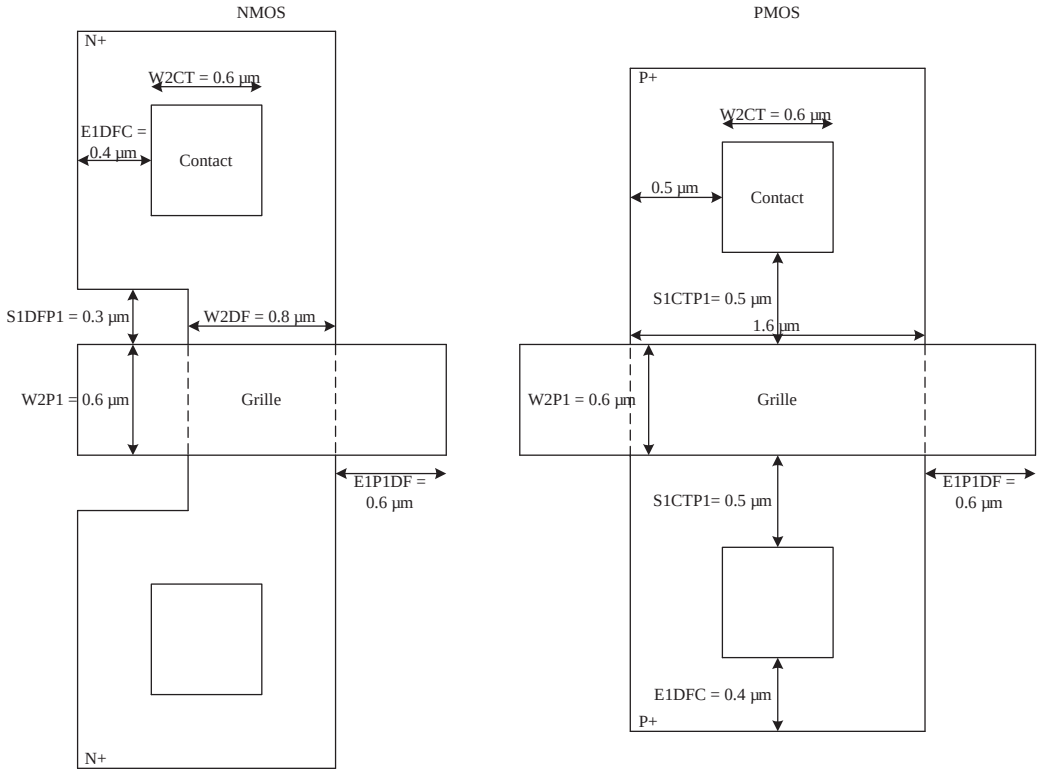


FIG. B.2 – Règles de conception des transistors NMOS et PMOS

Model). Le modèle de simulation utilisé est le modèle BSIM3v3. Il est paramétré avec les valeurs suivantes (Tableau B.1) pour chaque type de transistor :

Paramètre	Transistor N	Transistor P	Commentaire
W	$0.8\mu m$	$1.6\mu m$	Largeur du transistor
L	$0.6\mu m$	$0.6\mu m$	Longueur de canal
AS	$1.4^2 + 0.3 * 0.8$ $= 2.2\mu m^2$	$1.6 * 1.5$ $= 2.4\mu m^2$	Surface de la source
AD	$2.2\mu m^2$	$2.4\mu m^2$	Surface du drain
PS	$1.4 * 3 + 0.3 * 2 + 0.8$ $= 5.6\mu m$	$(1.6 + 1.5) * 2$ $= 6.2\mu m$	Périphérie de la source
PD	$5.6\mu m$	$6.2\mu m$	Périphérie du drain

TAB. B.1 – Paramètres des transistors NMOS et PMOS

Avec ces paramètres l'inverseur équilibré présente un temps de transition de la sortie d'environ $0.54ns$ pour les deux changements d'état (Figure B.3).

Les temps de propagation du signal d'entrée sont aussi équilibrés. Lorsque l'entrée passe de 1 à 0, la sortie passe de 0 à 1 avec un retard de 100 ps environ. De même, lorsque l'entrée passe de 0 à 1, la sortie passe de 1 à 0 avec un retard de 100 ps environ (Figure B.4).

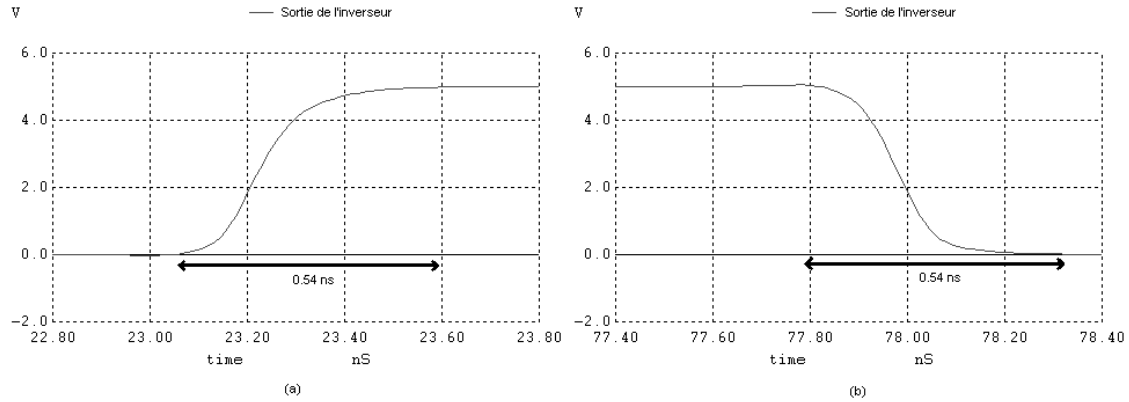


FIG. B.3 – Temps de transition en montée (a) et en descente (b) de l'inverseur

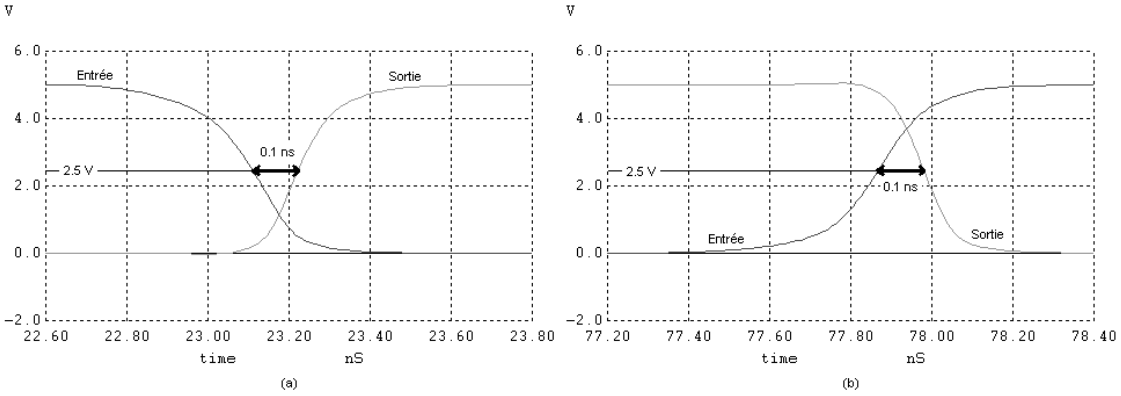


FIG. B.4 – Temps de propagation de l'inverseur d'un front montant (a) et d'un front descendant (b)

B.2 Caractérisation de l'inverseur

La première caractéristique cherchée est la sortance, c'est à dire le nombre d'inverseurs mis en parallèle en sortie qui peuvent être supportés sans dégrader celle-ci. La dégradation est considérée trop importante à partir du doublement du temps de transition par rapport à un seul inverseur en sortie. La figure B.5 illustre les circuits simulés.

Avec 6 inverseurs (Figure B.6) mis en parallèle avec la sortie, le temps de transition de montée de s6 dépasse la limite fixée précédemment, c'est à dire plus de deux fois le temps de transition avec un seul inverseur. La limite est donc égale à une charge équivalente à 5 inverseurs parallèles en sortie d'un inverseur (s5 sur le schéma).

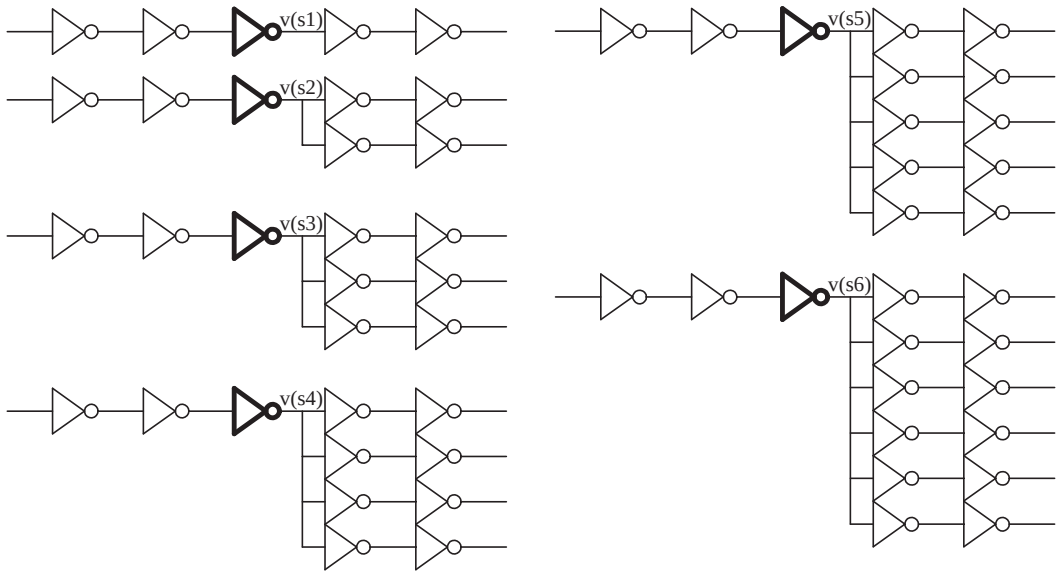


FIG. B.5 – Circuits de caractérisation de la sortance de l'inverseur

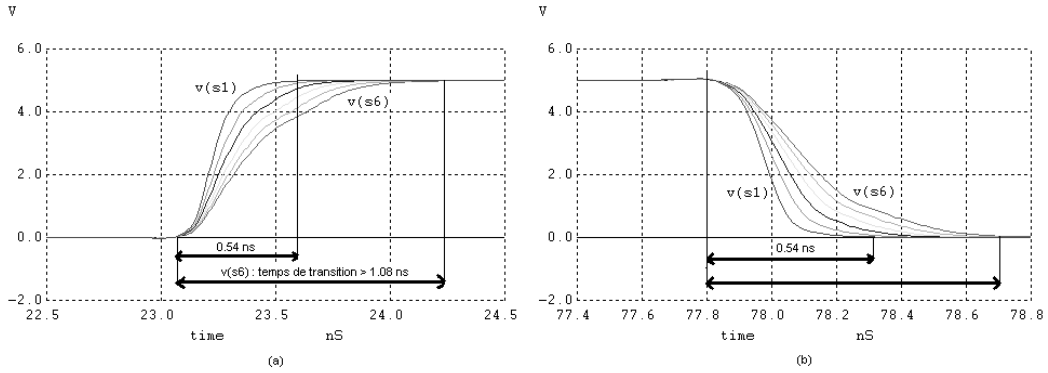


FIG. B.6 – Résultats de la caractérisation de la sortance de l'inverseur

La seconde caractéristique recherchée est la capacité de reformatage, c'est à dire la capacité d'un inverseur à reformater un signal dégradé par une sortance trop importante. Pour évaluer cette propriété, le signal de sortie d'un inverseur est progressivement dégradé par un nombre croissant d'inverseurs de charge connectés en sortie jusqu'à ce que la dégradation soit trop importante. La dégradation est considérée trop importante à partir du doublement du temps de transition ou du temps de propagation par rapport à un seul inverseur en entrée. La figure B.7 illustre les circuits simulés.

Avec 16 inverseurs mis en parallèle avec l'entrée, les temps de transition de descente et de montée dépassent la limite fixée c'est à dire plus de deux fois le temps de transition avec un seul inverseur (Figure B.8). Les autres temps en propagation et transition sont toujours respectés. La limite sera donc fixée à une charge équivalente à 15 inverseurs parallèles en entrée d'un inverseur (s15 sur le schéma).

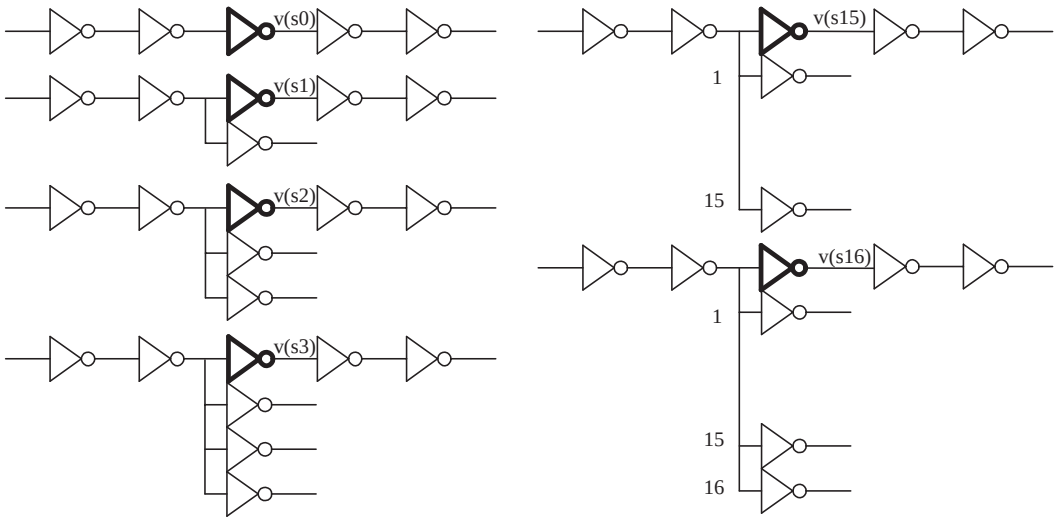


FIG. B.7 – Circuits de caractérisation de la capacité de reformatage de l'inverseur

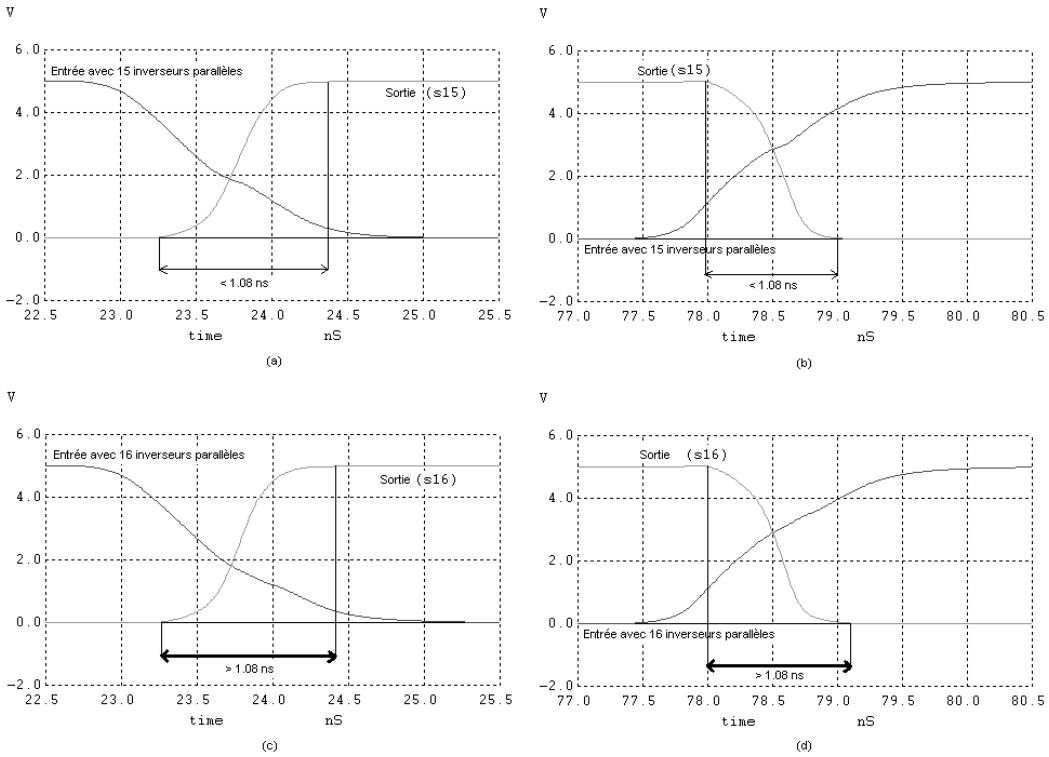


FIG. B.8 – Résultats de la caractérisation de la capacité de reformatage de l'inverseur

Annexe C

Notions de cryptologie

Sommaire

C.1 Terminologie

La cryptologie est étymologiquement la science du secret. Elle peut être dissociée en deux branches, la cryptographie pour l'art des écritures cachées qui assure la sécurité et la cryptanalyse qui contourne la cryptographie afin de déjouer la sécurité. On appellera chiffrement l'action de rendre inintelligible un message en texte clair afin d'assurer sa confidentialité. Le message ainsi rendu inintelligible est appelé message chiffré. On appellera déchiffrement l'action de retrouver le clair associé à un chiffré.

Fig. C.1 – Chiffrement et déchiffrement

La cryptographie n'assure pas seulement comme fonction la confidentialité d'un clair. Elle peut aussi permettre son authentification, c'est à dire de garantir l'origine d'un chiffré afin d'éviter une usurpation d'auteur. Le clair peut encore avoir été modifié à l'insu de l'émetteur. La cryptographie peut en assurer l'intégrité en garantissant que le clair obtenu est bien celui que l'émetteur a chiffré. Dans certains cas, le clair ne doit pas pouvoir être contesté par son émetteur. La cryptographie peut alors assurer la non répudiation en garantissant que l'émetteur ne peut nier à tort l'envoi par exemple.

Un algorithme cryptographique est utilisé afin de chiffrer et de déchiffrer. Si la sécurité repose sur le secret de l'algorithme, alors dès que celui-ci est divulgué ou si un utilisateur est révoqué lorsque celui-ci perd ses droits, tout chiffré devient potentiellement déchiffrable pour ceux qui l'ont intercepté. La cryptographie moderne a résolu ce problème en introduisant la notion de clé. Le secret ne repose plus sur l'algorithme mais sur la clé qui peut prendre un grand nombre de valeurs appelé espace des clés. Dans la suite, seul un algorithme cryptographique public sera considéré pour chiffrer et déchiffrer sachant que le secret repose sur une clé.

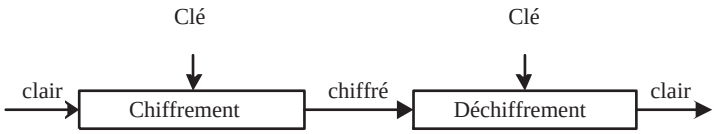


Fig. C.2 – Chiffrement et déchiffrement avec clé

La clé de chiffrement et la clé de déchiffrement peuvent être identiques ou différentes selon les algorithmes. La connaissance de l'algorithme, des clairs possibles, des chiffrés et de l'espace des clés forme un cryptosystème. On distinguera deux grands types d'algorithmes : à clé secrète ou à clé publique. Les algorithmes à clé secrète sont tels que la clé de déchiffrement est obtenue à partir de la clé de chiffrement par calcul ou inversement. Très souvent, les clés de chiffrement et de déchiffrement sont identiques et toute la sécurité repose sur le secret de cette clé commune. Si celle-ci est divulguée, alors tout le monde peut chiffrer ou déchiffrer dans un cryptosystème à clé secrète. Il existe deux sous-classes d'algorithmes à clé secrète. Ceux qui opèrent sur un bit de clair à la fois sont appelés algorithmes de chiffrement par flot ou en continu et ceux qui opèrent sur un groupe de bits de clair, nommé bloc et qui sont appelés algorithmes de chiffrement par bloc.

Les algorithmes à clé publique sont différents des algorithmes à clé secrète dans leur construction des clés de chiffrement et de déchiffrement. En effet, la connaissance de la clé de chiffrement ne permet pas de calculer simplement la clé de déchiffrement. Il y a bien une relation entre elles mais le problème mathématique à résoudre pour en déduire l'une de l'autre est difficile lorsque l'on ne possède pas le secret. Il est alors possible de diffuser publiquement la clé de chiffrement ce qui permet à tout le monde de chiffrer sachant que personne ne peut facilement déchiffrer, la clé de déchiffrement étant gardée secrète.

Comme cela fut déjà mentionné, la cryptanalyse s'emploie à contourner la cryptographie pour déjouer la sécurité que ce soit en confidentialité, authentification, intégrité et non répudiation. On appellera attaque, une tentative de cryptanalyse. Lorsque celle-ci vise à déjouer la confidentialité, l'action s'appellera décrypter. Au dix neuvième siècle, Kerckhoffs a posé un axiome fondamental de la cryptographie : l'attaquant possède tous les détails de l'algorithme sauf la clé de chiffrement. Le monde académique considère que si avec toute cette connaissance l'algorithme ne peut être cassé, c'est à dire que l'on ne peut retrouver le clair ou la clé de chiffrement totalement ou partiellement, alors il est impossible de le casser sans cette connaissance. En se plaçant dans le contexte de Kerckhoffs, il existe plusieurs types d'attaques possibles. L'attaque à texte chiffré seul considère que l'attaquant ne possède que des chiffrés. L'objectif est de retrouver de l'information sur les clairs ou mieux les clés de chiffrement. L'attaque à texte clair connu considère que l'attaquant possède non seulement les chiffrés mais aussi les clairs correspondants. L'objectif est alors de retrouver de l'information sur les clés de chiffrement. L'attaque à texte clair choisi considère que l'attaquant possède non seulement les chiffrés et les clairs correspondants mais qu'il a aussi la possibilité de choisir les clairs à chiffrer. L'objectif est de retrouver de l'information sur les clés de chiffrement. Elle peut être dite adaptative lorsque l'attaquant peut sélectionner en plus le clair à chiffrer en fonction du résultat précédent. Cette notion d'adaptabilité est un raffinement. L'attaque à texte chiffré choisi considère que l'attaquant peut déchiffrer certains chiffrés. Il s'en sert ensuite afin de retrouver de l'information sur les autres chiffrés ou les clés. La complexité des attaques est mesurée selon les critères suivants :

- la complexité en information est la quantité d'information nécessaire en entrée de l'algorithme pour réussir l'attaque,
- la complexité en temps est le temps nécessaire pour réaliser l'attaque,
- la complexité en espace est la quantité de mémoire nécessaire pour l'attaque.

C.2 Historique et fonctions fréquentes en cryptographie

Au IVe siècle avant notre ère à Sparte, Plutarque rapportait que les communications entre les Ephores et les chefs des armées en campagne étaient chiffrées à l'aide de la scytale. C'était un bâton dont le diamètre représentait la clé de chiffrement et de déchiffrement et sur lequel était enroulée une lanière en spires jointives. Le clair était rédigé puis la lanière était déroulée. Elle portait alors toujours les mêmes lettres que le clair mais dans un ordre différent. C'était le principe de transposition. Le procédé allemand ADFGVX utilisé pendant la première guerre mondiale était, entre autres, un chiffre à transposition. Il fut cassé par George Painvin qui réussit à reconstituer le tableau de transposition en plus de la clé de substitution. La technique de transposition permet en cryptographie de réaliser une fonction de diffusion. Cette fonction permet de disperser la redondance d'un clair en la répartissant dans le chiffré. A titre d'exemple de redondance à diffuser, il faut savoir que chaque langue possède une fréquence moyenne pour

chaque lettre de l'alphabet qui lui est propre. Un autre chiffre très connu est celui de César. Il consistait, pour chiffrer, à décaler d'un certain nombre de rangs la lettre claire. Si le rang est deux, alors A est substitué par B, B par C et ainsi de suite jusqu'à Z par A pour toutes les lettres du clair. C'était le principe de substitution. Il existe plusieurs types de substitution. La substitution simple consiste à remplacer chaque caractère du clair par un caractère correspondant dans le chiffré. C'est une permutation quelconque. Pour décrypter, il est possible d'essayer les vingt cinq rangs possibles. Mais chaque langue possède ses propres caractéristiques en fréquence. Par exemple, la lettre " e " revient avec une fréquence de dix sept pour cent en français. Il suffit pour chaque lettre du chiffré d'analyser sa fréquence et alors d'associer la lettre clair correspondant à la même fréquence de la langue. Plus le chiffré est long, plus sera pertinente l'analyse en fréquence. La substitution homophonique permet de remplacer chaque caractère du clair par plusieurs caractères dans le chiffré. Ce procédé a l'avantage de brouiller les fréquences des lettres du chiffré. Il fut utilisé dans le chiffre de Philibert Babou de 1558. La substitution polyalphabétique est réalisée à partir de plusieurs substitutions simples. Elle fut mise en œuvre par Vigenère dont le procédé était apparemment fondé sur le tableau de Trithème. Cela consiste à changer l'alphabet de substitution à chaque chiffrement d'une lettre du clair. Le décryptement consiste à se ramener à plusieurs substitutions simples en supposant la longueur de la clé. On peut utiliser le procédé de Kasiski (1863) qui identifie dans le chiffré une séquence de lettres identiques. Ceci indique probablement qu'il s'agit du même mot qui fut chiffré avec la même position relative de la clé. Ensuite il suffit de réarranger les groupes de lettres de longueur égale à celle de la clé. Puis, sur chaque groupe, il suffit d'appliquer le décryptement d'une substitution simple. La technique de substitution permet en cryptographie de réaliser une fonction de confusion. Cette fonction élimine les redondances et les biais statistiques du chiffré. De nos jours on ne traite plus des caractères mais des bits, ce n'est qu'un changement d'alphabet qui passe de 26 éléments à 2 éléments. Cependant, la plupart des algorithmes combinent toujours les deux principes de diffusion et confusion.

C.3 L'algorithme DES

C.3.1 Introduction

Le DES (Data Encryption Standard) fut homologué comme standard de chiffrement par l'ANSI (American National Standard Institute) pour le secteur privé sous la référence ANSI X3.92 et le nom DEA (Data Encryption Algorithm). La description officielle fut publiée sous la référence FIPS PUB 46 le 15 janvier 1977 par le NBS (National Bureau of Standards, renommé National Institute of Standards and Technology). Il s'agit d'un algorithme de chiffrement symétrique par bloc.

Cette publication décrit mathématiquement l'algorithme bloc pour chiffrer et déchiffrer une information sous une forme binaire. Ces deux opérations reposent sur une clé qui est, elle aussi, sous une forme binaire. La clé est composée de 64 bits dont seulement 56 sont aléatoires et utilisés par l'algorithme. Les 8 autres bits ont été présentés comme des bits servant à la détection d'erreur ce qui diminue au passage l'entropie de la clé de 64 à 56 bits. Celle-ci est découpée en 8 octets qui ont chacun un bit de parité vérifiant une parité impaire, c'est à dire que le nombre total de bits à 1 dans l'octet est un nombre impair. Pour déchiffrer il est nécessaire de posséder la clé de chiffrement. La connaissance de celle-ci permet donc à quiconque de retrouver le clair. Comme le DES est un algorithme bloc à clé secrète, toute la sécurité repose sur le secret de cette clé. Celle-ci doit être échangée de façon sécurisée.

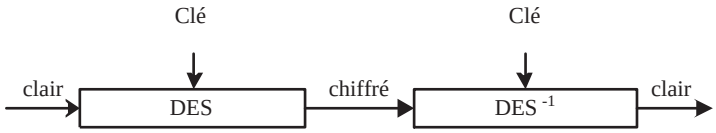


FIG. C.3 – Chiffrement et déchiffrement avec le DES

L'algorithme chiffre et déchiffre des blocs de donnée de 64 bits (clair) en fonction d'une clé de 56 bits. On omettra désormais les 8 bits de parité qui ne servent pas aux calculs. Le déchiffrement utilise la même clé que pour le chiffrement mais avec une séquence différente des bits de clé. Le clair passe tout d'abord dans une permutation initiale IP, puis dans un calcul paramétré par la clé et pour finir dans l'inverse de la permutation initiale IP-1. Le calcul paramétré par la clé est composé d'une boucle de 16 tours. Chaque tour met en œuvre une fonction f de chiffrement et une fonction de séquencement KS de la clé. Dans un premier temps, le chiffrement sera décrit avec la fonction f et les fonctions qui la composent. Les blocs sont découpés en 2 blocs de 32 bits appelés chacun L et R. LR est le bloc complet obtenu par concaténation, c'est à dire les 32 bits de L bits suivis des 32 bits de R. Dans un second temps, le déchiffrement sera décrit par différence avec le chiffrement.

C.3.2 Description de l'algorithme de chiffrement

La figure C.4 décrit l'algorithme de chiffrement. La permutation initiale permute les 64 bits du clair (entrée E) numérotés de 1 à 64. L'algorithme se termine par l'inverse de cette permutation. Par exemple, les bits 58 et 50 du clair deviennent les bits 1 et 2 (sortie S) comme défini dans le tableau C.1.

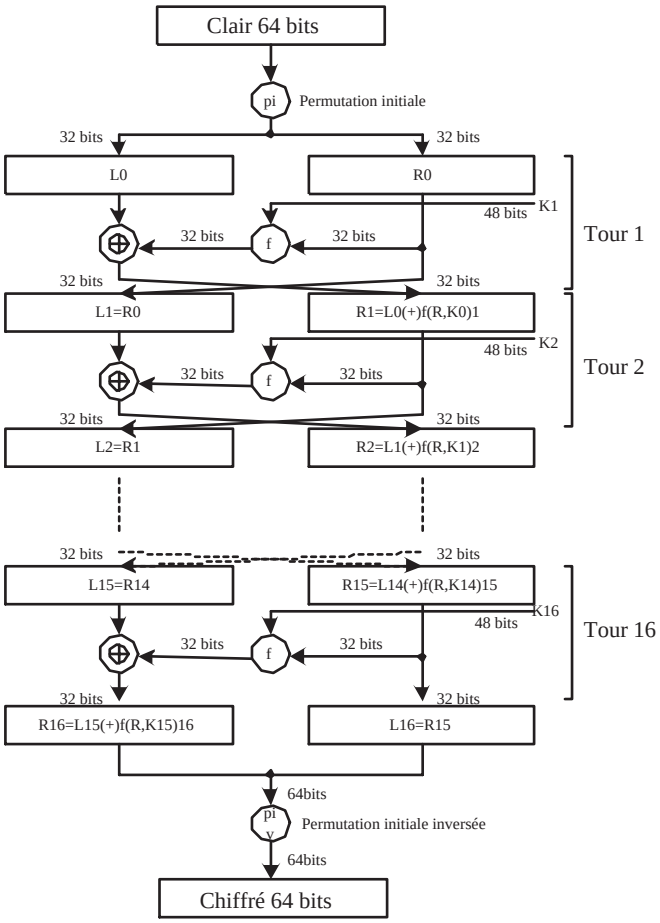


FIG. C.4 – Algorithme de chiffrement DES

Le bloc résultant de la permutation initiale est composé de 64 bits. Il est séparé en deux blocs de 32 bits chacun appelés L pour les bits de 1 à 32 et R pour les bits de 33 à 64. La concaténation de LR représente le bloc en cours. Chaque nouveau bloc R et L est obtenu selon le schéma de Feistel. Un nouveau bloc R est obtenu par un ou exclusif entre une fonction f, qui prend en argument le bloc R du tour d'avant et 48 bits provenant de la clé, et le bloc L du tour précédent. Chaque nouveau bloc L est obtenu à partir du bloc R précédent.

$$\begin{cases} L_0 & \textit{permutation initiale du clair} \\ R_0 & \textit{permutation initiale du clair} \\ L_n & = & R_{n-1} \\ R_n & = & L_{n-1} \oplus f(R_{n-1}, K_n) \end{cases} \tag{C.1}$$

Le bloc R16L16 est l'entrée de la permutation initiale inverse dont la sortie est le résultat chiffré qui est un bloc de 64 bits.

bit de sortie	bit d' entrée	bit de sortie	bit d' entrée	bit de sortie	bit d' entrée	bit de sortie	bit d' entrée
1	58	17	62	33	57	49	61
2	50	18	54	34	49	50	53
3	42	19	46	35	41	51	45
4	34	20	38	36	33	52	37
5	26	21	30	37	25	53	29
6	18	22	22	38	17	54	21
7	10	23	14	39	9	55	13
8	2	24	6	40	1	56	5
9	60	25	64	41	59	57	63
10	52	26	56	42	51	58	55
11	44	27	48	43	43	59	47
12	36	28	40	44	35	60	39
13	28	29	32	45	27	61	31
14	20	30	24	46	19	62	23
15	12	31	16	47	11	63	15
16	4	32	8	48	3	64	7

TAB. C.1 – Permutation initiale

C.3.3 Description de la fonction f

Le détail de la fonction f est donnée par la figure C.5. Le bloc R de 32 bits passe d'abord dans une fonction d'expansion E. Celle-ci étend un bloc de 32 bits en entrée en un bloc de 48 bits en sortie (Tableau C.2). Les 48 bits de sortie peuvent être écrits sous la forme de 8 blocs de 6 bits.

bloc 1		bloc 2		bloc 3		bloc 4	
E(R)	R	E(R)	R	E(R)	R	E(R)	R
1	32	7	4	13	8	19	12
2	1	8	5	14	9	20	13
3	2	9	6	15	10	21	14
4	3	10	7	16	11	22	15
5	4	11	8	17	12	23	16
6	5	12	9	18	13	24	17
bloc 5		bloc 6		bloc 7		bloc 8	
E(R)	R	E(R)	R	E(R)	R	E(R)	R
25	16	31	20	37	24	43	28
26	17	32	21	38	25	44	29
27	18	33	22	39	26	45	30
28	19	34	23	40	27	46	31
29	20	35	24	41	28	47	32
30	21	36	25	42	29	48	1

TAB. C.2 – Fonction E

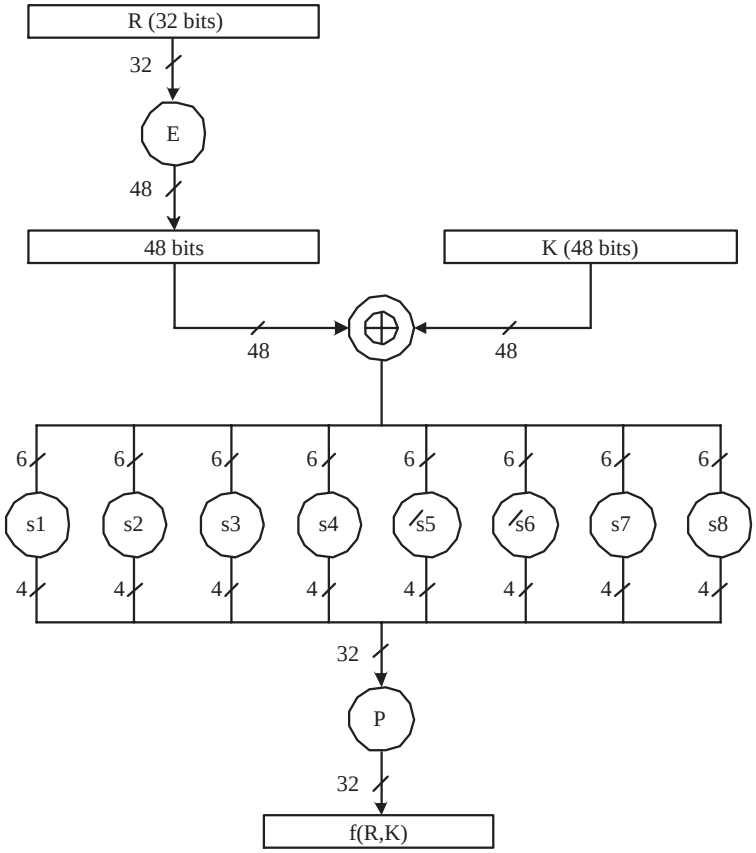


FIG. C.5 – Fonction f

Le résultat de la fonction E(R) est combiné par un ou exclusif bit à bit avec 48 bits de K qui proviennent de la fonction de séquencement KS. Celle-ci est présentée au paragraphe C.3.4.

Les 48 bits résultants sont alors substitués par 8 fonctions S1 à S8. Chaque fonction de substitution Si prend 6 bits en entrée et donne un résultat de 4 bits en sortie. Il n'y a pas de perte d'information parce qu'il faut se rappeler que la fonction d'expansion E précédente a dupliqué les bits de donnée pour passer de 32 à 48. La fonction de substitution S1 est présentée dans le tableau C.3.

S1																
	$b_1\ b_2\ b_3\ b_4$															
$b_0\ b_5$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

TAB. C.3 – Fonction de substitution S1

Les 6 bits en entrée de S_1 sont notés de $b_0\ b_1\ b_2\ b_3\ b_4\ b_5$ et sont interprétés de manière matricielle.

Les bits b_0 et b_5 représentent un nombre de 0 à 3 qui adresse une ligne. Les bits b_1, b_2, b_3 et b_4 représentent un nombre de 0 à 15 qui adresse une colonne. L'intersection entre la ligne et la colonne donne la valeur de substitution. Par exemple, si les 6 bits en entrée sont "011011", la ligne est définie par les bits "01" c'est dire la ligne 1 et la colonne est définie par les bits "1101" c'est à dire la colonne 13. La substitution de "011011" par S_1 a pour résultat les bits "0101". Les autres fonctions de substitution S_2 à S_8 sont construites de la même façon. Elles sont définies dans le standard.

Le résultat des 8 fonctions de substitution donne un bloc de 32 bits. Celui-ci est permuté par la fonction P définie par le tableau C.4. Les bits 16 et 7 passent en position 1 et 2 et les bits 4 et 25 passent en position 31 et 32.

P(S1..S8)	S1..S8	P(S1..S8)	S1..S8	P(S1..S8)	S1..S8	P(S1..S8)	S1..S8
1	16	9	1	17	2	25	19
2	7	10	15	18	8	26	13
3	20	11	23	19	24	27	30
4	21	12	26	20	14	28	6
5	29	13	5	21	32	29	22
6	12	14	18	22	27	30	11
7	28	15	31	23	3	31	4
8	17	16	10	24	9	32	25

TAB. C.4 – Permutation P

Le résultat de cette permutation est le résultat de la fonction f. Il sera combiné par un " xor " bit à bit avec le bloc L correspondant.

C.3.4 Description de la fonction KS

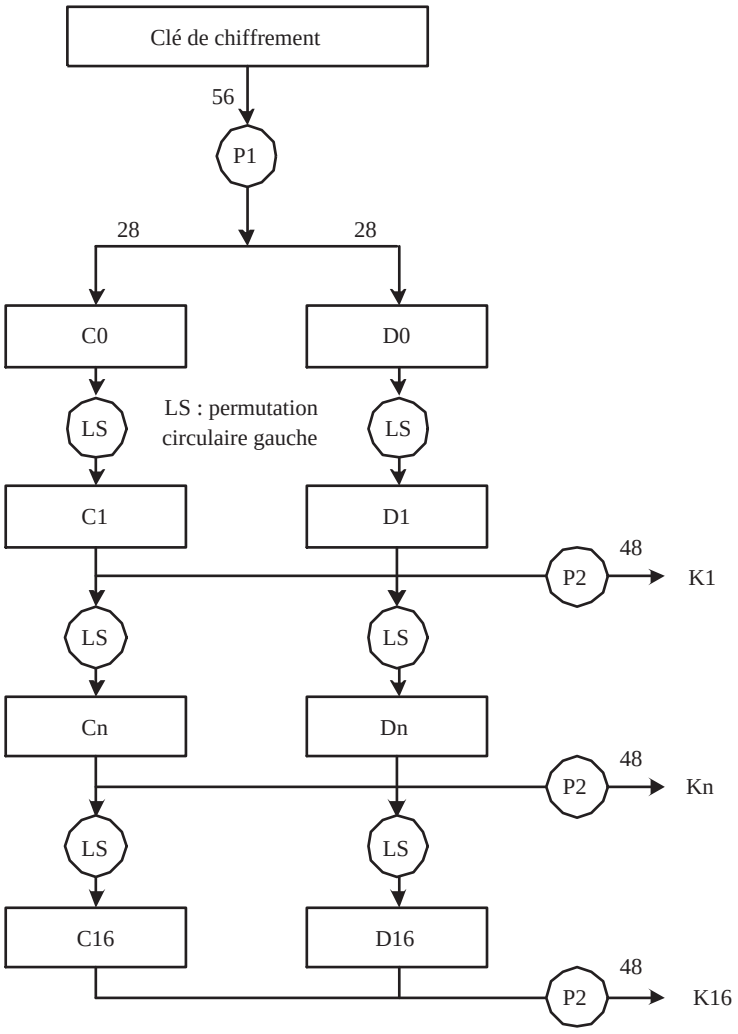


FIG. C.6 – Fonction KS

Le détail de la fonction KS est donné par la figure C.6. Le DES se compose de 16 tours. Chacun de ces tours met en œuvre une sous-clé Kn de 48 bits, n allant de 1 à 16, calculée à partir de la clé de 56 bits. Celle-ci passe tout d'abord dans une permutation P1 définie par le tableau C.5.

Le résultat de la permutation P1 est divisé en 2 blocs de 28 bits respectivement appelés C0 (bits 1 à 28,) et D0 (bits 29 à 56). Les blocs Cn et Dn correspondant au tour n sont obtenus à partir d'une permutation circulaire gauche des blocs Cn-1 et Dn-1. La permutation circulaire gauche décale les bits de Cn-1 et Dn-1 de 1 ou de 2 bits en fonction de n comme indiqué dans le tableau C.6.

Par exemple, C1 et D1 sont obtenus à partir de C0 et D0 en permutant leurs bits de 1 bit à gauche circulairement. C3 et D3 sont obtenus à partir de C2 et D2 en permutant leurs bits de 2 bits à gauche circulairement. On constate qu'au tour 16 les bits ont été décalés de 28 positions

P1(clé)	clé	P1(clé)	clé	P1(clé)	clé	P1(clé)	clé
1	57	15	10	29	63	43	14
2	49	16	2	30	55	44	6
3	41	17	59	31	47	45	61
4	33	18	51	32	39	46	53
5	25	19	43	33	31	47	45
6	17	20	35	34	23	48	37
7	9	21	27	35	15	49	29
8	1	22	19	36	7	50	21
9	58	23	11	37	62	51	13
10	50	24	3	38	54	52	5
11	42	25	60	39	46	53	28
12	34	26	52	40	38	54	20
13	26	27	44	41	30	55	12
14	18	28	36	42	22	56	4

TAB. C.5 – Permutation P1

Tour	Permutation circulaire gauche
n	Nombre de bits décalés
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

TAB. C.6 – Permutation circulaire gauche LS

à gauche et que C16 = C0 et D16 = D0.

Kn est obtenue à partir de Cn et Dn en permutant et en sélectionnant leurs bits par une fonction P2 définie dans le tableau C.7. Par exemple, les bits 14 et 17 du bloc CnDn passent en position 1 et 2 et les bits 29 et 32 passent en position 47 et 48.

P2(CnDn)	CnDn	P2(CnDn)	CnDn	P2(CnDn)	CnDn	P2(CnDn)	CnDn
1	14	13	23	25	41	37	44
2	17	14	19	26	52	38	49
3	11	15	12	27	31	39	39
4	24	16	4	28	37	40	56
5	1	17	26	29	47	41	34
6	5	18	8	30	55	42	53
7	3	19	16	31	30	43	46
8	28	20	7	32	40	44	42
9	15	21	27	33	51	45	50
10	6	22	20	34	45	46	36
11	21	23	13	35	33	47	29
12	10	24	2	36	48	48	32

Tab. C.7 – Permutation P2

C.3.5 Description de l'algorithme de déchiffrement

L'algorithme de déchiffrement est identique à celui de chiffrement, seul le séquencement des clés Ki est modifié. Le chiffré remplace le clair en entrée. La première clé de 48 bits utilisée est K16 et la dernière utilisée K1. Au tour 16, il y a eu 28 permutations circulaires à gauche. Après 16 tours, C16 est égal à C0 et D16 est égal à D0. Pour obtenir ce résultat, il suffit de les permuter circulairement à droite en respectant le décalage inverse du nombre de bits pour déchiffrer.

C.4 L'algorithme AES

C.4.1 Introduction

La description officielle de l'AES (Advanced Encryption Standard) fut publiée sous la référence FIPS PUB 197 [34] le 26 novembre 2001 par le NIST (National Institute of Standards and Technology) afin de remplacer l'algorithme DES. Il est le résultat d'une compétition internationale qui choisit l'algorithme Rijndael ([11] et [12]) comme remplaçant du DES.

Cette publication décrit l'algorithme bloc pour chiffrer et déchiffrer une information binaire par blocs de 128 bits. Ces deux opérations reposent sur une clé qui est, elle aussi, sous une forme binaire. La clé est composée de 128, 192 ou 256 bits. Pour déchiffrer il est nécessaire de posséder la clé de chiffrement. La connaissance de cette clé permet donc à tout le monde de déchiffrer. Comme pour le DES, l'AES est un algorithme bloc à clé secrète (Figure C.7), toute la sécurité repose sur le secret de cette clé. Celle-ci doit être échangée de façon sécurisée.

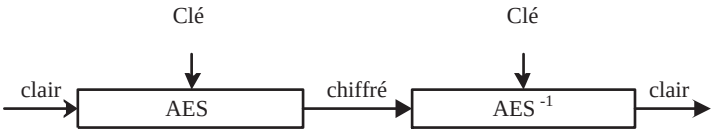


FIG. C.7 – Chiffrement et déchiffrement avec l'AES

C.4.2 Description de l'algorithme de chiffrement

Avant de passer à la description de chaque fonction, il est nécessaire d'introduire une notation. Le clair, les messages intermédiaires et le chiffré sont des blocs de 128 bits. Ils sont découpés en 16 octets et représentés sous la forme matricielle suivante appelée état (Figure C.8).

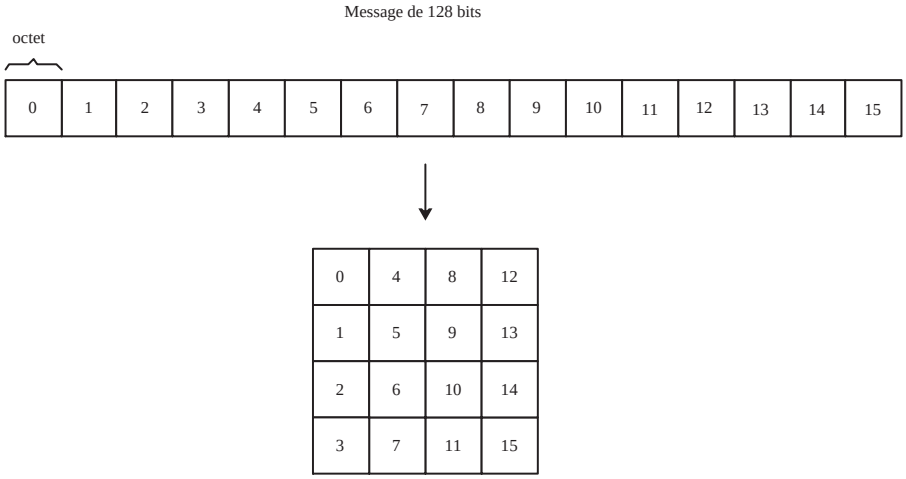


FIG. C.8 – Notation des messages clair, intermédiaire et chiffré

La figure C.9 décrit l'algorithme de chiffrement. Lorsque la clé est de longueur 128 bits le nombre

de tours est de 10. Lorsque celle-ci est de longueur 192 bits le nombre de tours est de 12 et lorsque celle-ci est de longueur 256 bits le nombre de tours est de 14. Le clair est tout d'abord combiné bit à bit par un " xor " avec la première sous-clé grâce à la fonction AddRoundKey. Puis le chiffrement est réalisé par un calcul paramétré par la clé. Le calcul paramétré par la clé est composé de 10, 12 ou 14 tours selon la longueur de la clé. Chaque tour intermédiaire met en œuvre de façon séquentielle les fonction suivantes :

– la fonction SubBytes de substitution d'octet,

– la fonction ShiftRows de permutation circulaire des octets de chaque ligne,

– la fonction MixColumns pour un calcul matriciel sur chaque colonne,

– la fonction AddRoundKey pour combiner bit à bit la sous-clé et le message,

– la fonction de séquencement Key Expansion de la clé pour générer chaque sous-clé.

Le dernier tour est quasi identique à un tour intermédiaire hormis le fait que la fonction MixColumns est supprimée.

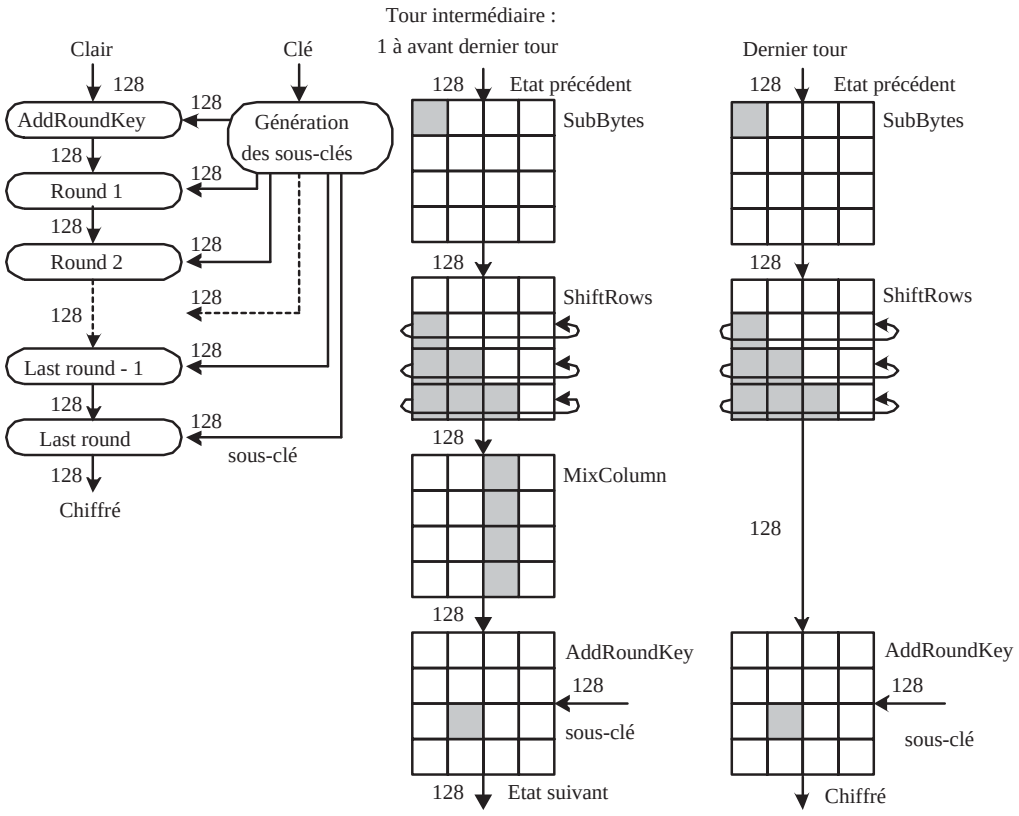


Fig. C.9 – Algorithme de chiffrement AES

C.4.3 Description de la fonction SubBytes

La fonction SubBytes (Figure C.10) remplace un octet par un autre octet défini par le standard.

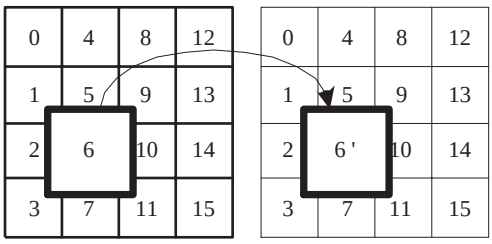


FIG. C.10 – Fonction SubBytes

C.4.4 Description de la fonction ShiftRows

La fonction ShiftRows réalise une permutation circulaire des lignes de l'état (Figure C.11).

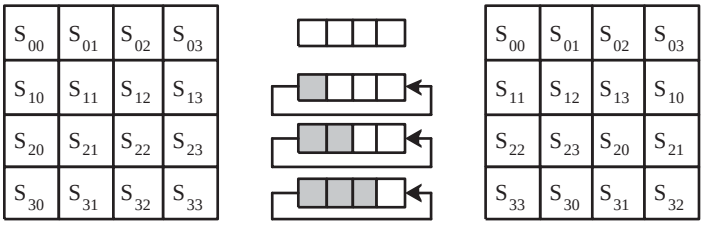


FIG. C.11 – Fonction ShiftRows

C.4.5 Description de la fonction MixColumns

La fonction MixColumns réalise une multiplication matricielle entre une matrice définie par le standard et une colonne de l'état (Figure C.12).

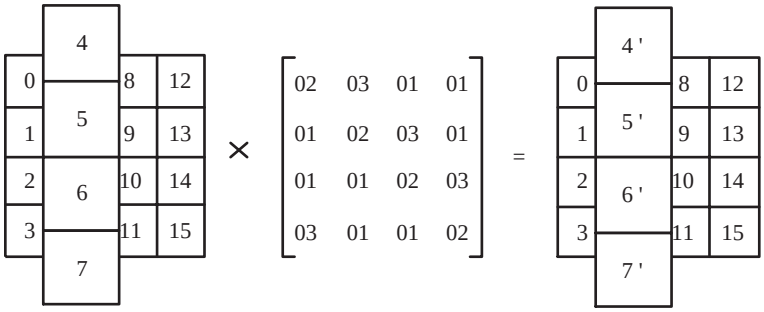


FIG. C.12 – Fonction MixColumns

C.4.6 Description de la fonction AddRoundKey

La fonction AddRoundKey réalise une combinaison bit à bit entre la sous clé et le message par un " xor ". La sous clé a une représentation en mots de 32 bits wi (Figure C.13).

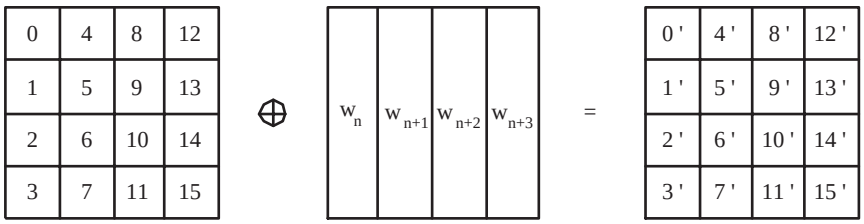


FIG. C.13 – Fonction AddRoundKey

C.4.7 Description de la fonction Key Expansion

La fonction Key Expansion génère les sous-clés pour chaque tour de l'AES (Figure C.14).

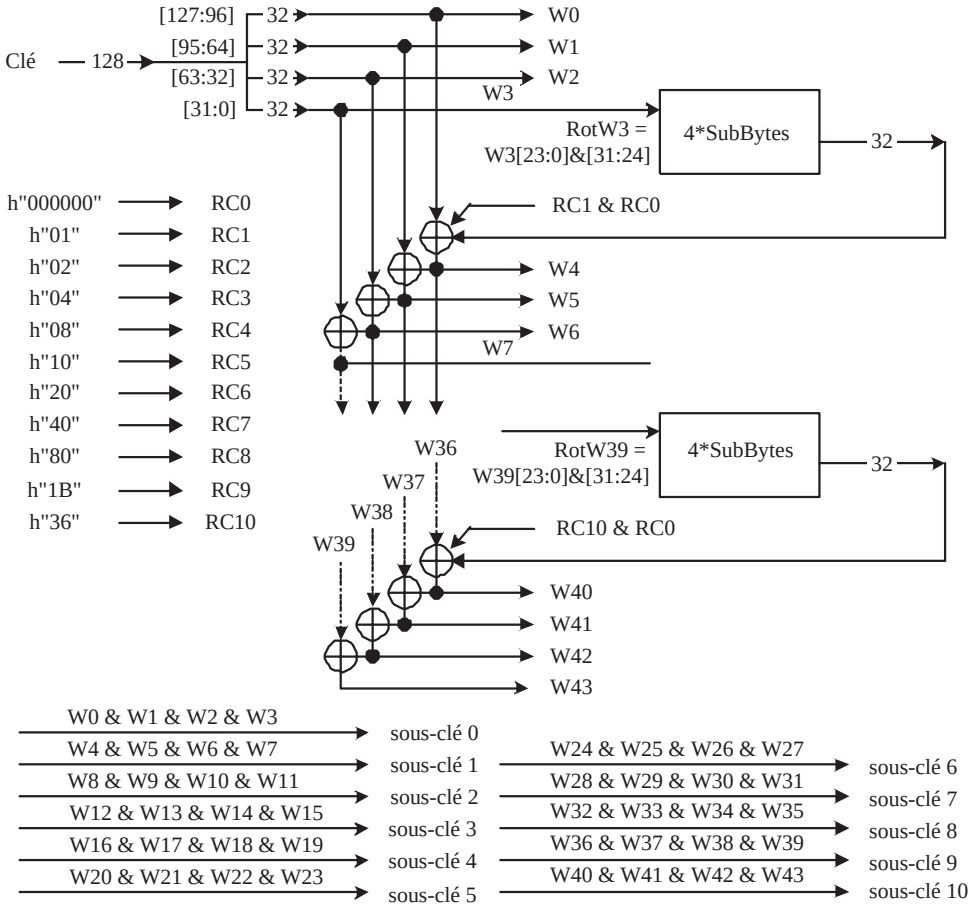


FIG. C.14 – Fonction Key Expansion

C.4.8 Description de l'algorithme de déchiffrement

L'algorithme de déchiffrement a la même structure qu'en chiffrement. Les fonctions sont modifiées en InvSubBytes, InvShiftRows et InvMixColumns.

La fonction InvSubBytes (Figure C.15) remplace un octet par un octet de l'état.

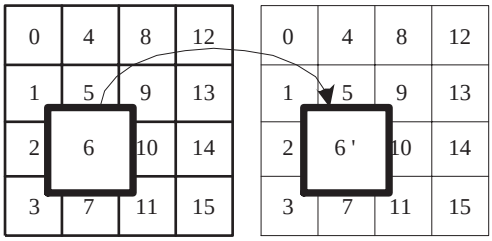


FIG. C.15 – Fonction InvSubBytes

La fonction InvShiftRows réalise la permutation circulaire des lignes de l'état (Figure C.16).

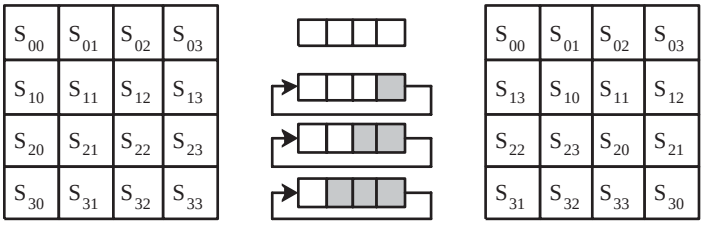


FIG. C.16 – Fonction InvShiftRows

La fonction MixColumns réalise une multiplication matricielle (Figure C.17).

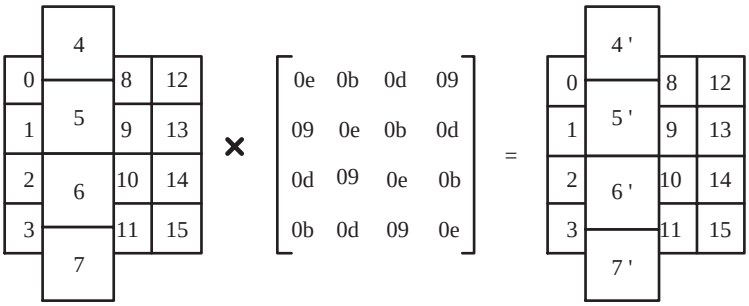


FIG. C.17 – Fonction InvMixColumns

Bibliographie

[1] M. Akkar and C. Giraud. An Implementation of DES and AES Secure Against Some Attacks. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-42521-7. Springer-Verlag, 2001.

[2] M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart. Power Analysis, What Is Now Possible ... In T. Okamoto, editor, *Advances in Cryptology – Asiacrypt '00*, volume 1976 of *Lectures Notes in Computer Science*, pages 489 – 502. Springer-Verlag, 2000.

[3] J. Blömer, J. Merchan, and V. Krummel. Provably Secure Masking of AES. Selected Areas in Cryptography, 2004.

[4] F. Bouesse, F. Germain, and M. Renaudin. Asynchronous AES Crypto-Processor Including Secured and Optimized Blocks. In *Journal of Integrated Circuits and Systems*, volume 1 of *ISSN 1807-1953*, pages 5 – 13, 2004.

[5] F. Bouesse, M. Renaudin, S. Dumont, and F. Germain. Formalizing and Improving DPA resistance of Quasi Delay Insensitive Asynchronous Circuits. Design, Automation and Test in Europe, 2005.

[6] F. Bouesse, M. Renaudin, A. Witon, and F. Germain. A Clock-less Low-voltage AES Crypto-processor. In *European Solid-Sstate Circuits Conference*, 2005.

[7] F. Bouesse et F. Germain et M. Renaudin. Conception d'un circuit AES en technologie asynchrone - Cahier des charges version 2. DCSSI-TIMA, 2003.

[8] F. Bouesse et F. Germain et M. Renaudin. Conception d'un circuit AES en technologie asynchrone - Rapport de conception. DCSSI-TIMA, 2003.

[9] F. Bouesse et F. Germain et M. Renaudin. Conception d'un circuit AES en technologie asynchrone - Spécification du composant AES et du banc de test version 2. DCSSI-TIMA, 2003.

[10] D. Buzon et B. Robisson. Attaque par canaux auxilliaires sur AES asynchrone - Résultats expérimentaux. CEA LETI-DCSSI, 2005.

[11] J. Daemon and V. Rijmen. AES proposal : Rijndael. AES Algorithm Submission, September 3, 1999.

[12] J. Daemon and V. Rijmen. The bloc cipher Rijndael. In Springer Verlag, editor, *Smart Card research and Applications*, LNCS 1820, pages 288–296.

[13] Rémy Daudigny, Hervé Ledig, Frédéric Muller, and Frédéric Valette. Scare of the DES. In *ACNS*, pages 393–406, 2005.

[14] D.Suzuki, M.Saeki, and T.Ichikawa. DPA Leakage Models for CMOS Logic Circuits. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-28474-5. Springer-Verlag, 2005.

[15] F. Germain et L. Duflot et P. Le Moigne. Dispositif formant porte logique adaptée pour minimiser les différences de comportement électrique ou électromagnétique dans un circuit intégré manipulant un secret. Demande de brevet n° 0504569. État français représenté par le secrétariat général de la défense nationale, 2005.

[16] J. Golic and C. Tymen. Multiplicative Masking and Power Analysis of AES. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-00409-2. Springer-Verlag, 2002.

[17] M. Gomułkiewicz and M. Kutyłowski. Hamming Weight Attacks on Cryptographic Hardware - Breaking Masking Defense. In *European Sympsosium on Research in Computer Security*, LNCS 2502. Springer-Verlag, 2002.

[18] S. Guillet, P. Hoogvorst, Y .Mathieu, and R .Pacalet. A Leakage-Proof Place-and-Route Strategy for ASICs. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-28474-5. Springer-Verlag, 2005.

[19] S. Guillet, P. Hoogvorst, Y .Mathieu, R .Pacalet, and J .Provost. CMOS Structures Suitable for Secured Hardware. Design, Automation and Test in Europe, 2004.

[20] J.Fournier, H.Li, S.Moore, R.Mullins, and G.Taylor. Security Evaluation of Asynchronous Circuits. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-40833-9. Springer-Verlag, 2003.

[21] J.Sparsø and S.Furber. *Principles of Asynchronous Circuit Design.* Kluwer Academic, 2001.

[22] K.J.Kulikowski, M.Su, A.Smirnov, A.Taubin, M.G.Karpovsky, and D.MacDonald. Delay Insensitive Encoding and Power Analysis : A Balancing Act. ASYNC, 2005.

[23] P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Others Systems. In N. Koblitz, editor, *Advances in Cryptology – Crypto '96*, volume 1109 of *LNCS*, pages 104 – 113. Springer-Verlag, 1996.

[24] P. C. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. http ://www.cryptography.com/dpa/technical, 1998.

[25] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology – Crypto '99*, volume 1666 of *LNCS*, pages 388 – 397. Springer-Verlag, 1999.

[26] K.Tiri, M.Akmal, and I.Verbauwhede. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withdraw Differential Power Analysis on Smart Cards. In *European Solid-Sstate Circuits Conference*, 2002.

[27] H. Kuo and I. Verbauwhede. Architecture Optimization for a 1.82 gbits/sec VLSI Implementation of the AES Rijndael Algorithm. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-42521-7. Springer-Verlag, 2001.

[28] H. Ledig, F. Muller, and F. Valette. Enhancing Collision Attacks. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-22666-4. Springer-Verlag, 2004.

[29] T. Messerges. Using a Second Order Power Analysis to Attack DPA Resistant Software. In *Cryptographic Hardware and Embedded Systems*, 2000.

[30] S. Morioka and A. Satoh. An Optimized SBOX Architecture for Low Power AES Design. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-00409-2. Springer-Verlag, 2002.

[31] N.H.E.Weste and K.Eshraghian. *Principles of CMOS VLSI DESIGN A Systems Perspective.* Addison-Wesley, 1994.

[32] NIST. An Optimized S-box Circuit Architecture for low Power Aes Design. Available at `http ://www.csrc.nist.gov/CryptoToolkit/aes/`.

[33] NIST. FIPS PUB 46 - Data Encryption Standard (DES), 1977.

[34] NIST. FIPS PUB 197 - Advanced Encryption Standard (AES), November 2001.

[35] C. Raynaud et J-C. Courrege. Attaque par canaux cachés sur un circuit AES sécurisé en technologie asynchrone - Théorie. THALES SECURITY SYSTEMS CEACI-DCSSI, 2004.

[36] C. Raynaud et J-C. Courrege. Attaque par canaux cachés sur un circuit AES sécurisé en technologie asynchrone - Résultats expérimentaux. THALES SECURITY SYSTEMS CEACI-DCSSI, 2005.

[37] V. Rijmen. Efficient Implementation of the Rijndael Sbox. Available at `http ://www.esat.ku-leuven.ac.be/ rijmen/rijndael/`.

[38] R.J.Baker. *CMOS Circuit Design, Layout, and Simulation.* IEEE Press Series on Microelectronic Systems.

[39] B. Robisson. Attaque par canaux auxilliaires sur AES asynchrone - Théorie. CEA LETI-DCSSI, 2004.

[40] K. Schramm, G. Leander, P. Felke, and C. Paar. A Collision-Attack on AES (Combining Side Channel and Differential-Attack.

[41] D. Shang, F. Burns, A. Bystrov, A. Kœlmans, D. Sokolov, and A. Yakovlev. A Low and Balanced Power Implementation of the AES Security Mechanism Using Self Timed Circuits. Power and Timing Modeling, Optimization and Simultation, 2004.

[42] S.Moore, R.Anderson, R.Mullins, G.Taylor, and J.Fournier. Balanced Self-Checking Asynchronous Logic for Smart Cards. In *Microprocessors and Microsystems Journal*, 2003.

[43] D. Sokolov, J. Murphy, A. Bystrov, and A. Yakovlev. Improving the Security of Dual Rail Circuits. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-22666-4. Springer-Verlag, 2004.

[44] K. Tiri, D.Hwang, A.Hodjat, B.Lai, S.Yang, P.Schaumont, and I. Verbauwhede. Prototype IC with WDDL and Differential Routing - DPA Resistant Assessment. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-28474-5. Springer-Verlag, 2005.

[45] K. Tiri and I. Verbauwhede. Securing Encryption Algorithms Against DPA at the Logic Level : Next Generation Smart Cards Technology. In *Cryptographic Hardware and Embedded Systems*, ISBN 3-540-40833-9. Springer-Verlag, 2003.

[46] K. Tiri and I. Verbauwhede. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. Design, Automation and Test in Europe, 2004.

[47] T. Verhœff. Delay-Insensitive Codes - An Overview. In *Distributed Computing*, 1988.

[48] J. Wolkerstorfer, E. Oswald, and M. Lamberger. An ASIC Implementation of the AES Sboxes. In *Proceedings of the Cryptographer's Track at the RSA Conference 2002*, LNCS 2271. Springer-Verlag, 2002.

[49] Z.C.Yu, S.Furber, and L.A.Plana. An Investigation into the Security of Self-timed Circuits. ASYNC, pages 206–215, 2003.

Résumé

L'analyse différentielle de consommation (notée DPA pour Differential Power Analysis) est une puissante attaque non intrusive par canal auxilliaire dont l'objectif est de retrouver des informations secrètes contenues dans des circuits intégrés en exploitant la consommation globale. Des clés de chiffrement peuvent alors être découvertes pendant l'exécution d'algorithmes cryptographiques. L'objet de cette thèse est de proposer une contre-mesure véritablement efficace basée sur la conception de portes logiques intrinsèquement résistantes à la DPA indépendamment des états logiques et électriques passés, présents et futurs. Il est alors théoriquement possible de concevoir des circuits intégrés résistants à l'attaque DPA. La contre-mesure proposée repose sur des bases microélectroniques précises qui permettent d'expliciter les sources de la DPA. La solution s'appuie sur la conception CMOS (Complementary Metal Oxide Silicon) de circuits intégrés réalisant des algorithmes cryptographiques tels que l'AES (Advanced Encryption Standard).

Mots-clés: DPA, sources microélectroniques de la DPA, dispersion instantanée des caractéristiques électriques, contre-mesure DPA.

Abstract

Differential Power Analysis (DPA) is a powerful side channel attack aiming at recovering secret information embedded into integrated circuits by monitoring overall power consumption. For instance, cipher keys can be recovered during cryptographic computations. The purpose of this these is to propose an efficient countermeasure based on the design of intrinsic DPA-proof gates with logic state-independent electrical consumption. With such a gates, designing a DPA-proof integrated circuit is theoretically possible. The proposed counter-measure is examined under precise microelectronics based considerations to explain the microelectronic sources of DPA. The proposed solution focuses on Complementary Metal Oxide Silicon (CMOS) integrated circuits implementing cryptographic algorithms such as the Advanced Encryption Standard (AES).

Keywords: DPA, microelectronic sources of DPA, instantaneous scattering of electrical characteristics, DPA counter-measure.