



HAL
open science

Analyse et amélioration de la robustesse des circuits asynchrones QDI

Florent Ouchet

► **To cite this version:**

Florent Ouchet. Analyse et amélioration de la robustesse des circuits asynchrones QDI. Autre. Université de Grenoble, 2011. Français. NNT : 2011GRENT062 . tel-00680544

HAL Id: tel-00680544

<https://theses.hal.science/tel-00680544v1>

Submitted on 19 Mar 2012

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

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

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Micro et Nano Electronique (MNE)**

Arrêté ministériel :

Présentée par

Florent Ouchet

Thèse dirigée par **Laurent Fesquet**
et codirigée par **Katell Morin-Allory**

préparée au sein du laboratoire **TIMA**
et de l'École Doctorale **Électronique, Électrotechnique, Automatique et de Traitement du Signal**

Analyse et amélioration de la robustesse des circuits asynchrones QDI

Thèse soutenue publiquement le **2 décembre 2011**,
devant le jury composé de :

Lorena Anghel

Professeur des universités, Grenoble-INP, Président

Sylvie Renaud

Professeur des universités, Institut Polytechnique de Bordeaux, Rapporteur

Michel Robert

Professeur des universités, Université de Montpellier 2, Rapporteur

Yves Leduc

Professeur, Chaire TI, Université de Nice Sophia Antipolis, Examineur

Laurent Fesquet

Maître de conférences Grenoble-INP, Directeur de thèse

Katell Morin-Allory

Maître de conférences Grenoble-INP, Co-Directrice de thèse



Sommaire

| | |
|--------------------------------------------------------------------------|------------|
| Glossaire | xii |
| 1 Introduction | 1 |
| 2 Généralités sur les circuits asynchrones | 5 |
| 2.1 Introduction aux circuits asynchrones | 6 |
| 2.1.1 Limitations des circuits synchrones | 6 |
| 2.1.2 Principe des circuits asynchrones | 7 |
| 2.2 Les échanges entre blocs fonctionnels | 8 |
| 2.3 Classement des circuits asynchrones | 9 |
| 2.3.1 Circuits de Huffman | 9 |
| 2.3.2 Circuits micropipelines | 9 |
| 2.3.3 Circuits indépendant de la vitesse "SI" | 10 |
| 2.3.4 Circuits quasi-insensibles aux délais "QDI" | 10 |
| 2.3.5 Circuits insensibles aux délais "DI" | 11 |
| 2.4 Codage données groupées | 11 |
| 2.4.1 Codage 4 états | 12 |
| 2.4.2 Codage 3 états, 1 parmi 2 | 12 |
| 2.4.3 Codage 1-parmi-N | 13 |
| 2.5 Protocoles d'échanges | 13 |
| 2.5.1 Protocole à 2 phases | 14 |
| 2.5.2 Protocole à 4 phases | 15 |
| 2.6 C-elements | 15 |
| 2.7 Synchronisation entre entrées et sorties d'un bloc logique | 17 |

| | | |
|----------|------------------------------------------------------------|-----------|
| 2.7.1 | Protocoles séquentiels | 19 |
| 2.7.2 | Protocole WCHB | 19 |
| 2.7.3 | Protocole PCHB | 21 |
| 2.7.4 | Protocole PCFB | 23 |
| 2.7.5 | Protocole FDFB | 24 |
| 2.7.6 | Mise en série de buffers | 26 |
| 2.8 | Aléas de commutation | 27 |
| 2.8.1 | Conception logique sans aléa | 28 |
| 2.9 | Fonctionnement des circuits asynchrones | 29 |
| 3 | Structure des portes C-elements | 33 |
| | Introduction | 34 |
| 3.1 | Implémentations full-custom | 34 |
| 3.1.1 | Structures habituelles | 35 |
| 3.1.2 | Autres structures | 37 |
| 3.2 | Implémentations structurelles | 38 |
| 3.2.1 | A base de portes logiques | 38 |
| 3.2.2 | Sur composants programmables | 39 |
| 3.3 | Implémentation de C-elements à n entrées | 39 |
| 3.3.1 | Décomposition avec les portes d'une bibliothèque | 40 |
| 3.3.2 | Rationalisation des cellules spécifiques | 41 |
| 3.4 | Fonctions apparentées au C-elements | 42 |
| 3.4.1 | Implémentation en logique différentielle | 42 |
| 3.4.2 | Fonction C-element généralisée | 43 |
| | Conclusion | 44 |
| 4 | Phénomène de création de jetons | 47 |
| | Introduction | 48 |
| 4.1 | Niveaux d'abstractions | 48 |
| 4.1.1 | Modélisations numériques | 48 |
| 4.1.2 | Modélisations analogiques | 49 |
| 4.1.3 | Modélisations des circuits asynchrones | 50 |
| 4.2 | Simulations sur fronts lents | 50 |
| 4.2.1 | Cause des pentes lentes | 50 |

| | | |
|----------|----------------------------------------------------------------|-----------|
| 4.2.1.1 | Faible tension d'alimentation | 51 |
| 4.2.1.2 | Influence de l'environnement | 51 |
| 4.2.1.3 | Causes liées à la conception | 52 |
| 4.2.1.4 | Causes liées à la fabrication | 54 |
| 4.2.1.5 | Causes liées au vieillissement | 54 |
| 4.2.2 | Dispositif expérimental complet | 54 |
| 4.2.3 | Dispositif expérimental simplifié | 55 |
| 4.2.4 | Résultats des simulations | 56 |
| 4.3 | Phénomène inverse | 59 |
| 4.4 | Conséquences sur le circuit | 59 |
| 4.4.1 | Erreurs de phases | 60 |
| 4.4.2 | Absorption de jetons | 62 |
| 4.4.3 | Création de jetons | 62 |
| | Conclusion | 62 |
| 5 | Seuils et sensibilité à la pente | 65 |
| | Introduction | 66 |
| 5.1 | Modèles numériques des seuils de commutations | 66 |
| 5.1.1 | Modèle di-atomique | 66 |
| 5.1.2 | Modèles à temps non-bornés | 68 |
| 5.2 | Niveau analogique | 69 |
| 5.2.1 | Logique combinatoire | 69 |
| 5.2.2 | Logique séquentielle | 70 |
| 5.3 | Vérification des bibliothèques | 70 |
| 5.3.1 | Protocoles de mesure | 70 |
| 5.3.2 | Précision de la mesure | 70 |
| 5.3.3 | Portes des bibliothèques TIMA | 71 |
| 5.3.4 | Autres structures de C-elements | 73 |
| 5.4 | Correction des seuils des portes | 74 |
| 5.4.1 | Redimensionnement des transistors | 74 |
| 5.4.1.1 | Structure <i>Weak-Feedback</i> | 74 |
| 5.4.1.2 | Structure conventionnelle | 76 |
| 5.4.1.3 | Structure Symétrique | 76 |
| 5.4.1.4 | Conclusions sur le redimensionnement des transistors | 77 |

| | | |
|----------|----------------------------------------------------------------|-----------|
| 5.4.2 | Réduction de la tension d'alimentation | 79 |
| 5.4.2.1 | Polarisation des transistors | 79 |
| 5.4.2.2 | Structure <i>Weak-Feedback</i> | 80 |
| 5.4.2.3 | Structures Conventiennelle et symétrique | 81 |
| 5.4.2.4 | Structure dynamique et structures composites | 81 |
| 5.4.3 | Variations simultanées | 82 |
| 5.4.3.1 | Structure <i>Weak-Feedback</i> | 82 |
| 5.4.3.2 | Structure Conventiennelle | 82 |
| 5.4.3.3 | Structure Symétrique | 83 |
| | Conclusion | 84 |
| 6 | Prise en compte des variations de process | 87 |
| | Introduction | 88 |
| 6.1 | Influence technologique | 88 |
| 6.1.1 | Variations des paramètres | 88 |
| 6.1.2 | Disparité des paramètres | 89 |
| 6.2 | Analyses des variations | 89 |
| 6.2.1 | Analyses statistiques | 89 |
| 6.2.1.1 | Indépendance des seuils | 90 |
| 6.2.2 | Marges de conception | 91 |
| 6.2.2.1 | Marges pour des variables aléatoires indépendantes | 91 |
| 6.2.2.2 | Prise en compte des corrélations | 92 |
| 6.3 | Résultats obtenus | 92 |
| 6.3.1 | Redimensionnement de l'étage de maintien | 92 |
| 6.3.2 | Réduction de la tension d'alimentation | 93 |
| 6.3.3 | Autres phénomènes influençant les tensions de seuils | 93 |
| | Conclusion | 95 |
| 7 | Formalisation de la vérification | 99 |
| | Introduction | 100 |
| 7.1 | Simulations numériques de circuits électroniques | 100 |
| 7.2 | Vérifications des circuits électroniques | 101 |
| 7.2.1 | Vérifications semi-formelles | 102 |
| 7.2.2 | Vérifications formelles | 102 |

| | | |
|----------|--------------------------------------------------------------------|------------|
| 7.2.3 | Limites des vérifications formelles analogiques | 102 |
| 7.3 | Coho | 103 |
| 7.3.1 | Principes des simulations avec intervalles | 104 |
| 7.3.2 | Structure du programme | 106 |
| 7.3.3 | Spécifications du circuit | 106 |
| 7.3.4 | Spécifications des entrées | 107 |
| 7.3.5 | Graphe de simulation | 108 |
| 7.4 | Application aux circuits asynchrones | 109 |
| 7.4.1 | Structure <i>Weak-Feedback</i> | 110 |
| 7.4.1.1 | Analyse des <i>Brockett Annuli</i> | 110 |
| 7.4.1.2 | Analyse des seuils de basculements | 112 |
| 7.4.2 | Structure Conventionnelle | 115 |
| 7.4.2.1 | Analyse de <i>Brockett Annuli</i> | 115 |
| 7.4.2.2 | Analyse des seuils de commutation | 116 |
| | Conclusion | 118 |
| 8 | Conclusion | 121 |
| A | CRC16 asynchrone | 125 |
| A.1 | Sous-composants du CRC16 | 126 |
| A.2 | Schematic principal | 130 |
| B | Fichiers de configuration de Coho | 133 |
| B.1 | Spécifications du circuit et des entrées d'une porte WM2 | 133 |
| B.2 | Spécifications de l'automate de simulation | 134 |
| B.3 | Spécifications du circuit et des entrées d'une porte CM2 | 136 |

Glossaire

Dans la suite de ce manuscrit, les mots suivis d'un asterisk (*) font l'objet d'une définition dans ce glossaire.

acquiescement indication de prise en compte de la [requête*](#) par le bloc logique, transmise au bloc ayant initié cette requête.

aléa phénomène transitoire généré lors des transitions des blocs combinatoires.

benchmark environnement de simulation analysant le comportement d'un circuit en stimulant ses entrées.

Brockett Annulus spécification des transitions d'un signal d'après des intervalles de niveaux électriques et de vitesse (bornes sur la dérivée) de transition.

bulle Encodage séparant deux [jetons de données*](#).

C-element point mémoire réalisant la synchronisation entre des signaux fortement désynchronisés : un tel point mémoire peut réaliser un rendez-vous entre les signaux provenant de deux blocs logiques asynchrones.

causal relation entre les entrées et les sorties d'un bloc logique réel : les sorties ne peuvent anticiper une variation des entrées.

circuit asynchrone circuit électronique où les données sont traitées quand elles sont effectivement prêtes.

circuit de Huffman famille de circuits à base de portes combinatoires rebouclées. La fonction séquentielle est réalisée grâce à l'implémentation de verrous *ad-hoc*.

circuit indépendant de la vitesse famille de circuits fonctionnant quel que soient les délais dans les portes logiques, les délais dans les interconnexions sont nulles.

circuit insensible aux délais famille de circuits fonctionnant quel que soient les délais dans les portes logiques et les interconnexions.

circuit micropipeline famille de circuits où les données sont automatiquement propagées dans un pipeline de type [flot de données*](#).

circuit quasiment insensible aux délais famille de circuits fonctionnant quel que soient les délais dans les portes logiques et les interconnexions, modulo l'hypothèse des [fourches isochrones*](#).

circuit synchrone circuit électronique où toutes les données sont cadencées par un signal de synchronisation global : l'horloge.

Dynamic Frequency Scheduling (DFS) variations dynamique de la fréquence de l'horloge d'un circuit synchrone en fonction des traitements à effectuer, principalement pour réduire sa consommation dynamique.

Dynamic Voltage Scheduling (DVS) variations dynamique de la tension d'alimentation d'un circuit synchrone en fonction des traitements à effectuer, principalement pour réduire sa consommation statique.

Early Threshold prise en compte des changements d'état dès le début des transitions.

effet de substrat Modification des caractéristiques d'un transistor MOS en fonction des potentiels de ses électrodes de source et de substrat.

effet *Early* Variation de la longueur effective du canal d'un transistor MOS à cause de sa tension V_{DS} .

flot de données structure de circuit directement issue de la description du circuit : les différentes opérations successives appliquées à une même donnée donnent lieu à autant d'étages dans un pipeline de traitement.

fourche isochrone interconnexion reliant un émetteur à plusieurs récepteur où les délais de propagation vers les différents récepteurs doivent être identiques.

Gate-Level niveau d'abstraction où le circuit est décrit comme un ensemble de *standard-cells** inter-connectées.

golden model modèle dont le comportement correspond parfaitement et dans tous les cas au comportement souhaité du circuit.

implication relation mathématique de la forme $A \Rightarrow B$ où A et B sont des relations booléennes. Lorsque la relation A est vraie, alors la relation B doit être vraie elle aussi..

Intellectual Property Partie d'un circuit électronique développée par un tiers et intégrée tel-quelle dans l'assemblage du circuit complet. Les IP peuvent être des descriptions comportementales (on parle alors de *soft IP*) ou des parties de *layout** déjà placées-routées (*hard IP*).

jeton de donnée ensemble des requêtes et des valeurs correspondant à un même cycle de fonctionnement se propageant dans le circuit.

keeper Étage de maintien de l'état d'un point mémoire statique.

layout dessin des masques utilisés pour la fabrication du circuit intégré.

liveness absence de blocage fatal dans un circuit.

mismatch déséquilibre de appariement de plusieurs composants liées à des variations indépendantes de leurs caractéristiques.

Model Checking Technique de vérification comparant un modèle à ses spécifications.

modem module traduisant des données analogiques ou numériques pour les envoyer ou les recevoir vers un canal de communication.

netlist modèle du circuit sous la forme de composants inter-connectés.

Network On Chip (NOC) réseau de communications numériques permettant à divers modules d'un même circuit intégré de communiquer entre eux.

Non-Retour-à-Zéro NRZ - encodage de données où les données sont transmises les unes à la suite des autres sans que les transitions intermédiaires n'aient de signification.

n -uplet (ou *tuple* en anglais) extension de la notion de couple de valeurs à l'ensemble de n composantes. Par exemple, un nombre complexe est un 2-uplet comportant une composante réelle et une composante imaginaire, les coordonnées d'un point dans un espace tridimensionnel forment un 3-uplet.

one-hot encodage de données où seul un fil parmi n peut être actif à un instant donné.

parallélotope droit extension des rectangles (dimension 2) et des parallélépipèdes rectangles (dimension 3) à n dimensions.

poignée de main échange fortement synchronisé entre deux blocs logiques basé sur des signaux de requête et d'acquittement.

Process-Voltage-Temperature (PVT) phénomènes couramment décrits dans la littérature comme modifiant substantiellement le comportement d'un circuit intégré.

pull type de canal de communication où les données sont associées à l'acquittement : elles sont "tirées" du bloc activé vers le bloc initiant la requête quand les traitements sont terminés.

push type de canal de communication où les données sont associées à la requête : elles sont "poussées" par le bloc initiant un transfert vers le bloc activé par cette requête.

push-pull type de canal de communication où les données transitent dans les deux sens. Elles sont associées à la requête et à l'acquittement.

record type composite, comprenant plusieurs champs stockant des données distinctes.

requête demande de traitement en entrée d'un bloc logique.

reset action de mise à 0 d'un signal logique.

Retour-à-Zéro RZ - encodage de données où un code spécial de remise à zéro est toujours transmis entre deux données.

set action de mise à 1 d'un signal logique.

simulation Monte-Carlo (micro-électronique) Ensemble des simulations d'un même dispositif modélisant statistiquement les variations possibles lors de la fabrication.

skew (*clock skew*), décalage temporel dans l'acheminement de l'horloge qui introduit des mémorisations décalées dans le temps avec potentiellement des données non stabilisées ou en avance de phase.

SPICE Moteur de simulation temporelle et analogique basé sur la résolution de modèles linéaires approchés du circuit.

standard-cell portes réalisant une fonction logique et optimisées pour une technologie donnée.

support intervalles de valeurs où une fonction (et par extension une distribution de valeurs aléatoires) est non-nulle.

Very Large Scale Integration (VLSI) circuits électroniques intégrés regroupant plusieurs millions de transistors.

watchdog dispositif détectant une inactivité plus longue qu'un temps limite dans un circuit, en cas de détection synonyme d'erreur de fonctionnement, la partie surveillée est généralement remise à zéro.

Introduction

La micro-électronique est un marché de masse. Dans la plupart des applications grand-public, les coûts de recherche et de développement d'un système sont amortis lorsque plusieurs milliers voire plusieurs millions d'exemplaires sont vendus. Pour chaque circuit, la phase de recherche et de développement nécessite plusieurs années-équivalent-ingénieur depuis l'établissement des spécifications jusqu'à la commercialisation en passant par les tests de prototypage. Cet investissement s'explique par la complexité de la conception, par la nécessité de vérifier consciencieusement le comportement du circuit et par les éventuelles itérations de prototypes pour obtenir les performances souhaitées. Chaque étape de la conception requiert des connaissances très spécialisées et donc chaque développement fait intervenir plusieurs personnes à des périodes différentes.

Dans le but de rationaliser les coûts, les industriels préfèrent les solutions technologiques stables et éprouvées. Cela permet de réutiliser une partie des acquis d'un projet dans une ou plusieurs réalisations futures. Ces acquis sont aussi bien techniques (méthodes, outils, codes) que humains (habitudes de travail, formations). Les industriels ont donc acquis au cours des quarantes dernières années un savoir-faire très important concernant les circuits dit "synchrones", qui représentent l'écrasante majorité des circuits vendus actuellement. Leur conception est facilitée par la disponibilité de nombreuses méthodes et de nombreux outils très répandus (suites logicielles de *Cadence*, *Synopsys* et de *Mentor Graphics* par exemple). L'offre créant la demande et inversement, le monde industriel est essentiellement intéressé par les solutions éprouvées et la formation des concepteurs est encore aujourd'hui essentiellement axée sur les circuits synchrones. Finalement, cette préférence pour les circuits synchrones entretient leur égémonie sur la micro-électronique. Mais au fût-et-à-mesure des évolutions et des contraintes technologiques grandissantes, de nombreuses limitations apparaissent.

Les premiers développements des circuits asynchrones ont commencé dès le début de l'intégration sur puce dans les années 1960. Ils ne sont restés que des objets de recherche pendant de nombreuses décennies. Leurs caractéristiques de robustesse les rendent néanmoins particulièrement adaptés aux nouvelles technologies de fabrication et incontournables pour l'avenir des circuits *Very Large Scale Integration** (*VLSI*) où

l'interconnexion de plusieurs blocs logiques conventionnels (synchrones) se fait de plus en plus à l'aide de réseaux sur puce (*Network On Chip**) basés sur un fonctionnement asynchrone.

Les circuits asynchrones offrent des caractéristiques intrinsèques pour lesquelles de lourdes modifications structurelles seraient nécessaires pour réaliser un circuit synchrone équivalent. Ces caractéristiques sont des points de recherche et de développement actuellement très actifs :

- la puissance consommée par le circuit est modulée automatiquement en fonction des variations de la tension d'alimentation et de l'activité des données à traiter ;
- la mise en veille des blocs inactifs fait partie du fonctionnement normal du circuit ;
- leur structure s'adapte automatiquement aux caractéristiques réelles du circuit et s'affranchit donc des variations des procédés de fabrication ;
- l'autocadencement implique que les performances du circuit sont très proches du maximum atteignable pour la technologie ;
- le spectre du rayonnement électromagnétique du circuit est étalé et ne contient que peu de pics de puissance ;
- les effets des perturbations électro-magnétiques peuvent être détectés dans certains cas ;
- l'intégration de blocs logiques tiers (*Intellectual Property**) dans la hiérarchie des blocs logiques d'une application complexe est facilité par les auto-synchronisations des échanges de données.

Des techniques similaires ont été mises au point pour les circuits synchrones : *Dynamic Voltage Scheduling**, *Dynamic Frequency Scheduling**, redondances spatiales ou temporelles des traitements pour éviter les effets électro-magnétiques de l'environnement. Ces techniques augmentent l'aire occupée et la consommation du circuit synchrone. Mais l'aspect intrinsèque de ces caractéristiques ont ravivé l'intérêt pour les techniques asynchrones dans la conception des circuits intégrés numériques.

Ces techniques asynchrones reposent sur des approches complètement différentes aussi bien pour le flot de conception numérique que pour les portes logiques finalement utilisées. Les circuits asynchrones sont historiquement décrits à l'aide de langages spécifiques basés sur la description de canaux communicants (*CSP*, *CHP*...). Plus récemment, les flots de conception se basent sur des descriptions au niveau transactionnel (*TLM*) [TIE]. Les outils de synthèse transforment ces descriptions dans le but d'obtenir une *netlist** de portes logiques de base. Ces primitives logiques comprennent les portes logiques combinatoires classiques (réalisant les fonctions et, ou, ou-exclusif...), mais aussi des portes spécifiques aux circuits asynchrones. Ces portes spécifiques permettent d'implémenter assez simplement les échanges dans ces canaux de communications. Les outils classiques de placement-routage terminent ensuite le flot de synthèse dans le but d'obtenir le dessin des masques pour la fabrication.

Les techniques de synthèse mises en œuvre pour les circuits asynchrones peuvent dans certains cas prouvées-correctes, directement au niveau des transformations effectuées par l'outil sur la description. C'est-à-dire que la description de tout circuit synthétisé avec cet outil vérifié correspondra parfaitement aux spécifications fournies par le concepteur.

Par contre ces vérifications s'arrêtent à la *netlist** obtenue par l'outil de synthèse.

Ce manuscrit porte sur l'étude de phénomènes analogiques pouvant intervenir dans les portes spécifiques aux circuits asynchrones et pouvant perturber leur comportement. Les perturbations analogiques étudiées peuvent être transitoires ou permanentes. Les environnements agressifs peuvent perturber temporairement le fonctionnement du circuit. De surcroît, les tolérances dans les procédés de fabrication, son vieillissement et son fonctionnement prolongé peuvent altérer durablement ses caractéristiques. Cette étude vise à améliorer la robustesse des circuits numériques asynchrones pendant toute la vie du produit en limitant ses dysfonctionnements.

Le but est d'augmenter encore la robustesse des circuits déjà très robustes afin de rendre leur fonctionnement numérique moins sensible aux phénomènes analogiques : le fonctionnement numérique du circuit doit rester fidèle à sa description logique. Les modifications proposées aux portes spécifiques visent à éviter les blocages des circuits (*deadlocks*) et donc améliorer leur immunité aux phénomènes analogiques. Idéalement, en présence de perturbations analogiques, le comportement numérique du circuit doit rester inchangé : une des variables d'ajustement pour absorber ces effets est par exemple le temps de réponse du circuit. Celui-ci pourrait être augmenté afin de garantir le fonctionnement correct du circuit en présence de phénomènes analogiques perturbants : la structure insensible aux délais du circuit asynchrone peut intrinsèquement s'accomoder d'éventuelles fluctuations dans les temps de traitement.

Les structures des circuits asynchrones seront d'abord détaillées dans le [chapitre 2](#). Ces circuits fonctionnent grâce à différentes structures possibles pour les cellules spécifiques de mémorisation dont les principales sont détaillées au [chapitre 3](#). Un phénomène analogique perturbant le fonctionnement d'une structure asynchrone est expliqué dans le [chapitre 4](#) et les cellules spécifiques sont ensuite caractérisées et corrigées dans le [chapitre 5](#). Afin de garantir leur comportement lors du processus de fabrication, les éventuelles variations dans leurs caractéristiques seront prises en compte dans le [chapitre 6](#). Enfin une approche plus formelle sera mise en œuvre dans le [chapitre 7](#) afin de vérifier mathématiquement le fonctionnement des circuits asynchrones.

Chapitre 2

Généralités sur les circuits asynchrones

| | | |
|------------|-------------------------------------------------------------------|-----------|
| 2.1 | Introduction aux circuits asynchrones | 6 |
| 2.1.1 | Limitations des circuits synchrones | 6 |
| 2.1.2 | Principe des circuits asynchrones | 7 |
| 2.2 | Les échanges entre blocs fonctionnels | 8 |
| 2.3 | Classement des circuits asynchrones | 9 |
| 2.3.1 | Circuits de Huffman | 9 |
| 2.3.2 | Circuits micropipelines | 9 |
| 2.3.3 | Circuits indépendant de la vitesse "SI" | 10 |
| 2.3.4 | Circuits quasi-insensibles aux délais "QDI" | 10 |
| 2.3.5 | Circuits insensibles aux délais "DI" | 11 |
| 2.4 | Codage données groupées | 11 |
| 2.4.1 | Codage 4 états | 12 |
| 2.4.2 | Codage 3 états, 1 parmi 2 | 12 |
| 2.4.3 | Codage 1-parmi-N | 13 |
| 2.5 | Protocoles d'échanges | 13 |
| 2.5.1 | Protocole à 2 phases | 14 |
| 2.5.2 | Protocole à 4 phases | 15 |
| 2.6 | C-elements | 15 |
| 2.7 | Synchronisation entre entrées et sorties d'un bloc logique | 17 |
| 2.7.1 | Protocoles séquentiels | 19 |
| 2.7.2 | Protocole WCHB | 19 |
| 2.7.3 | Protocole PCHB | 21 |
| 2.7.4 | Protocole PCFB | 23 |
| 2.7.5 | Protocole FDFB | 24 |
| 2.7.6 | Mise en série de buffers | 26 |
| 2.8 | Aléas de commutation | 27 |
| 2.8.1 | Conception logique sans aléa | 28 |
| 2.9 | Fonctionnement des circuits asynchrones | 29 |

2.1 Introduction aux circuits asynchrones

La désignation de *circuit synchrone** ou de *circuit asynchrone** est assez générale et masque une diversité de solutions techniques pour obtenir le fonctionnement du circuit respectivement *avec* et *sans* horloge de cadencement. Dans cette section, les différences structurelles entre les circuits synchrones et asynchrones seront abordées.

2.1.1 Limitations des circuits synchrones

Dans les circuits synchrones, tous les éléments du circuits sont cadencés par un signal de synchronisation commun (cas général). Cette synchronisation globale garantit l'évolution simultanée de tous les blocs logiques. Le fonctionnement global et correct de ces circuits est lié à des hypothèses temporelles devant être respectées lors de la conception et de la réalisation technologique des circuits :

- La distribution du signal de synchronisation doit être *instantanée* sur tout le circuit pour que l'échantillonnage soit effectué au même *instant* sur des signaux stables. Si un phénomène de *skew** apparaît, il y a risque d'échantillonner des valeurs en cours de propagation et pas encore stabilisées, voir d'erreur de phase si le *skew** est important.
- Le signal d'horloge doit être amplifié et distribué à un très grand nombre de bascules. L'arbre d'horloge est généralement constitué d'un nombre important d'amplificateurs : cette logique occupe de la surface sur la puce et consomme beaucoup d'énergie dynamique.
- Les mémorisations ne doivent avoir lieu qu'une fois la stabilisation des blocs combinatoires obtenue : les *aléas** transitoires sont automatiquement masqués par l'échantillonnage. Lors de la conception, le chemin critique doit être déterminé afin d'évaluer le temps nécessaire à cette stabilisation. On peut ainsi fixer la vitesse de cadencement maximale du circuit. Généralement des marges de sécurité sont appliquées sur la vitesse maximale de cadencement dans le but de garantir le fonctionnement correct de tous les circuits d'une série (dimensionnement pire cas). Lors de la fabrication, il est nécessaire de maîtriser parfaitement le processus pour éviter les disparités et respecter ce temps. Le vieillissement du circuit fabriqué dégrade également ses performances. Par conséquent, les circuits synchrones sont utilisés très souvent à une vitesse de cadencement moindre que leur capacité réelle. Ces marges de sécurité garantissent alors le fonctionnement correct de la plupart des circuits pendant la durée de vie du produit.
- Ce chemin combinatoire critique ne concerne que quelques blocs logiques mais vient limiter le cadencement de tout le circuit. L'exécution dans certains blocs pourrait être bien plus rapide, mais la synchronisation globale interdit ce genre d'optimisation. Les concepteurs de circuits synchrones utilisent dans ce cas des horloges multiples (multi-clocks) pour faire fonctionner chaque bloc à la vitesse qui lui est propre. La puce est donc composée de plusieurs domaines d'horloges. Dans un tel environnement, les communications entre les différents domaines d'horloge nécessitent des

resynchronisations coûteuses en latence et introduisent un risque de métastabilité (mémorisation de tension intermédiaire n'étant ni un niveau logique 0, ni un niveau logique 1).

- L'intégration modulaire de plusieurs composants synchrones est délicate car l'agencement en série de plusieurs blocs combinatoires est susceptible d'augmenter le chemin critique du circuit. Tous les blocs logiques de la description doivent être rigoureusement caractérisés, même si ils sont sous la forme d'*Intellectual Property** (IP).
- Le cadencement global se traduit par une absorption de courant par le circuit sous forme de pics cadencés avec l'horloge et par ailleurs cet état de fait produit un rayonnement électromagnétique contenant de nombreuses harmoniques.

Toutes ces limitations sont liées à la structure et au principe de fonctionnement des circuits synchrones. Au fur-et-à-mesure des évolutions technologiques, celles-ci prennent de plus en plus d'importance.

2.1.2 Principe des circuits asynchrones

De par leur approche complètement différente, les circuits asynchrones ne souffrent pas des symptômes observés sur les circuits synchrones.

L'approche asynchrone implique l'absence de synchronisation globale. Le fonctionnement du circuit ne fait appel qu'à des synchronisations locales : d'un bloc logique à un autre par le biais d'un canal de communication. Cette communication permet le passage des ordres de fonctionnement et des valeurs des données.

La synchronisation entre deux blocs logiques est intrinsèque à la communication. Cette caractéristique rend la conception modulaire immédiate car il suffit d'interconnecter les différents blocs sans se soucier d'une quelconque synchronisation globale. La mise en série de plusieurs blocs asynchrones va structurellement former un pipeline de traitement. Chaque bloc fonctionnel interconnecté démarre son calcul dès que des données sont disponibles à ses entrées et rend compte de la fin de ces traitements vers les blocs suivants connectés à ses sorties.

Le fonctionnement des circuits asynchrones est basé sur les événements transmis dans ces communications. Un bloc s'active dès que des données à traiter ont été transmises depuis le bloc précédent. Dès qu'un bloc a terminé le traitement des données, il rentre intrinsèquement dans une phase d'attente assimilable à une mise en veille. Son état est stable, ce qui aboutit à une consommation électrique très réduite. Ainsi la consommation dynamique d'un circuit asynchrone est directement lié à la fréquence d'occurrence des données à traiter.

Nous avons vu les avantages immédiats de l'utilisation d'une conception asynchrone. Il convient maintenant d'étudier plus en détails les mécanismes d'échanges permettant ce mode de fonctionnement.

2.2 Les échanges entre blocs fonctionnels

Les échanges entre deux blocs se font par un protocole de communication visant à synchroniser l'émission et la réception des données afin que le transfert puisse être effectué dès que les deux modules sont prêts. Par définition, le bloc actif initie la communication. Les protocoles utilisés sont bidirectionnels, ce qui signifie que chaque bloc fonctionnel doit disposer de ports d'entrée et de sorties pour chaque canal de synchronisation. La [figure 2.1](#) montre l'interconnexion entre deux blocs logiques dans un circuit asynchrone.

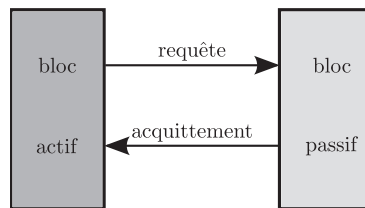


FIGURE 2.1 – Communication entre deux blocs asynchrones

Chaque sens de communication à une fonctionnalité bien définie :

- Le bloc actif envoie une *requête** vers le bloc passif pour demander un traitement ;
- Le bloc passif envoie un *acquiescement** vers le bloc actif pour l'informer de la prise en compte de cette *requête**. Cet acquiescement peut avoir lieu à différents instants du traitement du bloc passif. La [section 2.5](#) résume les protocoles les plus couramment présents dans les circuits asynchrones.

Outre la synchronisation du fonctionnement des blocs, les informations de *requête** et d'*acquiescement* permettent le passage de données. Par définition, une donnée part d'un bloc émetteur vers un bloc récepteur. Selon que les données soient groupées avec la requête ou avec l'*acquiescement*, il existe quatre familles de canaux de communications différenciées par les échanges de données :

- Dans les canaux de communication uniquement dédiés à la synchronisation, aucune donnée n'est échangée ;
- Dans les canaux de type *push**, l'émetteur est le bloc actif, il informe le bloc passif de la présence d'une donnée. Les données sont transmises du bloc actif vers le bloc passif en même temps que la *requête** : le signal de requête peut être considéré comme un signal de validité des données. Cette famille de canal est la plus utilisée dans la pratique ;
- Dans les canaux de type *pull**, le récepteur est le bloc actif ; il demande au bloc passif de transmettre une donnée. L'information de requête est orientée du bloc passif vers le bloc actif et la donnée inversement. Le signal d'*acquiescement* peut alors être considéré comme un signal de validité des données ;
- Dans les canaux de type *push-pull**, des données sont échangées dans les deux sens, ces canaux ne sont quasiment pas utilisés dans la pratique.

Les liaisons entre les deux blocs font intervenir un groupe de signaux codant la valeur des données, l'indicateur de requête et l'information d'*acquiescement*. En logique asyn-

chrone, tous les signaux sont unidirectionnels. Ils sont issus d'un étage basse impédance (la sortie d'une porte logique) et se propagent vers un étage haute impédance (l'entrée d'une porte logique). Chaque canal de communication fait intervenir des entrées et des sorties logiques pour les deux blocs connectés.

2.3 Classement des circuits asynchrones

Il existe plusieurs familles de circuits asynchrones correspondant à différentes hypothèses sur les communications entre blocs logiques. Les sections suivantes décrivent les familles de circuits asynchrones classées par importance des hypothèses temporelles.

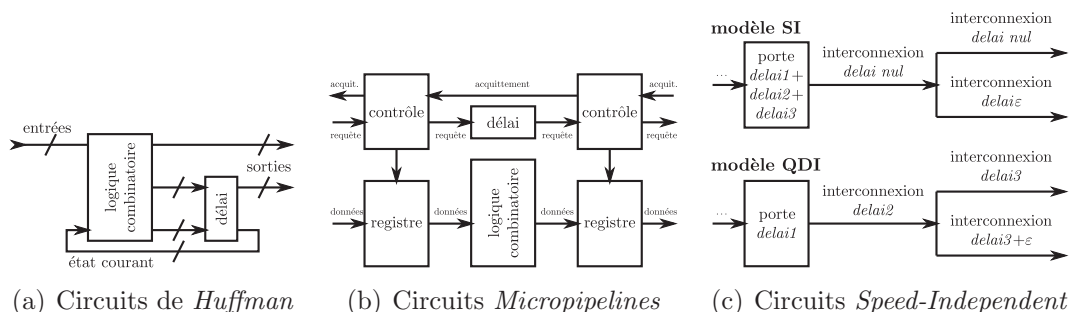


FIGURE 2.2 – Principales familles de circuits asynchrones

2.3.1 Circuits de Huffman

Le principe des *circuits de Huffman** (*Huffman circuits*) repose sur des hypothèses temporelles fortes assez similaires à celles des circuits synchrones : les sorties des étages combinatoires sont supposées être stables après un certain temps. Leur structure est formée d'une fonction combinatoire rebouclée sur elle-même [Mye01] représentée dans la figure 2.2(a). En fonctionnement normal, le délai inséré dans la boucle doit toujours être supérieur au temps de stabilisation du circuit combinatoire afin que les entrées du bloc combinatoire (état courant et entrées du système) restent inchangées pendant la phase de stabilisation.

Ce délai peut être réalisé à l'aide de logique combinatoire sous la forme d'une chaîne d'inverseurs lents par exemple. Dans la pratique, cette famille de circuits est très peu utilisée car son hypothèse temporelle de fonctionnement est très difficilement vérifiable : l'inactivité des entrées pendant la stabilisation n'est pas réalisable dans un circuit réel.

2.3.2 Circuits micropipelines

Les *circuits micropipelines** [Sut89] ont une structure de type *flot de données**. C'est-à-dire que les opérations successives effectuées sur un *jeton de donnée** (*data token*)

vont être placées les unes après les autres afin de former un pipeline de traitement. Les opérations successives sont séparées par des registres commandés par une horloge locale. Cette horloge, assimilable à une requête, est propagée en parallèle des données au travers d'un délai (figure 2.2(b)). Ce délai est supérieur au temps maximum de traitement des données et cette hypothèse garantit donc la stabilité des données au moment de l'échantillonnage. Les registres et les blocs combinatoires sont semblables à ceux utilisés en technologie synchrone. Par contre, l'implémentation des blocs de contrôle nécessitent des cellules spécifiques aux circuits asynchrones.

De tels circuits peuvent être synthétisés à partir de descriptions haut-niveau écrites en CHP [Rez04]. Ce langage de haut-niveau définit au niveau du bloc logique les actions à accomplir dès que des données sont disponibles dans un canal de communication. Ces règles sont alors compilées par l'outil TAST (TIMA Asynchronous Synthesis Tools) pour fournir la *netlist** du circuit correspondant.

2.3.3 Circuits indépendant de la vitesse "SI"

Contrairement aux deux familles vues précédemment, les *circuits indépendants de la vitesse** (*Speed Independent - SI*) n'imposent pas d'hypothèses temporelles sur le fonctionnement des parties combinatoires [Mil65] : le temps de traitement des données peut potentiellement être non-borné. Cette modélisation permet de s'affranchir des implémentations matérielles des blocs logiques : la structure du circuit garantit le fonctionnement global quels que soient les temps de traitement. Cependant, ce modèle pose des hypothèses fortes sur les interconnexions : les délais dans les fils sont supposés nuls. Les circuits SI sont équivalents [Hau95] aux circuits QDI expliqués dans la section suivante. Cette équivalence est illustrée dans la figure 2.2(c) : le délai réel des interconnexions peut être inclu dans le délai de la porte logique connectée à ce fil.

2.3.4 Circuits quasi-insensibles aux délais "QDI"

La famille des *circuits quasiment insensibles aux délais** (*Quasi-Delay-Insensitive - QDI*) ne pose pas d'hypothèses sur les temps de propagation dans les portes logiques et dans les interconnexions : ces temps sont non-bornés. Cependant toute diffusion d'un même signal à plusieurs destinataires peut introduire des différences de temps de propagation et par conséquent un comportement non souhaité lorsque cette donnée est acquittée par l'un des destinataires avant que les autres aient eu le temps de la traiter [vB92].

Afin de garantir le fonctionnement correct des circuits QDI quels que soient les délais dans les portes et dans les interconnexions, il faut s'assurer que toutes les *fourches isochrones** sont équilibrées [Mar90b].

Cette famille de circuits est dans la pratique la plus répandue. De nombreux microprocesseurs ont été fabriqués en technologie asynchrone QDI : mini-MIPS [MLM+97], ASPRO (cœur à structure RISC) [RVR98], MICA [ABR+01], Lutonium (cœur 8051) [MNP+03]. Plus récemment, de nombreux industriels proposent maintenant des

solutions complètement asynchrones : Microprocesseur 8-bit flexible à base de transistors TFT par Seiko Epson (sur support souple - 2005) [Shi05], Intellasys SEAForth 40C18 (40 cœurs asynchrones - 2008), microprocesseur 16-bit très faible bruit (IP pour cartes à puce) [IC09].

2.3.5 Circuits insensibles aux délais "DI"

Les **circuits insensibles aux délais*** (*Delta Insensitive* - DI) ont été formalisés dès les débuts de la micro-électronique intégrée [Cla67]. Leur fonctionnement ne repose sur aucune hypothèse temporelle au niveau des portes logiques ou au niveau des interconnexions : leurs délais peuvent être potentiellement non bornés.

La synchronisation forte entre tous les blocs logiques implique que toutes les transmissions de données sont acquittées. En cas de diffusion d'une donnée à plusieurs blocs logiques, il est nécessaire de s'assurer que tous les destinataires ont bien reçu la donnée. L'implémentation de cette vérification nécessite l'utilisation de portes logiques à plusieurs entrées et à plusieurs sorties (une par destinataire) [Hau95]. Cette conception non-standard rend très complexes les réalisations de circuits de cette famille.

2.4 Codage données groupées

Lorsque des données sont transférées en même temps que la requête ou l'acquittement, il est nécessaire de grouper les signaux orientés dans la même direction.

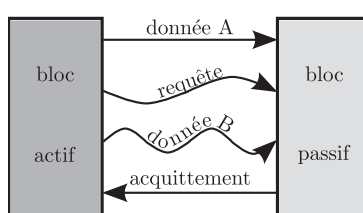


FIGURE 2.3 – Communication entre deux blocs asynchrones après placement-routage

En effet, après placement-routage du circuit (figure 2.3), tous les signaux risquent de ne pas être propagés à la même vitesse : certaines interconnexions seront plus longues que d'autres. Dans certains cas, la donnée (A) se propage plus rapidement que son information de validité (requête). Dans d'autres cas, la requête est propagée plus rapidement que la donnée (B). Ce dernier cas est problématique : le bloc récepteur risque de prendre en compte des données qui ne sont pas encore arrivées à destination. Pour éviter ce phénomène de *skew**, il est nécessaire de transférer les données et la requête selon un codage insensible au délai de propagation.

2.4.1 Codage 4 états

Dans le codage 4 états, chaque bit des données est dupliqué et transmis par deux signaux. La valeur de ces deux signaux à un instant de transmission est déterminée à partir de leurs valeurs à l'instant précédent et de la donnée à transmettre à cet instant. Cette décision peut se coder sous la forme d'un graphe de transition des valeurs atteignables représenté en [figure 2.4](#). Le nom de chaque état correspond à la valeur de chaque signal transmis : dans l'état "01", le premier signal transmis est à l'état logique bas alors que le second est à l'état logique haut.

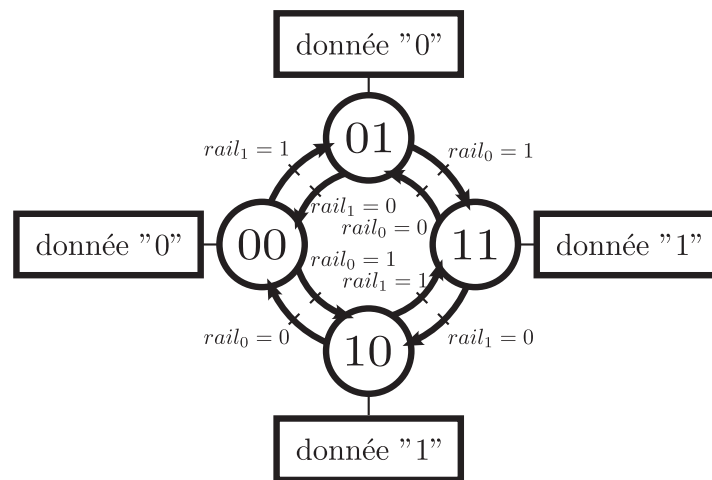


FIGURE 2.4 – Transition des signaux pour un codage 4 états

Ce codage est aussi appelé **Non-Retour-à-Zéro*** (*NRZ*) : à tout instant, les signaux transmis représentent une donnée. Chaque nouvelle transmission d'une donnée provoque un seul basculement d'un seul des deux signaux : l'activité du circuit sera très réduite.

Cet encodage est insensible aux délais : quelque soit le temps de propagation d'un des deux signaux, les valeurs du couple de signaux représentent toujours une donnée valide. Le récepteur est sensible aux changements d'états sur l'un ou l'autre des conducteurs : tant qu'aucun changement est détecté, il considère que l'ancienne donnée est toujours présente et il est en phase d'attente. Finalement, en cas de déséquilibre des longueurs des deux conducteurs, certaines données nécessiteront plus de temps pour être transmises mais leur transfert sera toujours sûr.

Par contre, les contrôleurs utilisant ce codage sont complexes. Ils doivent être capable de détecter des transitions montantes et descendantes sur deux conducteurs à la fois.

2.4.2 Codage 3 états, 1 parmi 2

Afin de réduire la complexité des contrôleurs, d'autres encodages ont été introduits. Le codage 3 états fait lui-aussi intervenir deux signaux pour communiquer un bit de données. Parmi les quatre états logiques atteignables par ces signaux, deux états caractérisent la donnée binaire (généralement "01" et "10"), l'état de repos ("00") correspond à l'absence

de données et le dernier état ("11") est interprétable comme une erreur de communication. La [figure 2.5](#) résume l'enchaînement possible des valeurs des deux signaux.

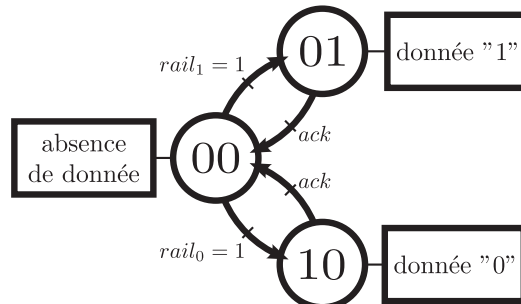


FIGURE 2.5 – Transitions des valeurs pour un codage 3 états

Lors de la transmission d'un bit de donnée, il y a deux transitions successives sur un des deux signaux transmis. Après chaque transmission d'un bit, les deux signaux reprennent leur valeur par défaut ("00"). Ce type d'encodage est aussi appelé [Retour-à-Zéro*](#) (*RZ*).

L'activité d'un des signaux correspond à la transmission d'une donnée de niveau logique "0", l'autre à la transmission d'un "1". Si le temps de propagation d'un des signaux est plus long que l'autre, alors les données "0" seront transmises plus rapidement que les "1" (ou inversement). Les protocoles de communications ([section 2.5](#)) assurent l'absence de recouvrement : un acquittement est nécessaire pour revenir à l'état de repos. Par conséquent, cet encodage est lui-aussi insensible aux délais de propagation.

2.4.3 Codage 1-parmi-N

L'encodage 3 états 1 parmi 2 vu dans la [section 2.4.2](#) peut être étendu à la transmission de données à N valeurs différentes. Il faut dans ce cas N signaux et l'encodage est [one-hot*](#) : c'est à dire que seul un des signaux passe à l'état logique "1" pour transmettre la donnée et repasse ensuite à son état de repos "0". La [figure 2.6](#) représente une généralisation du graphe des valeurs atteignables. Le mot de chaque état représente la valeur des N signaux transmis.

L'efficacité d'un tel encodage est faible puisqu'il faut N signaux pour transférer une donnée à N valeurs possibles. Les recherches actuelles s'orientent vers l'utilisation d'encodage M parmi N (avec $1 < M < N/2$). N signaux peuvent alors transférer toutes les combinaisons de M signaux actifs : $C_N^M = \frac{N!}{M!(N-M)!}$ valeurs. Par exemple un encodage 2 parmi 5 peut transférer 10 valeurs différents pour une donnée sur seulement 5 fils.

2.5 Protocoles d'échanges

Pour enchaîner le transfert de plusieurs données, les blocs de communication implémentent des protocoles d'échanges. Pour assurer la forte synchronisation entre

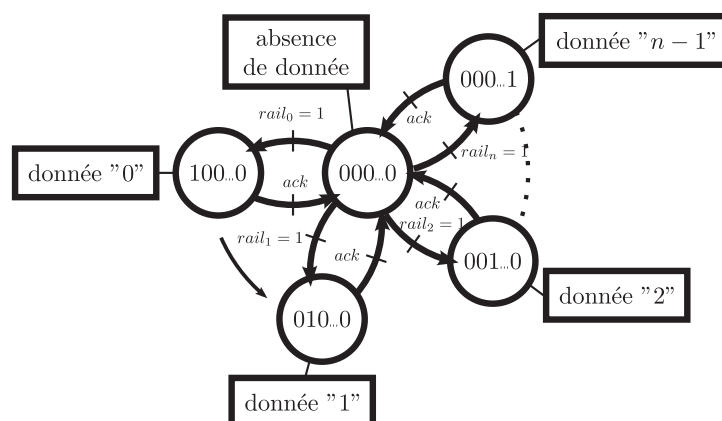


FIGURE 2.6 – Machine à état d'un codage 1 parmi N états

l'émetteur et le récepteur, ces protocoles sont du type **poignée de main*** (*handshake protocols*) : chaque action de l'émetteur est validée par le récepteur et inversement. Cela permet d'effectuer l'échange de données lorsque les deux blocs sont prêts et d'accuser réception de la bonne réception de celles-ci.

Ces protocoles sont gérés dans chacun des blocs asynchrones par des contrôleurs. Ces blocs logiques occupent de la surface et requièrent de l'énergie pour fonctionner, tout comme l'arbre d'horloge en synchrone, sans apporter de fonctionnalités supplémentaires au circuit. L'efficacité complète du circuit passe par un choix judicieux de protocole et de codage afin de réaliser un contrôleur minimisant la surface occupée et l'énergie utilisée.

2.5.1 Protocole à 2 phases

Dans ce protocole d'échanges, il n'y a pas d'étape de retour à l'état de repos : les données s'enchaînent et le signal d'acquittement change d'état pour autoriser le passage à la donnée suivante. Ce protocole de communication est généralement couplé à un encodage de type 4 états (section 2.4.1). L'implémentation de ce protocole nécessite une logique sensible sur les fronts montants et les fronts descendants, ce qui est généralement coûteux en terme de complexité du contrôleur.

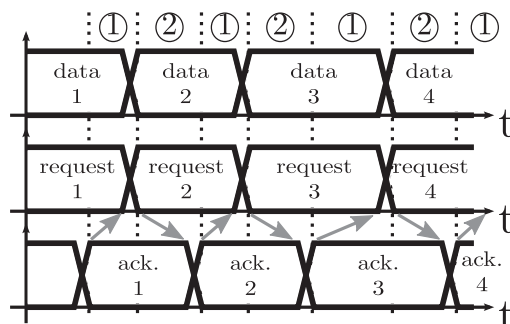


FIGURE 2.7 – Protocole 2 phases

La figure 2.7 représente l'enchaînement des données dans le canal de communication : l'émetteur place les données couplées encodées avec l'ordre requête dans la phase ①, le récepteur acquitte ces données dans la phase ②. Une fois cet acquittement reçu, l'émetteur peut transmettre la donnée suivante et le cycle continue.

2.5.2 Protocole à 4 phases

L'enchaînement ininterrompu des données dans le protocole à 2 phases pose des problèmes d'implémentation : les contrôleurs de protocole doivent être aussi bien sensibles à des fronts montants qu'à des fronts descendants pour un même signal. Voilà pourquoi les protocoles 4 phases ont été développés : chaque donnée et chaque acquittement sont séparées par des états de repos. Ces phases de repos sont les états de repos du protocole à 3 états vus dans la section 2.4.2 et du protocole 1 parmi N vu dans la section 2.4.3.

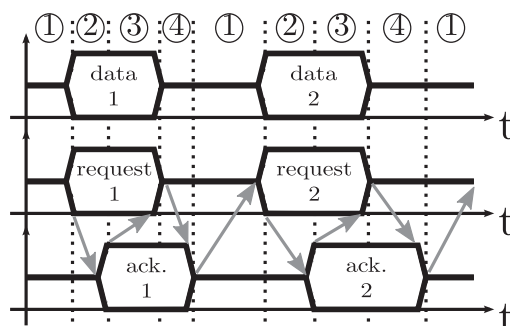


FIGURE 2.8 – Protocole 4 phases

La figure 2.8 montre l'évolution des signaux de requête et d'acquiescement pendant plusieurs transferts de données. La phase ① est l'état d'attente : le récepteur est prêt. En phase ②, l'émetteur place la première donnée et la requête dans le canal de communication. Ces données sont acquittées par le récepteur dans la phase ③. Voyant cet acquittement, l'émetteur supprime les données et la requête dans la phase ④. Le récepteur valide cette suppression en enlevant son acquiescement dans la phase ① suivante. Le canal de communication est prêt pour le prochain transfert.

L'utilisation d'un signal d'acquiescement actif à l'état bas facilite les implémentations matérielles des contrôleurs 4 phases. La phase ① correspond alors à l'absence d'une requête sur le canal (noté \overline{R}) et l'inactivité de l'acquiescement (noté A). Lors de la phase ②, on a alors $R.A$. La phase ③ correspond à $R.\overline{A}$ et finalement la phase ④ correspond à l'observation $\overline{R}.\overline{A}$.

2.6 C-elements

La réalisation de circuits asynchrones nécessite l'utilisation de portes logiques spécifiques afin de réaliser efficacement en terme de surface les contrôleurs de proto-

cole. Les circuits asynchrones sont réalisés à l'aide de points mémoire actifs sur état : les C-elements*.

Le C-element à deux entrées symétriques (appelé M2) est représenté dans la [figure 2.9\(a\)](#). Sa sortie prend la valeur des entrées lorsque celles-ci sont égales. La table de vérité fait donc apparaître deux états de mémorisation. Les règles de transitions sont une autre écriture plus formelle de cette table de vérité. Les conditions de l'*implication** (à gauche du symbole \Rightarrow) indiquent les états à respecter pour que la sortie bascule à l'état logique 0 ou à 1 (l'action est située à droite du symbole d'implication) : par exemple la sortie bascule à 0 lorsque les deux entrées sont à 0.

Le rôle des entrées d'un C-element peut également être différencié. Les C-elements dissymétriques permettent à certaines des entrées de n'intervenir lors du basculement de la sortie que vers l'état logique 0 ou que vers l'état logique 1 :

- Les dissymétries positives ([figure 2.9\(b\)](#)) empêchent les entrées affectées d'intervenir lors des basculement de la sortie vers l'état logique 0.
- Les dissymétries négatives ([figure 2.9\(c\)](#)) ont l'effet inverse : les entrées affectées n'interviennent pas lorsque la sortie bascule vers l'état logique 1.

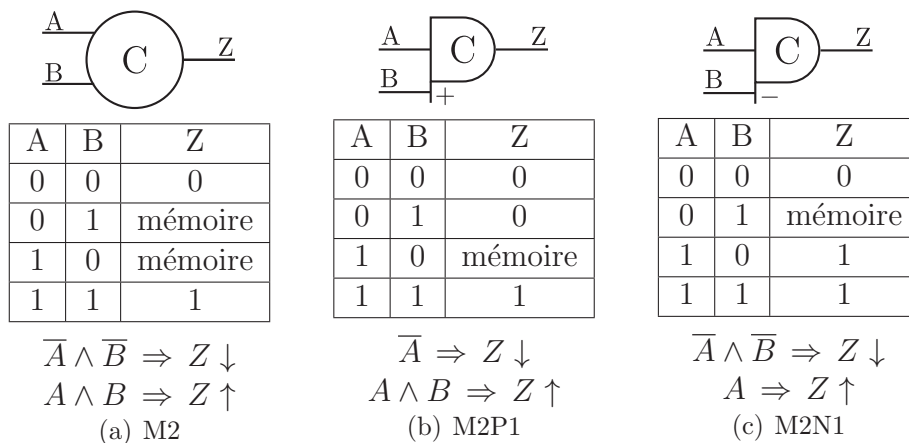


FIGURE 2.9 – Symbole, table de vérité et règles de transition de portes C-elements

La fonction C-element peut être généralisée à un nombre plus élevé d'entrées et des dissymétries peuvent être appliquées sur différentes entrées d'une même porte. Comme tout point mémoire, il est parfois nécessaire d'assurer une valeur déterminée lors de l'initialisation du circuit, une entrée *reset** ou *set** peut également être ajoutée à la fonction. La combinaison de toutes ces variantes implique que de nombreux C-elements distincts peuvent être nécessaires à la conception d'un circuit asynchrone.

Les fondeurs ne fournissent généralement pas ces primitives asynchrones. Ces portes sont alors soit réalisées en assemblant les portes standards disponibles afin de réaliser la fonction souhaitée, soit conçues en mode full-custom (du dimensionnement des transistors jusqu'au dessin des masques) dans la technologie du fondeur. La fonction C-element et les implémentations possibles sont décrits en détail dans le [chapitre 3](#) (page 33).

2.7 Synchronisation entre entrées et sorties d'un bloc logique

Un bloc logique asynchrone peut disposer de plusieurs entrées et de plusieurs sorties. Une fois les données disponibles sur une entrée, plusieurs choix sont possibles concernant l'action suivante à donner :

- le bloc logique peut commencer par acquitter l'entrée, mais celle-ci peut alors être supprimée par le bloc émetteur. Il faut donc préalablement sauvegarder cette valeur en interne au bloc logique ;
- il peut commencer les calculs, envoyer la valeur du résultat sur sa sortie et attendre que la sortie soit acquittée pour acquitter son entrée. . .

Le séquençement des requêtes et acquittements entre l'entrée et la sortie d'un bloc logique laisse de nombreux degrés de libertés [VSB10]. Dans le cas particulier d'un bloc logique réalisant la fonction buffer (représenté en figure 2.10), la figure 2.11 tirée de cet article représente l'ensemble des couples de phases atteignables pour le canal d'entrée et le canal de sortie d'un buffer asynchrone. Un buffer transfère une donnée du canal d'entrée vers le canal de sortie. Il la mémorise tant que le canal de sortie n'est pas prêt à accepter de données. Dans cette section, les deux canaux ont un protocole 4 phases.

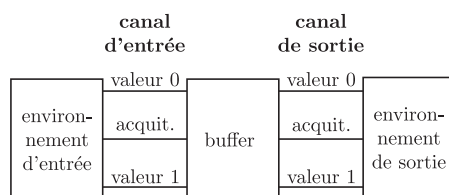


FIGURE 2.10 – Buffer asynchrone et son environnement

Un changement de phase sur le canal d'entrée correspond à un déplacement horizontal dans le graphe, un changement de phase sur le canal de sortie à un déplacement vertical. Tout buffer asynchrone, mémorisant au maximum une donnée en interne, a un fonctionnement s'inscrivant dans ce graphe de l'ensemble des phases atteignables. Les couples de phase possibles sont représentés par un cercle blanc à bord noir. Lorsqu'il n'existe qu'un passage unique par certaines phases, celles-ci sont marquées d'un point noir en leur centre : ce sont des phases de synchronisations fortes entre les canaux d'entrée et de sortie.

La phase colorée entièrement en noir représente l'état initial lorsque les deux canaux sont en phase d'attente. Le buffer est *causal**, c'est-à-dire que aucune donnée ne peut être émise en sortie tant qu'aucune donnée n'a été lue sur le canal d'entrée. Par conséquent, il faut attendre la phase ② du jeton n sur le canal d'entrée pour que la sortie correspondant à ce $n^{ième}$ jeton puisse être émise.

Pour chaque canal de communication, le passage à la phase suivante requiert une action, soit de l'émetteur, soit du récepteur, en alternance. R_E représente l'information de requête sur le canal d'entrée, A_E son acquittement, R_S la requête du canal de sortie

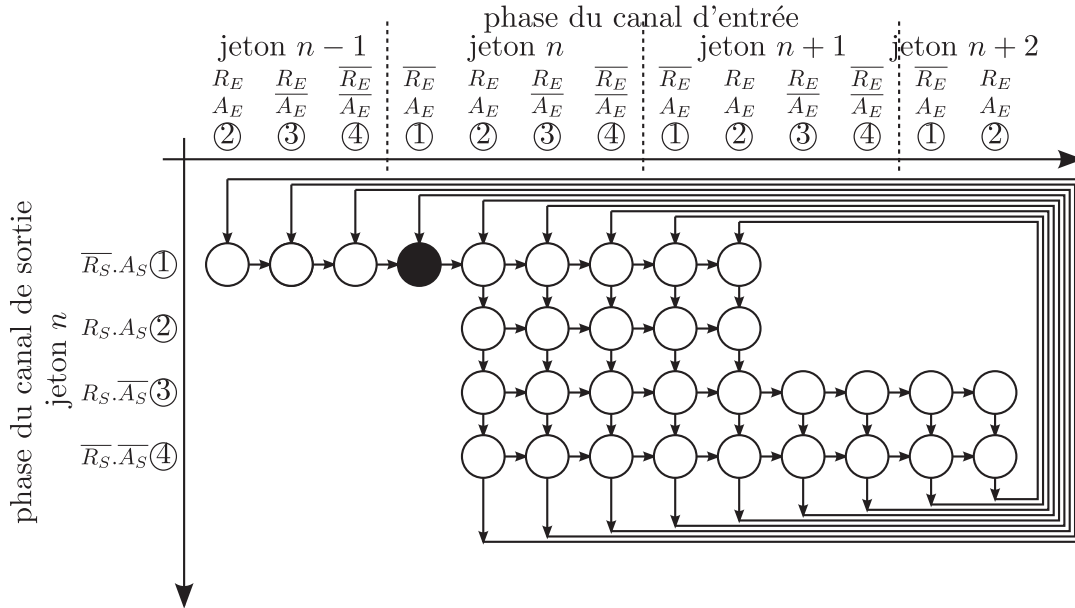


FIGURE 2.11 – Ensemble des phases atteignables en entrée et en sortie d'un Buffer asynchrone

et A_S l'acquittement correspondant. En encodage 1 parmi 2, une requête est présente dès que la valeur 0 ou la valeur 1 est active. Le passage de la phase ① à la phase ② du canal d'entrée nécessite une action de l'environnement d'entrée : celui-ci doit activer R_E . Il est automatiquement configuré pour placer une requête dès que le buffer est prêt (lorsque A_E est inactif). Cette règle peut être modélisée par l'action représentée par l'équation 2.1. Cette requête est également supprimée dès que le buffer l'a acquittée : l'activité de A_E provoque l'inactivation de R_E et donc la transition de la phase ③ à la phase ④ (équation 2.2).

$$A_E \Rightarrow R_E \uparrow \quad (2.1)$$

$$\overline{A_E} \Rightarrow R_E \downarrow \quad (2.2)$$

La transitions depuis la phase ② vers la phase ③ du canal d'entrée requiert le changement de l'activité de l'acquittement A_E , tout comme la transition depuis la phase ④ vers la phase ①. Ces deux transitions du canal d'entrée sont dirigées par le buffer et leurs conditions dépendront de son implémentation.

De même, les acquittements du canal de sortie sont générés automatiquement par l'environnement de sortie : la requête est automatiquement acquittée (passage de la phase ② à la phase ③, modélisé par l'équation 2.3) et l'acquittement est supprimé dès que la donnée est supprimée par le buffer (passage de la phase ④ à la phase ①, modélisé par l'équation 2.4). Les transitions depuis la phase ① vers la phase ② et depuis la phase ③ vers la phase ④ sont provoquées par le buffer et leurs conditions dépendront elles-aussi de son implémentation.

$$R_S \Rightarrow A_S \downarrow \quad (2.3)$$

$$\overline{R_S} \Rightarrow A_S \uparrow \quad (2.4)$$

Dans la pratique, l'implémentation d'un bloc asynchrone supportant l'ensemble des couples de phases de la [figure 2.11](#) occuperait trop de surface. Les concepteurs asynchrones ont élaboré un ensemble de synchronisations pouvant être facilement implémentées sur silicium. Pour chacun des protocoles suivants, l'ensemble des phases atteignables est restreint d'après les règles suivantes :

- la phase initiale est toujours activée.
- une transition est désactivée si la structure du circuit rend impossible le changement des signaux correspondants.
- une phase est désactivée si toutes les transitions partant de cette phase sont désactivées. Cette propriété peut s'apparenter à l'absence de dead-lock dans le circuit, cette propriété est également appelée *liveness**.
- une phase est désactivée si toutes les transitions arrivant vers cette phase sont désactivées : cette phase ne pourra jamais être observée.
- une transition est désactivée dès que sa phase d'origine ou sa phase de destination sont désactivées.

Dans les prochaines figures, les phases et les transitions désactivées seront représentées en grisé.

2.7.1 Protocoles séquentiels

Les protocoles séquentiels ne sont pas utilisés dans la pratique. Ils consisteraient en une synchronisation forte entre les canaux d'entrées et sorties de telle sorte que l'évolution de leurs phases ne serait possible que selon un seul enchaînement forcé lors de la conception du circuit.

La [figure 2.12](#) est un exemple de protocole séquentiel, il existe de nombreux autres tracés possibles. Tous les couples de phases sont des points de synchronisation forts entre les canaux d'entrée et de sortie, ils sont tous marqués d'un point noir central.

2.7.2 Protocole WCHB

Le protocole *Weak-Condition-Half-Buffer* est le protocole le plus séquentiel, après le protocole séquentiel. Les évolutions de phase des canaux d'entrée et de sortie sont très fortement liées. La [figure 2.13](#) représente une implémentation courante d'un buffer en protocole WCHB pour deux canaux encodés en dual-rail (1 parmi 2). Cette implémentation nécessite seulement trois portes logiques :

- une porte *NOR* (non-ou) à deux entrées, présente dans les bibliothèques standards du fondeur.

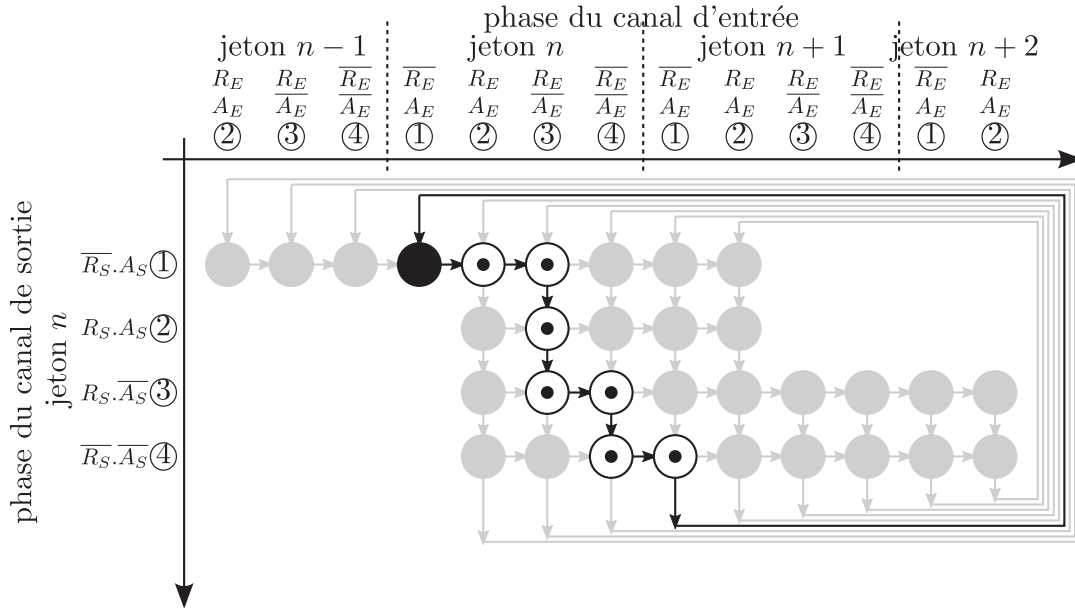


FIGURE 2.12 – Exemples de protocoles séquentiels

- deux portes C-elements (également appelé porte de Muller [MB59]) à deux entrées, présente dans la TIMA Asynchronous Library [FBFR07].

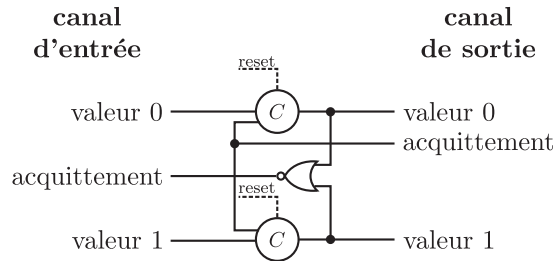


FIGURE 2.13 – Schematic d'un Buffer WCHB

Dans cette réalisation, l'entrée sera acquittée (équation 2.6) seulement après que la donnée ait été propagée dans le canal de sortie (équation 2.7). De même, l'acquittement de l'entrée ne sera enlevé (équation 2.5) qu'une fois la donnée enlevée du canal de sortie (équation 2.8).

$$\overline{R_S} \Rightarrow A_E \uparrow \quad (2.5)$$

$$R_S \Rightarrow A_E \downarrow \quad (2.6)$$

$$R_E \wedge A_S \Rightarrow R_S \uparrow \quad (2.7)$$

$$\overline{R_E} \wedge \overline{A_S} \Rightarrow R_S \downarrow \quad (2.8)$$

La figure 2.14 indique l'ensemble des phases atteignables lors de l'utilisation d'un protocole WCHB. Elle est obtenue à partir de la figure 2.11 en tenant compte des conditions

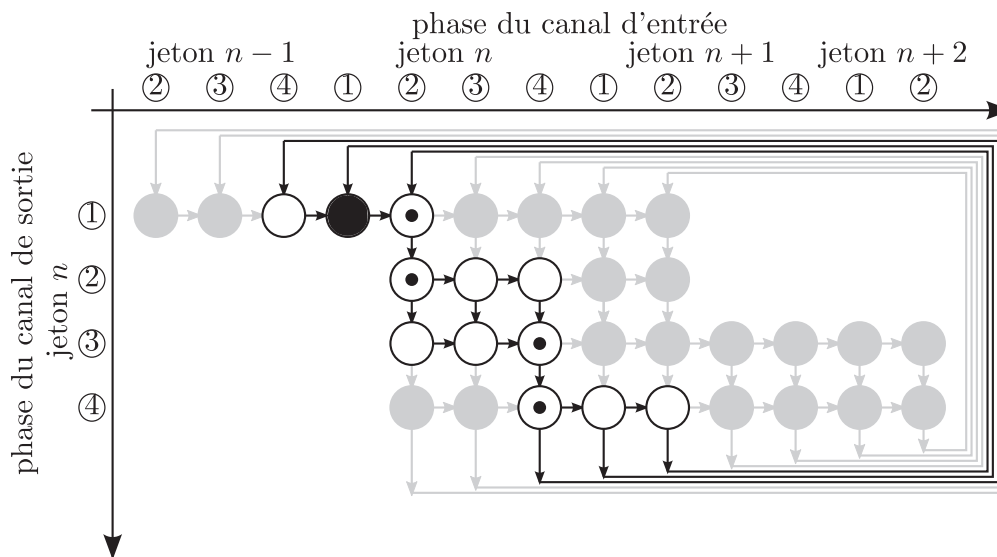


FIGURE 2.14 – Ensemble des phases atteignables d'un buffer WCHB

exprimées par les équations 2.5 à 2.8. Les relations de séquentialité entre les conditions créent des passages obligés par les phases marquées d'un point.

La séquentialité lors de la propagation des données (passage des phases ① à ②) est liée à la causalité du buffer. A l'opposé, la séquentialité lors de la suppression des données (passage des phases ③ à ④) limite fortement les performances d'un pipeline réalisé seulement avec des buffers WCHB : seulement un étage du pipeline sur deux peut traiter une donnée à un instant donné, l'étage précédent et l'étage suivant sont au repos le temps que la donnée traitée se propage. Dans un tel pipeline, on a autant de bulles* que de jetons de données.

Ce protocole découple très peu les phases des canaux d'entrée et de sortie. Le fonctionnement des différents blocs logiques est simultanément et peu étalé dans le temps. Un pipeline réalisé avec des buffers WCHB aura une émission EMI très marquée autour de sa fréquence de fonctionnement [PSR04].

2.7.3 Protocole PCHB

Le protocole *Pre-Charge-Half-Buffer* est une évolution du protocole WCHB dans laquelle les canaux d'entrée et de sortie sont un peu plus découplés : une requête peut être lue sur le canal d'entrée alors que le canal de sortie est toujours occupé par la requête précédente. Un tel buffer nécessite l'utilisation d'un point mémoire supplémentaire tel que représenté dans la figure figure 2.15 : l'acquittement du canal d'entrée est généré par un C-element supplémentaire.

Les règles de transitions de A_E et R_S sont exprimées dans les équations 2.9 à 2.12.

$$\overline{R_E} \wedge \overline{R_S} \Rightarrow A_E \uparrow \quad (2.9)$$

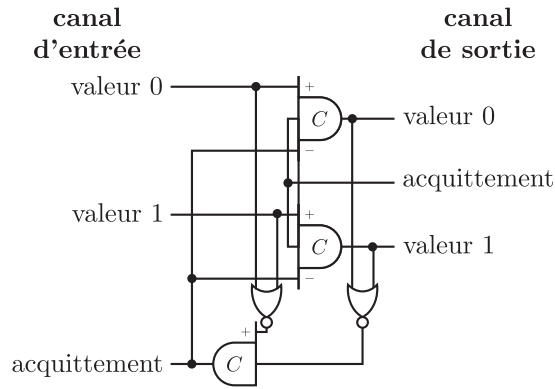


FIGURE 2.15 – Schematic d'un Buffer PCHB

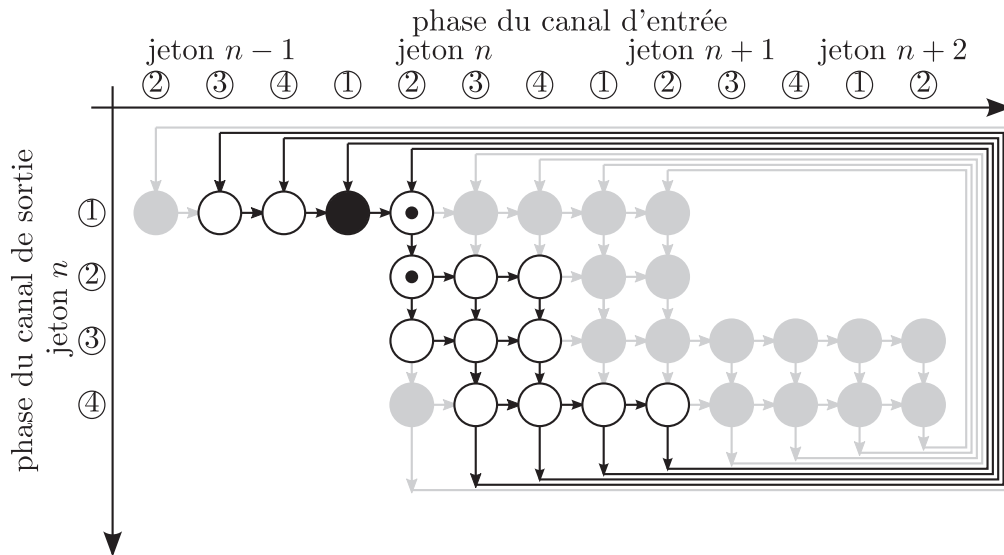


FIGURE 2.16 – Ensemble des phases atteignables d'un Buffer PCHB

$$R_S \Rightarrow A_E \downarrow \quad (2.10)$$

$$R_E \wedge A_S \Rightarrow R_S \uparrow \quad (2.11)$$

$$\overline{A_E} \wedge \overline{A_S} \Rightarrow R_S \downarrow \quad (2.12)$$

La figure 2.16 représente l'ensemble des couples de phases atteignables entre les canaux d'entrée et de sortie lorsque les équations 2.1 à 2.4 et les équations 2.9 à 2.12 sont prises en compte. On remarque que seulement les deux couples de phases marqués d'un point impliquent une forte séquentialité entre les deux canaux : la transmission des requêtes est fortement séquentielle alors que la phase de relaxation correspondant à la suppression des requêtes sur les canaux l'est beaucoup moins. Les requêtes peuvent être supprimées des canaux presque indépendamment l'une de l'autre.

Cette désynchronisation permet au buffer d'accéder à la requête suivante (phase ② correspondant au jeton $n + 1$ canal d'entrée) dès que la requête courante a été retirée du canal d'entrée (phase ④ correspondant au jeton n du canal de sortie). Au niveau du pipeline, cette caractéristique permet au bloc logique en entrée du buffer de commencer le calcul suivant dès que le bloc de sortie a acquitté la requête courante.

2.7.4 Protocole PCFB

Le protocole *Pre-Charge-Full-Buffer* désynchronise plus les canaux d'entrée et de sortie que le protocole PCHB. Le circuit correspondant est donné par la figure 2.17.

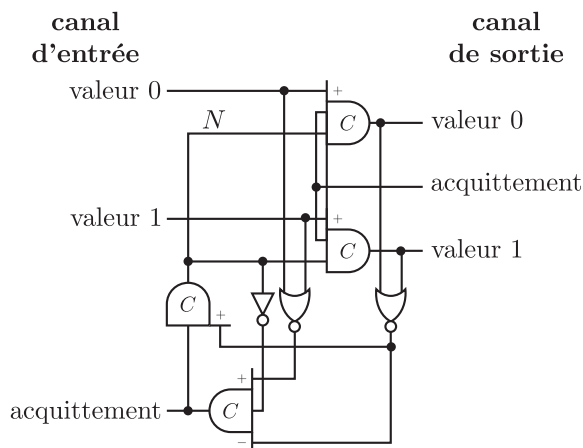


FIGURE 2.17 – Schematic d'un Buffer PCFB

Cette implémentation est régie par les règles de transitions décrites dans les équations 2.13 à 2.18. Le nœud interne, noté N , mémorise le déphasage entre les deux canaux. Sur l'ensemble des phases atteignables représentées en figure 2.18, on remarque que dès que la phase ② du canal de sortie est atteinte, un autre cycle complet de lecture peut commencer sur le canal d'entrée. Sur cette figure, les zones aux contours en pointillés délimitent les phases atteignables où la valeur de N est forcée à l'état logique haut (noté N) ou à l'état logique bas (noté \overline{N}). Dans les autres états atteignables, le

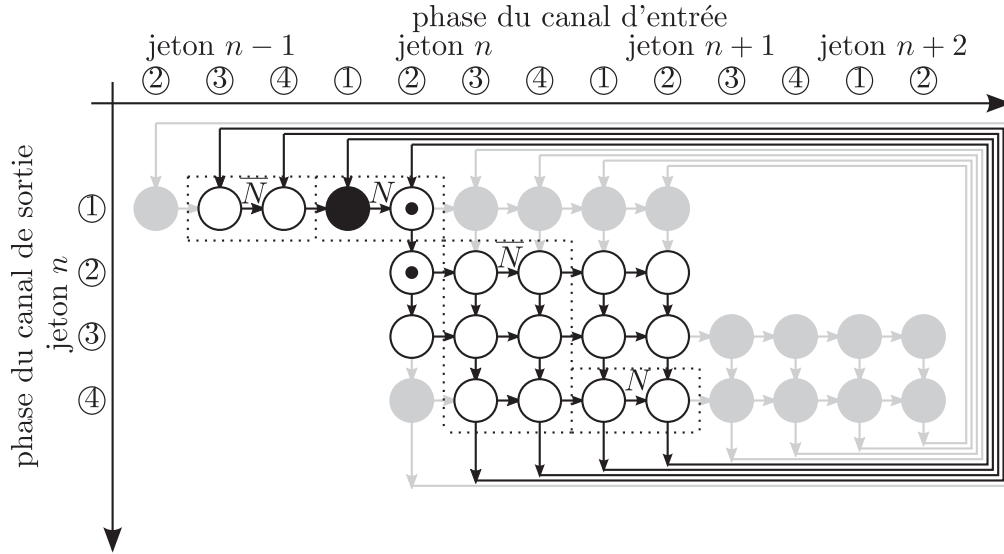


FIGURE 2.18 – Ensemble des phases atteignables d'un buffer PCFB

signal N garde sa valeur précédente. Le signal N indique alors si le canal de sortie peut accepter la requête présente sur le canal d'entrée.

$$A_E \wedge \overline{R_S} \Rightarrow N \uparrow \quad (2.13)$$

$$\overline{A_E} \Rightarrow N \downarrow \quad (2.14)$$

$$\overline{R_E} \wedge \overline{N} \Rightarrow A_E \uparrow \quad (2.15)$$

$$N \wedge R_S \Rightarrow A_E \downarrow \quad (2.16)$$

$$R_E \wedge A_S \wedge N \Rightarrow R_S \uparrow \quad (2.17)$$

$$\wedge \overline{A_S} \wedge \overline{N} \Rightarrow R_S \downarrow \quad (2.18)$$

Ce protocole force encore la synchronisation forte des deux canaux lorsque une requête est placée sur le canal d'entrée. Le buffer doit alors enchaîner les deux couples de phases marqués d'un point avant que les deux canaux puissent évoluer de manière indépendante jusqu'à la prochaine requête : la propagation des données est encore très séquentielle dans un pipeline réalisé avec des buffers PCFB. Lors de la relaxation (suppression de la requête), les deux canaux sont maintenant bien découplés : les données peuvent être enlevées du canal d'entrée ou du canal de sortie sans ordre particulier.

2.7.5 Protocole FDFB

Pour pallier à ce problème, le protocole *Fully-Decoupled-Full-Buffer* a été mis au point [YRL08]. Le circuit réalisant la fonction buffer est formé de deux étages représentés dans la figure 2.19. Les données sont d'abord propagées du canal d'entrée vers un canal intermédiaire avant d'être émises dans le canal de sortie. Le signal interne N contrôle l'avancement de ces données.

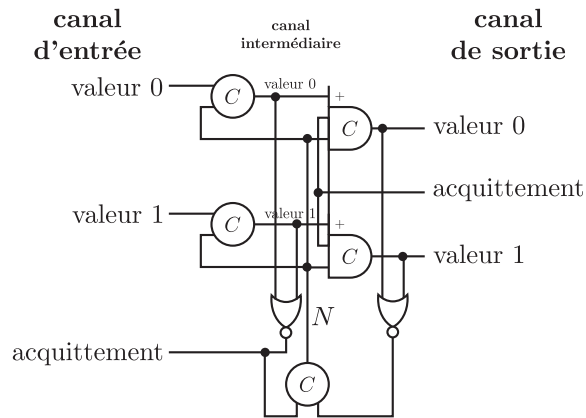


FIGURE 2.19 – Schematic d'un buffer FDFB

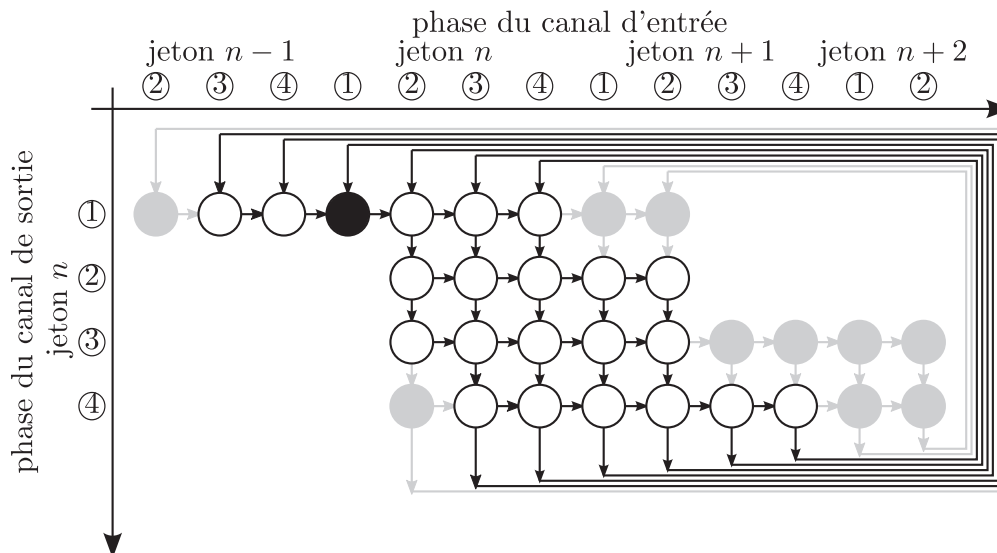


FIGURE 2.20 – Ensemble des phases atteignables d'un buffer FDFB

Les règles de transition suivantes permettent de déterminer l'ensemble des phases atteignables de la figure 2.20 :

$$R_E \wedge N \Rightarrow R_I \uparrow \quad (2.19)$$

$$\overline{R_E} \wedge \overline{N} \Rightarrow R_I \downarrow \quad (2.20)$$

$$\overline{R_I} \wedge \overline{R_S} \Rightarrow N \uparrow \quad (2.21)$$

$$R_I \wedge R_S \Rightarrow N \downarrow \quad (2.22)$$

$$\overline{R_I} \Rightarrow A_E \uparrow \quad (2.23)$$

$$R_I \Rightarrow A_E \downarrow \quad (2.24)$$

$$R_I \wedge A_S \wedge N \Rightarrow R_S \uparrow \quad (2.25)$$

$$\overline{A_S} \wedge \overline{N} \Rightarrow R_S \downarrow \quad (2.26)$$

Ce protocole garantit la propagation et la suppression des données d'une manière moins synchronisée que le protocole PCFB. Cette fois-ci, il n'est plus nécessaire de passer par certains couple de phases lors de la propagation des données. Une fois la première donnée transférée, les données suivantes peuvent être présentes sur le canal d'entrée avant que le canal de sortie ne soit libéré.

Cependant, cette indépendance entre le canal d'entrée et le canal de sortie est principalement liée à la structure du buffer formé de deux étages successifs. L'implémentation d'un seul buffer FDFB nécessite un nombre élevé de C-elements. Finalement la structure de ce circuit ressemble à la mise en série de deux half-buffers, structure étudiée dans la section suivante.

2.7.6 Mise en série de buffers

La mise en série de deux *Weak-Condition-Half-Buffer* (section 2.7.2) forme le buffer à deux étages représenté en figure 2.21.

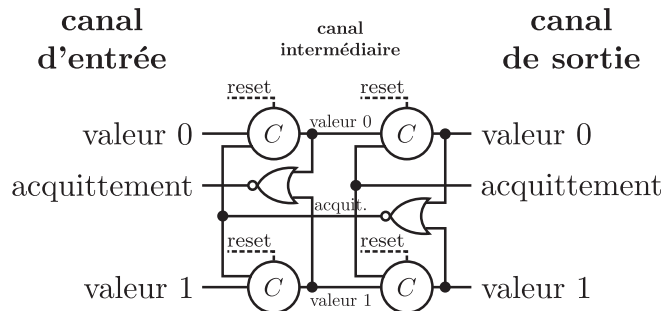
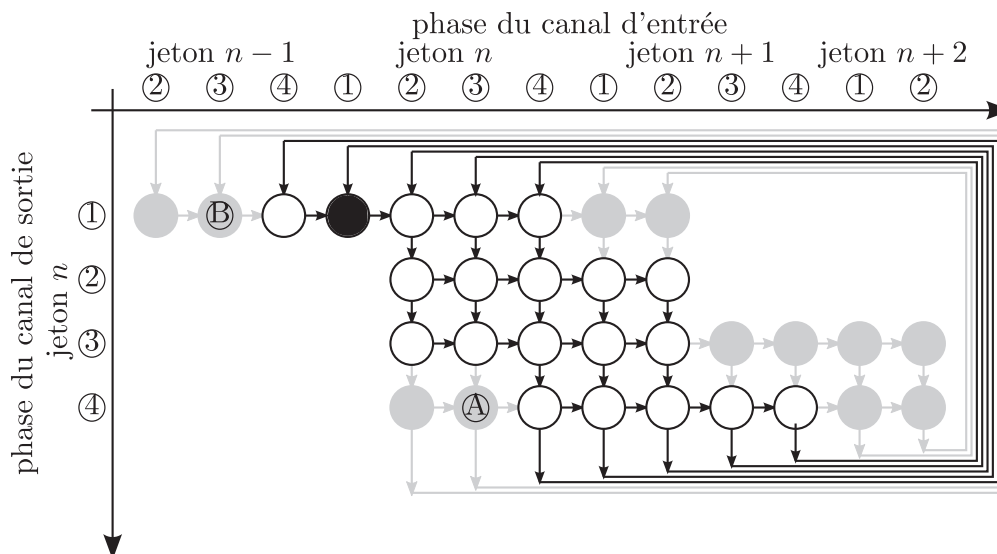


FIGURE 2.21 – Schematic de la mise en série de deux half-buffers WCHB

L'ensemble des couples de phases atteignable est un sous ensemble de celui obtenu avec le protocole FDFB (figure 2.20). Seuls deux couples ne sont pas accessibles :



Ensemble des phases atteignables par deux half-buffer WCHB en série

- les données ne peuvent être retirées du canal de sortie avant d'être retirées du canal d'entrée (couple \textcircled{A}).
- l'acquittement ne peut être retiré du canal de sortie tant que des données sont présentes sur ce canal (couple \textcircled{B}), la désactivation de ce couple de phase est la conséquence de la désactivation du couple \textcircled{A} .

Un circuit asynchrone peut comporter plusieurs encodages de données, plusieurs protocoles et/ou plusieurs synchronisations. Le choix de la synchronisation entre les entrées et les sorties d'un bloc influe sur la vitesse de traitement des jetons de données [Yah09]. Les synchronisations de protocole les plus contraignantes (telles que le WCHB) peuvent apporter de meilleures performances selon la structure du pipeline et la complexité des blocs logiques de traitement. En effet, l'introduction de buffers dans un pipeline permet d'augmenter la profondeur de celui-ci, et donc d'effectuer plus de tâches en parallèle. Cependant cette insertion s'accompagne de délais supplémentaires liés au fonctionnement du buffer. Les half-buffers WCHB ont une structure simple et un temps de cycle très faibles contrairement aux buffers complets tels que le FDFB formés de plusieurs étages.

2.8 Aléas de commutation

Dans la section précédente, les règles de transition dans les canaux de communication sont sensibles à chaque changement d'état. Lors de la conception d'un bloc logique, il est nécessaire de bien maîtriser l'évolution des différents signaux de requête et d'acquittement pour éviter la génération de jetons de donnée intempestives. Une donnée est présente dans un canal de communication dès que l'encodage représente une donnée valide, de même une donnée est considérée comme acquittée dès que le signal d'acquittement devient actif. Il faut donc s'assurer que les signaux générés par un bloc logique le seront volontairement et que les blocs logiques émettront jamais de variations intempestives.

tives lors de leur fonctionnement interne. On parle d'*aléa** de commutation lorsqu'une porte logique émet transitoirement une ou plusieurs valeurs incorrectes lors du passage entre deux valeurs correctes.

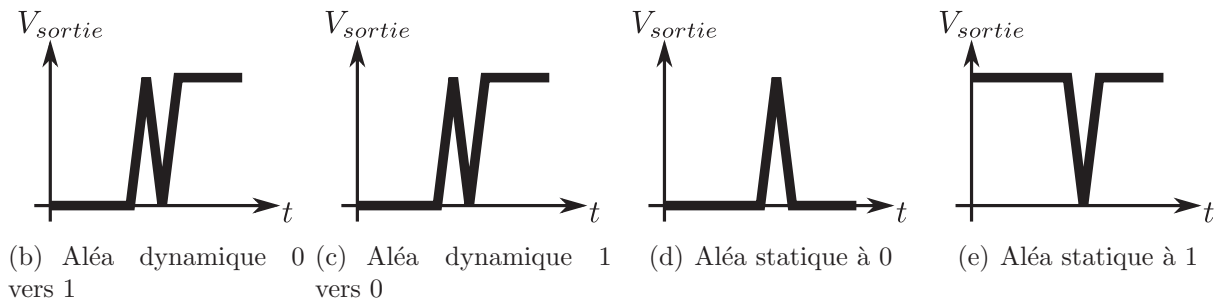


FIGURE 2.22 – Aléas statiques et dynamiques

Les aléas peuvent être classés en deux catégories :

- Les aléas statiques peuvent apparaître lors du changement du vecteur d'entrée d'un bloc combinatoire lorsque l'ancienne et la nouvelle valeur de sortie sont identiques (figures 2.22(d) et 2.22(e)). Pendant un court instant, la sortie peut changer d'état, le concepteur de circuits parle de ce phénomène sous le nom de *glitch*.
- Les aléas dynamiques interviennent lors du changement de la valeur après un changement du vecteur d'entrée. Les propagations dans les éléments logiques peuvent introduire des fronts temporaires inversant la tendance (figures 2.22(b) et 2.22(c)).

La conception d'un circuit asynchrone nécessite donc l'étude des cas transitoires liés aux variations des signaux d'entrées des blocs dans un ordre quelconque. Il faut s'assurer de l'absence de création d'aléas dans les blocs combinatoires quelques soient les transitions. Dans le cas contraire, ces aléas pourraient être interprétés comme autant de données supplémentaires.

2.8.1 Conception logique sans aléa

Les aléas peuvent intervenir lorsqu'une seule entrée varie à la fois, on parle alors de *Simple Input Change hazard* (SIC). Lorsque plusieurs entrées varient ensemble, les aléas sont alors qualifiés de *Multiple Input Change hazard* (MIC).

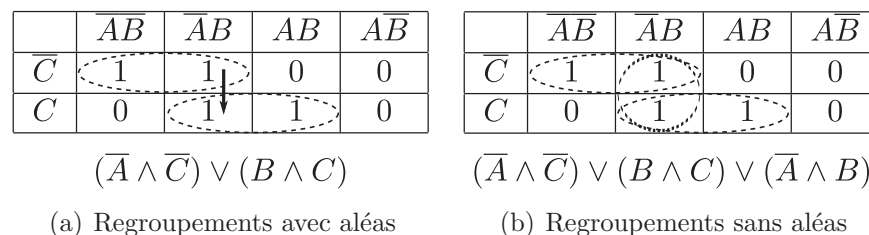


FIGURE 2.23 – Principe de suppression des aléas d'après une table de Karnaugh

Les aléas peuvent être éliminés en ajoutant des combinaisons logiques supplémentaires dans la table de Karnaugh correspondant à la fonction implémentée [Ung83, ND95, TNW96]. La figure 2.23(a) indique les états habituellement regroupés afin de minimiser le nombre de termes de l'expression logique. Cependant lors d'un passage de l'état $\overline{A} \wedge B \wedge \overline{C}$ vers l'état $\overline{A} \wedge B \wedge C$, la sortie peut transitoirement ne plus être maintenue à l'état logique 1. Pour résoudre ce problème, il est nécessaire d'ajouter le terme $\overline{A} \wedge B$ qui sera actif tout au long de cette transition. On obtient alors les regroupements représentés dans la figure 2.23(b).

D'autres méthodes basées sur les CSG [JMG00, MJ01] permettent de concevoir des C-elements généralisés (gC) spécifiques à la fonction implémentée. Lors de transitions simultanées sur de multiples entrées, le point mémoire du C-element est mis à contribution pour éviter les basculements intempestifs. L'utilisation de ces C-elements spécifiques rend plus complexe la conception d'un circuit intégré avec cette méthode : il faut réaliser le *layout** de toutes ces *standard-cells** avant de réaliser le placement-routage du circuit. La réalisation du *layout** des *standard-cells** est une opération généralement effectuée manuellement, donc très coûteuse en temps de développement. Il existe maintenant plusieurs outils permettant d'automatiser la génération des vues *layout** tels que Synopsys Cadabra [GMD⁺97, CAD].

2.9 Fonctionnement des circuits asynchrones

A un niveau de modélisation numérique *Gate-Level**, les entrées des portes sont des valeurs binaires, dont la valeur est soit 0, soit 1. Leurs valeurs sont discrètes mais ces signaux évoluent à temps continu.

Or lors d'un front montant ou descendant, les tensions représentant ces différents signaux varient continuellement. Les capacités intrinsèquement présentes dans le circuit (capacités des grilles des transistors et capacités parasites entre-lignes ou avec le substrat) interdisent les discontinuités de tension. Une signal basculant d'un niveau logique 0 (niveau généralement associé à la tension V_{SS}) vers un niveau logique 1 (associé à V_{DD}) voit donc sa tension augmenter sans discontinuité jusqu'à atteindre sa valeur finale.

Les portes logiques asynchrones sont actives sur état et non sur front comme le sont les bascules synchrones. La valeur de leur sortie peut varier dès que les tensions de leurs entrées sont assez marquées :

- soit proche de V_{SS} . Dans ce cas l'entrée est vue comme un niveau logique 0. La borne haute de cette zone sera notée V_{TL} dans la suite du manuscrit : elle est définie comme la tension en dessous de laquelle la porte interprète toujours un niveau logique 0 (équation 2.27).

$$V_{SS} < V_{nœud} < V_{TL} \tag{2.27}$$

- soit proche de V_{DD} . Elle sera alors vue comme un niveau logique 1. La borne basse de cette zone notée V_{TH} est définie comme la plus petite tension au dessus de laquelle la porte interprète toujours un niveau logique 1 (équation 2.28).

$$V_{TH} < V_{nœud} < V_{DD} \quad (2.28)$$

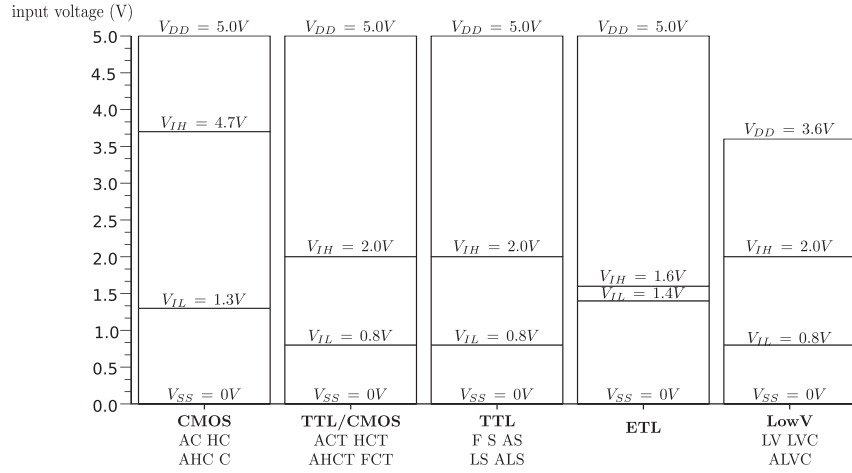


FIGURE 2.24 – Tensions de basculements dans les principales technologies de logique discrète

Un parallèle peut être établi avec les tensions de basculements des technologies logiques discrètes représentées dans la figure 2.24 [TEC]. Mais ces tensions de références technologiques ont une définition différente :

- V_{IL} est définie comme étant la plus grande tension qui sera *toujours* interprétée comme un niveau logique 0 par toutes les portes de la technologie ;
- V_{IH} est la plus petite tension *toujours* interprétée comme un niveau logique 1 dans les mêmes conditions.

Les tensions de basculements V_{TL} et V_{TH} sont des tensions de basculement effectives. Elles sont spécifiques à chaque entrée de chaque porte logique (*standard-cell**) et dépendent des conditions d'opération du circuit : tension d'alimentation, variations de caractéristiques de la technologie de fabrication, vieillissement du circuit . . . Des *standard-cells** différentes n'auront donc pas effectivement les mêmes tensions de basculement. De même, différentes instances d'une même *standard-cell** peuvent avoir des tensions de basculement différentes à cause des mismatches et des gradients spatiaux de variations de process dus à la fabrication du circuit.

Cependant, pour toutes les portes d'une technologie donnée, les relations des inéquations 2.29 et 2.30 doivent être respectées afin que d'éviter les incompatibilités entre sorties et entrées des portes logiques.

$$V_{IL} < V_{TL} \quad (2.29)$$

$$V_{TH} < V_{IH} \quad (2.30)$$

La disparité des tensions de basculements effectives V_{TL} et V_{TH} est à éviter dans les circuits asynchrones QDI : leurs différences participent au déséquilibre des fourches isochrones [vB92]. Cependant, Il est physiquement impossible de les éliminer complètement.

Plusieurs portes logiques peuvent potentiellement interpréter une tension donnée comme un niveau logique 0 alors que d'autres l'interprèteront comme un 1.

Nous avons vu précédemment que les circuits asynchrones fonctionnent à temps continu et que leur comportement peut varier en fonction de tensions analogiques. Par conséquent, seule une modélisation analogique peut reproduire fidèlement le fonctionnement d'un circuit asynchrone. Une modélisation purement numérique ne peut pas représenter tous les comportements des circuits asynchrones dans le cas général. Ces modélisations numériques sont néanmoins possibles si le comportement du circuit vis-à-vis des tensions analogiques est ignoré.

Chapitre 3

Structure des portes C-elements

| | |
|------------------------------------------------------------------|-----------|
| Introduction | 34 |
| 3.1 Implémentations full-custom | 34 |
| 3.1.1 Structures habituelles | 35 |
| 3.1.2 Autres structures | 37 |
| 3.2 Implémentations structurelles | 38 |
| 3.2.1 A base de portes logiques | 38 |
| 3.2.2 Sur composants programmables | 39 |
| 3.3 Implémentation de C-elements à n entrées | 39 |
| 3.3.1 Décomposition avec les portes d'une bibliothèque | 40 |
| 3.3.2 Rationalisation des cellules spécifiques | 41 |
| 3.4 Fonctions apparentées au C-elements | 42 |
| 3.4.1 Implémentation en logique différentielle | 42 |
| 3.4.2 Fonction C-element généralisée | 43 |
| Conclusion | 44 |

Introduction

L'implémentation de circuits asynchrones SI, QDI et DI nécessite l'utilisation de portes spécifiques : les C-elements. La fonction C-element correspond à un point mémoire dont la valeur de sortie est affectée par la valeur des entrées lorsque celles-ci sont égales. Son comportement numérique a été détaillé et formalisé dans la [section 2.6 \(page 15\)](#). Le nombre d'entrées, la présence de signaux de reset ou de set, les dissymétries entre les entrées induisent un nombre très importants de portes logiques différentes.

Il existe de nombreuses implémentations possibles pour la fonction C-element dans l'état de l'art de la logique asynchrone. Une réalisation utilisera généralement une seule des structures suivantes. Ce choix de structure est essentiellement guidé par la technologie de fabrication du circuit et par les performances souhaitées.

3.1 Implémentations full-custom

Lorsque le circuit asynchrone est un ASIC (*Application Specific Integrated Circuit*), le flot de conception comprend la génération d'un modèle *Gate Level* : c'est une structure à base de cellules logiques normalisées (*standard-cells*) inter-connectées afin de réaliser la fonction souhaitée. L'étape de placement-routage permet ensuite d'obtenir les "plans" des différents masques utilisés pendant les étapes de fabrications.

Certaines technologies permettent d'ajouter des standard-cells personnalisées dans le flot de conception. Ces bibliothèques additionnelles peuvent alors contenir les fonctions C-elements souhaitées que l'outil de placement-routage utilisera au même titre que les cellules logiques fournies avec la technologie : la sortance des cellules sera ajustée "aux charges" connectées à leurs sorties.

Par exemple, lorsque de nombreuses entrées sont connectées à une même porte, ou lorsque le fil connecté à la sortie est long, ou lorsque les contraintes temporelles impliquent des transitions très rapides afin de transmettre l'information le plus vite possible, l'outil de placement-routage va augmenter la sortance de cette porte. Cela correspond à utiliser une variante de la porte où l'étage de sortie est surdimensionné afin de supporter des courants plus élevés. Le concepteur doit alors fournir les vues *layout* (dessin de tous les masques) correspondant à toutes ces cellules C-elements utilisées, mais aussi décliner ces cellules en plusieurs sortances.

Les bibliothèques de la TIMA Asynchronous Library (TAL) [[FBFR07](#)] développées au sein de l'équipe CIS proposent des standard-cells pour l'asynchrones pour les technologies STMicroelectronics. Les TAL se déclinent en deux technologies : la version 130nm contient entre-autre 98 cellules C-element différentes et la version 65nm a 42 C-elements.

Il est nécessaire de vérifier le comportement correct de toutes ces cellules afin d'éviter les dysfonctionnements de circuits asynchrones conçus avec ces bibliothèques.

3.1.1 Structures habituelles

L'état de l'art asynchrone fait apparaître deux structures principales de C-elements : la structure *Weak-Feedback* [Sut89] dont le schéma au niveau transistor est représenté en figure 3.1(a) et la structure conventionnelle [Mar90a] représentée en figure 3.1(b). Ces structures forment des points de mémoire *statique*, où la donnée est mémorisée pendant un temps non-borné. Ce modèle de temps permet à ces structures d'implémenter des circuits asynchrones SI, QDI ou DI sans restriction.

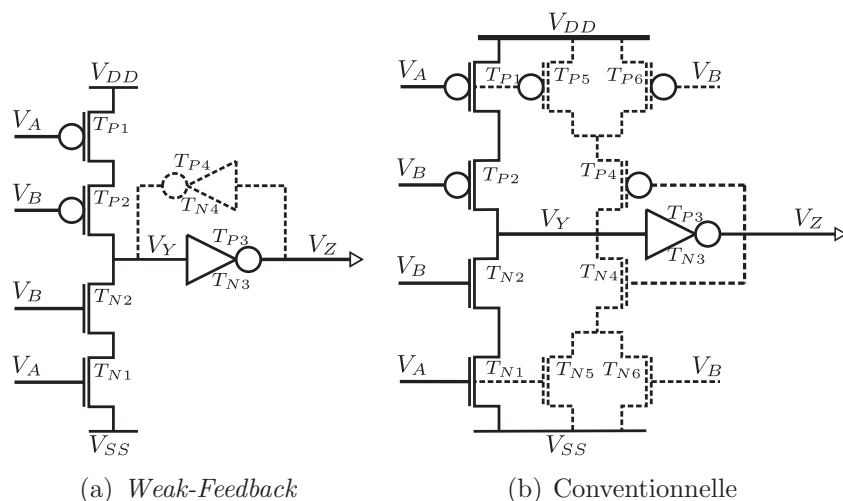


FIGURE 3.1 – Principales structures de C-elements en full-custom

Ces deux structures ont un étage d'entrée, formé des transistors T_{N1} , T_{N2} , T_{P1} et T_{P2} , capable de faire basculer le point mémoire formé des autres transistors. La structure de ce point mémoire est classique, il s'agit de deux inverseurs placés tête bêche. Un de ces inverseurs, formé de T_{N3} et T_{P3} est également responsable de la sortie de la porte : les largeurs de ces transistors seront augmentées pour assurer cette sortance. L'autre inverseur (appelé *keeper**) est interne à la structure. Afin de minimiser l'amplitude des pics de courants lors des basculements du point mémoire, les transistors le composant auront un dimensionnement minimal.

Dans la structure conventionnelle, le *keeper** est modifié afin de faciliter les transitions : il est automatiquement désactivé lorsque les deux entrées ont des valeurs égales : l'étage d'entrée peut alors fixer la tension V_Y sans avoir à absorber le courant du *keeper** dont le rôle est de conserver la valeur mémorisée et donc de lutter contre le basculement.

Les C-elements de la bibliothèque TAL 130nm sont du type *Weak-Feedback*. Ce choix était principalement la conséquence de contraintes de surface : une M2 (C-element à deux entrées symétriques) *Weak-Feedback* nécessite huit transistors alors que la structure conventionnelle de cette même fonction en requiert douze. A l'opposé, la TAL 65nm ne contient que des structures conventionnelles, celle-ci permet des transitions plus douces avec des pics de courant plus faibles.

Le dimensionnement de ces transistors est effectué manuellement : leur longueur (notée

L) est fixée aux dimensions minimales de la technologie, leur largeur (notée W) est établie par itération afin d'obtenir les caractéristiques en délai de propagation, en temps de transition et en consommation comparables avec celles des cellules standards de la technologie visée. Ce processus manuel est non déterministe et plusieurs résultats peuvent être obtenus pour une même porte. Ces critères de dimensionnement ne comprennent pas les mesures des seuils tels que décrits dans le [chapitre 5 \(page 65\)](#).

Ces structures de C-elements peuvent aisément être adaptées pour ajouter des entrées, des dissymétries, des signaux de set/reset ou augmenter la sortance :

- l'ajout d'une entrée supplémentaire se traduit par deux transistors supplémentaires insérés dans l'étage d'entrée : un transistor NMOS et un transistor PMOS insérés en série respectivement avec T_{N1} et T_{N2} et en série avec T_{P1} et T_{P2} . Dans le cas d'un C-element conventionnel, il faut également ajouter un transistor NMOS en parallèle avec T_{N5} et T_{N6} et un PMOS avec T_{P5} et T_{P6} .
- l'ajout d'une dissymétrie sur une entrée correspond à la suppression d'un des transistors correspondant de l'étage d'entrée : une dissymétrie négative est implémentée en supprimant le transistor NMOS de cette entrée et une dissymétrie positive le transistor PMOS.
- les signaux de set ou de reset se traduisent par l'ajout d'un transistor entre le nœud Y et un rail d'alimentation : un set actif à l'état haut est implémenté en ajoutant un transistor NMOS entre Y et V_{SS} , un reset actif à l'état bas avec un PMOS entre Y et V_{DD} .
- la sortance de la porte est assurée par les transistors T_{N3} et T_{P3} . Leur rapport W/L fixe le courant maximal admissible par la sortie de cette porte.

Toutes ces modifications influent sur le comportement de la porte. Il est donc nécessaire d'itérer à nouveau le dimensionnement des transistors tel qu'expliqué précédemment afin de rectifier les caractéristiques souhaitées. Chaque variante de la fonction C-element correspond à un réseau de transistor différent en structure et/ou en dimensionnement, par conséquent les vues *layout* sont à redessiner pour chaque cellule de la bibliothèque.

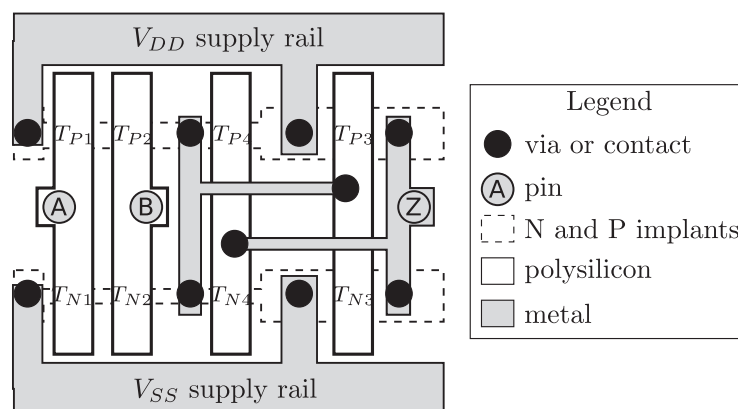


FIGURE 3.2 – Layout d'une structure *Weak-Feedback*

La [figure 3.2](#) représente la disposition classique du *layout* pour une structure *Weak-*

Feedback à deux entrées. Chaque transistor est obtenu à l'intersection d'une ligne de dopage (*implants*) et d'une colonne de conducteur de grille (*polysilicon*). Les règles de dessin imposent de placer les transistors PMOS près du rail d'alimentation V_{DD} et les NMOS près de V_{SS} . Chaque série de transistors est donc alignée et des motifs en métal réalisent les interconnexions entre les grilles et les drains des transistors, les sources étant reliées au conducteur d'alimentation le plus proche.

3.1.2 Autres structures

D'autres structures de C-elements full-custom existent dans la littérature, telle que la structure dynamique représentée en figure 3.3(b) [MND91] ou la structure symétrique [vB92] (figure 3.3(a)).

La structure symétrique est un point de mémoire statique, souvent cité en exemple de réalisation pour la fonction C-element. Sa structure se complique considérablement dès que la porte a plus de deux entrées. Il semble que cette structure n'ait jamais été utilisée pour réaliser un circuit réel.

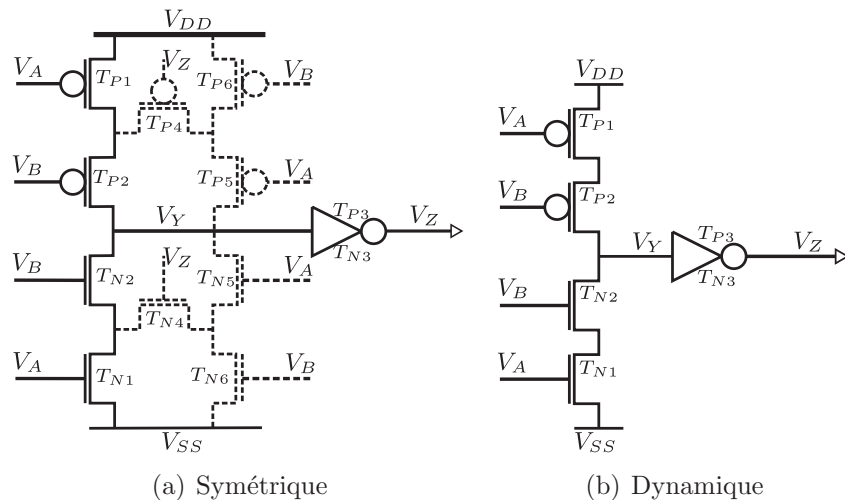


FIGURE 3.3 – Autres structures de C-elements en full-custom

La structure dynamique permet d'implémenter la fonction C-element avec seulement quatre transistors. Elle a un fonctionnement très rapide car l'absence de *keeper** réduit la capacité équivalente du nœud Y : l'étage d'entrée est donc capable d'inverser très rapidement la tension V_Y pour basculer le point mémoire. Par contre l'absence de *keeper** implique que la donnée mémorisée n'est que temporairement valide. Les courants de fuite et les éventuelles perturbations peuvent perturber le potentiel V_Y et donc altérer la donnée mémorisée au fur et à mesure du temps. Les fonctions C-elements implémentées avec cette structure ont donc un modèle de fonctionnement borné dans le temps : après un certain temps de validité, caractéristique de la technologie employée et du dessin du *layout*, la donnée *doit* être considérée comme invalide. Ce temps caractéristique peut être augmenté en augmentant la capacité du nœud Y vis-à-vis de la masse principalement au détriment de la vitesse et de l'énergie de commutation.

La structure dynamique peut être utilisée pour la conception de circuits micropipelines où les très faibles hypothèses temporelles nécessitent un modèle de temps borné pour les portes.

Leur utilisation dans des circuits asynchrones SI, QDI ou DI n'est pas possible car ces classes de circuits sont à temps non-borné : les C-elements doivent être en mesure de conserver leur donnée pendant un temps potentiellement infini. Il est cependant possible d'utiliser des C-elements dynamiques dans une structure de type SI, QDI ou DI tant que le temps de cycle est inférieur à la durée de rétention de l'information dans le C-element. On parle alors de circuits *burst-mode* : la donnée est toujours valide pendant un *court* cycle et est toujours mise à jour pour passer au cycle suivant. La conception de tels circuits asynchrones burst-mode se rapproche de la conception traditionnelle de circuits synchrones [YD92].

3.2 Implémentations structurelles

L'utilisation de standard-cells pour l'asynchrone telles que les structures au niveau transistor n'est pas toujours possible. Certaines technologies d'ASIC ne permettent pas des vues *layout* personnalisées. Lorsque la cible est un composant programmable de type FPGA, il faut obtenir le même comportement en s'adaptant aux ressources disponibles.

3.2.1 A base de portes logiques

La conception et le dimensionnement des transistors puis le dessin des masques du *layout* nécessitent d'avoir un design-kit de la technologie cible et de respecter les règles de dessin édictées par le fondeur. En l'absence de ces pré-requis, il est néanmoins possible de réaliser des assemblages à partir de portes standard (NAND/AND/NOR/OR/... issues du fondeur) réalisant la fonction C-element recherchée.

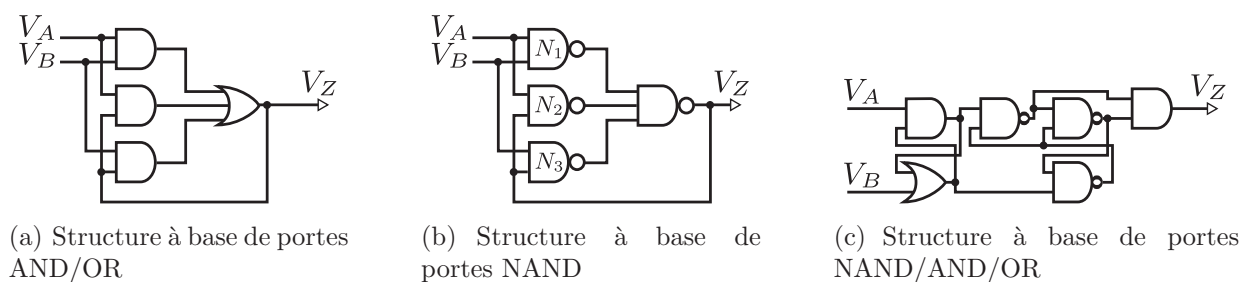


FIGURE 3.4 – Structures à base de portes logiques existantes

Il existe de nombreuses combinaisons de portes réalisant la fonction C-element à deux entrées [Fig98, Rig02, SB05]. Celles-ci sont généralement extraites de la description comportementale de la fonction représenté par l'équation 3.1. Cette structure de base est donnée en figure 3.4(a).

$$Z = (A \wedge B) \vee (A \wedge Z) \vee (B \wedge Z) \quad (3.1)$$

La logique CMOS étant naturellement complémentée, il est plus aisé de réaliser des implémentations à bases de portes inverseuses telles que représentées en [figure 3.4\(b\)](#) : son fonctionnement sera généralement plus rapide que celui d'une structure à base de portes non-inverseuses.

D'une manière générale, le fonctionnement correct de ces structures doit être assuré par des contraintes de placement routage très fortes : les interconnexions à l'intérieur du C-element doivent être courtes comparées aux autres interconnexions de l'environnement de la porte. Cela permet à la structure de mémoriser la donnée avant que celle-ci puisse être retirée par l'environnement.

Des structures plus complexes telles que représentées dans la [figure 3.4\(c\)](#) sont également possibles. Il existe un nombre potentiellement infini de réalisations basées sur une interconnexion de cellules standards. Ces trois structures semblent les plus intéressantes pour la suite de l'étude.

3.2.2 Sur composants programmables

Les circuits programmables classiques de type FPGA peuvent également être utilisés pour implémenter des circuits asynchrones [[Rig02](#)]. Le principe est le même que celui de la section précédente : il s'agit de déterminer l'équation "statique" de la porte et d'utiliser un outil de synthèse classique pour réaliser la synthèse.

Le processus de placement-routage à l'intérieur d'un FPGA est difficilement contrôlable et il est très difficile voire impossible d'insérer des règles concernant la longueur des interconnexions pour certains conducteurs. Afin de garantir le comportement correct des C-elements implémentés sur FPGA, il est nécessaire de forcer leur synthèse de chaque C-element à l'intérieur d'une macro-cell, les fabricants de FPGA permettent généralement l'ajout de contraintes de ce type. Les macro-cells possèdent intrinsèquement un rebouclage entre leur sortie et les entrées de la Look-Up-Table (LUT), celle-ci est mise à contribution pour assurer le rebouclage de nos structures de C-elements.

3.3 Implémentation de C-elements à n entrées

Les limitations technologiques interdisent généralement la mise série de plus de quatre transistors lors d'une implémentation *full-custom*, cela limite le nombre d'entrées pour une cellule donnée : chaque insertion d'un transistor en série provoquant une chute de tension supplémentaire, cet empilement de transistors (*transistor stacking*) peut compromettre l'intégrité du niveau logique obtenu. Sur FPGA, le nombre d'entrées de la *macro-cell* limite également le nombre maximal d'entrées de la fonction.

Or, le flot de synthèse peut potentiellement nécessiter des portes avec un nombre d'entrées bien supérieur à ce que la technologie permet de réaliser en une seule cellule. Les

outils de synthèse utilisent différentes techniques pour générer des structures équivalentes à l'aide des cellules disponibles.

3.3.1 Décomposition avec les portes d'une bibliothèque

La fonction C-element peut être décomposée afin de réaliser des portes avec plus d'entrées. Cette décomposition doit se faire sans introduction d'aléas et en utilisant au mieux les ressources disponibles [FBFR07, Rig02, SDM94].

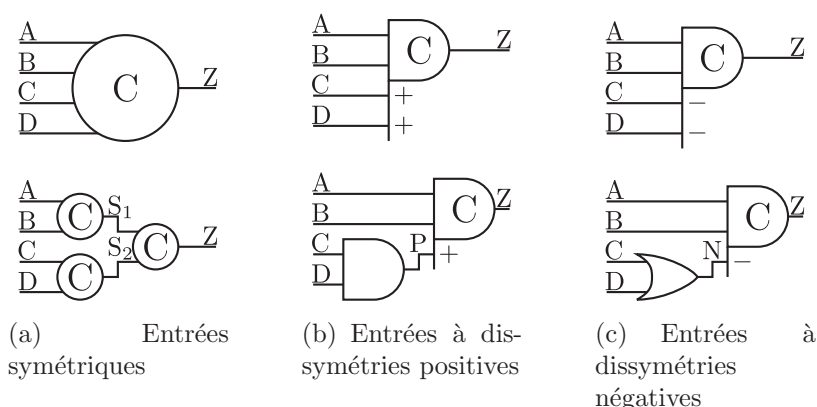


FIGURE 3.5 – Décomposition des C-elements

Plusieurs entrées symétriques peut être regroupées à l'aide de plusieurs C-elements symétriques (figure 3.5(a)). Plusieurs entrées avec une dissymétrie positives peuvent être regroupées à l'aide d'une porte logique AND (figure 3.5(b)). De même, plusieurs dissymétries négatives peuvent être regroupées à l'aide d'une porte logique OR (figure 3.5(c)).

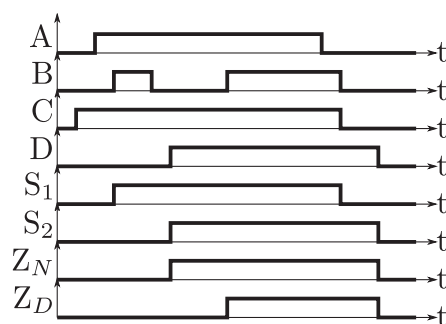


FIGURE 3.6 – Aléas dans la décomposition d'une porte à entrées symétriques

Le regroupement de plusieurs entrées à l'aide d'autres portes logiques est correct si et seulement si il n'y a pas d'aléas sur les entrées. Dans le cas contraire, le comportement de la fonction décomposée peut devenir différent de celui de la fonction souhaitée (non décomposée). La figure 3.6 expose ce problème. Les quatre signaux A , B , C et D correspondent aux entrées d'un C-element à quatre entrées symétriques. S_1 et S_2 sont

les signaux internes de la fonction décomposée (figure 3.5(a)) dont la sortie est notée Z_D . La sortie de la fonction C-element non-décomposée est tracée sous le nom Z_N . Un aléa sur l'entrée B fait basculer le nœud interne S_1 au niveau logique 1. Cette fausse information étant mémorisée, elle est propagée jusqu'à la sortie et provoque un basculement de la sortie de la fonction décomposée avant celui de la fonction de référence non-décomposée. D'une manière générale, tout changement d'état dans un circuit doit être acquitté avant d'être retiré. Dans le cas contraire, ces changements intempestifs sont considérés comme *aléas* et peuvent provoquer des dysfonctionnements du circuit.

3.3.2 Rationalisation des cellules spécifiques

Les méthodes basées sur les décompositions de fonctions requièrent un nombre élevé de standard-cells afin d'être efficaces. Afin de limiter le temps de développement des bibliothèques de cellules spécifiques aux circuits asynchrones, il est intéressant de se limiter à un nombre très limité de cellules spécifiques autour desquelles toutes les variations sont réalisées avec des cellules standards (NAND/NOR...).

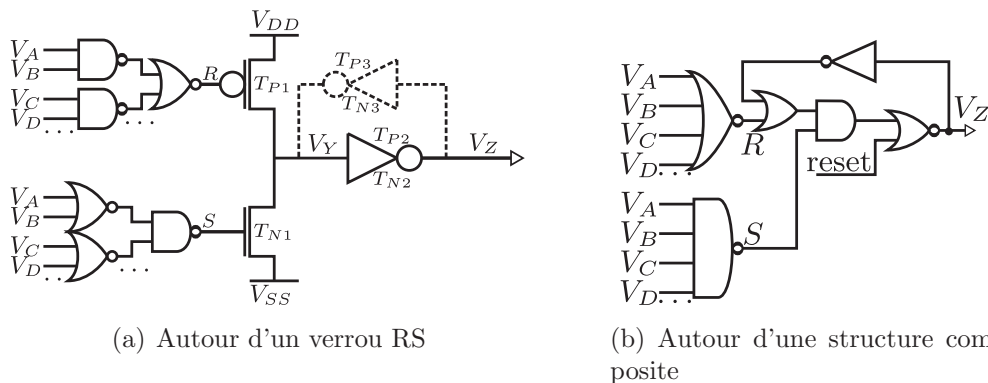


FIGURE 3.7 – Composition des C-elements à l'aide d'arbres NAND/NOR

La structure représentée par la figure 3.7(a) peut être décomposée en deux parties distinctes [YS10] : un verrou RS (reset/set) asynchrone [vB92] assez similaire à une porte C-element *Weak-Feedback* et un étage de spécialisation pour générer la fonction de mémorisation souhaitée. Ce découpage autorise d'une part la conception *full-custom* du verrou (avec peu de variations : seulement pour créer plusieurs sortances) et d'autres part la réutilisation de cellules standards NAND et NOR qui sont disponibles avec la technologie. Ces cellules standards ont elles-aussi un nombre maximal d'entrées, limitation contournable en réalisant une structure d'arbres. Dans cette structure, il est possible de créer des dissymétries positives en retirant une entrée de l'arbre R (de même pour les dissymétries négatives : en retirant l'entrée de l'arbre S).

Il est également possible de créer des fonctions mémorisantes à n entrées seulement avec des cellules standards [WV93] (figure 3.7(b)). Comme précédemment, la fonction logique sera spécifiée par l'outil de synthèse au travers des structures des arbres S et R .

Ces structures formées d'arbres NAND/NOR ne souffrent pas de sensibilités aux aléas

autant que les arbres de C-elements vus dans la [section 3.3.1](#) : aucune porte formant l'arbre n'est mémorisante. Au contraire, les fonctions NAND/NOR ont un comportement absorbant : un aléa sur une entrée sera la plupart du temps masqué par l'état des autres entrées interdisant le basculement du point mémoire.

3.4 Fonctions apparentées au C-elements

La fonction C-element vue dans ce chapitre est très proche d'autres points mémoires présents dans l'état de l'art de la conception logique. L'étude du fonctionnement des C-elements présenté ultérieurement dans ce manuscrit révèle des problèmes également présents dans ces approches.

3.4.1 Implémentation en logique différentielle

Les structures statiques des C-elements sont construites autour d'un point mémoire statique formé de deux inverseurs tête bêche. Dans une telle structure, à l'exception des instants de basculement, les niveaux logiques des deux nœuds (notés Y et Z dans les figures précédentes) sont toujours opposés. Cette caractéristique permet de concevoir à moindre coût des circuits asynchrones à logique différentielle. Cependant, lors des instants de basculement, les potentiels de Y et Z peuvent transitoirement représenter la même valeur logique. Ce phénomène, assimilable à un aléa, est dû aux délais de propagation de l'inverseur de Y à Z . Pour réaliser une implémentation asynchrone à logique différentielle exempte de ce problème, il est nécessaire de concevoir d'autres structures de C-elements [SEE97, Shi09].

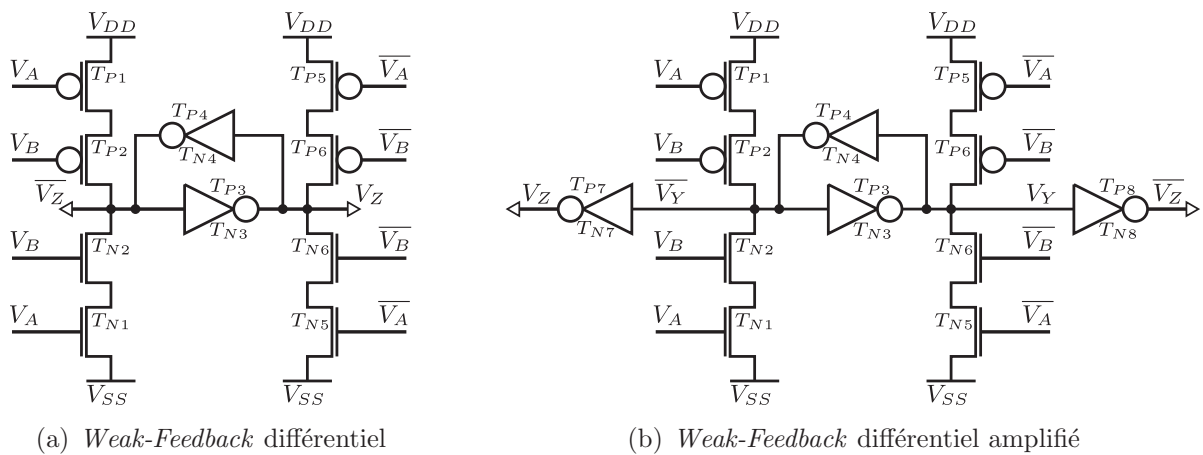


FIGURE 3.8 – C-elements en logique différentielle

La [figure 3.8\(a\)](#) représente une des implémentations proposées en logiques différentielles. Les potentiels marqués d'une barre ($\overline{V_X}$) représentent le fil "négatif", ils forment une paire différentielle avec le fil "positif" associé (V_X). Cette variante de la structure *Weak-Feedback* commute plus rapidement que la structure originelle : puisque l'étage

d'entrée dupliqué bascule en même temps les potentiels V_Z et $\overline{V_Z}$, il n'y a pas de délai de propagation dans les inverseurs du point mémoire. Dans cette structure, les sorties de la portes sont directement reliées aux inverseurs du point mémoire. Par conséquent, les inverseurs devront donc être (sur)dimensionnés pour apporter la sortance nécessaire à la porte.

De même, il est nécessaire de sur-dimensionner les deux étages d'entrée afin qu'ils soient en mesure d'absorber tout le courant émis par les inverseurs et ainsi provoquer la commutation. Ce surdimensionnement peut devenir très important lors de la conception de portes à sortance élevée. Afin de résoudre ce problème, d'autres structures telles que celle représentée en [figure 3.8\(b\)](#) ont été proposées. Elles comprennent deux inverseurs supplémentaires pour émettre les signaux de sortie, le dimensionnement du point mémoire et des étages d'entrées est donc beaucoup moins contraints par la sortance et peut se faire à de très faibles largeurs de transistors.

En logique asynchrone, il est nécessaire de limiter le décalage entre les deux rails positifs et négatifs. En effet, une propagation d'information plus rapidement sur un des deux rails est susceptible de causer les mêmes effets d'aléas que vus précédemment. Cette contrainte s'applique :

- au niveau des interconnexions : les deux fils de la paire différentielle doivent avoir le même tracé pour éviter les différences de longueurs et d'environnements.
- au niveau des transistors de chaque porte logique : en cas de variations locales des caractéristiques pendant la fabrication de la puce, certains transistors risquent d'avoir des performances meilleures que les autres et pourraient "commuter" plus rapidement.

Ces contraintes rendent les réalisations différentielles peu compatibles avec les circuits asynchrones SI, QDI et DI. En effet, ces structures introduisent des contraintes de type *temps borné* qui sont contradictoires avec les efforts faits pour s'affranchir des hypothèses temporelles.

3.4.2 Fonction C-element généralisée

La fonction C-element généralisée (notée gC) étend les règles de transitions des C-elements ([figure 2.9\(a\)](#) - [page 16](#)) en autorisant plusieurs conditions distinctes de basculement à 0 et à 1. Cela se traduit concrètement par plusieurs regroupements distincts dans la table de Karnaugh régissant le fonctionnement de la cellule. Les gC sont réalisables si et seulement si il n'existe aucune combinaison d'entrées produisant simultanément un basculement à 0 et un basculement à 1.

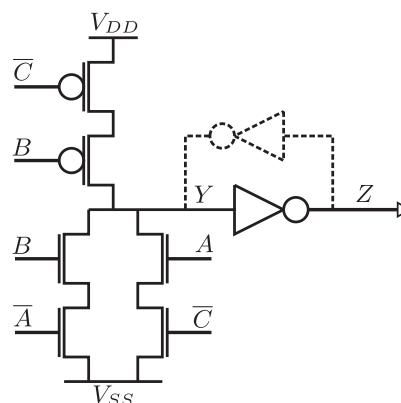
La [figure 3.9\(a\)](#) est un exemple de table de Karnaugh à plusieurs regroupements, la structure de la gC correspondante est ensuite automatiquement synthétisée [[JMG00](#)] ([figure 3.9\(b\)](#)).

| | $\overline{A}\overline{B}$ | $A\overline{B}$ | AB | $\overline{A}B$ |
|----------------|----------------------------|-----------------|---------|-----------------|
| \overline{C} | mémoire | 1 | 1 | 1 |
| C | 0 | 0 | mémoire | 1 |

$$\overline{B} \wedge C \Rightarrow Z \downarrow$$

$$(\overline{A} \wedge B) \vee (A \wedge \overline{C}) \Rightarrow Z \uparrow$$

(a) Table de Karnaugh



(b) Schematic correspondant

FIGURE 3.9 – Exemple de gC

Conclusion

De nombreuses structures ont été créées pour réaliser des points de synchronisation dans les circuits asynchrones. Elles ont été déclinées en de nombreuses variantes pour optimiser l'implémentation matérielle de certains circuits, par exemple des entrées de mise à 1, de remise à 0 ou des dissymétries dans les entrées peuvent être ajoutées ou leur sortance peut être ajustée.

Certaines structures ont été implémentée matériellement sous la forme de *standard cells* incluses dans la *TIMA Asynchronous Library* (TAL) et ont déjà permis la réalisation de circuits intégrés au sein de l'équipe CIS du laboratoire TIMA.

Bien que le comportement de ces cellules soit identique aux niveaux de modélisations numériques, les différentes structures fonctionnent intrinsèquement de manières différentes. Ces différences sont d'ailleurs visibles lors des simulations analogiques. Un circuit asynchrone réalisé à base de portes C-element à structure *Weak-Feedback* aura un comportement légèrement différent du même circuit dont l'implémentation des C-elements est réalisé avec des structures conventionnelles.

Chapitre 4

Phénomène de création de jetons

| | |
|----------------------------------------------|-----------|
| Introduction | 48 |
| 4.1 Niveaux d'abstractions | 48 |
| 4.1.1 Modélisations numériques | 48 |
| 4.1.2 Modélisations analogiques | 49 |
| 4.1.3 Modélisations des circuits asynchrones | 50 |
| 4.2 Simulations sur fronts lents | 50 |
| 4.2.1 Cause des pentes lentes | 50 |
| 4.2.1.1 Faible tension d'alimentation | 51 |
| 4.2.1.2 Influence de l'environnement | 51 |
| 4.2.1.3 Causes liées à la conception | 52 |
| 4.2.1.4 Causes liées à la fabrication | 54 |
| 4.2.1.5 Causes liées au vieillissement | 54 |
| 4.2.2 Dispositif expérimental complet | 54 |
| 4.2.3 Dispositif expérimental simplifié | 55 |
| 4.2.4 Résultats des simulations | 56 |
| 4.3 Phénomène inverse | 59 |
| 4.4 Conséquences sur le circuit | 59 |
| 4.4.1 Erreurs de phases | 60 |
| 4.4.2 Absorption de jetons | 62 |
| 4.4.3 Création de jetons | 62 |
| Conclusion | 62 |

Introduction

Les niveaux d'abstraction d'un circuit numérique ne prennent pas en compte tous les phénomènes présents dans le circuit réel. Puisqu'ils fonctionnent à temps continu et en prenant en compte les niveaux intermédiaires des signaux (section 2.9), les circuits asynchrones sont assez proches des circuits analogiques. Or la plupart du temps, le concepteur se limite à des niveaux d'abstractions ne modélisant pas fidèlement les phénomènes analogiques.

Dans ce chapitre, un phénomène non modélisable à l'aide des modèles numériques usuels est mis en évidence grâce à des simulations analogiques sur un petit circuit numérique asynchrone.

4.1 Niveaux d'abstractions

Un circuit logique peut être décrit à différents niveaux d'abstractions, du plus comportemental au plus physique. La figure 4.1 représente les niveaux couramment utilisés lors de la conception de circuits intégrés.

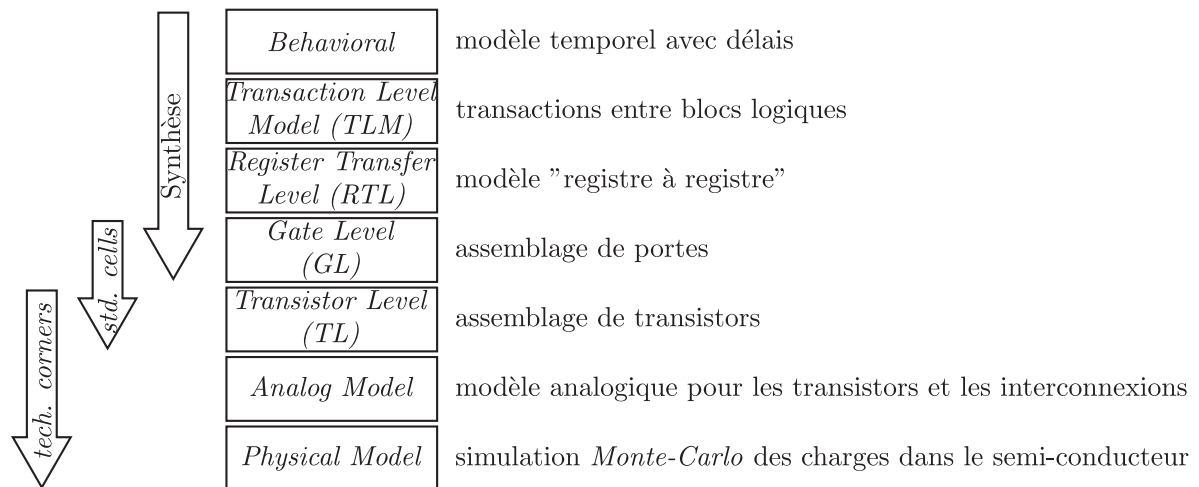


FIGURE 4.1 – Niveaux d'abstraction utilisés dans la conception de circuits intégrés

4.1.1 Modélisations numériques

Dans les niveaux d'abstraction les plus hauts, le comportement du circuit est entièrement numérique et peut être facilement simulé :

- Modèle comportemental : les différentes actions du circuit sont séparées par des délais explicites.
- TLM (*Transaction Level Modeling*) : le circuit est modélisé comme un ensemble de blocs logiques inter-connectés par des canaux de communications. Le modèle décrit

les sensibilités des blocs logiques aux activités des canaux et les actions à effectuer lorsqu'un canal sensible est actif. Ce niveau d'abstraction est très proche des langages de description des circuits asynchrones de type CSP [Hoa85] ou CHP [Mar91]. Les langages de description comme SystemC-TLM peuvent également être appliqués à la synthèse de circuits asynchrones [KH09]. Ce niveau d'abstraction permet de modéliser fidèlement le comportement numérique des circuits asynchrones micro-pipeline, SI, QDI ou DI mais aussi des circuits synchrones.

- RTL (*Register Transfer Level*) : c'est une modélisation décrivant la logique combinatoire placée entre la sortie d'un registre et l'entrée du registre suivant. La description peut encore faire référence à des types de haut niveau, tels que les entiers ou les *records**. Ce niveau d'abstraction est généralement utilisé pour décrire les structures séquentielles du type machine à état de Moore et de Mealy utilisées dans les circuits synchrones.
- GL (*Gate Level*) : le circuit est représenté par une structure de portes interconnectées. Cette description est généralement un raffinement d'un modèle de plus haut niveau (TLM ou RTL) réalisé avec un outil de synthèse dans le but d'éliminer tous les types de haut niveau et ne conserver que des *bits*. Cette liste de portes peut faire référence aux bibliothèques standards spécifiques à une technologie cible. Ces portes peuvent également faire partie d'une bibliothèque générique, sans cible technologique particulière [Syn10]. Le comportement numérique des portes est généralement décrit en langage VHDL [IEEb] ou Verilog [IEEa] afin de permettre la simulation numérique du circuit entier à l'aide de ModelSim par exemple.
- TL (*Transistor Level*) : toutes les portes logiques du niveau *Gate Level* sont décrites sous la forme d'un réseau de transistors.

4.1.2 Modélisations analogiques

Il existe principalement deux niveaux d'abstractions analogiques. Ceux-ci sont issus du niveau *Gate Level* dans lequel les portes sont décomposées en transistors et ont un modèle analogique : le simulateur doit alors résoudre le modèle approché du transistor défini pour la technologie souhaitée. La complexité du modèle et des calculs sous-jacents rend la simulation analogique de circuits très longue au delà de 10 000 transistors : la simulation de quelques nanosecondes à l'échelle du circuit nécessite alors plusieurs heures de calculs. Cette explosion des temps de simulation interdit l'utilisation de ce niveau d'abstraction à l'échelle d'un circuit intégré entier. Il est néanmoins possible de valider analogiquement le comportement d'une petite partie critique d'un circuit.

Ce modèle technologique du transistor n'étant qu'une approximation basée sur un modèle et des caractéristiques issues de la technologie, il est remplacé au niveau physique par une simulation des porteurs de charges afin d'avoir le comportement le plus fidèle possible au matériel. Ce niveau d'abstraction n'est pas utilisé par les concepteurs de circuits intégrés numériques.

4.1.3 Modélisations des circuits asynchrones

Dans toutes les familles de circuits synchrones et asynchrones, les hypothèses temporelles de fonctionnement sont exprimées au niveau des délais de propagation des valeurs numériques dans les portes et dans les interconnexions (cette particularité est également vraie pour les circuits synchrones). Dans les familles de circuits asynchrones dont le fonctionnement est à temps borné (circuits de Huffman et micro-pipeline), la stabilisation des valeurs numériques dans les parties combinatoires doit se faire dans un temps limite pour que le circuit fonctionne correctement. A l’opposé, les familles speed-insensitive et (quasi-)delay-insensitive ont un fonctionnement correct quelque soit le temps de propagation des valeurs numériques dans les portes et dans les interconnexions (modulo l’hypothèse de la *fourche isochrone** dans le cas des circuits QDI).

Les flots de conception transforment la description initiale du concepteur dans le but d’obtenir le circuit correspondant. Ces transformations doivent être effectuées sans changer le comportement du circuit. Elles doivent donc respecter la sémantique du langage dans lequel la description est écrite. Afin de garantir cette propriété, les flots de synthèse sont adaptés afin de garantir l’exactitude des transformations effectuées [BBD⁺03]. Il est également possible de garantir mathématiquement la correction de toutes transformations. On parle alors de transformations prouvées-correctes. Des techniques de preuves formelles ont été appliquées aux outils de synthèse pour les circuits asynchrones CAST [Mar89], TAST [Bre07] et Petrifly [CKK⁺97].

Le raffinement du niveau *Gate Level* au niveau *Transistor* a également été abordé formellement pour les *standard-cells* synchrones [Fig91] et asynchrones [Fig98].

Les vérifications n’interviennent qu’aux niveaux numériques. Des différences de comportement peuvent apparaître avec les niveaux d’abstraction inférieurs. La [section 2.9 \(page 29\)](#) introduit la notion de seuils de basculement et de leurs disparités à différents points du circuit. Le circuit réel se comporte d’une façon proche à la simulation analogique qui prend en compte des phénomènes n’apparaissant pas pendant les simulations aux niveaux d’abstractions logiques. Nous exposons dans la suite de ce chapitre un problème mis-en évidence lors de simulations analogiques d’un circuit asynchrone.

4.2 Simulations sur fronts lents

4.2.1 Cause des pentes lentes

Dans un circuit électronique, qu’il soit synchrone ou asynchrone, les temps de transition sont généralement considérés comme rapides (instantanés par rapport aux temps de propagation dans les portes) ou ne sont simplement pas considérés. Cette assertion n’est plus vraie lors d’une utilisation sous faible tension ou en environnement agressif.

4.2.1.1 Faible tension d'alimentation

En effet, la diminution de la tension d'alimentation réduit considérablement les performances du circuit. Cette caractéristique étant utile pour réduire la puissance consommée par le circuit, des approches comme le *Dynamic Voltage Scaling* (DVS) ont été appliquées à des circuits synchrones [FFRP04, SRH05] mais aussi à des circuits asynchrones [NNSvB94, ESFR02, RSMF05]. Les hypothèses de fonctionnement des circuits asynchrones à temps non-borné (famille SI, QDI et DI) rend leur vitesse de fonctionnement intrinsèquement adaptable à la tension d'alimentation. Cette réduction de vitesse se traduit par une augmentation des délais de propagation dans les portes, c'est le phénomène le plus souvent mis en avant. Mais aussi par une augmentation des temps de transitions.

4.2.1.2 Influence de l'environnement

L'environnement dans lequel fonctionne le circuit peut également perturber son fonctionnement : des couplages électromagnétiques (EM) peuvent se créer entre une partie du circuit et son environnement, voire entre deux parties du même circuit. Ces couplages se traduisent par une modification de la distribution des charges existantes dans les semi-conducteurs et dans les conducteurs. Les déséquilibres dans la distribution de ces charges impliquent des différences de potentiels et le déplacement de ces charges introduisent des courants parasites. L'environnement peut également être radiatif et soumettre le circuit à un bombardement de particules de hautes énergies. Au contact des éléments du circuit, ces particules peuvent modifier son état.

Ces phénomènes sont naturellement plus présents en haute altitude [Sho94] : les applications pour l'aéronautique et le spatial doivent donc être durcies [GGD00]. Il est également possible de rendre un environnement volontairement agressif dans le but de compromettre l'intégrité du circuit et accéder à des données cachées [Mon07].

Ces perturbations électromagnétiques se traduisent par des modifications de comportement dans les parties combinatoires (*Single or Multiple Event Transient* - SET ou MET) des circuits numériques. Ces valeurs altérées peuvent éventuellement ensuite être mémorisées. Lorsque la perturbation attaque directement les points de mémoire, ceux-ci peuvent alors basculer (*Single or Multiple Event Upset* - SEU ou MEU). Ces modifications transitoires et permanentes peuvent compromettre le fonctionnement des circuits synchrones [LPMC09] et asynchrones [Mon07].

Les circuits asynchrones sont néanmoins réputés plus robustes aux fautes que leurs équivalents synchrones [PN95] et les transmissions *point-à-point* dans les canaux de communication peuvent encore être rendues plus sûres grâce à l'adjonction de codes correcteurs d'erreurs [AN10]. L'environnement peut provoquer des SET dans un circuit asynchrone, mais la structure des circuits asynchrones fait que ces changements d'états sont détectés dans une grande majorité de cas : un changement de la valeur d'un fil introduit des erreurs de phases dans les protocoles de communications et/ou des erreurs d'encodage dans les données transmises. Ces ruptures de synchronisation aboutissent

généralement à l'arrivée prématurée d'une requête ou d'un acquittement : un jeton de données est alors inséré ou supprimé par rapport au fonctionnement normal du circuit asynchrone. Celui-ci va généralement se bloquer complètement (les causes de ce blocage sont détaillées dans la [section 4.4](#)).

Si un signal est perturbé pendant une transition, les effets de la perturbation s'ajoutent à sa variation normale. Celle-ci peut potentiellement avoir une durée augmentée et donc présenter un front plus lent que prévu.

4.2.1.3 Causes liées à la conception

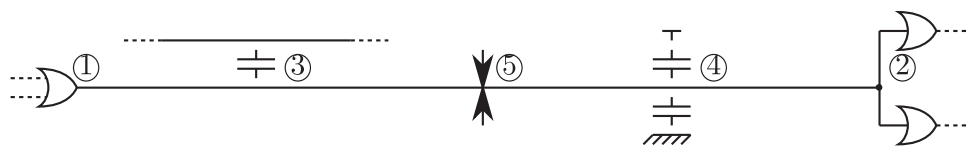


FIGURE 4.2 – Caractéristiques intervenant dans la vitesse des transitions

Des fronts lents peuvent également être la conséquence d'une mauvaise conception du circuit. Au niveau électronique, une transition sur une interconnexion correspond à l'apport ou à l'extraction de charges électroniques pour faire basculer le potentiel du conducteur. La vitesse des transitions dans une interconnexion dépend de plusieurs caractéristiques ([figure 4.2](#)) :

- de la sortance de la porte provoquant la transition ① : les portes à sortance élevée ont un courant en sortie plus élevé et peuvent donc apporter ou extraire plus de charges par unité de temps ;
- de la charge de l'interconnexion ② : en logique CMOS, ces charges sont des entrées de portes logiques, donc des capacités de grille de transistors MOS. Si plusieurs entrées de portes sont reliées à une même sortie, la capacité de l'interconnexion est la somme de toutes leurs capacités d'entrée ;
- de l'environnement de l'interconnexion : des capacités parasites existent avec les autres conducteurs ③ et les plans d'alimentation ④ proches. Ces capacités s'ajoutent aux capacités de charge du point précédent ;
- aux caractéristiques propres de l'interconnexion ⑤ : une interconnexion de forte longueur et de faible section aura une résistance linéaire importante et éventuellement une inductance propre importante elle-aussi.

Généralement, les capacités de charges ont une influence plus importante que les inductances et résistances linéaires de l'interconnexion. Une transition peut alors se simplifier à la charge-décharge d'une capacité par les transistor MOS à la sortie de la porte logique.

La [figure 4.3](#) représente les détails d'une transition obtenue pour un inverseur de sortance minimale avec la technologie *STMicroelectronics* 130nm HS. La courbe v correspond au potentiel de l'interconnexion et le courant i le courant dirigé de la sortie de la porte vers l'entrée des portes connectées. La transition fait apparaître quatre périodes distinctes :

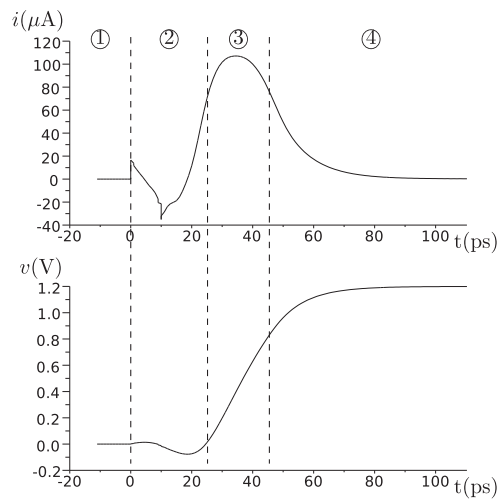


FIGURE 4.3 – Détails d'une transition

- Dans la période ①, le potentiel v est stable (aucune transition récente). Puisque $v = 0$, le transistor NMOS en sortie de la porte est en régime ohmique, le transistor PMOS est bloqué.
- La transition commence au temps $t = 0\text{ps}$. A cet instant, les transistors de sortie doivent changer d'état : le NMOS se bloque et le PMOS devient passant pour commencer une transition montante. Ces commutations nécessitent la charge/décharge de leurs capacités de grille respectives, étape qui s'accompagne d'un déplacement des charges à l'intérieur des canaux des transistors. Ces phénomènes complexes interviennent pendant la période ②.
- Dans la période ③, le transistor NMOS est bloqué et le PMOS est passant. Le faible potentiel v implique que le PMOS est en régime *source de courant* : le courant i est élevé. Ce courant peut *abusivement* être considéré constant et fixé par les dimensions du transistor PMOS et par les caractéristiques de la technologie. Dans cette période, le potentiel v augmente linéairement. Cette particularité confirme l'hypothèse de l'influence des capacités pendant les transitions : la capacité équivalente des entrées des portes connectées se chargent à courant constant.
- Finalement, pendant la période ④, le potentiel v est suffisamment élevé pour basculer le transistor PMOS en régime *ohmique* : le courant i décroît et la transition s'effectue plus lentement. La limite entre le régime *ohmique* et le régime *source de courant* est lié au V_{TP} du transistor PMOS, dépendant de ces dimensions et des caractéristiques technologiques.

Les outils de placement-routage sont normalement capables d'évaluer les temps de transition à partir d'une estimation des différentes caractéristiques des interconnexions. Si un temps de transition est estimé trop long, l'outil de placement routage peut le réduire à l'aide de plusieurs techniques classiques :

- réduire la longueur de l'interconnexion,
- augmenter la sortance si plusieurs variantes de la porte sont à disposition,
- ajouter des amplificateurs à sortance élevée si aucune variante de la porte ne permet

de réduire suffisamment le temps de transition.

Il est néanmoins nécessaire de vérifier les rapports de l'outil de placement-routage pour s'assurer que tous les nœuds ont été correctement étudiés (et corrigés si nécessaire).

4.2.1.4 Causes liées à la fabrication

La finesse des motifs des technologies de semi-conducteurs actuelles est de l'ordre de la dizaine de nanomètres. Les oxydes de grilles des transistors sont généralement plus fins d'au moins un ordre de grandeur. La fabrication de matériaux ayant des dimensions si réduites s'accompagne de grandes disparités et de grandes variations de caractéristiques. Le circuit fabriqué peut potentiellement avoir un comportement différent de celui espéré lors de la phase de conception.

4.2.1.5 Causes liées au vieillissement

Les technologies submicroniques actuelles ont des caractéristiques physiques évoluant fortement dans le temps. Ces modifications impactent les caractéristiques électriques et les temps de transition estimés lors de la conception peuvent devenir éloignés de la réalité.

Le vieillissement des conducteurs peut conduire à des phénomènes d'électromigrations : une partie du conducteur est arraché par le passage d'électrons à répétition. La section du conducteur est alors plus faible que prévue. Cela se traduit par une résistivité accrue qui peut augmenter le temps de charge des capacités.

Les tensions de basculement des transistors NMOS (V_{TN} - valeur positive) et PMOS (V_{TP} - valeur négative) ont tendance à augmenter en valeur absolue au fur et à mesure du vieillissement du circuit. Les transistors basculent alors plus précocement en zone ohmique lors des transitions. Or, le courant de charge est moins élevé dans cette zone. Donc la charge nécessitera plus de temps et la transition sera d'autant plus longue.

4.2.2 Dispositif expérimental complet

Le phénomène de création de jeton a d'abord été mis en évidence dans un circuit réalisant la fonction CRC (Contrôle de Redondance Cyclique). Le code CRC est une somme de contrôle permettant de détecter la présence d'erreurs dans des données transmises. Son calcul est basé sur un polynôme générateur : l'ordre et les termes de celui-ci caractérisent la capacité à détecter plus ou moins de bits d'erreurs, consécutifs ou non.

Le CRC étudié est basé sur le polynôme $X^{16} + X^{12} + X^5 + 1$ (également appelé CCITT), utilisé notamment pour le *bluetooth*, pour les *modems** en téléphonie analogique, pour les cartes mémoires au format *SD*... Dans le polynôme, l'opérateur + correspond à une opération *XOR* et les puissances aux retards successifs dans les bits de données. Ces retards sont implémentés matériellement avec des registres en série. La [figure 4.4](#) représente une implémentation de ce polynôme en logique synchrone [PB61].

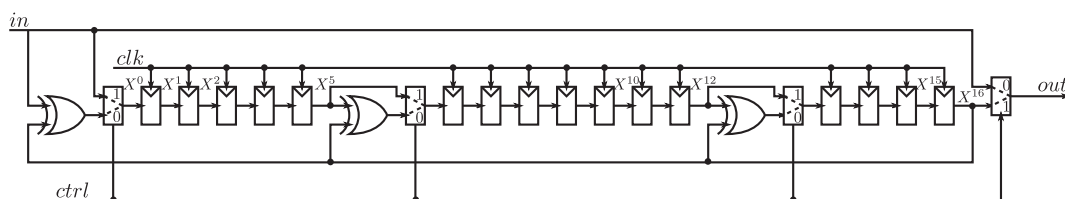


FIGURE 4.4 – Implémentation d'un CRC16 CCITT en logique synchrone

Ce dispositif calcule le CRC sur une liaison série de données et celui-ci est émis à la suite des données :

- lorsque *ctrl* vaut 0, chaque bit entrée sur le fil *in*, cadencé par l'horloge *clk*, est transmis au fil *out* et la somme de contrôle est mise à jour en interne ;
- dès que *ctrl* passe un 1, le CRC est émis sur *out* à chaque front d'horloge.

Cette description est traduite en logique asynchrone ([annexe A](#)) d'après les transformations suivantes :

- la notion de bit de donnée cadencé par l'horloge est remplacée par la notion de jeton de donnée dans un canal dual-rail (codage 1 parmi 2) ;
- chaque bascule est remplacée par un *full-buffer* asynchrone (mise en série de deux *half-buffers* WCHB) ;
- les fonctions XOR et multiplexeurs sont remplacées par leurs équivalents asynchrones ;
- les fourches sont conservées si les données sont toujours lues en même temps sur les deux branches ;
- les fourches où les données ne sont utilisées que par une des branches sont remplacées par des démultiplexeurs ;
- les données initiales dans les bascules sont remplacées par des jetons initialement présents dans les *half-buffers* (l'utilisation des entrées set/reset permet de forcer certaines sorties de C-elements à 0 et d'autres à 1).

La structure du circuit fait apparaître un grand nombre de *half-buffers* WCHB. Dans le but de raccourcir le temps des simulations analogiques et d'étudier les causes et effets des créations de jetons, le circuit a été simplifié.

4.2.3 Dispositif expérimental simplifié

La [figure 4.5](#) représente le dispositif simplifié : deux *Half-Buffers* WCHB double-rails sont mis en série afin de simuler le comportement d'un *Full-Buffer*. Seulement trois canaux de communication composent ce dispositif.

L'environnement d'entrée est supposé lent et émet une seule requête : le modèle de simulation est généré à l'aide de sources de tensions.

L'environnement de sortie est rapide : il accepte automatiquement toutes les requêtes transmises par le buffer. Il est matériellement implémenté avec une porte *NOR*.

Les données présentes sur le canal d'entrée du *Full-Buffer* doivent être réémises dans l'ordre d'arrivée et sans altération de leurs valeurs sur son canal de sortie. En fonction-

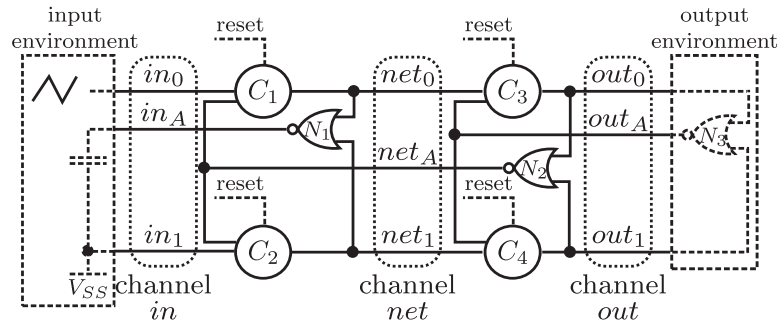
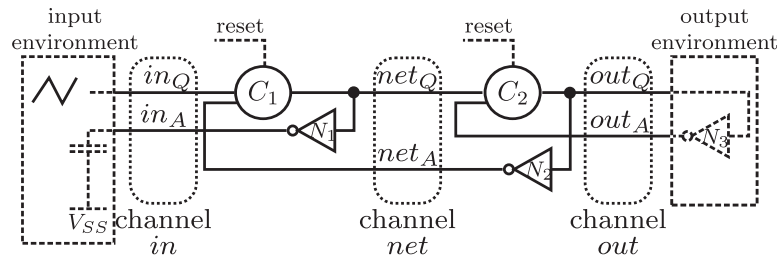


FIGURE 4.5 – Mise en série de deux half-buffers dual-rail WCHB

nement normal, c'est-à-dire lorsque les hypothèses de fonctionnement sont respectées, un buffer asynchrone doit émettre autant de jetons de données qu'il en a reçu. Dans le cas contraire, le circuit dysfonctionne.

Dans ce dispositif, seul un des rails des half-buffers est effectivement utilisé. Il est donc possible de simplifier encore le dispositif en supprimant complètement le rail inutilisé (correspondant à la transmission des données 1) comme représenté sur la figure 4.6.


 FIGURE 4.6 – Circuit équivalent *single-rail*

4.2.4 Résultats des simulations

Le fonctionnement escompté est une transmission normale des requêtes depuis le canal d'entrée jusqu'au canal de sortie au fur et à mesure de leur émission par l'environnement d'entrée et de leur réception par l'environnement de sortie. Les transitions sur le fil in_Q doivent donc être transmises sur le fil net_Q puis sur out_Q . Lors de ces transmissions, les portes logiques devraient atténuer l'effet des pentes lentes. En effet, pour chaque étage logique traversé non affecté par les pentes lentes, la transition est de plus en plus rapide.

La figure 4.7 trace les résultats souhaités pour les signaux de requête et d'acquittement des trois canaux de communication du dispositif simplifié : les résultats de la zone ① sont représentés plus en détails dans la zone ②. La pente lente de la requête issue de l'environnement (in_Q) est *redressée* après son passage dans le premier *half-buffer* (net_Q). A l'issue du deuxième *half-buffer*, la pente de la requête est à nouveau du même ordre de grandeur que les recommandations technologiques du fondeur.

La figure 4.8 représente les requêtes et acquittements effectivement mesurés lors des

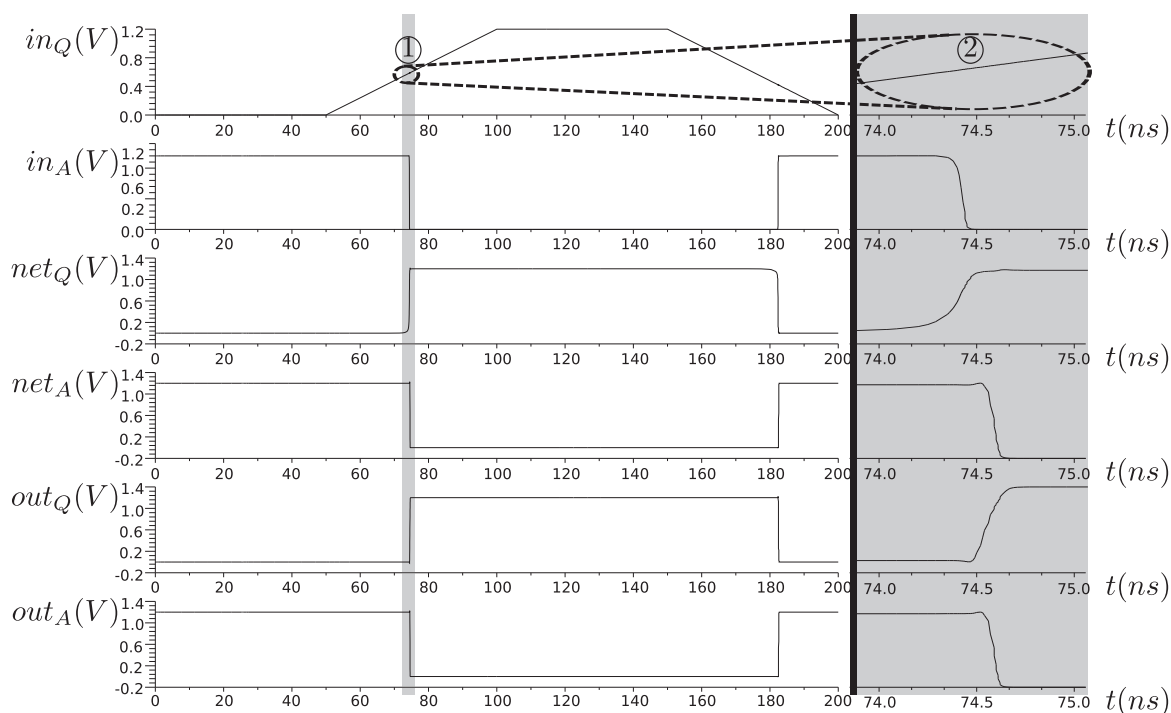


FIGURE 4.7 – Résultats escomptés

premières mesures. Le circuit est très actif lors des transitions lentes. L'agrandissement de la zone d'activité ① est visible dans la zone ②. Dans cette période de forte activité, présente aussi bien pendant les fronts montants que descendants, de nombreuses requêtes sont émises sur le canal intermédiaire net et transmises au canal de sortie out . Toutes celles-ci sont automatiquement acquittées par l'environnement de sortie.

Ce comportement peut s'apparenter à une création d'aléas dans le C-element C_1 mais la cause est ici différente. Un aléa en sortie d'une porte correspond à la création de transitions intempestives liées à des hypothèses temporelles ou à un effet de l'environnement non souhaitée sur la sortie de la porte après qu'une de ses entrées ait basculé. Un aléa ne correspond pas au fonctionnement normal d'une porte (tel que représenté dans sa table de vérité par exemple).

Or, dans le cas présent, ce sont les transitions du signal net_A qui provoquent les basculements de la sortie de C_1 , conformément à la table de vérité de la fonction C-element. Ces transitions sur l'entrée de C_1 sont elles-même causées par les transitions de la sortie de C_1 par l'intermédiaire des portes C_2 et N_2 (figure 4.9). Cet anneau oscillant comporte deux portes non-inverseuses (C_1 et C_2) et une porte inverseuse (N_2).

On est en présence d'un circuit oscillant lorsque la tension de in_Q est à un niveau analogique intermédiaire, ne représentant clairement ni un niveau logique 0, ni un niveau logique 1. La boucle formée par les composants C_1 , C_2 et N_2 est une structure classique rencontrée dans les circuits asynchrones : l'acquiescement d'un canal de communication reflète toujours l'activité du signal de requête du même canal. Le problème est donc à chercher dans le composant C_1 : celui-ci ne prend pas correctement en compte les tensions

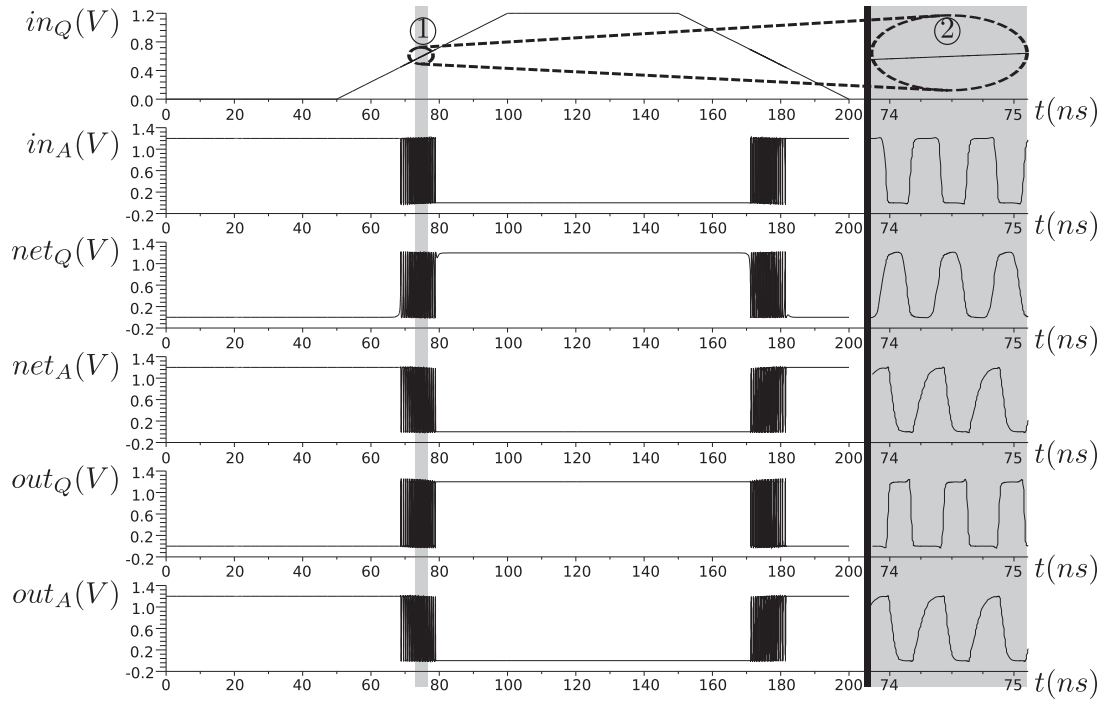


FIGURE 4.8 – Résultats obtenus

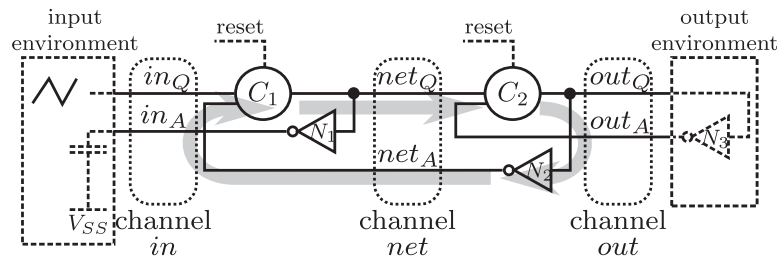


FIGURE 4.9 – Boucle des oscillations dans le circuit équivalent single-rail

analogiques intermédiaires sur ses entrées.

4.3 Phénomène inverse

Les pentes lentes peuvent également perturber le fonctionnement du circuit asynchrone lorsque l'environnement de sortie est plus lent que l'environnement d'entrée : des jetons de donnée sont alors absorbés.

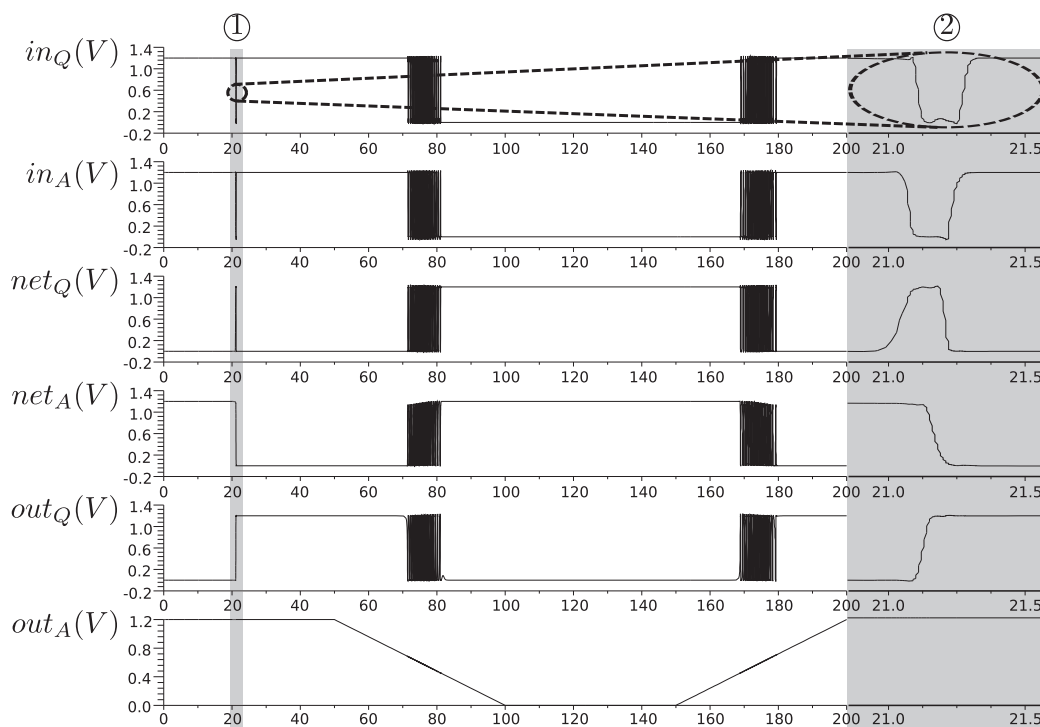


FIGURE 4.10 – Résultats obtenus avec un environnement de sortie lent

La figure 4.10 décrit le fonctionnement du circuit dans ces circonstances. A l'origine, le buffer est vide : ces deux étages ne mémorisent aucune requête. Dans la zone ①, agrandie dans la zone ②, un premier jeton de donnée est transféré jusqu'au canal de communication de sortie. L'environnement de sortie va alors placer un acquittement dont les fronts sont lents. Comme précédemment, les valeurs intermédiaires du signal d'acquittement sont mal interprétées et ce seul acquittement provoque la destruction de plusieurs requêtes émises successivement par l'environnement d'entrée.

4.4 Conséquences sur le circuit

Les jetons de données ainsi créés ou absorbés dans le circuit agissent comme des aléas, même si leur origine est ici différente. Dans cet exemple simple d'un circuit à structure linéaire, les jetons de données additionnellement créés sont automatiquement absorbés

par l'environnement de sortie. Une telle structure ne se bloquera pas en présence de ce phénomène de création de jetons. Par contre, de nombreux calculs erronés seront effectués par les étages suivants de traitement.

Dans un circuit réel, les étages suivants peuvent éventuellement former une boucle (c'est le cas par exemple dès qu'un traitement itératif est implémenté). Le nombre de requêtes présentes dans la boucle est une grandeur connue dès la conception, qui sert à dimensionner le nombre d'étages nécessaires dans cette boucle. Les apparitions de requêtes peuvent saturer la boucle et empêcher la progression des données. A l'inverse, leur disparition implique un blocage des blocs logiques nécessitant leur présence : ils seront indéfiniment en phase d'attente de ces données.

Ces erreurs, même localisées dans une partie du circuit, se traduiront par son dysfonctionnement généralisé. En effet, tous les blocs logiques étant fortement synchronisés avec leurs voisins (cette proximité est issue de la description algorithmique et non de la localisation géographique sur le circuit fabriqué), la modification du comportement d'un des blocs contaminera tout le circuit de proche en proche.

4.4.1 Erreurs de phases

Les fautes transitoires placées sur les signaux de requêtes ou d'acquittement aboutissent à des sauts de phase dans les protocoles de communication. La figure 4.11(a) représente les sauts de phases possibles pour un aléa initialement introduit sur le signal d'acquittement, la figure 4.11(b) les phases possibles à cause d'un aléa initialement sur la requête.

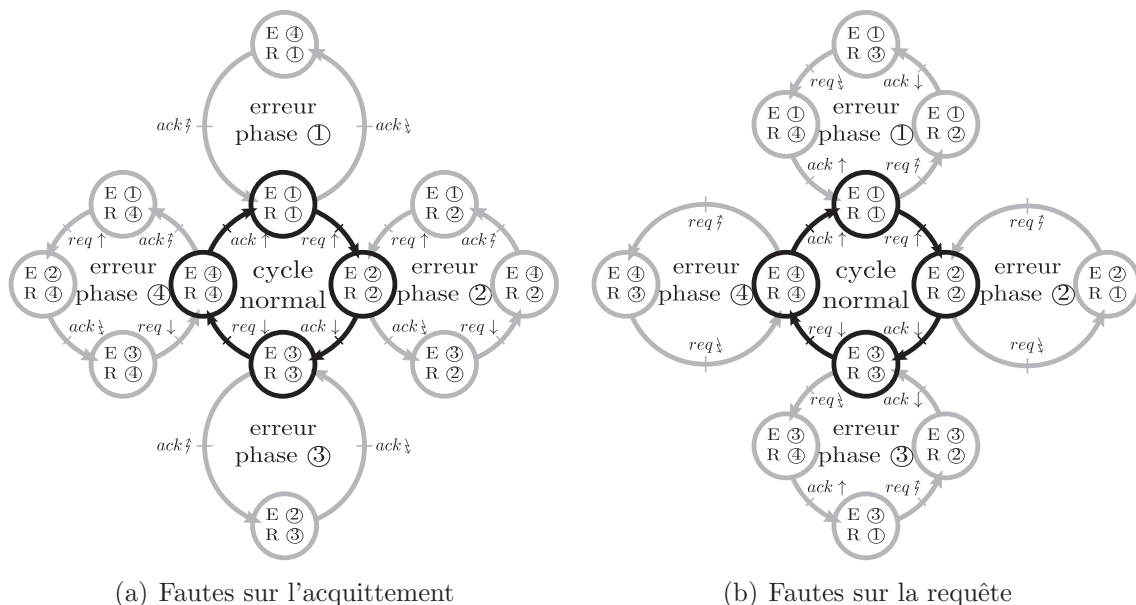


FIGURE 4.11 – Conséquences des fautes sur la synchronisation d'un canal de communication

Dans ces deux figures sous forme de machine à état, les états représentent les couples de phases dans un canal de communication à quatre phases. Chaque état est caractérisé par le deux phases vues par le bloc émetteur (le numéro entouré après "E") et le bloc récepteur (le numéro entouré après le "R") de ce canal de communication. Par exemple, une phase notée "E ① R ②" indique que le bloc émetteur est dans la phase ① du protocole quatre phases alors que le bloc récepteur est dans la phase ②.

Pendant un cycle normal (sans aléa), représenté en noir au centre de ces figures, les phases vues par les blocs émetteur et récepteur sont identiques : le canal de synchronisation est verrouillé et les données sont bien transférées. A l'inverse, une différence entre les deux phases dans un couple indique une désynchronisation du canal de communication : les blocs émetteur et récepteur peuvent évoluer indépendamment et potentiellement corrompre le transfert des données.

Les transitions entre les états indiquent les transitions de signaux nécessaires pour le passage à la phase suivante. Chaque transition indique le signal concerné : *ack* ou *req*. Les transitions montantes et descendantes volontaires (issues des blocs logiques) sont représentées par les symboles respectifs \uparrow et \downarrow . Ces transitions font évoluer les phases des blocs émetteur et récepteur simultanément. A l'opposé, les transitions subies (liées aux perturbations sur le conducteur) sont représentées par $\uparrow\downarrow$ et $\downarrow\uparrow$. Les perturbations changent la phase vue du coté haute-impédance mais n'ont pas d'effet sur le coté basse-impédance.

Les perturbations sont temporaires et une transition $\uparrow\downarrow$ est toujours suivie d'une transition $\downarrow\uparrow$ (et inversement). Les perturbations peuvent intervenir dans chacune des quatre phases du cycle normal et sont représentées en grisés sur les figures.

Par exemple, lorsqu'une faute intervient sur le signal d'acquittement pendant la phase ② (cycle grisé à partir de l'état noté "E ② R ②" du cycle principal), le canal de communication est désynchronisé et son fonctionnement est régi par ce cycle grisé : le bloc d'entrée peut enlever la requête et en placer une autre dès que la perturbation est finie. On a dans ce cas l'absorption d'un jeton de données. Il est également probable que la donnée lue par le bloc récepteur ne soit pas valide car issue d'un mélange entre les données de l'ancien jeton (absorbé) et du nouveau jeton. Le même comportement est observable pendant la phase ④. Une autre faute sur le même signal d'acquittement pendant la phase ① (ou ③) a des conséquences plus limitées : le bloc émetteur n'est pas sensible au signal d'acquittement à ce moment et la perturbation sera ignorée.

De même, les fautes sur le signal de requête sont masquées pendant les phases ② et ④ mais produisent des jetons supplémentaires pendant les phases ① et ③.

Les effets des erreurs de phases liées aux fautes transitoires dans un circuit asynchrone peuvent donc produire les mêmes effets de création ou absorption de jetons liées aux pentes lentes.

4.4.2 Absorption de jetons

Dans un circuit asynchrone, un déséquilibre entre le nombre de jetons effectivement présents dans le circuit et le nombre de jetons prévus dans sa description se traduit par un blocage. Pour un bloc fonctionnel consommant deux données issues de deux canaux de communications (un additionneur à deux entrées par exemple, [figure 4.12\(a\)](#)), il est généralement nécessaire d'attendre la présence de ces deux données pour calculer le résultat. L'absence d'une des deux données peut provoquer l'absence de résultat.

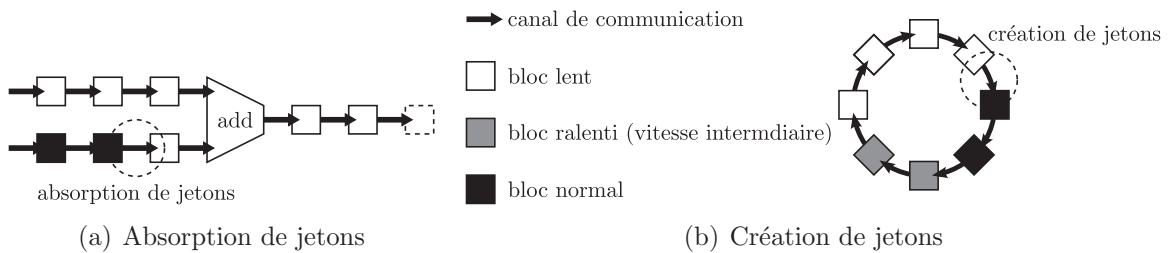


FIGURE 4.12 – Conséquences des blocs lents dans un circuit asynchrone

Les hypothèses de fonctionnement des circuits asynchrones étant généralement à temps non-borné, les blocs suivants seront en attente, de durée potentiellement infinie, de jetons absents (puisqu'absorbés). Pour permettre au circuit de récupérer d'un tel blocage, le concepteur doit par exemple ajouter des blocs détectant les phases trop longues d'inactivité (de type *watchdog**).

4.4.3 Création de jetons

Dans un circuit bouclé, les jetons peuvent progresser dans les étages successifs tant qu'un étage au moins de la boucle est vide (sans jeton, on parle alors de bulle). Les jetons additionnels insérés dans un circuit bouclé peuvent le saturer ([figure 4.12\(b\)](#)) : les jetons ne peuvent plus évoluer dans la boucle. Ce blocage peut alors contaminer les autres blocs nécessitant des jetons progressant dans cette boucle. Il est nécessaire de le remettre à zéro (*reset**) pour récupérer de cette situation.

Conclusion

Le comportement analogique des circuits asynchrones peut mettre à défaut leur fonctionnement tel que décrit, modélisé et simulé aux niveaux d'abstractions numériques courants. Les points mémoires "C-elements" couramment utilisés dans les circuits asynchrones peuvent ne pas prendre en compte correctement la tension d'une de leurs entrées si elle est à un niveau intermédiaire (ne représentant clairement ni un niveau logique 0, ni un niveau logique 1. Or, les pentes lentes peuvent imposer ces niveaux intermédiaires suffisamment longtemps pour que ce phénomène intervienne.

Dans cette situation, des anneaux oscillants se forment à cause des structures bouclées naturellement présentes dans les circuits asynchrones. Les oscillations, inhérentes à cette structure, aboutissent à la création d'un nombre plus élevé que conçu de requêtes ou d'acquittements bloquant complètement le circuit. Une remise à zéro de celui-ci peut rétablir temporairement son fonctionnement, mais les blocages reviendront dès que les conditions de pentes lentes seront à nouveau présentes.

Chapitre 5

Seuils et sensibilité à la pente

| | |
|--------------------------------------------------------------|-----------|
| Introduction | 66 |
| 5.1 Modèles numériques des seuils de commutations | 66 |
| 5.1.1 Modèle di-atomique | 66 |
| 5.1.2 Modèles à temps non-bornés | 68 |
| 5.2 Niveau analogique | 69 |
| 5.2.1 Logique combinatoire | 69 |
| 5.2.2 Logique séquentielle | 70 |
| 5.3 Vérification des bibliothèques | 70 |
| 5.3.1 Protocoles de mesure | 70 |
| 5.3.2 Précision de la mesure | 70 |
| 5.3.3 Portes des bibliothèques TIMA | 71 |
| 5.3.4 Autres structures de C-elements | 73 |
| 5.4 Correction des seuils des portes | 74 |
| 5.4.1 Redimensionnement des transistors | 74 |
| 5.4.1.1 Structure <i>Weak-Feedback</i> | 74 |
| 5.4.1.2 Structure conventionnelle | 76 |
| 5.4.1.3 Structure Symétrique | 76 |
| 5.4.1.4 Conclusions sur le redimensionnement des transistors | 77 |
| 5.4.2 Réduction de la tension d'alimentation | 79 |
| 5.4.2.1 Polarisation des transistors | 79 |
| 5.4.2.2 Structure <i>Weak-Feedback</i> | 80 |
| 5.4.2.3 Structures Conventionnelle et symétrique | 81 |
| 5.4.2.4 Structure dynamique et structures composites | 81 |
| 5.4.3 Variations simultanées | 82 |
| 5.4.3.1 Structure <i>Weak-Feedback</i> | 82 |
| 5.4.3.2 Structure Conventionnelle | 82 |
| 5.4.3.3 Structure Symétrique | 83 |
| Conclusion | 84 |

Introduction

Les synchronisations locales des circuits asynchrones reposent essentiellement sur les C-elements. Une rupture de synchronisation dans un canal de communication a d'abord des conséquences locales dans les deux blocs logiques connectés : le bloc récepteur peut interpréter des données différentes de celles émises ou alors le bloc émetteur peut interpréter à tort un acquittement.

Ces modifications locales de comportements seront ensuite propagées vers les blocs suivants du flot de donnée par le biais des signaux de *requête*^{*}, mais également vers les blocs précédents à cause des signaux d'*acquiescement*^{*}. Dans les circuits asynchrones, un dysfonctionnement local à tout endroit du circuit a donc des répercussions dans l'ensemble du circuit. Les points mémoires C-elements doivent donc faire l'objet de toutes les attentions afin de ne pas compromettre le comportement global du circuit.

Ces problèmes sont détectables une fois le circuit fabriqué. Mais dans le but de diminuer les rebuts, il est nécessaire de corriger ou de diminuer la cause de ces défaillances dès la phase de conception : il faut prendre en considération l'ordre dans les seuils de commutations V_{TL} et V_{TH} .

5.1 Modèles numériques des seuils de commutations

5.1.1 Modèle di-atomique

Les seuils de commutation des portes logiques ne sont généralement pas pris en compte dans les modèles numériques des circuits logiques, et plus spécifiquement asynchrones dans notre cas. Ces seuils ont cependant des conséquences sur le comportement du circuit réel que les abstractions numériques habituelles ne permettent pas de modéliser. Le modèle di-atomique [Pap05] consiste en la modification des modèles numériques dans le but de prendre en compte ces seuils de basculement.

Ce modèle permet de différencier le seuil de basculement haut (V_{TH}) du seuil bas (V_{TL}) dans les règles de transitions des portes logiques. Le modèle numérique du circuit est modifié : chaque signal x du circuit réel est dupliqué en deux signaux numériques virtuels couplés x_p et x_n . L'état logique bas du signal virtuel x_n modélise l'état logique bas du signal réel x et l'état logique haut du signal virtuel x_p l'état logique haut du signal réel x . La [table 5.1](#) résume la correspondance entre le signal réel x et les niveaux des signaux virtuels x_n et x_p : les états logiques 0 et 1 du signal x sont atteints lorsque les deux signaux virtuels x_n et x_p ont tous les deux cette même valeur. Lorsque $x_n = 1$ et $x_p = 0$, le signal x ne contient aucune information (ni 0, ni 1). Le dernier cas où $x_n = 0$ et $x_p = 1$ correspond à un signal x pouvant être interprété à la fois comme à 0 et à 1 par les portes logiques dont une entrée est connectée à ce fil x .

Les transitions sur le signal x sont modélisées comme la succession de deux transitions sur les signaux équivalents x_n et x_p . La [figure 5.1](#) représente les correspondances entre les seuils de basculements analogiques et les signaux équivalent x_n et x_p lorsque $V_{TL} < V_{TH}$.

| x_n | x_p | description pour x |
|-------|-------|-----------------------------------------|
| 0 | 0 | état logique 0 |
| 0 | 1 | conflit entre les états logiques 0 et 1 |
| 1 | 0 | ni 0, ni 1 |
| 1 | 1 | état logique 1 |

Table 5.1 – Correspondance des niveaux dans le modèle di-atomique

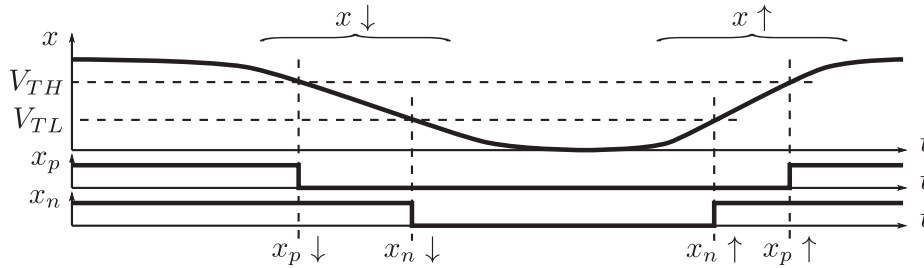


FIGURE 5.1 – Correspondances entre une transition analogique et le modèle di-atomique

Les équations 5.1 et 5.2 représentent l'ordonnancement des transitions de x_n et x_p lorsque $V_{TH} > V_{TL}$.

$$x \downarrow \Rightarrow x_p \downarrow; x_n \downarrow \quad (5.1)$$

$$x \uparrow \Rightarrow x_n \uparrow; x_p \uparrow \quad (5.2)$$

L'opérateur ";" indique la séquentialité des transitions de x_n et x_p : une transition descendante d'un signal x du circuit réel correspond d'abord à une transition descendante du signal équivalent positif : le signal x n'est alors plus modélisé au niveau logique haut ; transition suivie d'une transition descendante du signal équivalent négatif : x est alors modélisé au niveau logique bas.

Dans le cas où $V_{TH} < V_{TL}$, les transitions sont modélisées dans un ordre différent (équations 5.3 et 5.4).

$$x \downarrow \Rightarrow x_n \downarrow; x_p \downarrow \quad (5.3)$$

$$x \uparrow \Rightarrow x_p \uparrow; x_n \uparrow \quad (5.4)$$

Cette modélisation au niveau numérique des seuils de commutation permet de modéliser le comportement du circuit pendant des transitions lentes après les modifications préalables :

- de la description du circuit pour dédoubler chaque signal logique ;
- de la description comportementale des portes logiques afin de supporter les entrées dédoublées.

La simulation de cette description modifiée permet alors de répéter le phénomène de création de jetons vu dans la figure 4.8 (page 58). Ces simulations permettent de déterminer à partir de quel temps de transition minimum le phénomène apparaît. Cette

mesure met en évidence une borne assurant le comportement correct du circuit ([hypothèse 5.1](#)).

Hypothèse 5.1. *La transition doit être plus rapide qu'une limite (modèle temps-borné), dont la valeur dépend de la structure du circuit et des tensions de seuils des portes logiques.*

Si cette hypothèse n'est pas satisfaite dans un circuit asynchrone, des désynchronisations peuvent intervenir dans les canaux de communications et perturber le transfert des jetons de données entre les blocs logiques.

Cette méthode à temps borné ne donne des résultats valables que pour un circuit donné et elle doit être répétée pour tous les cycles possibles d'un circuit asynchrone. Donc pour toute la logique intercalée entre les étages de mémorisation.

5.1.2 Modèles à temps non-bornés

Les circuits dont le fonctionnement est régi par des hypothèses de fonctionnement à temps non-bornés telles que les circuits QDI ([section 2.3.4 - page 10](#)) et DI ([section 2.3.5 - page 11](#)) tirent leur robustesse de l'indépendance de leur comportement vis-à-vis du temps. Or, leurs hypothèses de fonctionnement (décrites au niveau numérique) ignorent complètement les phénomènes analogiques tels que les pentes.

Dans ces modèles à temps non-borné, l'ajout d'hypothèses de type temps-borné, comme le modèle di-atomique vu précédemment, permet de modéliser plus fidèlement le comportement du circuit mais ne permet en aucun cas de garantir son fonctionnement dans un environnement dégradé. Il est nécessaire d'introduire une nouvelle hypothèse de fonctionnement concernant les pentes [hypothèse 5.2](#).

Hypothèse 5.2. *Le circuit fonctionne correctement quels que soient les temps de transition dans les interconnexions et sur les entrées du circuit.*

Les deux classes de circuits suivantes permettent de s'affranchir complètement des temps de transition [[OMAF10](#)] :

- la classe de circuit QDSI (*Quasi-Delay-and-Slope-Insensitive*) ajoute aux hypothèses de délais issues de la classe QDI la restriction liée au temps de transitions ([hypothèse 5.2](#)) ;
- la classe de circuit DSI (*Delay-and-Slope-Insensitive*) restreint la classe DI en ajoutant cette même [hypothèse 5.2](#). Ces circuits seraient alors complètement insensibles aux délais et aux pentes.

Pour réaliser de tels circuits, il est nécessaire de respecter toutes les hypothèses de fonctionnement de la classe initiale : ce prérequis est généralement assuré par l'outil ou la technique de synthèse utilisée pour obtenir la *netlist** de portes. L'hypothèse des fourches isochrones de la classe QDI requiert des contraintes additionnelles sur le placement-routage du circuit. Pour être insensible aux pentes ([hypothèse 5.2](#)), il est nécessaire de concevoir avec précaution les portes logiques des bibliothèques.

5.2 Niveau analogique

L'état de l'art de la logique asynchrone indique que les seuils de commutations doivent être sensiblement identiques pour toutes les portes logiques d'un circuit [vB92]. Cependant cette particularité n'a fait l'objet que de peu de publications par la suite et ces valeurs de seuils de commutations n'ont pas été prises en compte lors de la conception des circuits existants.

5.2.1 Logique combinatoire

Les circuits asynchrones utilisent des portes logiques combinatoires standards également utilisées pour les circuits synchrones. Ces bibliothèques étant initialement dédiées à la conception de circuits synchrones où l'influence des seuils de commutations est négligée, les cellules ont, de par leur conception, des disparités dans leurs tensions de basculement (table 5.2). Les tensions de basculement V_{TL} et V_{TH} sont égales pour une même entrée d'une porte logique combinatoire, cette valeur mesurée est rapportée pour chaque entrée de portes logiques NAND et NOR ayant de deux à quatre entrées.

| porte | entrée A | entrée B | entrée C | entrée D |
|-------|--------------|----------|----------|----------|
| NAND2 | 0.512 | 0.540 | N/A | N/A |
| NAND3 | 0.484 | 0.510 | 0.531 | N/A |
| NAND4 | 0.463 | 0.486 | 0.505 | 0.523 |
| NOR2 | 0.598 | 0.558 | N/A | N/A |
| NOR3 | 0.632 | 0.596 | 0.566 | N/A |
| NOR4 | 0.650 | 0.617 | 0.589 | 0.566 |

Table 5.2 – Tensions de basculement des portes logiques combinatoires (ST 130nm HS)

Par exemple, dans la technologie *STMicroelectronics* 130nm High-Speed ($V_{DD} = 1.2V$), le V_{TH} est $0.512V$ pour l'entrée A de la porte NAND à deux entrées et $0.598V$ pour cette même entrée de la porte NOR à deux entrées. De part leur structure interne où plusieurs transistors sont mis en série (plusieurs transistors NMOS pour la fonction NAND et plusieurs PMOS pour la NOR), les tensions de basculement pour une même porte ne sont pas les mêmes sur ses différentes entrées. Dans la fonction NAND, plus une entrée est connectée à un transistor NMOS éloigné du rail d'alimentation V_{SS} , plus sa tension de basculement est élevée. Dans la fonction NOR, l'évolution est inverse : la tension de basculement est plus faible pour les entrées connectées aux transistors éloignés du rail V_{DD} .

L'utilisation de ces portes logiques aux extrémités d'une fourche isochrone entraînera irrémédiablement des différences dans les instants de basculement de leur entrées. Les différences de potentiel entre les tensions de basculement peuvent être particulièrement importantes : si une fourche isochrone est connectée aux entrées A d'une NAND4 et d'une NOR4, la différence de basculement est de $166mV$, soit une erreur de 25.5%.

5.2.2 Logique séquentielle

A l'inverse des portes logiques combinatoires où les seuils de basculement V_{TL} et V_{TH} sont égaux, la structure des portes utilisées pour l'asynchrone (C-element ou fonction C-element généralisée) font apparaître des disparités dans les seuils de basculement. Il est nécessaire de les concevoir en prenant en compte les phénomènes de pente lente mais aussi de la différence entre ces seuils de basculement.

5.3 Vérification des bibliothèques

5.3.1 Protocoles de mesure

Les portes logiques séquentielles de la bibliothèque TAL ont été caractérisées pour mesurer leurs tensions de basculement. Cette mesure fait un lien entre les transitions de la sortie de la porte et les tensions à ses entrées. Elle doit par conséquent être effectuée en négligeant les délais de propagation de la porte sous test. Cette assertion peut être satisfaite en appliquant des signaux très lents sur les entrées afin de se mettre en situation quasi-statique, les temps de propagation dans la porte étant alors négligeables.

La porte à caractériser est placée dans un *benchmark** simulant des entrées avec des fronts lents. Sa sortie est chargée par un inverseur de taille minimale qui correspond à la charge réaliste la plus faible pouvant être présente à sa sortie. La sortie de la porte pourra alors commuter rapidement, ce qui réduit l'imprécision sur la mesure de l'instant de commutation et donc sur la valeur de l'entrée à cet instant.

Les entrées sont d'abord supposées commuter indépendamment les unes des autres et les seuils sont mesurés pour une entrée à la fois selon la séquence suivante :

- Toutes les entrées sont placées à un état logique bien établi (V_{SS} pour le niveau logique 0, V_{DD} pour 1) afin d'initialiser l'état logique mémorisé par la porte testée.
- Les entrées non-mesurées sont alors inversées afin de permettre à la porte de basculer dès que la dernière entrée aura atteint une tension suffisante.
- L'entrée mesurée est stimulée par une pente lente : la sortie de la porte bascule lorsque la tension de cette entrée aura atteint son seuil de basculement.

Ce procédé de mesure est répété afin de déterminer les tensions de seuils hautes et basses pour chaque entrée de chaque porte.

5.3.2 Précision de la mesure

La transition de l'entrée mesurée doit se dérouler suffisamment lentement pour que la mesure de l'instant de commutation ne soit pas affectée par les délais de transitions dans la porte. La [figure 5.2](#) est le relevé des valeurs des tensions de transitions mesurées, pour une structure *Weak-Feedback* à deux entrées (WM2) issue de la TAL en technologie ST 130nm, pour différentes durées (en échelle logarithmique) de transitions linéaires.

On remarque que plus le temps de la transition est court, plus les valeurs mesurées

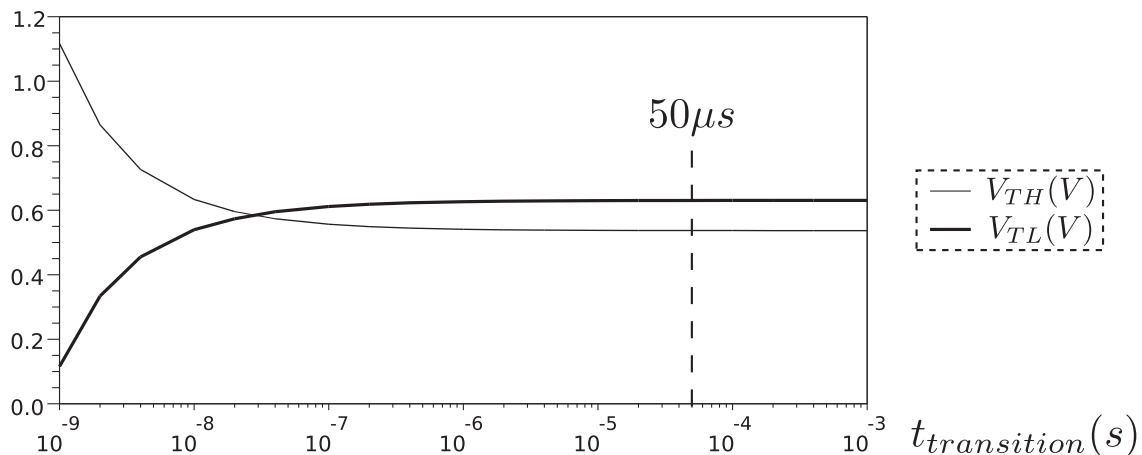


FIGURE 5.2 – Valeurs mesurées des seuils pour différents temps de transitions

divergent : lors des courtes transitions, le délai de la porte est suffisamment long pour que la tension d'entrée évolue pendant le basculement du point mémoire. Le délai de la porte perturbe la mesure : ce retard entre l'instant où les entrées permettent effectivement à la porte de basculer (qui correspond à la tension de seuil réelle) et l'observation de ce basculement sur la sortie (qui correspond à la valeur mesurée de ce seuil) permet à la tension d'entrée d'être plus importante pendant les fronts montants, décalant ainsi le V_{TH} mesuré vers V_{DD} . À l'inverse V_{TL} est décalée vers le bas car elle est mesurée sur un front descendant.

Les mesures effectuées pendant les transitions rapides ont pour conséquences une sur-estimation de V_{TH} et une sous-estimation de V_{TL} . Une mesure de l'écart $\Delta V_T = V_{TH} - V_{TL}$, d'après des mesures effectuées pendant une transition rapide, risque d'être positive alors que la différence des seuils est réellement négative. Le comportement de la porte serait mesuré comme correct alors qu'il ne l'est effectivement pas dans ce cas.

Afin d'éviter ces mauvaises caractérisations, il est nécessaire de dimensionner le temps de transition pour se situer dans les plateaux de la [figure 5.2](#) : la précision de la mesure reste quasi-inchangée lorsque le temps de mesure augmente. Dans le reste de ce manuscrit, la transition sera une variation linéaire s'étalant sur $50\mu s$.

5.3.3 Portes des bibliothèques TIMA

Les portes de la bibliothèque TAL ont été caractérisées selon cette méthode. De nombreuses portes fonctionnaient correctement : pour toutes leurs entrées, $V_{TL} < V_{TH}$. Comme pour les portes logiques combinatoires, l'étage d'entrée de la fonction C-element est réalisé par la mise en série d'un transistor par entrée logique. L'étude des portes logiques combinatoires a mis en évidence l'influence de la mise en série de nombreux transistors dans les tensions de commutations mesurées. Les portes C-elements comportant de nombreuses entrées risquent donc plus d'avoir des problèmes au niveau des tensions de basculement.

La [table 5.3](#) indique les tensions de basculement V_{TL} et V_{TH} mesurées pour chacune des quatre entrées logiques (notées A, B, C et D) des portes de structure *Weak-Feedback* à quatre entrées (WM4). Ces portes sont issues de la TAL en technologie 130nm *Low Leakage* (LL) et déclinée avec cinq sortances différentes. Les valeurs en gras indiquent les cas problématiques où pour une entrée i , les tensions de basculement sont inversées ($V_{TLi} > V_{THi}$). Ces mesures mettent en évidence les inversions des tensions de basculement pour les entrées connectées aux transistors les plus éloignés des rails d'alimentation (C et D).

| sortance | V_{TLA} | V_{THA} | V_{TLB} | V_{THB} | V_{TLC} | V_{THC} | V_{TLD} | V_{THD} |
|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| WM4X0 | 0.525 | 0.732 | 0.562 | 0.691 | 0.605 | 0.646 | 0.647 | 0.601 |
| WM4X1 | 0.511 | 0.683 | 0.552 | 0.651 | 0.596 | 0.613 | 0.641 | 0.576 |
| WM4X2 | 0.504 | 0.605 | 0.546 | 0.585 | 0.592 | 0.560 | 0.637 | 0.534 |
| WM4X4 | 0.548 | 0.547 | 0.584 | 0.535 | 0.622 | 0.518 | 0.661 | 0.501 |
| WM4X8 | 0.631 | 0.504 | 0.654 | 0.497 | 0.680 | 0.486 | 0.706 | 0.474 |

Table 5.3 – Caractérisation des quatre entrées de portes WM4 en technologie ST 130nm LL

Les portes à sortances élevées (dont le suffixe est X4 et X8) ont un comportement plus assujéti à l'inversion des seuils de basculement. En effet, leurs étages de sortie sont constitués de transistors de largeur élevée afin de respecter les contraintes de sortance. La grille de ces transistors présente donc une surface élevée. Cela se traduit dans le modèle électronique par une capacité équivalente élevée sur le nœud Y ([figures 3.1\(a\) et 3.1\(b\)](#) - [page 35](#)). Dans le but de minimiser le délai de la porte, il est nécessaire de charger cette capacité le plus rapidement possible afin de permettre le basculement de la tension de sortie. Le concepteur met habituellement en œuvre un compromis entre les deux solutions suivantes :

- L'étage d'entrée est surdimensionné afin de charger cette capacité avec un courant élevé. Cela se traduit par un pic très important de courant absorbé sur l'alimentation, ce qui n'est généralement pas souhaitable.
- L'étage d'entrée et l'étage *keeper** sont redimensionnés afin d'anticiper les transitions sur les entrées (*Early Threshold** [vB92]) : l'étage d'entrée commence plus tôt la charge de la capacité du nœud Y et cela réduit le délai de la porte. Cette méthode aboutit généralement à une prise en compte trop précoce des fronts montants et descendant et donc à l'inversion des seuils de basculement ; c'est exactement ce que nous souhaitons éviter.

Ces portes ayant été conçues sous des contraintes de consommation et de rapidité, le compromis tend vers l'utilisation de la seconde solution pour les portes à sortances élevées. Les portes WM4X4 et WM4X8 ont donc ce problème sur toutes leurs entrées.

La [table 5.4](#) indique les valeurs relevées pour les portes de la TAL en technologie ST 65nm LL. Ces cellules asynchrones ont une structure conventionnelle. Toutes les entrées des C-elements de deux à quatre entrées et déclinées en trois sortances ont été caractérisées. Les entrées C et D ne sont disponibles que pour les portes ayant respectivement au moins trois et quatre entrées.

| porte/sortance | V_{TLA} | V_{THA} | V_{TLB} | V_{THB} | V_{TLC} | V_{THC} | V_{TLD} | V_{THD} |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| CM2X4 | 0.630 | 0.496 | 0.628 | 0.498 | <i>N/A</i> | <i>N/A</i> | <i>N/A</i> | <i>N/A</i> |
| CM2X18 | 0.697 | 0.416 | 0.701 | 0.416 | <i>N/A</i> | <i>N/A</i> | <i>N/A</i> | <i>N/A</i> |
| CM2X35 | 0.740 | 0.379 | 0.747 | 0.378 | <i>N/A</i> | <i>N/A</i> | <i>N/A</i> | <i>N/A</i> |
| CM3X4 | 0.628 | 0.482 | 0.618 | 0.486 | 0.617 | 0.487 | <i>N/A</i> | <i>N/A</i> |
| CM3X18 | 0.691 | 0.419 | 0.686 | 0.421 | 0.690 | 0.421 | <i>N/A</i> | <i>N/A</i> |
| CM3X35 | 0.726 | 0.379 | 0.724 | 0.379 | 0.730 | 0.379 | <i>N/A</i> | <i>N/A</i> |
| CM4X4 | 0.680 | 0.428 | 0.674 | 0.430 | 0.669 | 0.432 | 0.672 | 0.431 |
| CM4X18 | 0.745 | 0.356 | 0.743 | 0.356 | 0.742 | 0.357 | 0.748 | 0.356 |
| CM4X35 | 0.745 | 0.381 | 0.743 | 0.382 | 0.742 | 0.383 | 0.748 | 0.382 |

Table 5.4 – Caractérisation de CM en technologie ST 65nm LL

Toutes les portes testées sont affectées par le phénomène d'inversion de seuils. Il est possible d'expliquer ce comportement en analysant la structure conventionnelle : l'étage *keeper** est désactivée dès que les deux entrées ont un niveau logique suffisamment marqué : les transistors T_{N5} , T_{N6} , T_{P5} et T_{P6} (figure 3.1(b) - page 35) deviennent bloquants et l'étage d'entrée peut alors librement basculer le nœud interne Y . Ce comportement intrinsèque de la structure conventionnelle implique dans la plupart des cas une inversion des seuils de commutation.

5.3.4 Autres structures de C-elements

Les tests ont été étendus à des C-elements implémentant d'autres structures :

- la structure symétrique à deux entrées représentée dans la figure 3.3(a) (page 37) ;
- la structure dynamique à deux entrées représentée dans la figure 3.3(b) (page 37) ;
- la structure composite à deux entrées réalisée à base de portes NAND (NM2) représentée dans la figure 3.4(b) (page 38) ;
- la structure composite à deux entrées réalisée à base de portes AND et OR (OM2) représentée dans la figure 3.4(a) (page 38) ;
- la structure composite à deux entrées réalisée à base de portes NAND, AND et OR (AM2) représentée dans la figure 3.4(c) (page 38).

Ces portes ne sont pas issues de la TAL et ont été conçues spécialement pour les besoins de nos mesures en technologie ST 130 nm.

| structure | ST 130nm LL | | ST 130nm HS | |
|-----------------------------|--------------|--------------|--------------|--------------|
| | V_{TL} | V_{TH} | V_{TL} | V_{TH} |
| symétrique (SM2) | 0.713 | 0.466 | 0.767 | 0.427 |
| dynamique (DM2) | 0.976 | 0.131 | 0.980 | 0.112 |
| à base de NAND2 (NM2) | 0.548 | 0.544 | 0.531 | 0.521 |
| à base de AND/OR (OM2) | 0.531 | 0.531 | 0.507 | 0.507 |
| à base de NAND/AND/OR (AM2) | 0.531 | 0.531 | 0.507 | 0.508 |

Table 5.5 – Caractérisation des autres structures de C-elements

Les structures composites sont moins sensibles aux phénomènes d'inversion de seuils que les structures de type standard-cell full-custom. De part l'utilisation des portes combinatoires où les seuils de basculements sont identiques pour une même entrée, les structures composites auront ces mêmes propriétés. Les différences dans les tensions de basculement des C-elements sont liées à la différence de chemin pris pour aboutir à la transition de la sortie.

Dans la porte NM2, les transitions montantes sur l'entrée A (V_A) de la porte passent par la porte N_1 et les transitions descendantes par N_2 .

Les structures symétriques et dynamiques ont un comportement intrinsèque à l'inversion de seuils. Ces portes sont conçues pour commuter très rapidement et ont été, dès leur introduction, qualifiées de *Early Threshold** [vB92].

5.4 Correction des seuils des portes

Afin de corriger le "mauvais" comportement des C-elements étudiés précédemment, deux méthodes de corrections ont été mises en œuvre afin de réordonner les tensions de basculement [ATRY10].

Le basculement du point mémoire fait appel à l'action de l'étage d'entrée qui doit annuler l'effet de l'étage de maintien (le *keeper**). Le basculement du point mémoire intervient lorsque l'étage d'entrée tire un courant supérieur à celui que peut écouler l'étage de maintien. A ce moment là seulement, le potentiel du nœud interne peut basculer. Afin de rendre la commutation plus difficile, il faut augmenter la force de l'étage de maintien face à l'étage d'entrée.

5.4.1 Redimensionnement des transistors

Le redimensionnement des transistors du *keeper** est possible pour les structures WM2, CM2 et SM2. Ces structures seront donc étudiées lorsque leur étage de maintien est renforcé.

La largeur initiale de tous les transistors de maintien, aussi bien NMOS que PMOS, est multipliée par un facteur d'échelle noté k . Cela a pour conséquence d'augmenter le courant de saturation des transistors de maintien et donc de retarder l'instant de commutation : les transistors de l'étage d'entrée ne pourront sortir un courant supérieur que plus tardivement dans la transition d'entrée.

5.4.1.1 Structure *Weak-Feedback*

La figure 5.3 représente les variations typiques des seuils de basculements en fonction du facteur d'échelle k pour une structure WM2. Les résultats ont été obtenus avec la technologie ST 130 nm HS alimentée à $V_{DD} = 1.2V$ mais des résultats comparables sont obtenus pour d'autres technologies.

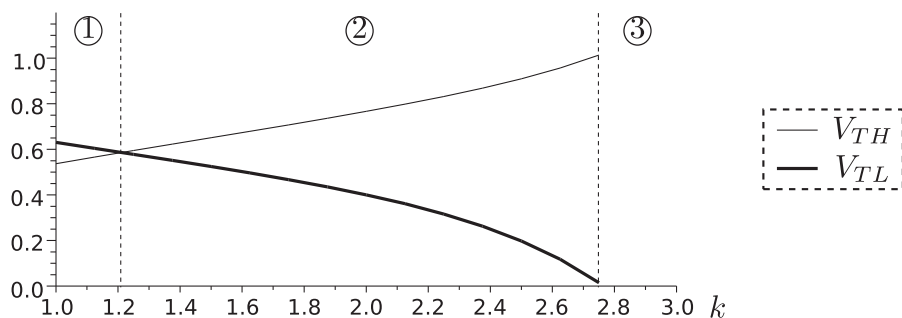


FIGURE 5.3 – Variation des seuils de commutations lors du redimensionnement d'une WM2

Ces résultats font apparaître trois zones caractéristiques :

- Dans la zone ①, la porte présente les symptômes des seuils inversés. Le cas particulier $k = 1.0$ correspond au dimensionnement initial de la porte ;
- Dès que l'étage de maintien devient assez puissant ($k > 1.2$), les instants de basculement sont suffisamment retardés pendant les transitions pour satisfaire l'hypothèse $V_{TL} < V_{TH}$ (zone ②) ;
- Si l'étage de maintien devient trop puissant ($k > 2.75$), l'étage d'entrée n'est plus en mesure de faire basculer le point mémoire, ce cas est représenté dans la zone ③.

Il est nécessaire de placer le point de fonctionnement de la porte dans la zone ②. Ces changements ont un coût en terme de délai et d'énergie de commutation. Le décalage de l'instant de décision et la plus forte réaction de l'étage de maintien augmentent ces grandeurs au fur-et-à-mesure que k augmente. La figure 5.4(a) indique l'évolution du délai de propagation lorsque le facteur d'échelle k est augmenté, la figure 5.4(b) l'énergie de commutation.

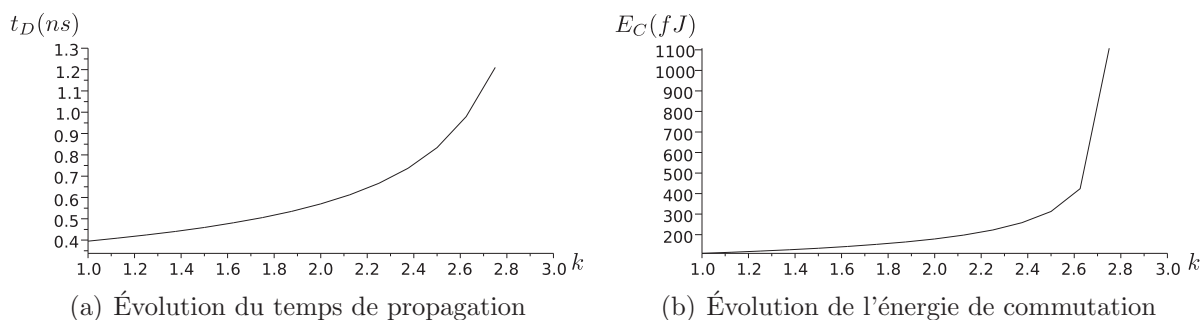


FIGURE 5.4 – Coût des redimensionnements dans une structure WM2

Ces deux courbes ont des variations semblables lorsque le facteur k augmente et présentent de très fortes augmentations pour les k élevés. Les asymptotes horizontales pour les faibles redimensionnements et finalement un facteur k de l'ordre de 1.2 a un coût de 6.7% en terme de délai et 8.2% pour l'énergie de commutation. Ces deux coûts de fonctionnement sont assez limités. Le coût de conception lié à la consommation en surface de la *standard-cell** est faible, voire nul : la modification de la largeur des transistors de maintien se traduit par une augmentation de la hauteur des zones de diffusion N et P

de ces transistors (figure 3.2 - page 36). Les largeurs de ces transistors étant initialement faible, ce redimensionnement peut se faire sans augmenter la surface de la *standard-cell**.

5.4.1.2 Structure conventionnelle

Lorsque la technique de redimensionnement de l'étage de maintien est appliquée à la structure conventionnelle, les résultats obtenus sont semblables à la différence près que la porte ne cesse jamais de fonctionner : la zone ③ vue précédemment n'existe pas : les redimensionnements permettent toujours à la sortie de la porte de basculer. La figure 5.5 trace l'évolution des tensions de basculement obtenue pour une CM2 en technologie 65nm LL alimentée à $V_{DD} = 1.2V$.

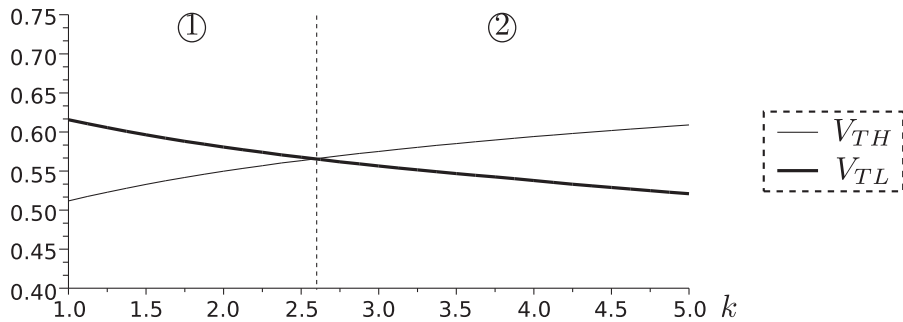


FIGURE 5.5 – Variation des seuils de commutations lors du redimensionnement d'une CM2

Les tensions de basculement varient moins rapidement en fonction de k que dans le cas de la structure *Weak-Feedback* : il est nécessaire d'augmenter ce coefficient jusqu'à $k = 2.6$ pour avoir le comportement souhaité. A ce point, le coût en terme de délai de propagation est 21.1% et celui en énergie de commutation est de 22.7%. Malgré ces importants coefficients, l'aire de la structure n'est pas augmentée par le redimensionnement des transistors.

5.4.1.3 Structure Symétrique

Dans la structure symétrique (figure 3.3(a) - page 37), le coefficient de redimensionnement k est appliqué aux seuls transistors T_{N4} et T_{P4} . Les mesures représentées dans la figure 5.6 sont obtenues avec une cellule conçue en technologie 45nm LL alimentée à $V_{DD} = 1.2V$.

Ces changements ne permettent pas d'influencer suffisamment sur les seuils de basculements pour corriger le comportement de la porte. Cela vient du fait que les étages d'entrée et de maintien ne sont pas indépendants dans cette structure : il n'est pas possible de changer le comportement de l'un sans influencer sur l'autre simultanément.

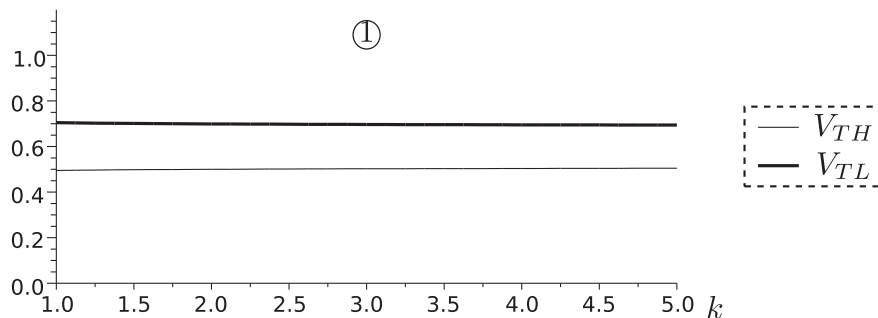


FIGURE 5.6 – Variation des seuils de commutations lors du redimensionnement d’une SM2

5.4.1.4 Conclusions sur le redimensionnement des transistors

La technique de redimensionnement des transistors peut être appliquée à toutes les structures disposant d’un étage de maintien distinct pour améliorer son comportement lors des transitions. Les portes CM2, SM2, WM2 ont été conçues dans différentes technologies afin de dégager des tendances.

La [table 5.6](#) représente les valeurs de seuils mesurées pour les structures WM2 et CM2. Comme précédemment, les tensions de seuils en gras indiquent les cas où elles sont inversées. Les structures SM2 ayant toutes ce comportement et le redimensionnement des transistors n’apportant aucune amélioration, elles n’ont pas été rapportées dans cette table. Pour chaque porte dont le comportement n’est pas correct, le coefficient de redimensionnement k est alors borné par une valeur minimum (k_{min} correspondant à la limite entre les zones ① et ② vues précédemment) et éventuellement une valeur maximum (k_{max} correspondant à la limite entre les zones ② et ③ lorsque cette dernière existe). Les pourcentages C_D et C_E représentent respectivement les coûts en délai de propagation et en énergie de commutation.

Le dimensionnement des cellules est identique pour les variantes de chaque famille technologique : les tailles des transistors sont les mêmes dans les variantes LH, LS et LL d’une même structure en technologie 65nm. Les technologies rapides étudiées (130nm HS - *High Speed* -, 65nm LL - *Low power Low V_T* - et 45nm LL - *Low power Low V_T*) ont un comportement plus propices aux inversions de seuils : à finesse de gravure équivalente, les V_{TL} (resp. V_{TH}) obtenus dans ces variantes sont inférieurs (resp. supérieurs) à ceux obtenus dans les technologies un peu moins rapides. Elles facilitent donc les inversions des seuils.

La structure *Weak-Feedback* semble plus robuste que la structure conventionnelle, seules les cellules WM2 réalisées en technologie ST 130 nm et optimisées en vitesse présentent l’inversion de seuil. Dès que cette contrainte est relâchée lors de la conception, le comportement des portes est alors correct.

| Technologie | <i>Weak-Feedback</i> | | Conventionnelle | |
|-------------|----------------------------------------------------------------|-----------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------|
| 130nm HS | $V_{TL} = \mathbf{0.631}$ $k_{min} = 1.2$ $C_D = 6.66\%$ | $V_{TH} = \mathbf{0.537}$ $k_{max} = 2.75$ $C_E = 8.23\%$ | $V_{TL} = \mathbf{0.705}$ $k_{min} = 3.4$ $C_D = 11.3\%$ | $V_{TH} = \mathbf{0.396}$ $k_{max} = N/A$ $C_E = 7.98\%$ |
| 130nm LL | $V_{TL} = 0.526$ $k_{min} = N/A$ $C_D = N/A$ | $V_{TH} = 0.657$ $k_{max} = N/A$ $C_E = N/A$ | $V_{TL} = \mathbf{0.655}$ $k_{min} = 3.5$ $C_D = 10.6\%$ | $V_{TH} = \mathbf{0.469}$ $k_{max} = N/A$ $C_E = 7.83\%$ |
| 65nm LL | $V_{TL} = 0.355$ $k_{min} = N/A$ $C_D = N/A$ | $V_{TH} = 0.847$ $k_{max} = N/A$ $C_E = N/A$ | $V_{TL} = \mathbf{0.628}$ $k_{min} = 3.1$ $C_D = 26.7\%$ | $V_{TH} = \mathbf{0.498}$ $k_{max} = N/A$ $C_E = 29.1\%$ |
| 65nm LS | $V_{TL} = 0.306$ $k_{min} = N/A$ $C_D = N/A$ | $V_{TH} = 0.925$ $k_{max} = N/A$ $C_E = N/A$ | $V_{TL} = \mathbf{0.624}$ $k_{min} = 3.6$ $C_D = 27.8\%$ | $V_{TH} = \mathbf{0.529}$ $k_{max} = N/A$ $C_E = 36.7\%$ |
| 65nm LH | $V_{TL} = 0.251$ $k_{min} = N/A$ $C_D = N/A$ | $V_{TH} = 1.00$ $k_{max} = N/A$ $C_E = N/A$ | $V_{TL} = \mathbf{0.632}$ $k_{min} = N/A$ $C_D = N/A$ | $V_{TH} = \mathbf{0.544}$ $k_{max} = N/A$ $C_E = N/A$ |
| 45nm LL | $V_{TL} = 0.309$ $k_{min} = N/A$ $C_D = N/A$ | $V_{TH} = 0.715$ $k_{max} = N/A$ $C_E = N/A$ | $V_{TL} = \mathbf{0.613}$ $k_{min} = 1.6$ $C_D = 8.79\%$ | $V_{TH} = \mathbf{0.536}$ $k_{max} = N/A$ $C_E = 6.89\%$ |
| 45nm LS | $V_{TL} = 0.301$ $k_{min} = N/A$ $C_D = N/A$ | $V_{TH} = 0.716$ $k_{max} = N/A$ $C_E = N/A$ | $V_{TL} = \mathbf{0.622}$ $k_{min} = 1.7$ $C_D = 8.37\%$ | $V_{TH} = \mathbf{0.566}$ $k_{max} = N/A$ $C_E = 7.79\%$ |

Table 5.6 – Corrections des seuils dans différentes structures de C-element

5.4.2 Réduction de la tension d'alimentation

La modification de la tension permet aussi de modifier le comportement analogique des portes logiques. Les seuils de basculement dépendent des dimensions des transistors mais aussi de leur point de polarisation. Celui-ci évolue fortement en cas de modification de la tension d'alimentation.

5.4.2.1 Polarisation des transistors

Lors de la réduction de la tension d'alimentation, les caractéristiques des transistors ne varient pas uniformément : la figure 5.7(a) est un tracé des variations relatives des courants de saturations. Les transistors sont polarisés dans leur zone saturée comme indiqué dans la figure 5.7(b). Plusieurs transistors NMOS de mêmes longueurs mais de largeurs différentes ont été simulés.

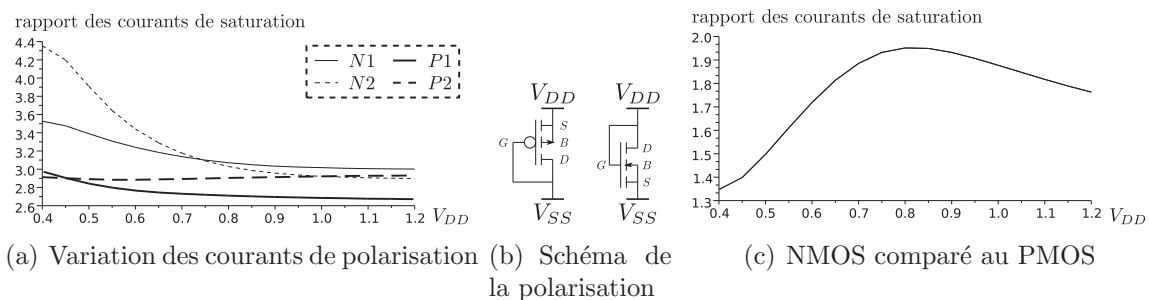


FIGURE 5.7 – Caractérisations des transistors en fonction de V_{DD}

La courbe N_1 trace l'évolution du rapport des courants entre un transistor NMOS de largeur minimale (grandeur notée W_{min}) et un transistor NMOS ayant trois fois cette largeur ($3*W_{min}$). La courbe N_2 est le rapport des courants entre deux transistors NMOS de largeurs $9*W_{min}$ et $3*W_{min}$. Les courbes P_1 et P_2 ont été obtenues avec des transistors de mêmes dimensions.

Théoriquement, le courant d'un transistor MOS en régime linéaire est donné par la relation bien connue de l'équation 5.5.

$$I_{DS} \approx \mu_0 \frac{C_{ox}}{2} \frac{W_E}{L_E} \frac{(V_{GS} - V_T)^2}{(1 + D)} (1 + \lambda_E V_{DS}) \quad (5.5)$$

Les paramètres de cette équation sont :

- I_{DS} est le courant entre le drain et la source ;
- μ_0 est la mobilité des porteurs de charges dans le canal du transistor ;
- C_{ox} est la capacité surfacique de la grille ;
- W_E est la largeur effective du transistor, celle-ci peut-être différente de la largeur dessinée à cause des déformations liées à la fabrication (diffusion du dopant en dehors de la zone prévue. . .) ;
- L_E est la longueur effective du transistor ;

- V_{GS} est la tension de commande appliquée entre la grille et la source ;
- V_T est la tension de seuil du transistor ;
- D modélise l'effet de substrat*, ce paramètre est lié à la polarisation du substrat (pôles B dans la figure 5.7(b)) ;
- λ_E modélise l'effet *Early** où la longueur du canal varie en fonction de la tension V_{DS} .

Cette approximation ne modélise pas fidèlement les résultats obtenus. En effet, seul le paramètre W_E varie entre les différents transistors NMOS simulés. Le rapport entre les courants I_{DS} des différents transistors NMOS devrait donc être égal au rapport de leurs largeurs W et donc être constant en fonction de V_{DD} . Le même raisonnement peut être tenu au sujet des transistors PMOS.

Dans la pratique, on observe dans la figure 5.7(a) que le rapport n'est pas constant en fonction de V_{DD} : au fur et à mesure que V_{DD} baisse, N_1 et N_2 augmentent. Les courants de saturations des transistors de petites largeurs baissent moins que les courants de saturations des transistors de largeurs plus élevées. Cet effet est moins marqué pour les transistors PMOS : la courbe P_2 est sensiblement constante et l'augmentation de la courbe P_1 pour les petits V_{DD} est plus faible.

La tension d'alimentation est réduite pour tous les étages des portes. L'équilibre entre l'étage d'entrée et l'étage de maintien sera donc modifié dans les structures C-elements lorsque la tension est diminuée. Cela se traduit par des modifications importantes dans les tensions de seuils. Par exemple, lors d'une transition descendante de la sortie de la porte, la transition montante du nœud interne Y (figures 3.1(a) et 3.1(b) - page 35 et figure 3.3(a) - page 37) fait appel aux transistors NMOS de l'étage d'entrée et aux transistors PMOS de l'étage de maintien. La figure 5.7(c) trace le rapport des courants de saturation entre deux transistors NMOS et PMOS de dimensions minimales : ce rapport tend à diminuer lorsque V_{DD} diminue. Donc à priori, l'étage d'entrée devient de moins en moins puissant par rapport à l'étage de maintien lorsque la tension est réduite. Cette particularité rend la commutation du point mémoire plus difficile et peut avoir pour conséquence une amélioration du comportement. Les conséquences de la réduction de V_{DD} doivent donc être étudiées pour les différentes structures.

5.4.2.2 Structure *Weak-Feedback*

Lors de la réduction de l'alimentation d'un C-element *Weak-Feedback*, l'écart entre les tensions de seuil tend à être constant. La figure 5.8(a) illustre ce cas lorsque le comportement de la porte est initialement correct. Cette non modification de l'écart induit le comportement correct des portes tant que la tension d'alimentation reste supérieure à cet écart.

Pour les portes dont le comportement n'était pas correct à V_{DD} nominal, l'effet tend à s'inverser pour les V_{DD} plus faibles (figure 5.8(b)). Pour $V_{DD} < 1.01V$, le comportement devient correct (zone ②). Les variations des tensions de seuils demeurent très faibles en fonction de la tension d'alimentation : il est nécessaire de dégrader les performances du circuits pour avoir un comportement correct. A ce point, le coût du délai de propagation

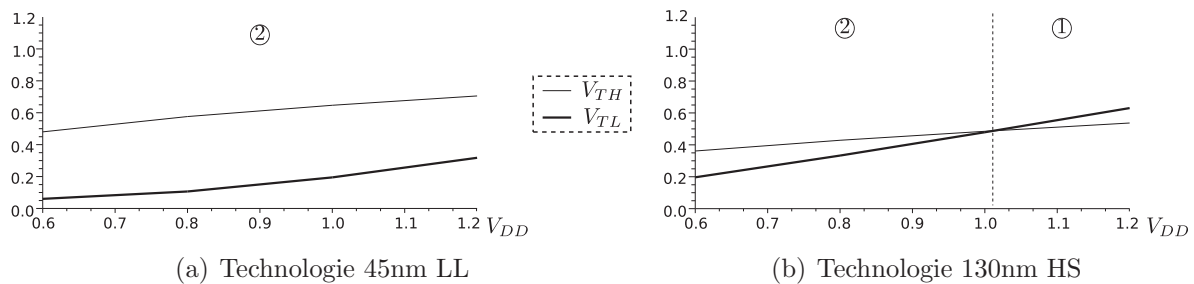


FIGURE 5.8 – Seuils de commutations d’une WM2 lors des variations de la tension d’alimentation

est de 14.1%. A l’opposé, l’énergie nécessaire au basculement est diminuée de 31.8%.

5.4.2.3 Structures Conventiennelle et symétrique

Les structures statiques conventiennelle et symétrique ne présentent pas le même comportement face à la réduction de leur tension d’alimentation. Leur comportement n’étant pas initialement correct, la réduction de la tension d’alimentation ne va pas corriger l’ordre des seuils (figures 5.9(a) et 5.9(b)).

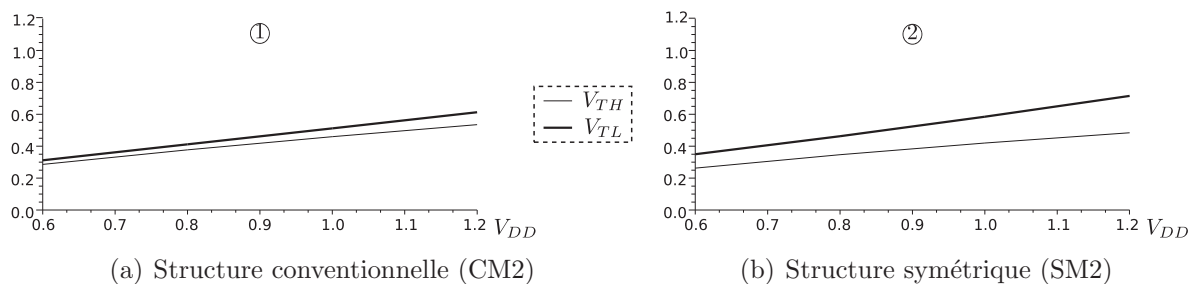


FIGURE 5.9 – Seuils de commutations en fonction de la tension d’alimentation

L’utilisation de la seule méthode de réduction de la tension d’alimentation ne peut donc pas corriger le comportement des portes que nous avons conçues. Il est donc nécessaire de coupler les deux méthodes pour corriger leurs comportements, cette étude est menée dans la section 5.4.3.

5.4.2.4 Structure dynamique et structures composites

Dans ces structures, le seul curseur possible est la réduction de la tension d’alimentation car elles ne possèdent pas d’étage de maintien. La figure 5.10 représente l’évolution des tensions de seuils en fonction de la tension d’alimentation pour une porte de structure dynamique. Le comportement de cette porte n’est pas amélioré lors de la réduction de la tension d’alimentation.

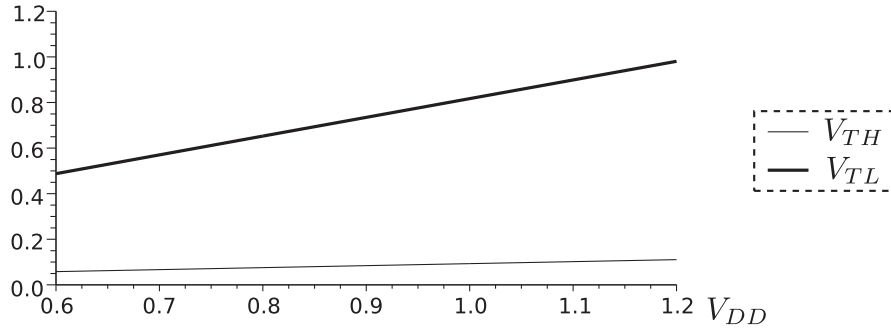


FIGURE 5.10 – Variation des seuils de commutations dans une DM2

Pour les structures composites à base de portes logiques NAND et NOR, les tensions de seuils sont toujours sensiblement identiques. Leurs valeurs diminuent lors de la réduction de la tension d'alimentation mais leur ordre reste inchangé.

5.4.3 Variations simultanées

L'utilisation indépendante des deux méthodes de correction permet de corriger le comportement de certaines portes. Dans le but de corriger le comportement des structures conventionnelles et symétriques, les deux méthodes peuvent être couplées. Quant à la structure dynamique, les écarts entre les tensions de seuils sont si importants qu'il semble impossible de corriger son comportement.

5.4.3.1 Structure *Weak-Feedback*

Les deux méthodes appliquées individuellement sur une structure *Weak-Feedback* permettent de satisfaire l'exigence sur l'ordre des tensions de basculement. La figure 5.11(a) trace les tensions de basculement lorsque les deux paramètres varient simultanément pour la technologie ST 130nm HS. La courbe de V_{TH} passe au dessus de V_{TL} dès que l'on s'éloigne un peu du point correspondant à $V_{DD} = 1.2V$ et $k = 1$.

L'intersection entre ces deux courbes permet de déterminer des compromis entre les deux méthodes. La figure 5.11(b) est la projection de la courbe précédente sur le plan (V_{DD}, k) . La zone ① correspond à la zone où V_{TL} est au dessus (comportement non correct), la zone ② au comportement correct. La jointure entre ces deux zones est quasiment une droite : aux imprécisions de la représentation près, celle-ci coupe l'axe k à la limite du comportement correct obtenu lors du redimensionnement des transistors (section 5.4.1) et l'axe V_{DD} à la limite du comportement correct obtenue avec la technique de réduction de la tension d'alimentation (section 5.4.2).

5.4.3.2 Structure Conventiennelle

Dans la structure conventionnelle, le cumul des deux méthodes est représenté dans la figure 5.12. Ces résultats ont été obtenus dans la technologie ST 65nm LL, les zones

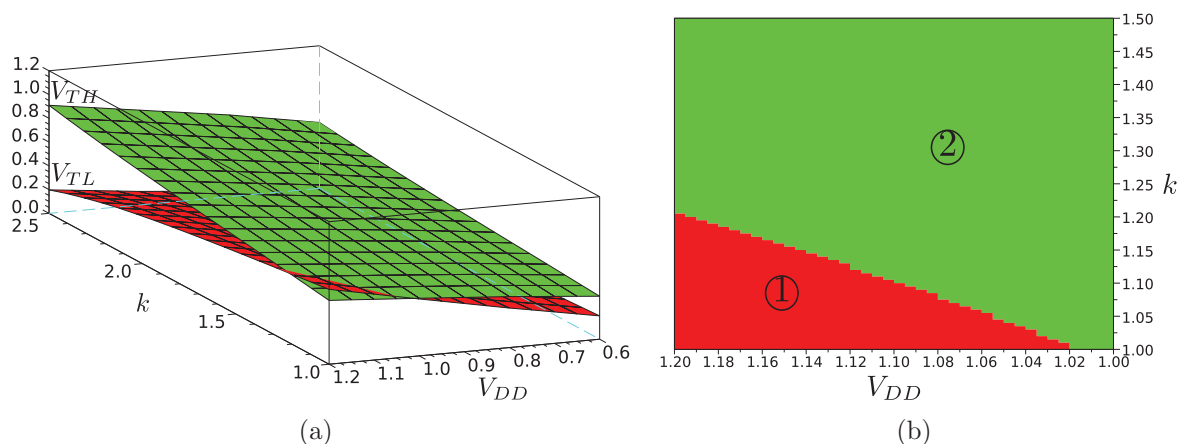


FIGURE 5.11 – Variation des seuils de commutations dans une WM2

sont projetées sur le plan (V_{DD}, k) .

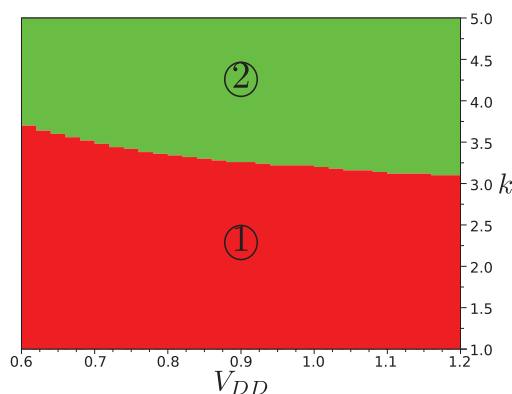


FIGURE 5.12 – Variation des seuils de commutations dans une CM2

Dans cette structure, les effets des deux méthodes ne s'additionnent pas : plus V_{DD} diminue, plus k doit être important pour obtenir le comportement correct. Ce phénomène est dû à l'augmentation des courants de saturation des petits transistors lorsque la tension d'alimentation est réduite (section 5.4.2.1) :

- à dimensionnement initial, les transistors de l'étage de maintien sont moins larges que les transistors de l'étage d'entrée : l'étage de maintien devient de plus en plus fort par rapport à l'étage d'entrée lorsque la tension d'alimentation est réduite ;
- à fort paramètre de redimensionnement k , les transistors de l'étage de maintien sont plus larges que les transistors de l'étage d'entrée : l'étage de maintien devient de moins en moins fort par rapport à l'étage d'entrée lorsque la tension d'alimentation est réduite.

5.4.3.3 Structure Symétrique

Les deux méthodes appliquées individuellement ne permettent pas de corriger le comportement des cellules à structures symétriques. Leur application simultanée est

représentée dans la figure 5.13 pour la technologie ST 45nm LS. Dans cette technologie, le comportement de la porte est toujours le même : la courbe de V_{TL} est toujours au dessus de la courbe de V_{TH} et par conséquent le comportement de la porte ne peut être corrigé.

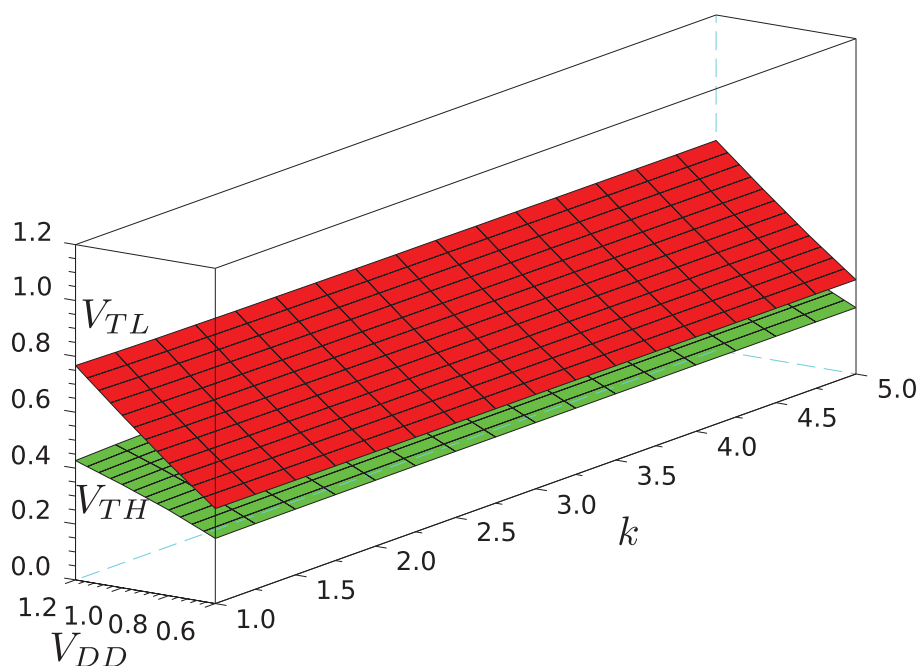


FIGURE 5.13 – Variation des seuils de commutations dans une SM2

Quelques soient les technologies, toutes les portes réalisées avec cette structure de porte exhibent un phénomène d'inversion de seuils.

Conclusion

Les différentes structures de C-elements ont des caractéristiques différentes dès que leur comportement analogique est étudié. Deux techniques ont été mises en œuvre pour corriger les portes dont la conception initiale impliquait l'inversion des seuils de commutation : la réduction de la tension d'alimentation et le redimensionnement des transistors de l'étage de maintien. Les différentes structures étudiées réagissent différemment à ces modifications :

- la structure dynamique bascule très rapidement au tout début de la transition. Elle est très affectée par le phénomène d'inversions des seuils de commutations et les méthodes employées n'ont pas permis de corriger ce comportement ;
- la structure symétrique a également un basculement précoce pendant les transitions (caractéristique *Early-Threshold*). Les corrections employées permettent de réduire l'écart entre les tensions de basculement mais celles-ci resteront toujours inversées.
- les seuils de commutation de la structure conventionnelle sont modifiés lorsque les transistors du *Keeper* sont redimensionnés, mais il est nécessaire de réduire la

tension d'alimentation pour obtenir l'ordre correct dans les tensions de basculement synonyme de comportement correct ;

- la tension d'alimentation n'a quasiment aucun effet sur l'écart entre les tensions de basculements de la structure *Weak-Feedback*, par contre le redimensionnement des transistors de l'étage de maintien permet de corriger aisément le comportement de la porte dans les cas où sa conception initiale était incorrecte ;
- les structures composites ne sont pas très affectées par l'inversion des seuils de commutations. En effet, les *standard cells* utilisées ont généralement la même tension de seuil.

Il est donc possible d'obtenir ou de corriger le comportement de nombreuses structures de C-elements afin que les seuils de basculement soient toujours correctement ordonnés. Cette précaution permet d'assurer l'immunité aux pentes lentes des circuits asynchrones réalisés à l'aide de ces portes.

Chapitre 6

Prise en compte des variations de process

| | |
|------------------------------------------------------------|-----------|
| Introduction | 88 |
| 6.1 Influence technologique | 88 |
| 6.1.1 Variations des paramètres | 88 |
| 6.1.2 Disparité des paramètres | 89 |
| 6.2 Analyses des variations | 89 |
| 6.2.1 Analyses statistiques | 89 |
| 6.2.1.1 Indépendance des seuils | 90 |
| 6.2.2 Marges de conception | 91 |
| 6.2.2.1 Marges pour des variables aléatoires indépendantes | 91 |
| 6.2.2.2 Prise en compte des corrélations | 92 |
| 6.3 Résultats obtenus | 92 |
| 6.3.1 Redimensionnement de l'étage de maintien | 92 |
| 6.3.2 Réduction de la tension d'alimentation | 93 |
| 6.3.3 Autres phénomènes influençant les tensions de seuils | 93 |
| Conclusion | 95 |

Introduction

Les seuils de commutation mesurés dans les chapitres précédents sont déterminés dans les conditions nominales des technologies employées. Les modèles de simulation représentent le cas moyen des caractéristiques qui peuvent être obtenues dans la réalité. Lors des opérations de fabrication, des disparités de caractéristiques apparaissent et peuvent changer le comportement électrique du circuit.

Le kit de développement de *STMicroelectronics* permet d'estimer ces variations de caractéristiques dès la phase de conception. Ces changements peuvent donc être anticipés afin de garantir le bon fonctionnement d'un grand nombre des circuits fabriqués en appliquant des marges dans l'écart entre les seuils de basculement.

6.1 Influence technologique

Lors des étapes de fabrication, diverses variations physiques impactent les caractéristiques électriques des transistors et des interconnexions (liste non exhaustive) :

- les dimensions effectives des motifs du *layout** dessinés lors de la conception des standard-cells peuvent être altérées lors du report du masque : déformations, changements de dimensions. . .
- les alignements des différents masques successifs peuvent introduire des décalages entre les différentes couches d'un transistor ou des interconnexions ;
- les différents dopages nécessaires pour le fonctionnement du transistor peuvent varier en intensité, en répartition et en diffusion.

Ces changements sont d'autant plus importants que la technologie de gravure est fine. En effet une variation de $5nm$ aura une importance relative plus élevée en $45nm$ qu'en $130nm$.

6.1.1 Variations des paramètres

Les circuits réalisés dans les technologies fines actuelles ($45nm$, $40nm$. . .) auront des disparités de caractéristiques électriques impactant le comportement du circuit. Les fondeurs ont modélisé ces variations de manière statistique et il est possible de simuler le comportement de plusieurs circuits "réels" par le biais de plusieurs simulations successives (*simulations Monte-Carlo**). Pour chacune d'entre elles, les paramètres physiques du circuit sont fixées à des valeurs statistiquement possibles selon les procédés de fabrication.

Ces simulations se font dans le même environnement que précédemment, les modèles des composants électroniques du circuit sont automatiquement modifiés pour refléter les dispersions. Chaque simulation permet de reproduire le comportement d'une combinaison particulière de modifications physiques. Il est nécessaire de reproduire un nombre important de simulations (500 dans notre cas) pour quantifier statistiquement assez

précisément les variations du circuit réel.

6.1.2 Disparité des paramètres

Les paramètres du circuit peuvent varier globalement : tous les transistors ou toutes les interconnexions sont alors affectées du même phénomène parasite. Des variations locales existent également, on parle alors de *mismatches**. Ces variations locales peuvent être dues à des gradients ou à des absences d'uniformité de caractéristiques physiques sur la totalité de la surface du circuit.

Les effets de ces variations relatives sont d'autant plus importants que la distance sur le circuit est importantes entre les composants. A l'échelle de la cellule logique, les effets des *mismatches** sont très limités puisque tous les composants sont regroupés sur quelques micromètres.

Les modèles de simulations fournis par le fabricant permettent de prendre en compte ces variations locales. Les simulations montrent qu'elles n'impactent que de quelques millivolts les tensions de basculement des portes logiques. Leur ajout étant assez aisé et peu coûteux, elles seront prises en compte dans la suite de ce chapitre.

6.2 Analyses des variations

Le protocole de simulation est identique aux mesures du [chapitre 5](#) : la caractérisation des seuils de basculement s'effectue pour chaque entrée à l'aide de transitions lentes d'une durée de $50\mu s$. Le seuil de basculement est atteint lorsque la tension de sortie de la porte franchit la limite de la moitié de la tension d'alimentation. La simulation est répétée 500 fois pour prendre en compte un nombre important de variations globales et locales.

6.2.1 Analyses statistiques

Les tensions de seuils mesurées pour chaque simulation sont analysées statistiquement afin d'évaluer leur distribution et ainsi borner l'intervalle des valeurs possibles. Les tensions de seuils sont alors considérées comme des variables aléatoires. Pour toutes les mesures réalisées, leurs distributions sont de type gaussiennes caractérisables par leur valeur moyenne et leur écart-type : pour une variable aléatoire x , la valeur moyenne (également appelée espérance) est noté \bar{x} et l'écart-type σ_x .

La [figure 6.1](#) représente les distributions obtenues pour une structure conventionnelle (CM2) en technologie 45nm LS. L'écart des tensions de seuils V_{TL} et V_{TH} , noté ΔV_T , est calculé pour chaque mesure et fait également l'objet d'une étude statistique. Pour chacune de ces tensions, l'écart-type calculé est assez important (plusieurs dizaines de millivolts) et implique des disparités importantes de comportements dans différents circuits fabriqués dans cette technologie.

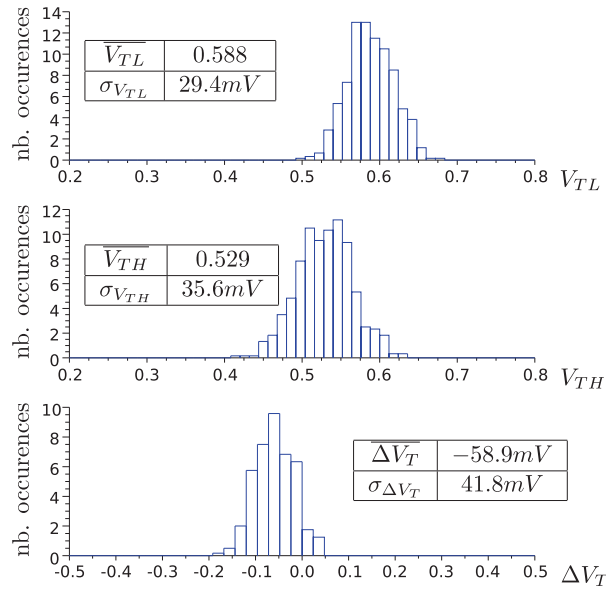


FIGURE 6.1 – Distribution des tensions de seuils

Le support* de la distribution ΔV_T s'étend de $-200mV$ à $50mV$: pour certains circuits fabriqués, le comportement de la porte sera correct (lorsque $\Delta V_T > 0$), dans d'autres cas ($\Delta V_T < 0$) il ne le sera pas. Afin de garantir le bon fonctionnement des circuits, il faut dès la conception s'assurer que le ΔV_T sera toujours positif [OFMA11].

6.2.1.1 Indépendance des seuils

Les différentes valeurs mesurées pour les deux tensions de seuil sont liées aux variations des procédés de fabrication. Ces variations peuvent avoir des effets simultanés sur les deux tensions de seuils. Par exemple, une augmentation de la force des transistors PMOS par rapport aux transistors NMOS peut avoir des effets sur les deux seuils :

- les transistors PMOS de l'étage de maintien sont renforcés par rapport aux transistors NMOS de l'étage d'entrée : le basculement du nœud interne Y depuis le niveau logique 1 vers le niveau logique 0 sera rendue plus difficile, la tension de seuil V_{TH} sera augmentée ;
- de même, les transistors PMOS de l'étage d'entrée sont renforcés par rapport aux transistors NMOS de l'étage de maintien : le basculement du nœud interne Y depuis le niveau logique 0 vers le niveau logique 1 sera plus facile, la tension de seuil V_{TL} sera elle aussi augmentée.

Ces relations impliquent donc une corrélation entre les deux tensions de seuils mesurées. La figure 6.2 trace les valeurs mesurées de V_{TH} en fonction de V_{TL} pour une structure conventionnelle à deux entrées (CM2) en technologie 45nm LS (*Low-power Standard- V_T* pour une tension d'alimentation $V_{DD} = 1.1V$). Ce nuage de point est assez diffus, mais une régression linéaire entre ces deux grandeurs montre une certaine corrélation.

La droite tracée dans cette figure a pour expression $V_{TH} = 0.223 * V_{TL} + 0.398$ et le coefficient de régression (noté r) évaluant la corrélation des deux grandeurs est $r = 0.184$. Ce coefficient est toujours compris dans l'intervalle $[0 \dots 1]$. Une valeur proche de 0 indique l'indépendance des deux grandeurs alors qu'une valeur proche de 1 une très forte corrélation. Dans ce cas, les valeurs sont faiblement corrélées.

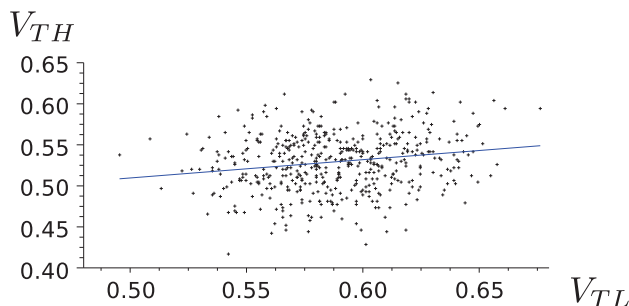


FIGURE 6.2 – Corrélation des tensions de seuils

6.2.2 Marges de conception

Afin de garantir le bon fonctionnement des portes dans une grande majorité lors de la fabrication, il est nécessaire de définir des marges lors de la conception des portes afin de garantir un fonctionnement correct de tous les circuits fabriqués. Les distributions considérées étant assimilables à des gaussiennes, la règle des 3σ permet de regrouper plus de 99.7% des occurrences d'une variable x dans l'intervalle $[\bar{x} - 3\sigma_x \dots \bar{x} + 3\sigma_x]$.

D'autres règles telles que la règle des 5σ permettraient d'augmenter la fiabilité du circuit aux dépens de contraintes encore plus élevées sur les tensions de basculement. La règle des 3σ est un compromis arbitraire garantissant que les occurrences en dehors de cet intervalle représentent moins de 1% des cas obtenus par simulations.

6.2.2.1 Marges pour des variables aléatoires indépendantes

La condition de fonctionnement correct $V_{TH} > V_{TL}$ peut être traduite sous la forme : tous les V_{TH} possibles doivent toujours être supérieurs à tous les V_{TL} possibles (équation 6.1).

$$\overline{V_{TL}} + 3\sigma_{V_{TL}} < \overline{V_{TH}} - 3\sigma_{V_{TH}} \quad (6.1)$$

Cependant la corrélation entre les tensions de seuil implique une très faible probabilité d'obtenir simultanément un fort V_{TL} (proche de la borne $\overline{V_{TL}} + 3\sigma_{V_{TL}}$) et un faible V_{TH} (proche de la borne $\overline{V_{TH}} - 3\sigma_{V_{TH}}$). Il est donc préférable de tenir compte de cette relation entre les deux variables aléatoires pour déterminer les marges de sécurité du circuit.

Une règle de conception basée sur l'hypothèse d'indépendance des deux variables aléatoires V_{TL} et V_{TH} seraient sous optimale : il faudrait lors de la conception prendre en

compte des cas très peu probables qui nécessitent une augmentation superflue de l'écart entre ces deux tensions de seuils. D'après les technique de correction vue dans la [section 5.4 \(page 74\)](#), cela nécessiterait un facteur de redimensionnement k plus important que celui qui est réellement requis.

6.2.2.2 Prise en compte des corrélations

Le calcul de la grandeur ΔV_T pour chaque itération de la *simulation Monte-Carlo** permet d'évaluer de la distribution de cet écart. Cette grandeur est elle aussi une variable aléatoire dont la distribution caractérise directement le comportement du circuit : le C-element a un comportement correct lorsque cet écart est positif, sinon il ne l'est pas. Ce calcul respecte les éventuelles corrélations entre les tensions de seuil et permet de corriger au plus juste le comportement du C-element. Dans la suite de ce chapitre, le comportement du circuit sera analysé en fonction de la grandeur ΔV_T .

6.3 Résultats obtenus

Les méthodes de corrections vues dans le chapitre précédent ont été appliquées en prenant en compte les variations de process.

6.3.1 Redimensionnement de l'étage de maintien

La [figure 6.3](#) trace les variations de l'écart de tensions de seuil obtenues dans une structure conventionnelle à deux entrées (CM2) en technologie ST 45nm LS pour une tension d'alimentation $V_{DD} = 1.1V$. La relation de bon fonctionnement $\Delta V_T > 0$ implique que les bornes de l'intervalle de 3σ soient toutes les deux positives pour que le comportement des portes soit considéré comme correct. Cette contrainte est satisfaite dès que $k > 4.25$, le comportement de la porte C-element est alors correct (zone ②) quelles que soient les variations de fabrication.

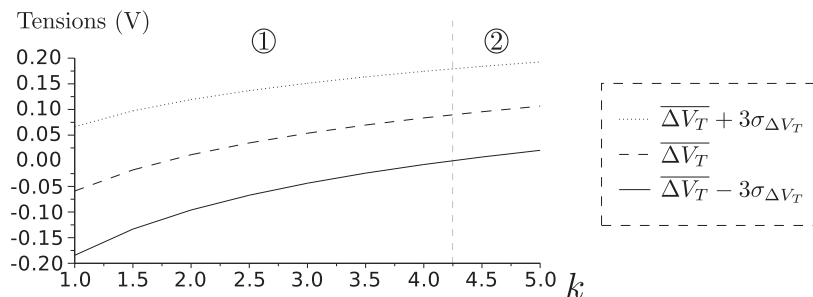


FIGURE 6.3 – Redimensionnement dans une structure CM2

A cette limite, la relation $\Delta V_T > \overline{\Delta V_T} - 3\sigma_{\Delta V_T}$ est vraie pour plus de 99.8% des cas d'après le modèle gaussien.

6.3.2 Réduction de la tension d'alimentation

Lorsque le redimensionnement de l'étage de maintien est couplé à une réduction de la tension d'alimentation, les effets des deux corrections ne se cumulent pas indépendamment. La section 5.4.3 (page 82) reflète le comportement moyen des cellules lors de ces changements simultanés. Les figures 6.4(a) et 6.4(b) sont le tracé de la borne basse de l'intervalle des 3σ (d'expression $\overline{\Delta V_T} - 3\sigma_{\Delta V_T}$) mesurée en fonction de la tension d'alimentation V_{DD} et du facteur de redimensionnement k . Lorsque cette borne est positive, le comportement de la porte est correct (zone ②), sinon il est incorrect (zone ①).

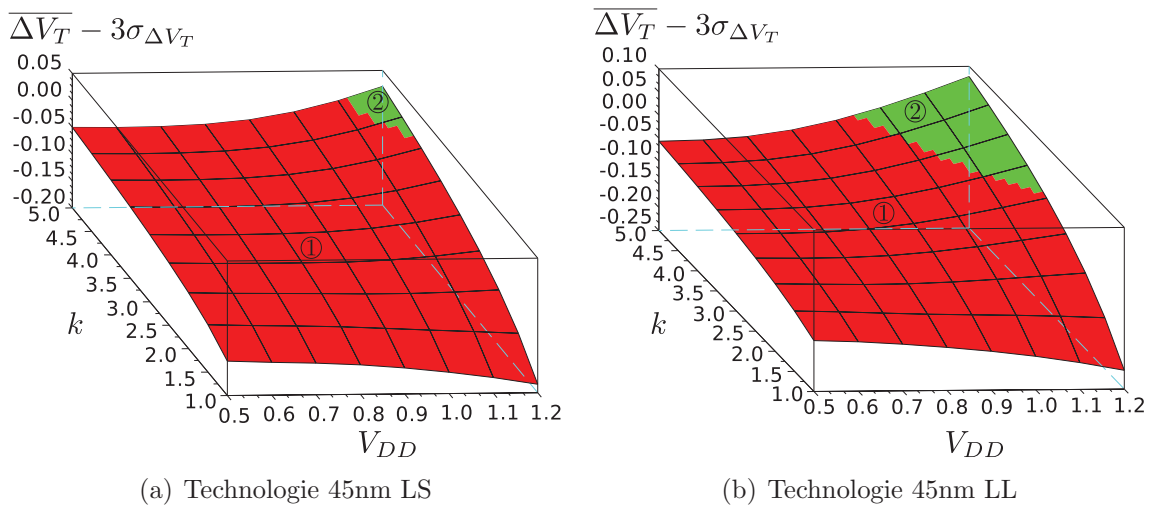


FIGURE 6.4 – Différences des seuils dans une structure CM2

Les figures 6.5(a) et 6.5(b) sont les projections des surfaces des figures 6.4(a) et 6.4(b) sur le plan (V_{DD}, k) . Les technologies où les transistors ont de faibles tensions de seuils (45nm LL) ont un comportement plus favorable au phénomène d'inversion de seuils. La zone obtenue où le comportement est correct est néanmoins plus étendue : il est nécessaire donc moins nécessaire de réduire la tension d'alimentation ou de redimensionner les transistors pour corriger le comportement des portes.

La position de cette zone, placée autour d'un facteur de redimensionnement élevé et d'une tension d'alimentation proche de la tension nominale, permet de simplifier l'étude en considérant que la réduction de la tension d'alimentation n'apporte pas de bénéfices supplémentaires en terme de robustesse de fonctionnement. Seule la technique de redimensionnement de l'étage de maintien permet de corriger le comportement des C-elements de structures *Weak-Feedback* et conventionnelles.

6.3.3 Autres phénomènes influençant les tensions de seuils

Les sections précédentes décrivent le comportement des portes en présence de variations de process de fabrication et de leur tension d'alimentation. La température

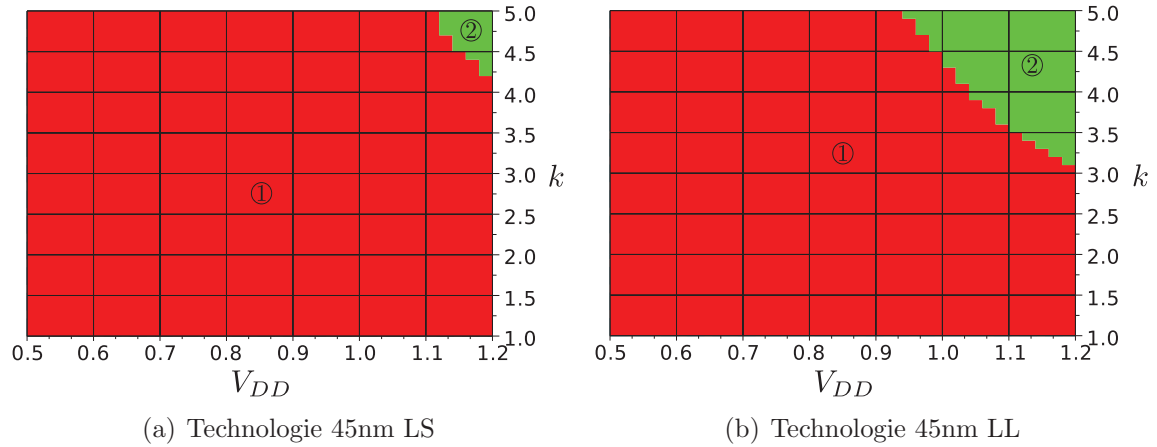


FIGURE 6.5 – Zones de comportements corrects et incorrects d'une structure CM2

est également un des paramètres déterminant du comportement du circuit dans la dénomination courante *Process-Voltage-Temperature**.

Une augmentation de la température physique d'un circuit intégré modifie ses caractéristiques électriques. Généralement, les blocs logiques voient leur vitesse de traitement légèrement diminuer [Tie95]. Dans les circuits asynchrones où les traitements sont auto-cadencés en fonction des caractéristiques de ces blocs logiques, cela peut aboutir à un emballement thermique : une augmentation de température diminue les temps de cycle des blocs logiques, l'activité de ceux-ci est alors augmentée, ce qui consomme plus de puissance, cette puissance augmentera encore la température du circuit...

La figure 6.7 représente les variations en fonction de la température de ΔV_T constatées dans une structure CM2 en technologie ST 45nm LL. Des simulations ont été effectuées de $27^\circ C$ à $107^\circ C$ sur une version dont le comportement est correct d'après la figure 6.5(b) : $k = 3.75$ et $V_{DD} = 1.1V$. L'écart des tension de seuil ΔV_T tend à augmenter lorsque la température augmente : l'écart-type de la distribution reste constant, les valeurs mesurées sont décalées linéairement au fur et à mesure de l'augmentation de la température. Un design initialement correct restera donc correct lorsque la puce fonctionne à plus haute température.

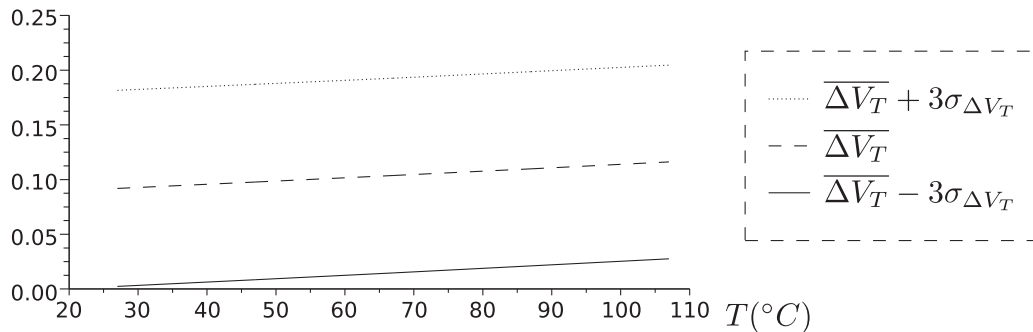


FIGURE 6.6 – Influence de la température sur une CM2 en ST 45nm LL

Cette évolution en fonction de la température n'est cependant pas uniforme, pour le dimensionnement initial de la porte ($k = 1$ et $V_{DD} = 1.1V$), les variations des seuils sont opposées (figure 6.7) : l'écart entre les tensions de seuils tend à diminuer lorsque la température augmente. Cette variation tend alors à augmenter l'inversion de seuil.

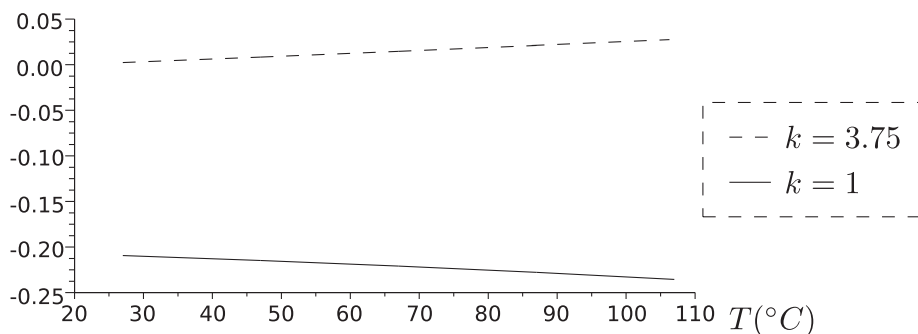


FIGURE 6.7 – Influence de la température sur deux dimensionnements de CM2

Dans la technologie ST 45nm LL, l'augmentation de la température tend à amplifier le courant de saturation des transistors dont la largeur élevée. Dans le dimensionnement initial de la porte ($k = 1$), les largeurs des transistors de l'étage d'entrée sont supérieures à celles des transistors de l'étage de maintien. L'étage d'entrée sera donc renforcé lorsque la température augmente, ce qui renforce le phénomène d'inversion des seuils.

La relation d'ordre entre les largeurs des transistors étant inverse dans le dimensionnement au comportement correct, l'étage de maintien est donc renforcé par rapport à l'étage d'entrée et le basculement du point mémoire nécessite donc une séparation des tensions de seuil plus importante.

Conclusion

La prise en compte des variations des procédés de fabrication et de température de fonctionnement lors de la conception des portes logiques permet de s'assurer de leur comportement correct dans la quasi-totalité des circuits fabriqués et dans la plupart des cas de fonctionnement.

Les comportements incorrects des structures *Weak-Feedback* et conventionnelles sont corrigés à l'aide de deux techniques vues au chapitre précédent. Lorsque les variations des procédés de fabrication sont prises en compte, il est nécessaire d'augmenter le facteur de redimensionnement des transistors de l'étage de maintien afin d'obtenir le comportement correct de la structure *Weak-Feedback*. Dans la structure conventionnelle, l'ajout de marges de sécurité réduit la zone de comportement correct. Mais cette zone étant située autour de la tension d'alimentation nominale et d'un facteur de redimensionnement important, la technique de réduction de la tension d'alimentation n'a finalement que peu d'intérêt pour corriger le comportement des structures *Weak-Feedback* et conventionnelles.

Les caractérisations des portes présentées dans ce chapitre sous la forme de graphiques nécessitent de nombreuses simulations pour obtenir une bonne précision : 500 mesures ont été effectuées pour chaque couple de tension d'alimentation et de facteur de redimensionnement. Pour vérifier le comportement d'une porte existante, toutes les mesures paramétriques en fonction du paramètre de redimensionnement et de la tension d'alimentation ne sont pas nécessaires si la conception initiale est bonne. Cette conformité est déterminée à l'aide d'une seule *simulation Monte-Carlo** nécessitant quelques minutes.

La correction d'une porte à structure *Weak-Feedback* ou conventionnelle se fait donc en redimensionnant les transistors de son étage de maintien. La borne basse de ce redimensionnement peut alors être déterminée rapidement par dichotomie.

Formalisation de la vérification

| | |
|-------------------------------------------------------------|------------|
| Introduction | 100 |
| 7.1 Simulations numériques de circuits électroniques | 100 |
| 7.2 Vérifications des circuits électroniques | 101 |
| 7.2.1 Vérifications semi-formelles | 102 |
| 7.2.2 Vérifications formelles | 102 |
| 7.2.3 Limites des vérifications formelles analogiques | 102 |
| 7.3 Coho | 103 |
| 7.3.1 Principes des simulations avec intervalles | 104 |
| 7.3.2 Structure du programme | 106 |
| 7.3.3 Spécifications du circuit | 106 |
| 7.3.4 Spécifications des entrées | 107 |
| 7.3.5 Graphe de simulation | 108 |
| 7.4 Application aux circuits asynchrones | 109 |
| 7.4.1 Structure <i>Weak-Feedback</i> | 110 |
| 7.4.1.1 Analyse des <i>Brockett Annuli</i> | 110 |
| 7.4.1.2 Analyse des seuils de basculements | 112 |
| 7.4.2 Structure Conventiionnelle | 115 |
| 7.4.2.1 Analyse de <i>Brockett Annuli</i> | 115 |
| 7.4.2.2 Analyse des seuils de commutation | 116 |
| Conclusion | 118 |

Introduction

Afin d'avoir une plus grande couverture des cas possibles d'utilisation des portes logiques utilisées dans les circuits asynchrones, il est nécessaire de compléter les simulations par une méthode de vérification plus formelle. Ces méthodes permettent de vérifier le bon comportement des portes C-element sur l'ensemble des traces possibles de leurs entrées. Cette systématisation de la vérification permet de détecter des dysfonctionnements qui pourraient ne pas être détectés par les vérifications ponctuelles sur les seuils de commutation par exemple.

Ces outils semi-formels utilisent les mêmes principes de simulation que *SPICE*, mais sont capables de générer un ensemble de traces plutôt qu'une seule à chaque simulation.

7.1 Simulations numériques de circuits électroniques

Les simulations analogiques classiques, telles que présentées dans les [chapitres 5 et 6](#) sont fondées sur un moteur de simulation de type *SPICE**. La simulation consiste en la résolution numérique de systèmes d'équations mathématiques linéaires approximant le comportement réel du circuit [[Fod88](#)]. Ces solutions numériques ne sont pas des valeurs exactes, mais elles permettent de représenter avec une bonne précision le fonctionnement du circuit réel.

Pour être simulé, le circuit est représenté comme un réseau de nœuds interconnectés : pour chaque nœud, le simulateur va déterminer son potentiel en fonction des composants les reliant aux nœuds adjacents.

La simulation numérique ne s'effectue pas en temps continu, mais seulement à quelques instants déterminés par calcul. Pour chaque instant temporel t_i où la résolution numérique a lieu, un modèle linéaire équivalent du circuit est déterminé à partir des modèles des composants et des interconnexions. La résolution de ce système linéaire donne les variations des différents potentiels (leurs dérivées calculées pour cet instant temporel t_i). Ces variations, associées à un intervalle de validité de l'approximation permettent l'interpolation linéaire du signal jusqu'au prochain instant temporel de simulation t_{i+1} . La valeur de ce pas temporel entre les instants t_i et t_{i+1} est également déterminée lors de la résolution du système linéaire représentant le circuit à l'instant t_i .

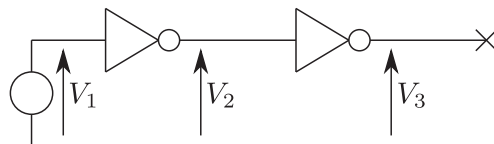


FIGURE 7.1 – Chaîne d'inverseurs

Pour le circuit électrique représenté dans la [figure 7.1](#), l'état du système est formé par le n -uplet* de ses variables. Chaque variable est une valeur scalaire représentant par exemple le potentiel d'un conducteur par rapport à la masse. Dans ce circuit, l'état

est le $\mathcal{3}$ -uplet $\langle V_1, V_2, V_3 \rangle$ où V_1 , V_2 et V_3 sont les potentiels du circuit. Il est possible de tracer lors de la simulation l'évolution des valeurs numériques de ces n -uplets de valeurs scalaires dans l'espace. Par exemple, la figure 7.2(a) présente les résultats obtenus lorsque la tension de sortie du deuxième inverseur est tracée en fonction de sa tension d'entrée. Le tracé suivi pendant un front montant et un front descendant de l'entrée sont différents, cet hystérésis est causé par le délai de l'inverseur. Ce tracé peut être étendu à trois dimensions (figure 7.2(b)) représentant les trois composants du $\mathcal{3}$ -uplet $\langle V_1, V_2, V_3 \rangle$.

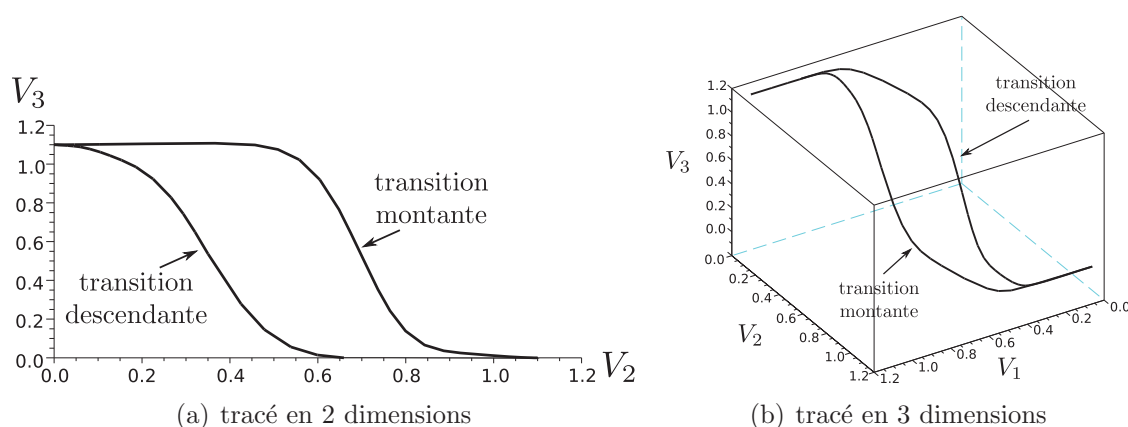


FIGURE 7.2 – Simulation dans des espaces de différentes dimensions (SPICE)

Chaque simulation "à la *SPICE*" ne prend en compte qu'une seule combinaison des entrées. Pour prendre en compte plusieurs combinaisons d'entrées, plusieurs décalages dans les instants de transitions par exemple, il est nécessaire de lancer de nombreuses simulations en faisant varier le paramètre étudié. Bien que chaque simulation individuelle soit généralement très rapide, leurs répétitions induisent des temps de simulation assez conséquents (plusieurs heures). La discrétisation d'un tel balayage des paramètres de simulation ne permet pas de garantir la fiabilité des vérifications car il est toujours possible de ne pas avoir pris en compte une valeur particulière d'un paramètre amenant un comportement différent du système.

7.2 Vérifications des circuits électroniques

Afin de garantir le fonctionnement du circuit en prenant en compte toutes les valeurs particulières possibles de ses paramètres, il est nécessaire d'utiliser des méthodes de vérification. Dans le domaine de la vérification du fonctionnement des circuits électroniques, il existe deux stratégies autorisant différents degrés de couverture des cas possibles d'utilisation.

7.2.1 Vérifications semi-formelles

Les vérifications semi-formelles sont fondées sur l'analyse de traces de fonctionnement obtenues lors de simulations à divers niveaux d'abstraction ou lors du fonctionnement du circuit réel. Diverses propriétés caractérisant le comportement correct du circuit peuvent alors être mathématiquement prouvées sur ces mesures. Ces propriétés permettent de vérifier le comportement du circuit d'après ce nombre fini de traces de fonctionnement, elles ne permettent pas d'extrapoler cette correction pour les autres cas non testés. La couverture des méthodes semi-formelles est donc incomplète et il est nécessaire de générer un grand nombre de jeux de tests afin de couvrir au mieux les cas réels possibles.

7.2.2 Vérifications formelles

La vérification formelle d'un circuit consiste en la vérification de l'adéquation du circuit avec un modèle. On parle généralement de vérification d'équivalence avec un modèle dont le comportement est correct. Cette référence est généralement dénommée *golden model**.

La vérification formelle de circuits électroniques numériques est un domaine très actif dans la recherche universitaire et industrielle depuis de nombreuses années. De nombreux outils et de nombreuses méthodes ont été mises en œuvre pour s'assurer du comportement correct de circuits de tailles conséquentes. Les vérifications sont généralement assurées automatiquement par des démonstrateurs de théorèmes ou des outils dédiés à la vérification de modèles (*Model Checking**).

La vérification formelle de circuits électroniques analogiques est un domaine assez peu répandu. Les techniques employées se limitent pour le moment à l'étude de petits circuits formés de quelques nœuds (généralement moins de 10) et de quelques composants (généralement moins de 10 également). Leur fonctionnement est vérifié d'après des modèles de composants approchés permettant de déterminer l'espace atteignable du circuit. Cet espace atteignable représente l'ensemble des valeurs possibles par l'état de ce circuit. Il est ainsi possible de vérifier l'inclusion d'un comportement souhaité dans cet espace atteignable ou l'exclusion d'un comportement rejeté.

7.2.3 Limites des vérifications formelles analogiques

Pour vérifier le comportement de circuits analogiques, il est nécessaire d'obtenir un modèle de leur fonctionnement et de vérifier son adéquation avec le comportement souhaité. Les circuits asynchrones ont des comportements analogiques fortement non-linéaires car les transistors composant les différentes portes logiques passent dans toutes les zones de fonctionnement : zone bloquée, ohmique ou saturée pour les transistors MOS. Les modèles des transistors sont différents dans chacune de ces zones de fonctionnement et ils sont tous fortement non-linéaires (ils comportent des expressions quadratiques ou exponentielles par exemple).

De surcroît, ces modèles de fonctionnement ne sont que des approximations du comportement physique du transistor et devront toujours être considérés comme tels. Ils représentent fidèlement les relations entre les courants et les tensions des différentes électrodes mais ils ne prendront pas en compte les éventuels couplages liés au bruit de fonctionnement du reste du circuit par exemple. Il est possible d'améliorer la fiabilité du modèle en incorporant certains phénomènes physiques mais la couverture de tous ces phénomènes physiques ne sera toujours que partielle et très dépendante de la technologie visée.

D'une manière générale, la vérification d'un circuit analogique est limitée à la vérification de son modèle vis à vis de son modèle de référence. La fidélité et la précision de ceux-ci sont prédominants pour l'obtention de résultats fiables et représentatifs du circuit réel. Si un phénomène analogique du circuit réel n'est pas pris en compte dans son modèle, la vérification peut potentiellement conclure sur des résultats faux.

Plusieurs outils ont déjà été développés dans des projets de recherche académique. La plupart d'entre eux utilisent une représentation mathématique du circuit que l'utilisateur doit écrire. Par exemple, l'outil d/dt [ADM02] résout des équations différentielles du premier ordre (équation 7.1) dans lesquelles x représente les variables d'état du circuit et le facteur du second membre u est un espace atteignable.

$$Ax + B \frac{dx}{dt} = Cu \quad (7.1)$$

PHAVer [Fre05] est un autre outil utilisant le principe *Assume-Guarantee* dans lequel des propriétés additionnelles (hypothèses de départ - *Assume* - sous la forme d'équations ou d'inéquations mathématiques) permettent de simplifier le modèle par le biais de déductions *Guarantee*.

Ces logiciels de vérifications ne sont pas dédiés à la vérifications de circuits électroniques et ne proposent pas de modèles directement utilisables pour les transistors par exemple. L'utilisateur doit déterminer le modèle du circuit avant de le vérifier.

7.3 Coho

Coho est un outil de simulation dédié aux circuits électroniques réalisé par CHAO YAN pendant son *Master Degree* [Yan06] et pendant sa thèse sous la direction du Professeur MARK GREENSTREET au sein du *Department of Computer Sciences* de l'*University of British Columbia* à Vancouver (Canada). Depuis notre collaboration, Coho est disponible en *open-source* [Yan].

Ce simulateur a permis de vérifier le comportement de primitives de circuits synchrones telles qu'une bascule [YG07] ou un arbitre [YG08b]. Son principe de fonctionnement n'étant pas limité aux composants avec horloge, il peut être utilisé pour vérifier le comportement des primitives asynchrones.

Coho permet la simulation de l'espace atteignable des sorties d'un circuit électronique

en fonction de ses entrées. Les résultats de simulations obtenus avec Coho ne sont pas limités à quelques traces des sorties du circuit (de tels résultats ont été précédemment obtenus dans les [chapitres 5 et 6](#) dont les méthodes peuvent s'apparenter à des vérifications semi-formelles de fonctionnement). Les résultats obtenus à l'aide de Coho représentent l'ensemble des traces possibles sur les sorties et peuvent servir à une vérification formelle du comportement du circuit simulé.

7.3.1 Principes des simulations avec intervalles

Un n -uplet* de valeurs scalaires ne représente qu'un point de fonctionnement. Afin de représenter efficacement l'ensemble des états possibles et l'ensemble des variations possibles, il est nécessaire d'utiliser une autre représentation de l'état du système.

Contrairement aux simulations conventionnelles décrites dans la [section 7.1](#), Coho réalise une simulation d'un système à n dimensions où chaque variable x_i du système ($i \in [1..n]$) est décrite comme appartenant à un intervalle, défini principalement par une valeur minimale $x_{min,i}$ et une valeur maximale $x_{max,i}$, ces deux valeurs étant des nombres scalaires.

A chaque instant de simulation t_i , l'état du système est donc représenté par un volume à n dimensions où chaque dimension correspond à l'intervalle de variation d'une variable du système. L'état du système est toujours inclus dans le [paralléloptope droit](#)* de coordonnées $[X_{min,1} \dots X_{max,1}] \dots [X_{min,n} \dots X_{max,n}]$. Par exemple, la [figure 7.3\(a\)](#) représente un paralléloptope droit dans un espace à deux dimensions : c'est un rectangle dont les arêtes sont alignées sur les axes. La [figure 7.3\(b\)](#) représente un paralléloptope dans un espace à trois dimensions.

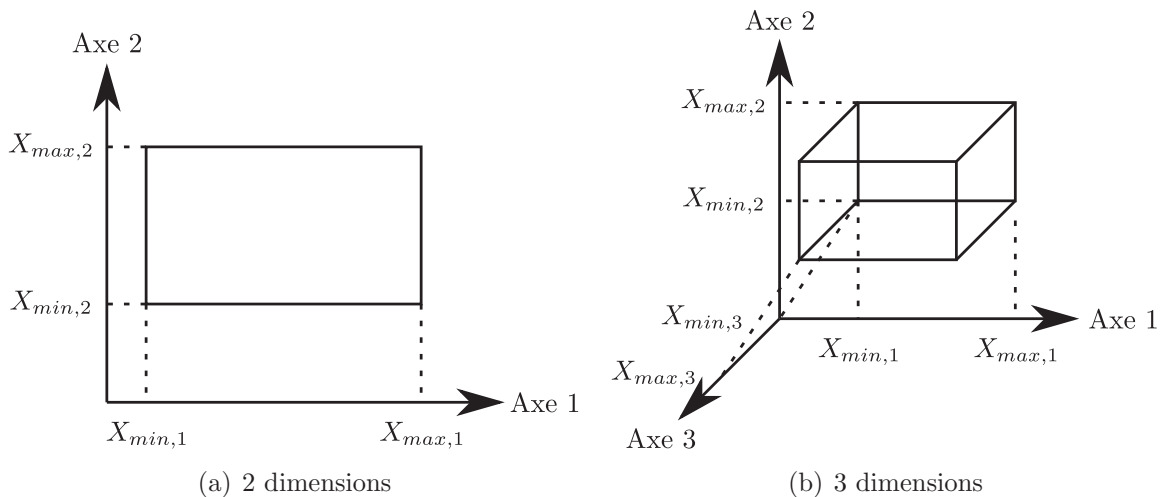


FIGURE 7.3 – Exemples de parallélotopes

Dans l'espace à n dimensions correspondant au circuit, ce paralléloptope est toujours une sur-approximation de l'espace effectivement atteignable par le circuit. Cette sur-approximation permet de garantir l'absence de cas non pris en compte lors des simulations

et ainsi d'avoir une couverture complète des intervalles simulés.

A l'opposé, la présence de faux-positifs est néanmoins possible, il est nécessaire de vérifier les résultats de simulations afin de comprendre les causes d'un possible dysfonctionnement du circuit. L'outil ne fournit pas de trace simple en cas de dysfonctionnement du système, c'est au concepteur d'analyser l'erreur en essayant d'éventuellement déterminer la surapproximation responsable.

Une sur-approximation trop importante peut également provoquer des divergences dans les simulations : les intervalles des différentes variables tendent alors vers toutes les valeurs possibles, ils sont alors quasi-équivalents à l'intervalle $[V_{SS} \dots V_{DD}]$ bornant l'évolution des tensions dans le circuit.

Dans le but de réduire les sur-approximations, *Coho* permet de réduire le volume de l'espace atteignable en le contraignant par projections sur un plan défini par un couple d'axes [YG08a]. La figure 7.4 représente le principe des projections : le volume de l'espace atteignable (représenté en grisé) est formé par l'intersection dans l'espace des différentes contraintes sur les différents plans de projection (représentées comme des polygones sur chacun des plans).

L'ajout de plans de projection n'est pas obligatoire : en cas d'absence de projection sur un plan défini par un couple d'axes correspondant aux variables i et j ($i, j \in [1 \dots n]^2$), le programme supposera que la projection implicite est le rectangle de coordonnées $[x_{min,i} \dots x_{max,i}][x_{min,j} \dots x_{max,j}]$. Dans ce cas, les variables seront considérées comme indépendantes : leurs évolutions ne sont pas liées.

Le choix de ces plans de projection dépend de la structure du circuit : lorsque les potentiels de deux signaux sont liés, l'ajout d'un plan de projection réduit l'indépendance des deux variables correspondant aux deux axes dans les résultats de la simulation mais augmente le temps de simulation. De part la structure de *Coho*, le polygone projeté sur ce plan doit toujours être convexe.

Cette représentation simplifiée permet un bon compromis entre la durée des calculs, la précision et la compacité des résultats. Les calculs de géométrie dans l'espace (de dimension n) étant particulièrement gourmands en temps de calcul, la simplification des transformations dans des plans de projections réduit énormément la complexité des problèmes.

Le principe de la simulation numérique est assez similaire à une simulation *SPICE** à la différence que chaque variable est définie par son intervalle de variations (et ses interactions avec les autres variables sont limitées par d'éventuelles projections). Le logiciel établit pour chaque pas temporel de simulation un modèle linéaire approché du circuit et résout ce système linéaire d'après l'état atteignable.

Lors de la simulation temporelle, les coordonnées des angles des projections vont être transformées par les variations calculées par la résolution du système linéaire (voire la figure 2.2 page 16 dans [Yan06] pour le synoptique complet d'un pas temporel de simulation).

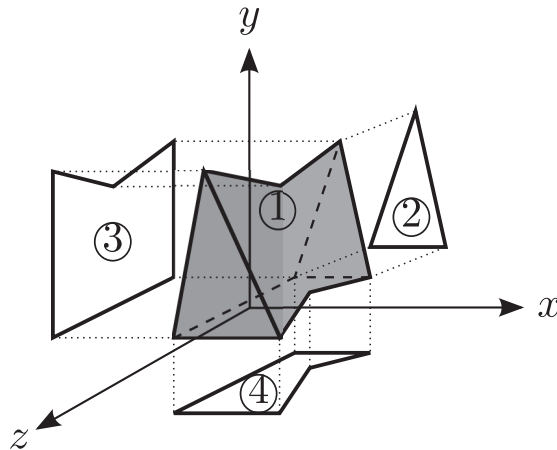


FIGURE 7.4 – Projections dans d'un volume 3D sur des plans 2D

7.3.2 Structure du programme

Coho se présente sous la forme d'un script Matlab s'interfaçant avec un programme écrit en Java par le biais d'une couche de communication écrite en C (figure 7.5).

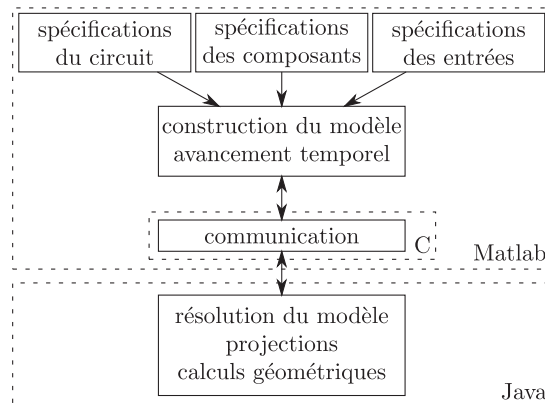


FIGURE 7.5 – Structure de Coho

L'utilisateur doit fournir les spécifications de son circuit, les modèles des composants de base tels que les transistors et les spécifications des entrées du système sous la forme de scripts Matlab. Le programme réalise alors automatiquement la simulation en transmettant au programme Java fonctionnant en parallèle les systèmes à résoudre. Pour chaque pas temporel de simulation, les différents polygones de projections sont sauves dans un fichier au format Matlab (.mat) et peuvent être retraités *a posteriori*.

7.3.3 Spécifications du circuit

Le circuit doit être spécifié sous la forme d'une *netlist** où les composants sont connectés à des nœuds préalablement alloués. Dans l'exemple du C-element de structure *Weak-Feedback* à deux entrées (WM2 - figure 3.1(a) - page 35) dont la *netlist**

est reportée en [annexe B.1](#), les nœuds sont alloués en lignes 30 à 35. Le nœud noté n correspond au contact entre les deux transistors N de l'étage d'entrée, le nœud p au contact entre les deux P. Les composants transistors sont ensuite créés en paramétrant leurs longueurs et largeurs respectives en lignes 37 à 49.

Finalement, les capacités des différents nœuds sont spécifiées en lignes 51 à 54. Par défaut les modèles des transistors N et P dans Coho surestiment leurs capacités de drain et de source, il est nécessaire de réduire leurs valeurs pour obtenir des résultats cohérents avec les simulations classiques.

7.3.4 Spécifications des entrées

Comme tous les nœuds du circuit, les entrées du système sont spécifiées par intervalle. Les transitions des signaux numériques sont découpées en différentes zones ([figure 7.6\(a\)](#)) :

- tant que la tension d'entrée est comprise dans l'intervalle $[V_{SS}, V_L]$, le potentiel peut éventuellement varier (augmenter et diminuer), mais c'est un niveau logique 0 compatible avec les sorties des portes de la technologie (zone ①) ;
- pendant la transition montante (zone ②), la variation est supposée monotone (potentiel strictement croissant) ;
- dans l'intervalle $[V_H, V_{DD}]$, c'est un niveau logique 1 (zone ③) ;
- pendant la transition descendante (zone ④), le potentiel de l'entrée est strictement décroissant.

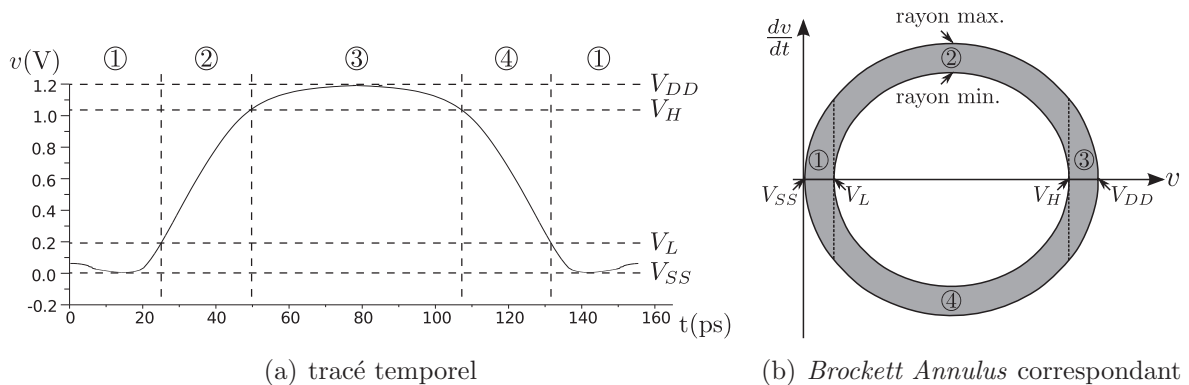


FIGURE 7.6 – Spécifications du *Brockett Annulus*

Ce prototype de ces transitions est représenté sous la forme d'un *Brockett Annulus** équivalent représenté en [figure 7.6\(b\)](#). L'utilisation d'un *Brockett Annulus* permet de spécifier des contraintes de tensions et de variations de tensions pendant les transitions sous la forme d'un anneau : c'est un tracé de la variation de la tension d'entrée dV_x/dt en fonction de cette tension d'entrée V_x . Dans la [figure 7.6\(b\)](#), les zones ① à ④ représentent les mêmes étapes que dans la [figure 7.6\(a\)](#).

La face intérieure de cet anneau correspond aux transitions les plus lentes considérées pendant la simulation et sa face extérieure aux transitions les plus rapides. Cet anneau est

défini par les bornes des niveaux logiques 0 (V_{SS} et V_L) et 1 (V_H et V_{DD}) mais également par des bornes concernant les vitesses maximales de variations (rayons minimum et maximum). Des contraintes sur la rapidité de transition peuvent également être spécifiées : elles impliquent des durées minimales et maximales des transitions.

Dans Coho, les zones ② et ④ sont séparées automatiquement en plusieurs sous-zones (cinq dans notre cas) afin de réduire l'intervalle des valeurs possibles sur les entrées et ainsi faciliter la convergence des simulations. L'anneau correspondant est représenté dans la figure 7.3.4.

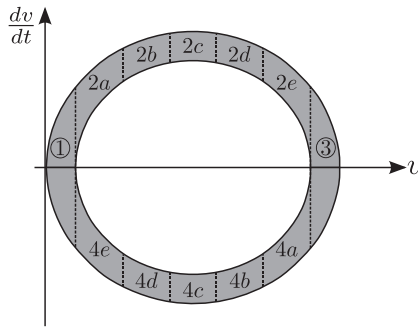


FIGURE 7.7 – Séparations des fronts montants et descendants

Ces anneaux sont spécifiés dans l'annexe B.1 dans les lignes 12 à 20 ($V_{SS} = 0.0V$, $V_L = 0.2V$, $V_H = 1.6V$, $V_{DD} = 1.8V$, $radius_{min} = 1.5e10$, $radius_{max} = 2e10$). Afin d'éviter les passages trop furtifs dans les zones ① et ③, Coho permet de spécifier un temps minimum de maintien ($minT = 1ns$ dans notre cas).

7.3.5 Graphe de simulation

La figure 7.8 représente l'ensemble des couples de zones à simuler pour un C-element pour prendre en compte les treize zones des transitions montantes et descendantes de ses deux entrées. Ce graphe est orienté : les zones d'une entrée sont toujours parcourues dans l'ordre 1, 2a, 2b... , 4. Il comporte donc un couple de zones initial (de coordonnées $in_A = 1, in_B = 1$). Chaque couple de zones a jusqu'à deux couples "suivants" et deux couples "précédents". Cet ordre est indiqué par les flèches orientées d'un couple précédent vers le couple suivant.

De part les protocoles de communications et les rendez-vous implémentés entre les blocs émetteurs et récepteurs, il ne peut y avoir une transition montante et une transition descendante simultanées sur les entrées d'un C-element, sauf si le fonctionnement est déjà incorrect : ce principe n'est vrai que si le comportement des portes est correct ($V_{TL} < V_{TH}$ vu dans le chapitre 5). Le *golden model** permet donc de simplifier l'ensemble des simulations à effectuer sur le modèle à vérifier. L'impossibilité d'avoir une transition montante sur une entrée simultanément à une transition descendante sur une autre retire de la simulation tous les couples d'états grisés de la figure 7.8.

Pour chacun des couples de zones et sous-zones de chacune des entrées, Coho calcule

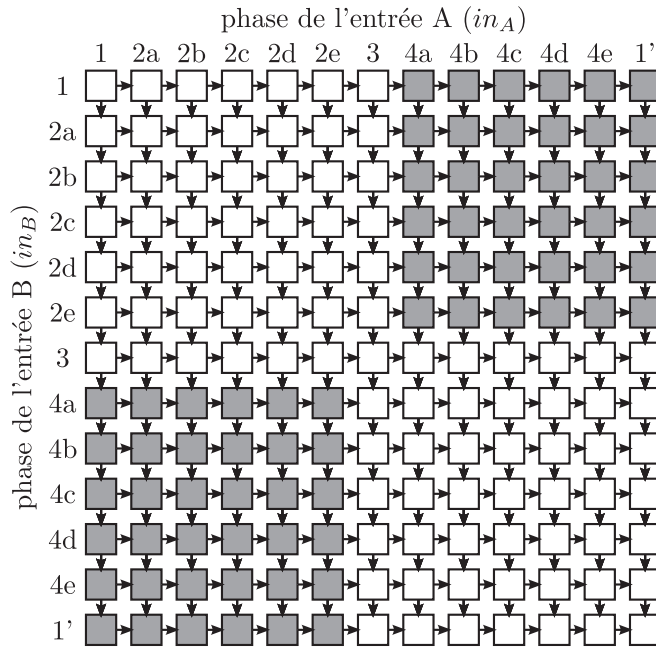


FIGURE 7.8 – Graphe de simulation

l'espace atteignable de chacun des signaux du circuits. Ce calcul itératif modifie un espace atteignable initial en fonction des valeurs possibles des entrées. La valeur initiale de cet espace atteignable est issu soit des spécifications du circuit pour le couple initial, soit pour les couples non-initiaux de l'union des espaces atteignables de leurs couples précédents respectifs. Pour chaque couple de zones, la simulation temporelle s'arrête dès que l'espace atteignable est stable : c'est le cas lorsque son volume (à n -dimensions) à l'instant t_i est inclus dans son volume à l'instant t_{i-1} . Coho permet de spécifier un nombre maximum de pas temporels au cas où ce calcul itératif de l'espace atteignable ne converge pas.

La configuration de ce graphe est indiquée dans l'[annexe B.2](#). Afin d'accélérer les calculs, les simulations sont scindées en deux graphes distincts : un graphe pour les fronts montants regroupant les zones 1 à 3 des deux entrées et un graphe pour les fronts descendants correspondant aux zones 3 à 1'. Cette optimisation permet de lancer les calculs en parallèle.

7.4 Application aux circuits asynchrones

Les structures conventionnelles à deux entrées (CM2) et *Weak-Feedback* à deux entrées (WM2) ont été créées et simulées [[YOFMA11](#)] dans la technologie TSMC 180nm. L'obtention de tout l'espace atteignable par la sortie de la porte de Muller permet de s'assurer de son comportement correct quelques soient ses entrées.

Le *golden model** d'une porte C-element est une porte respectant :

- la table de vérité de la fonction C-element ;
- un ordonnancement correct des seuils de transition : l'absence de transition lors-

- qu'une entrée n'a pas encore franchi $V_{DD}/2$;
- des vitesses de transitions compatibles entre les entrées et les sorties : la porte ne ralentira pas les transitions des données qu'elle synchronise.

Ce modèle de référence prend en compte, *a priori*, les symptômes d'un C-element pouvant amener à un dysfonctionnement du circuit asynchrone.

7.4.1 Structure *Weak-Feedback*

La structure WM2 comporte 6 nœuds (figure 3.1(a) - page 35). Sa *netlist**, comprenant notamment les dimensions des différents transistors, est disponible en annexe B.3). Cette simulation correspond aux transformations d'un volume en six dimensions.

7.4.1.1 Analyse des *Brockett Annuli*

Les figures 7.9 et 7.10 représentent respectivement les transitions descendantes et montantes de la sortie de la porte obtenue lors de cette vérification. Les tracés en gris-clair sont réalisés à partir des résultats de simulation sur le nœud Z : la dérivée de cette tension de sortie est tracée en fonction de sa valeur. Ce graphe de transition peut être comparé au *Brockett Annulus** des stimuli appliqués sur les entrées de la porte. Ce *Brockett Annulus* est formé par les deux ellipses noires de chaque figure.

Au début de chaque transition (pour $V_Z \approx 0V$ dans le front montant et $V_Z \approx 1.8V$ pour le front descendant), l'intervalle des valeurs de la dérivée de la tension de sortie est très important. Il existe des cas où la dérivée peut être négative pendant le front montant et positive pendant le front descendant.

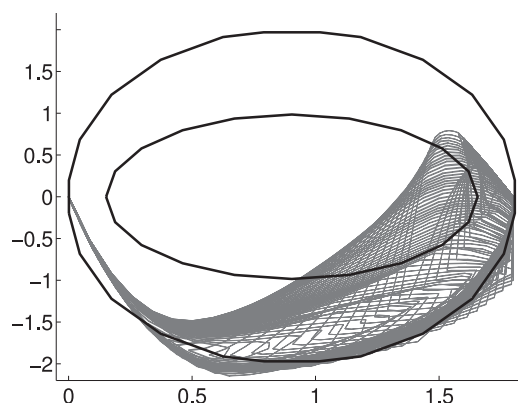
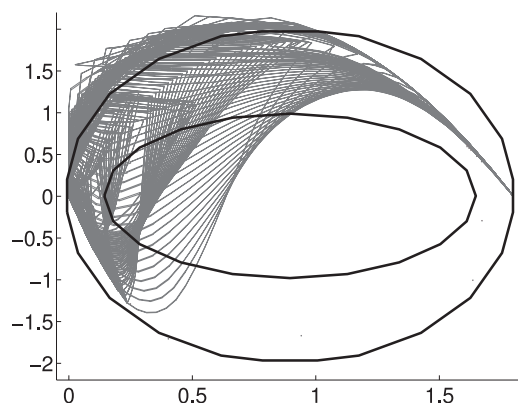


FIGURE 7.9 – *Brockett Annulus* de la sortie d'une WM2 - front descendant

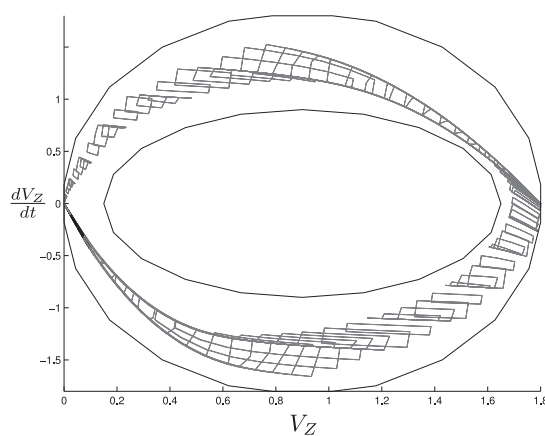
Ces inversions des variations peuvent s'apparenter à des rebonds ou à des métastabilités issues de l'incapacité pour l'étage d'entrée de basculer efficacement et rapidement le point mémoire. Malgré tous les tests mis en œuvre pour reproduire ce problème dans une simulation analogique conventionnelle, il n'a pas été possible de reproduire ce

FIGURE 7.10 – *Brockett Annulus* de la sortie d'une WM2 - front montant

phénomène sur les portes existantes des bibliothèques TAL. Ce comportement est donc très probablement lié aux approximations du modèle.

Cependant, il est important qu'une porte ne produise pas de transitions plus lentes sur sa sortie par rapport aux transitions de ces entrées. Si de telles portes logiques sont utilisées dans une structure de type pipeline, ces ralentissements successifs auraient pour conséquence l'apparition spontanée et reproductible de fronts très lents dans le circuit. Ces fronts pourraient alors aboutir au phénomène de création ou d'absorption de jetons vu au [chapitre 4](#).

Il est également possible de corriger le comportement de cette porte pendant les transitions de sa sortie en connectant un inverseur à sa sortie pour remettre le signal en forme. Les transitions de cette sortie inversée s'inscrivent alors dans le *Brockett Annulus** des tensions des entrées de la porte ([figure 7.11](#)).

FIGURE 7.11 – *Brockett Annulus* de la sortie d'une WM2 suivi d'un inverseur

7.4.1.2 Analyse des seuils de basculements

Afin de vérifier l'absence de basculements précoces dans cette structure *Weak-Feedback*, il est possible d'observer l'état de son point mémoire statique (formé par les deux inverseurs tête-bêche) pendant la première phase du front montant de chaque. L'état du point mémoire correspond aux potentiels des nœuds Y et Z de la structure *Weak-Feedback* (figure 3.1(a) - page 35).

La première phase du front montant correspond aux zones 1, 2a, 2b et 2c du *Brockett Annulus* de l'entrée observée et à tous les potentiels possible pour l'autre entrée (pendant cette observation, le potentiel de l'autre entrée est indéterminé et peut prendre toutes les valeurs de l'intervalle $[V_{SS} \dots V_{DD}]$).

Ainsi pour s'assurer de l'absence de basculements précoces sur son entrée A , il faut observer l'ensemble des valeurs atteignables des potentiels V_Y et V_Z lorsque le potentiel de l'entrée A est compris dans l'intervalle $[V_{SS} \dots V_{2c,max}]$, où $V_{2c,max}$ est la borne haute de la zone 2c ($V_{2c,max} = 1.04V$ dans notre cas), et lorsque le potentiel de l'entrée B est compris dans l'intervalle $[V_{SS} \dots V_{DD}]$. Cet ensemble de valeurs atteignable correspond à l'union des valeurs atteignables dans tous les couples de zones marqués d'une croix dans la figure 7.12(a).

Les potentiels V_Y et V_Z forment un plan de projection contraignant l'espace atteignable du circuit pendant la simulation, l'espace atteignable à chaque pas temporel est donc un polygone convexe délimitant les valeurs atteignables des potentiels V_Y et V_Z . La figure 7.12(b) représente tous les polygones déterminés pour chaque instant temporel de simulation de chaque couple de zones marqués d'une croix dans la figure 7.12(a).

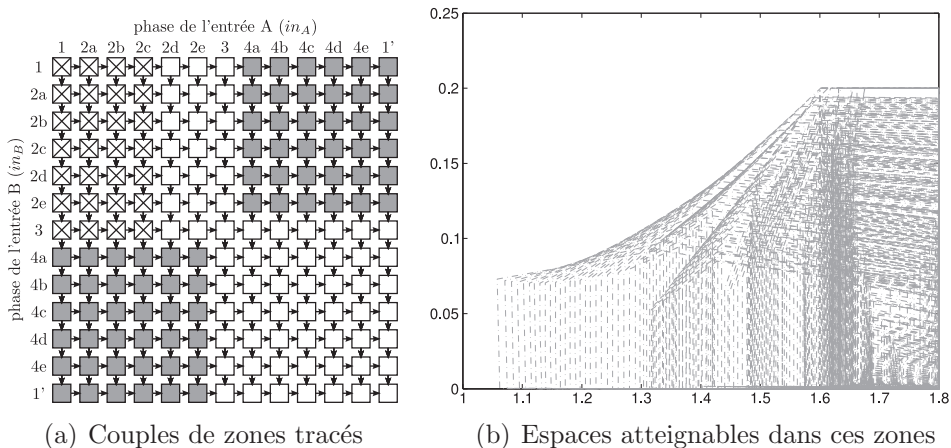


FIGURE 7.12 – Front montant de l'entrée A d'une WM2 (VTHA)

Lorsque tous ces couples de zones observés, le potentiel du nœud Y est toujours supérieur à $1.05V$ et le potentiel du nœud Z est toujours inférieur à $V_L = 0.2V$: le point mémoire a commencé un basculement mais celui-ci ne se termine pas tant que la tension de l'entrée observée est inférieure à $V_{2c,max} = 1.04V$, et ce quelque soit la valeur

de l'autre entrée. Par conséquent, on peut déduire que le seuil de basculement V_{THA} est supérieur à $V_{2c,max}$, quelque soit la valeur de l'autre entrée B .

Les figures 7.13(a) et 7.13(b) représentent respectivement les couples de zones et les observations pour le début du front montant de l'entrée B pendant que l'entrée A peut prendre toutes les valeurs possibles. Cette observation est le pendant de l'observation précédente pour les transitions montantes sur l'entrée B .

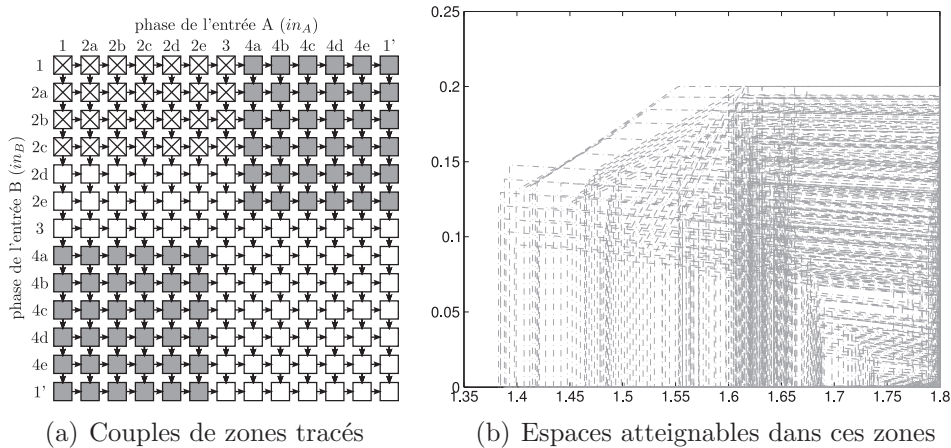


FIGURE 7.13 – Front montant de l'entrée B d'une WM2 (VTHB)

Comme précédemment, le potentiel V_Y ne représente plus clairement un niveau logique haut, mais le potentiel du nœud Z est toujours inférieur à V_L . Par conséquent, on peut également déduire que le seuil de basculement V_{THB} est supérieur à $V_{2c,max}$, quelque soit la valeur de l'autre entrée A .

Le début du front descendant correspond aux zones 3, 4a, 4b et 4c du *Brockett Annulus*. L'observation de ces zones pour une entrée quelque soit le potentiel de l'autre entrée permet de s'assurer de l'absence de basculements précoces pendant les transitions descendantes de cette entrée.

Les figures 7.14(a) et 7.14(b) représentent respectivement les couples de zones observés lors du début d'une transition descendante sur l'entrée A (V_A appartient à l'intervalle $[V_{4c,min} \dots V_{DD}]$ où $V_{4c,min}$ correspond à la borne basse de la zone 4c : $V_{4c,min} = 0.76V$ pendant que le potentiel de l'entrée B est inclus dans l'intervalle $[V_{SS} \dots V_{DD}]$.

Le potentiel du nœud Y est toujours inférieur à $0.5V$ et V_Z est toujours considéré comme un état haut (V_Z est inclus dans $V_H \dots V_{DD}$) : le point mémoire statique a commencé son basculement mais celui-ci ne peut se terminer tant que la tension de l'entrée A est supérieure à $V_{4c,min}$, quelque soit la valeur de l'entrée B . Par conséquent, le seuil de basculement V_{TLA} est inférieur à $V_{4c,min}$, quelque soit le potentiel V_B .

Les figures 7.15(a) et 7.15(b) sont respectivement la représentation des couples de zones observés pour le début de la transition descendante de l'entrée B quelque soit l'entrée A et les valeurs atteignables pour les potentiels V_Y et V_Z dans ces conditions.

Comme précédemment, le niveau du nœud Z est toujours compris dans l'intervalle

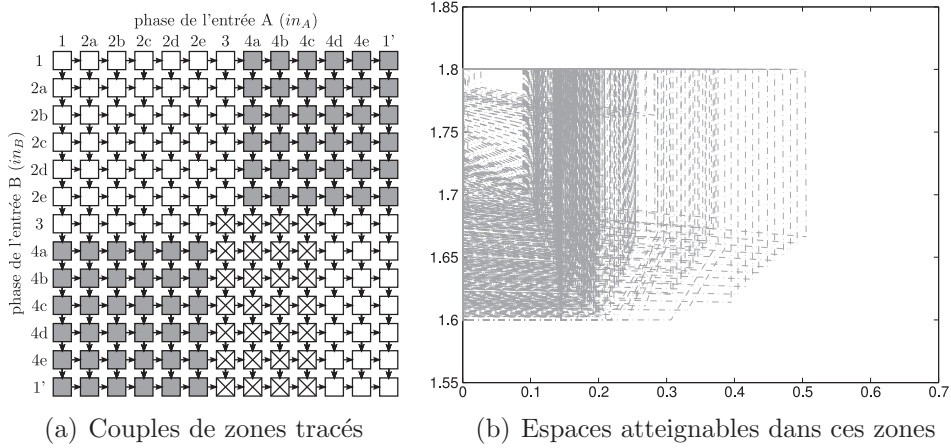


FIGURE 7.14 – Front descendant de l'entrée A d'une WM2 (VTLA)

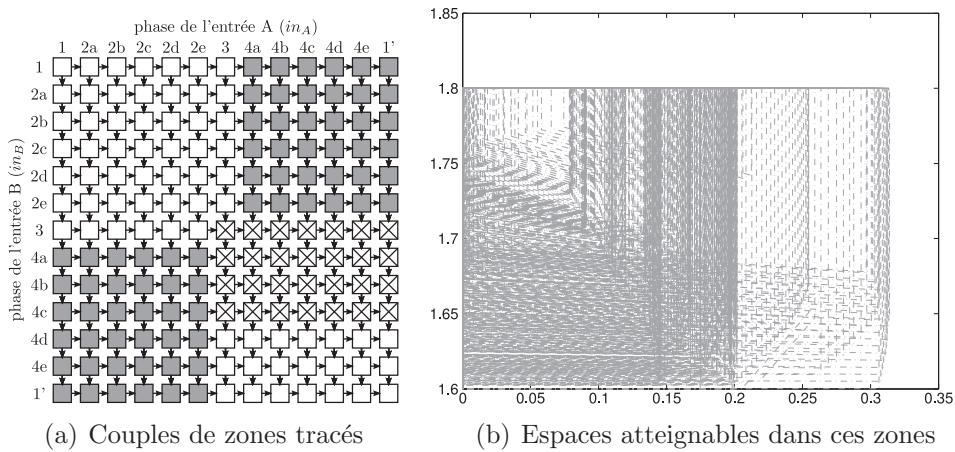


FIGURE 7.15 – Front descendant de l'entrée B d'une WM2 (VTLB)

$[V_H \dots V_{DD}]$, le point mémoire n'a pas encore basculé. Le seuil de basculement V_{TLB} est donc inférieur à $V_{4C,min}$.

Finalement, pour les deux entrées A et B , étant donné les propriétés mesurées sur les seuils de commutations hauts et bas, ces tensions de seuil sont correctement ordonnées (équations 7.2 et 7.3), quelques soient les activités de l'autre entrée au moment de la transition.

$$V_{TLA} < V_{4C,min} < V_{2C,max} < V_{THA} \quad (7.2)$$

$$V_{TLB} < V_{4C,min} < V_{2C,max} < V_{THB} \quad (7.3)$$

Cette étude formelle des tensions de basculement permet d'affirmer que dans tous les cas de fonctionnement, les tensions de seuils seront correctement ordonnées. Ce fonctionnement prouvé correct de la porte lorsque ses deux entrées varient simultanément permet de garantir son comportement dans les circuits asynchrones où les transitions en provenance des différents blocs logiques ne sont pas synchronisés entre elles.

7.4.2 Structure Conventiennelle

La structure CM2 a huit nœuds internes (figure 3.1(b) - page 35). La netlist de la porte est donnée en annexe B.3.

7.4.2.1 Analyse de *Brockett Annuli*

Les figures 7.16 et 7.17 représentent l'évolution de la tension de sortie (nœud Z) respectivement lors de la transition descendante et de la transition montante.

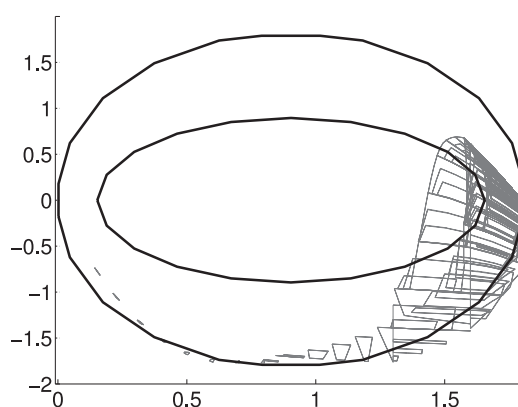
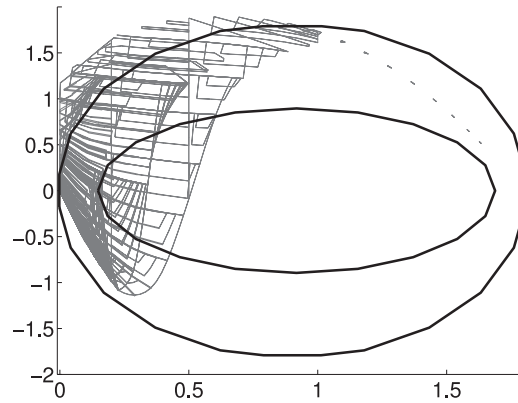


FIGURE 7.16 – *Brockett Annulus* de la sortie d'une CM2 - front descendant

FIGURE 7.17 – *Brockett Annulus* de la sortie d'une CM2 - front montant

La simulation d'un front montant et d'un front descendant de la sortie d'une structure conventionnelle, correspondant à l'étude des 97 états non grisés de la [figure 7.8](#), prend plusieurs jours de simulations sur un cœur processeur cadencé à $3GHz$. Ces performances sont liées à la complexité des calculs et à l'utilisation d'un solveur linéaire peu optimisé. La durée de simulation dépend très fortement du nombre de nœuds du circuit, déterminant le nombre de dimensions du volume atteignable. La vitesse de simulation dépend également du nombre de pas temporels de simulation nécessaires dans chaque état pour atteindre la convergence des résultats. Il faut généralement plusieurs dizaines de pas temporels pour chaque état du graphe de simulation.

Dans le cas particulier de la structure conventionnelle, la stabilisation de la sortie après que les deux entrées ont fini leurs transitions nécessite plusieurs centaines de pas temporels de simulations pour obtenir la convergence des résultats.

Les intervalles des valeurs possibles pour la dérivée de la tension de sortie sont moins importants dans cette structure que dans les résultats de la structure *Weak-Feedback*. Les écarts dans l'intervalle de la dérivée observés au début des transitions sont dus aux sur-approximations dans la représentation de l'espace atteignable : toutes les simulations analogiques conventionnelles faites sur cette structure ont montré la monotonie des variations de la sortie tant que les variations des entrées sont monotones.

7.4.2.2 Analyse des seuils de commutation

Comme pour la structure *Weak-Feedback*, les seuils de commutation de la structure conventionnelle ont été estimés d'après les espaces atteignables des différents couples de zones simulées. Pour chacun des résultats de simulation présenté dans cette section, l'ensemble des couples de zones pris en compte est équivalent à ceux de la simulation correspondante pour la structure *Weak-Feedback* ([section 7.4.1.2](#)).

Les [figures 7.18\(a\)](#) et [7.18\(b\)](#) représentent l'ensemble des espaces atteignables projetés sur le plan d'axes V_Y et V_Z en présence d'un front montant sur une entrée pendant que l'autre entrée est indéterminée.

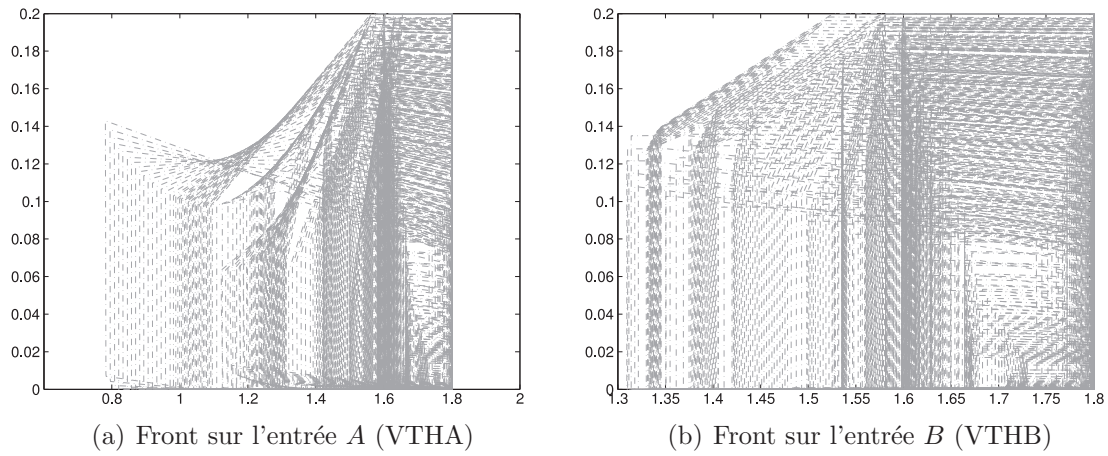


FIGURE 7.18 – Espaces atteignables lors des fronts montants pour une CM2

Dans ces deux cas de fonctionnement, la tension sortie de la porte est toujours confinée dans l'intervalle $[V_{SS} \dots V_L]$ synonyme d'état bas : la sortie ne bascule pas tant que la tension de ses deux entrées n'est pas inférieure à $V_{2c,max} = 1.04V$. Par déduction, les deux tensions de seuils V_{THA} et V_{THB} synonymes de basculement de la sortie à l'état logique 1 sont supérieures à cette borne.

Les polygones représentés dans les figures 7.19(a) et 7.19(b) sont les espaces atteignables par le point mémoire lorsqu'une des entrées de la porte CM2 présente un début de front descendant et pendant que l'autre entrée est indéterminée (toutes les valeurs possibles dans l'intervalle $[V_{SS} \dots V_{DD}]$).

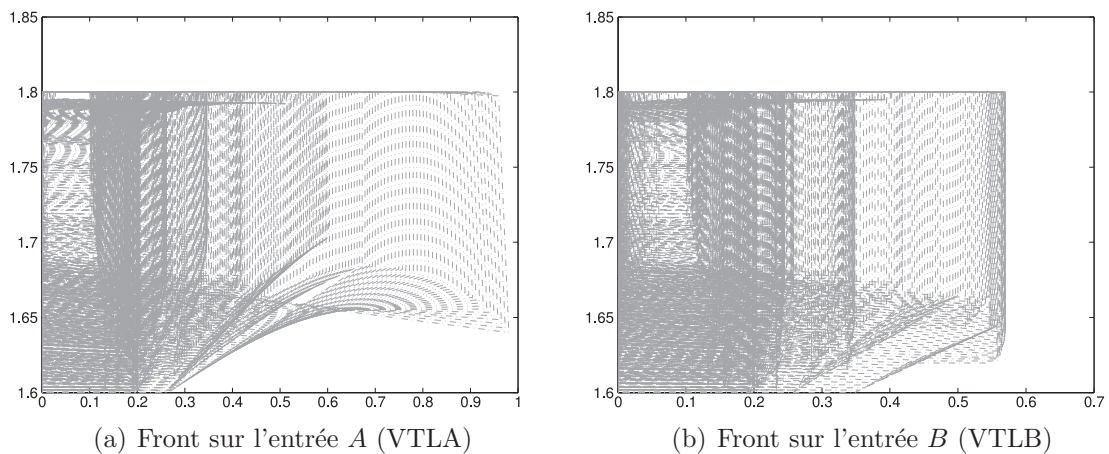


FIGURE 7.19 – Espaces atteignables lors des fronts descendants pour une CM2

Lors de ces fronts descendants, la sortie de la porte est toujours incluse dans l'intervalle $[V_H \dots V_{DD}]$ (niveau logique haut) : la sortie de la porte ne peut pas basculer tant que les tensions d'entrée ne sont pas inférieures à $V_{4c,min} = 0.76V$. Comme précédemment,

il a été prouvé que les seuils de basculement de cette structure CM2 sont correctement ordonnés d'après les relations d'ordre vues dans les équations 7.2 et 7.3.

Conclusion

Les méthodes formelles appliquées à la simulation d'un circuit analogique permettent de vérifier certaines propriétés d'un circuit exprimées dans un modèle de référence. Dans ce chapitre, ces propriétés ont été déduites d'après les simulations effectuées sur toute une classe de traces d'entrées. Cette classe représente toutes les transitions s'inscrivant dans un prototype spécifié sous la forme d'un *Brockett Annulus*.

L'outil Coho est utilisé pour calculer l'espace atteignable pour toutes les tensions de tous les nœuds des deux structures de C-element : la structure *Weak-Feedback* et la structure conventionnelle. Pour chacune de ces structures, il est vérifié :

- que la porte bascule toujours quelques soient les transitions appliquées sur ses entrées ;
- que les évolutions de la tension de sortie des cellules s'inscrivent dans ce même prototype de transition ; cette vérification permet de s'assurer que ces cellules respectent les gabarits d'une technologie donnée ;
- que les tensions de seuils sont correctement ordonnées : une estimation des bornes minimum des tensions de basculement hautes (V_{TH}) et des bornes maximum des tensions de basculement basses (V_{TL}) permet de s'assurer de leur relation d'ordre.

Ces vérifications formelles garantissent la zone de fonctionnement du modèle simulé. Si le modèle simulé est conforme au circuit réel, alors le comportement d'un circuit réel s'inscrira dans l'ensemble des traces atteignables. A l'opposé, il est impossible que le circuit réel fonctionne en dehors des espaces atteignables, tant que le modèle simulé dans l'outil est conforme au circuit fabriqué.

Conclusion

La très grande majorité des circuits intégrés actuellement fabriqués est basée sur des structures de calculs de type *synchrones* où l'horloge vient cadencer globalement l'avancement des données entre les blocs logiques. Les développements actuels intègrent sur un même substrat des systèmes de calcul et de mémorisation de plus en plus complexes. Cependant, l'intégration est de plus en plus difficile dans les technologies modernes dites "sub-microniques profondes" (*Deep submicronic*) où la finesse des gravures et des matériaux sont maintenant de l'ordre de la dizaine de nanomètres. Dans ces conditions, la fabrication de telles structures ne peut se faire qu'au prix d'une variabilité importante des circuits fabriqués. Cette variabilité remet en cause le paradigme du *tout synchrone* et favorise l'émergence des structures asynchrones où les hypothèses sur les délais sont tellement faibles que la structure fonctionne quels que soient les temps de propagation dans les portes logiques et dans les interconnexions.

Les circuits asynchrones, de par leur fonctionnement autocadencé localement, s'adaptent particulièrement bien aux caractéristiques requises pour les circuits intégrés actuels :

- leur structure modulaire permet l'intégration de modules tiers (de type *Intellectual Properties*) ;
- leur fonctionnement autocadencé s'adapte aux caractéristiques réelles du circuit fabriqué tout en exploitant au maximum sa vitesse de traitement et procure de la robustesse.

Dans les structures asynchrones, les données sont transférées entre les blocs logiques par le biais de canaux de communication. Ces transferts point-à-point font appel à des protocoles de communication de type poignée de main (*Handshake protocols*) pour synchroniser le bloc émetteur et le bloc récepteur lors de chaque transfert. Chaque bloc logique joue simultanément le rôle d'émetteur sur certains canaux et de récepteur sur d'autres. La fonction logique implémentée dans chaque bloc doit prendre en compte la phase de tous ces canaux pour fonctionner correctement. Or, les signaux issus des blocs précédents et suivants ne sont pas synchronisés entre eux. La fonction C-element implémente la fonction de *rendez-vous* nécessaire à cette synchronisation locale.

La conception de toutes ces portes de synchronisation doit être méthodique afin de garantir leur bon fonctionnement et par extension le bon fonctionnement de l'ensemble du circuit asynchrone. Cette vérification du fonctionnement au niveau de la porte logique n'a fait l'objet que de peu de travaux de recherche. Puisqu'une mauvaise synchronisation dans un circuit asynchrone se traduit généralement par un blocage complet du circuit, un circuit asynchrone robuste devrait pouvoir fonctionner (ne pas se bloquer) et réaliser la fonction souhaitée dans des conditions de fonctionnement clairement sous-optimales.

Un problème de synchronisation a été mis en évidence dans un circuit complet lorsque les temps de transition sont lents sur les entrées des C-elements. Ce phénomène est relié aux tensions de seuil permettant au C-element de basculer. Lorsque ces tensions de seuil sont incorrectement ordonnées, cela induit une mauvaise prise en compte des niveaux logiques sur les entrées des C-elements et finalement des basculements intempestifs des points mémoire du circuit.

Plusieurs variantes de la fonction C-element sont nécessaires afin de concevoir efficacement les diverses synchronisations possibles et il existe de nombreuses implémentations matérielles pour la fonction C-element. Pour chaque cellule décrite au niveau transistor, les tensions de seuil sont mesurées lors de simulations analogiques réalisées dans l'environnement de conception *Cadence* d'après les caractéristiques de la technologie. Ces mesures font apparaître des différences de comportements analogiques entre les différentes structures bien que celles-ci soient équivalentes dans les modèles numériques usuels.

L'étude a été effectuée pour différentes structures de C-element réalisées dans différentes technologies de *STMicronics*. Certaines de ces cellules sont issues de la *TIMA Asynchronous Library* (TAL) et ont déjà permis la réalisation de circuits intégrés fonctionnels au sein de l'équipe CIS du laboratoire TIMA mais également au CEA-LETI. Les méthodes employées ne sont pas spécifiques à ces technologies et les mêmes techniques de caractérisation et de correction du comportement peuvent être appliqués à d'autres technologies et plus généralement à tout point mémoire statique.

Cette étude a été étendue pour prendre en compte des variations des procédés de fabrication afin d'assurer le bon fonctionnement de la presque totalité des circuits fabriqués dans des technologies à fortes variabilités. La distribution de l'écart entre les tensions de seuil a été mesurée dans une simulation *Monte-Carlo* considérant 500 cas différents de fabrication. Ces jeux de mesures permettent d'appliquer la méthode statistique des 3σ qui garantit que moins de 1‰ des circuits sera en dehors de la zone correcte et pourront potentiellement être assujettis au phénomène de désynchronisation.

Néanmoins, ces vérifications ont été faites dans le cas de pentes lentes sur l'entrée des portes logiques. Il est possible que d'autres phénomènes viennent interférer avec le comportement correct du circuit. Dans le but d'améliorer la couverture de toutes ces possibilités de fonctionnement, il est nécessaire d'utiliser des méthodes de vérifications plus formelles.

Coho est un simulateur symbolique permettant d'obtenir la réponse d'un circuit électronique analogique pour toute une classe d'entrées (dont le gabarit est donné sous la forme d'un *Brockett Annulus**). Si le modèle du circuit simulé et le modèle des signaux

d'entrée est fidèle au circuit réel, alors ce circuit réel aura un comportement inclu dans les résultats symboliques de la simulation. Ces résultats formels permettent la vérification de plusieurs propriétés qui garantissent le comportement correct de tous les circuits fabriqués dans des conditions de fonctionnement non simulées lors des caractérisations précédentes dans l'environnement *Cadence*.

De plus, l'insensibilité (ou la quasi-insensibilité) des circuits asynchrones aux délais de propagation dans les portes et dans les interconnexions n'implique pas leur insensibilité aux temps de transition. Deux nouvelles classes de circuits : *Quasi-Delay-and-Slope-Insensitive* (QDSI) et *Delay-and-Slope-Insensitive* (DSI) ont été respectivement définies pour caractériser le fonctionnement de circuits asynchrones de structure QDI et DI qui sont conçus à partir de cellules C-elements dont le comportement est correct, y compris en présence de pentes lentes.

Ces techniques de caractérisation et de vérification appliquées à différentes structures de C-elements réalisées dans différentes technologies permettent de dégager des tendances dans le comportement des différentes portes nécessaires au fonctionnement des circuits asynchrones. Les résultats obtenus sont spécifiques aux technologies et aux dimensionnement des portes réalisées. Néanmoins, les méthodes employées peuvent être transposées aisément à d'autres technologies et à d'autres structures de portes réalisant des fonctions de synchronisation.

Cette étude permet donc la conception de points de synchronisation robustes dont les caractéristiques sont peu sensibles aux procédés de fabrication et aux conditions de fonctionnement. Ces propriétés sont requises pour l'usage des circuits asynchrones dans les environnements hostiles et dans les technologies à très fortes variabilités comme les technologies nanométriques ou sur les substrats non-conventionnels tels que les substrats souples.

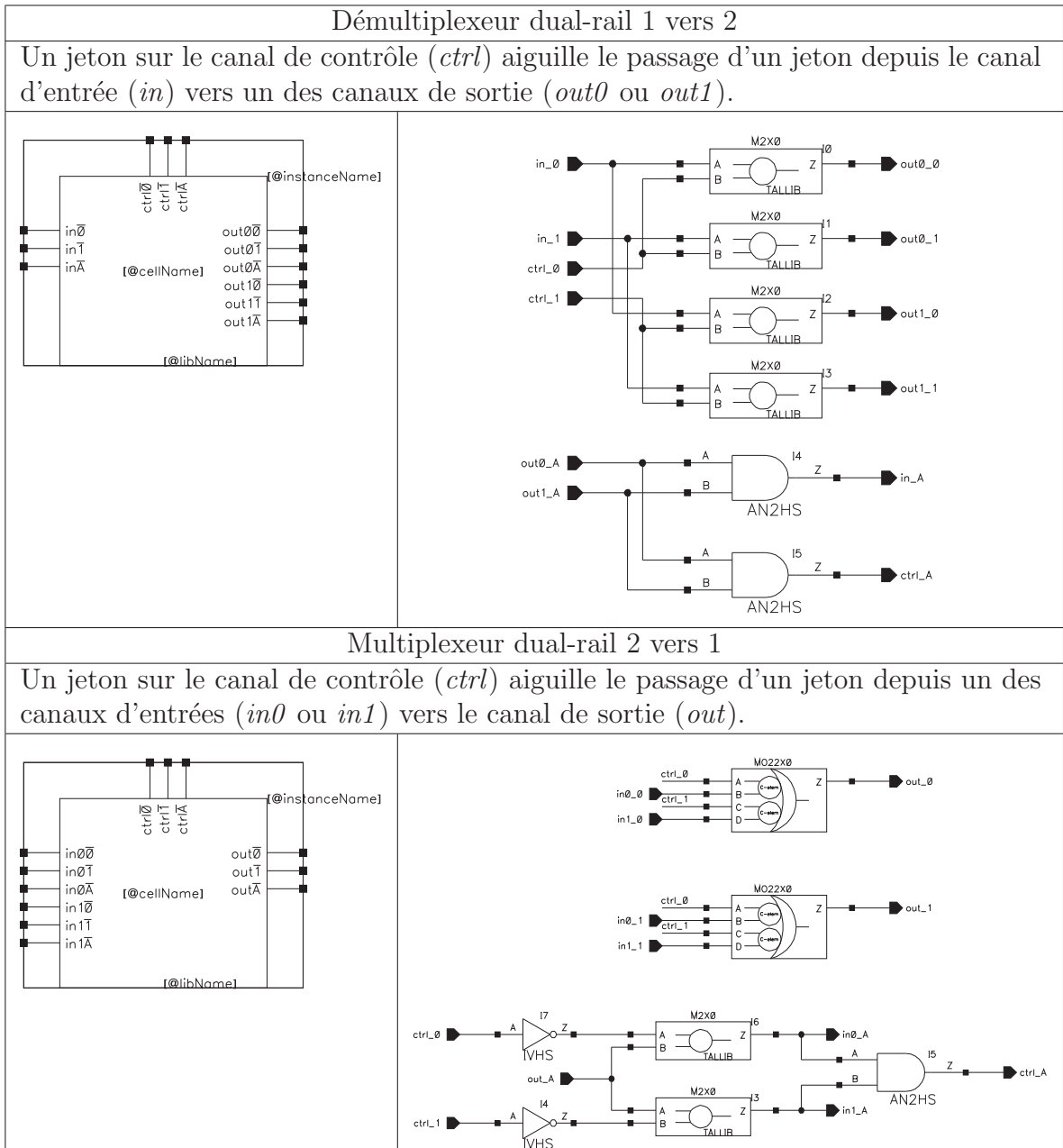
Les vérifications effectuées avec des simulations analogiques conventionnelles ou avec des techniques semi-formelles permettent de prendre en compte les effets analogiques dans le fonctionnement des circuits numériques. Ces simulations ne peuvent pas être envisagées actuellement sur des circuits entiers. Mais la vérification des blocs logiques critiques et des C-elements permet de garantir la correction du circuit entier, même lorsque les conditions de fonctionnement sont dégradées.

En milieu hostile, un durcissement des circuits synchrones est très coûteux en terme de temps de conception et de surface. Les circuits asynchrones, de par leurs caractéristiques, sont insensibles aux délais, ce qui leur confère déjà une excellente robustesse. Grâce à ces travaux, l'ajout de l'insensibilité aux pentes permet de les rendre encore plus robustes. Cette méthode permet l'obtention de circuits intégrés asynchrones plus robustes pouvant être utilisés pour des applications sensibles comme le domaine médical ou l'aéronautique...

Annexe **A**

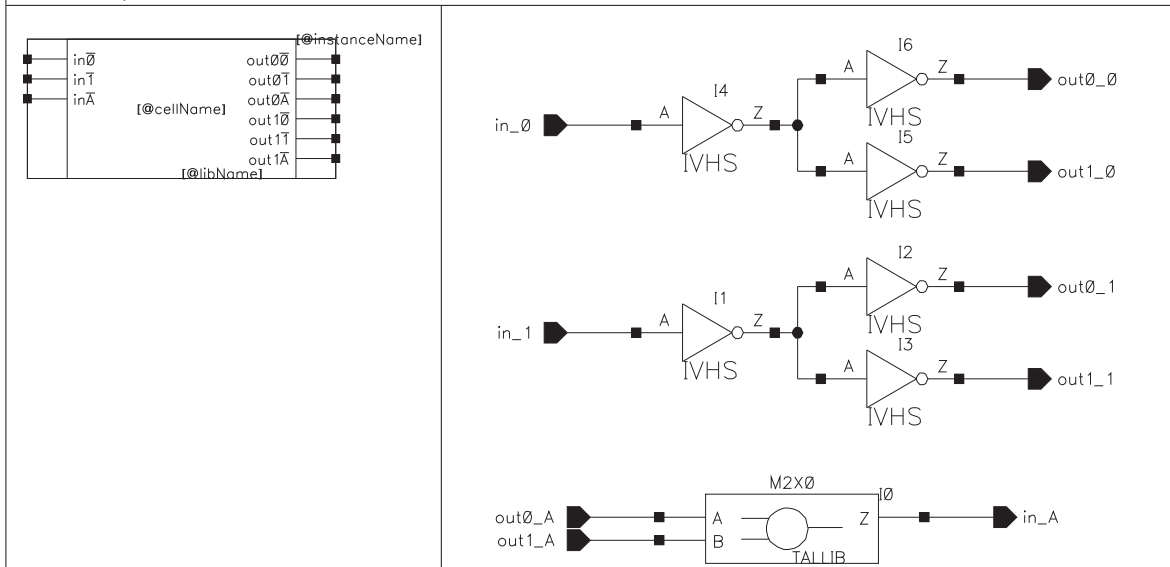
CRC16 asynchrone

A.1 Sous-composants du CRC16



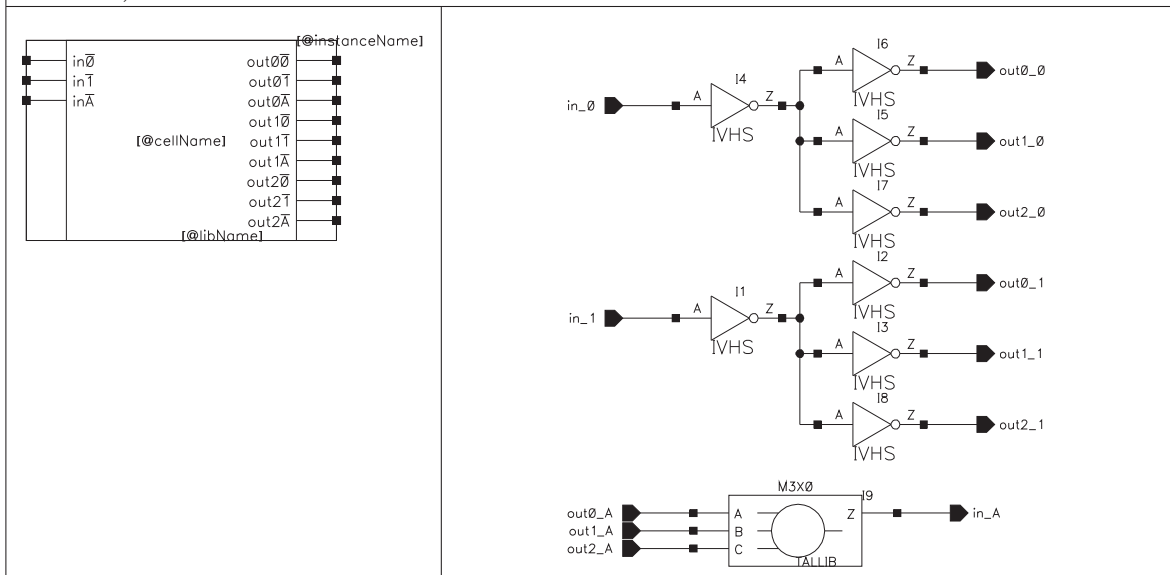
Fork dual-rail 1 vers 2

Chaque jeton du canal d'entrée (*in*) est envoyé aux deux canaux de sortie (*out0* et *out1*).



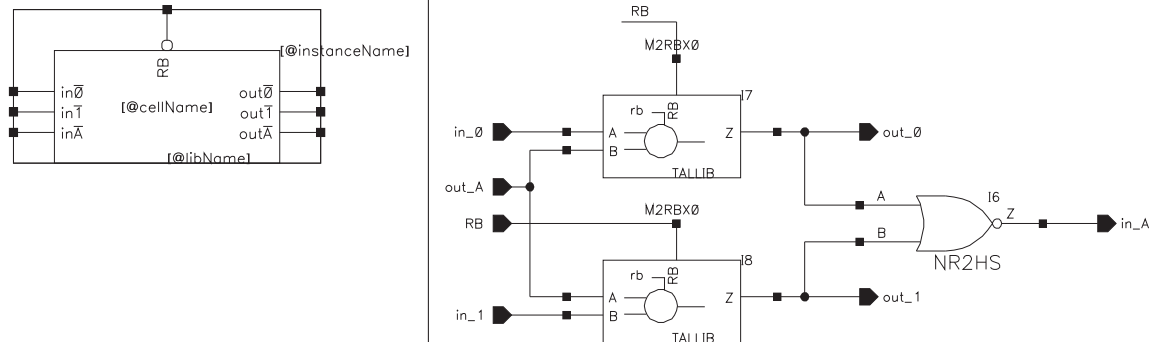
Fork dual-rail 1 vers 3

Chaque jeton du canal d'entrée (*in*) est envoyé aux trois canaux de sortie (*out0*, *out1* et *out2*).



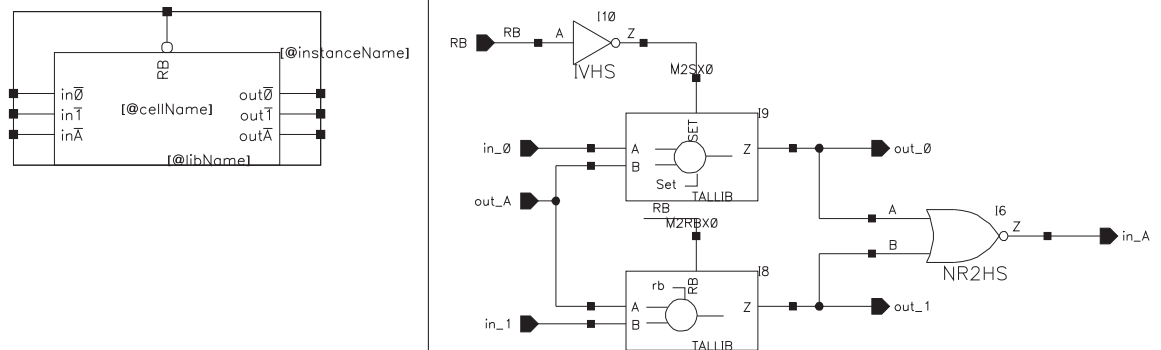
Half-buffer WCHB initialement vide

Chaque jeton du canal d'entrée (*in*) est mémorisé et envoyé au canal de sortie (*out*).



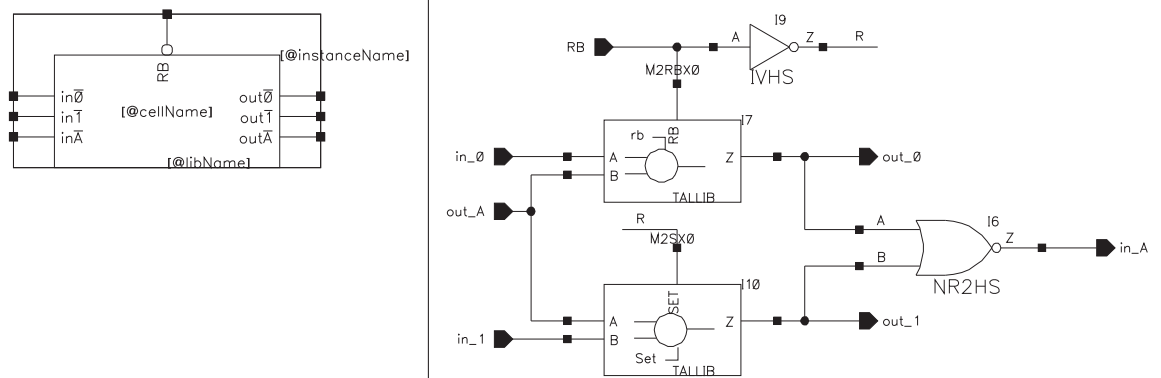
Half-buffer WCHB initialement à 0

Chaque jeton du canal d'entrée (*in*) est mémorisé et envoyé au canal de sortie (*out*).
Le buffer émet initialement la donnée 0.



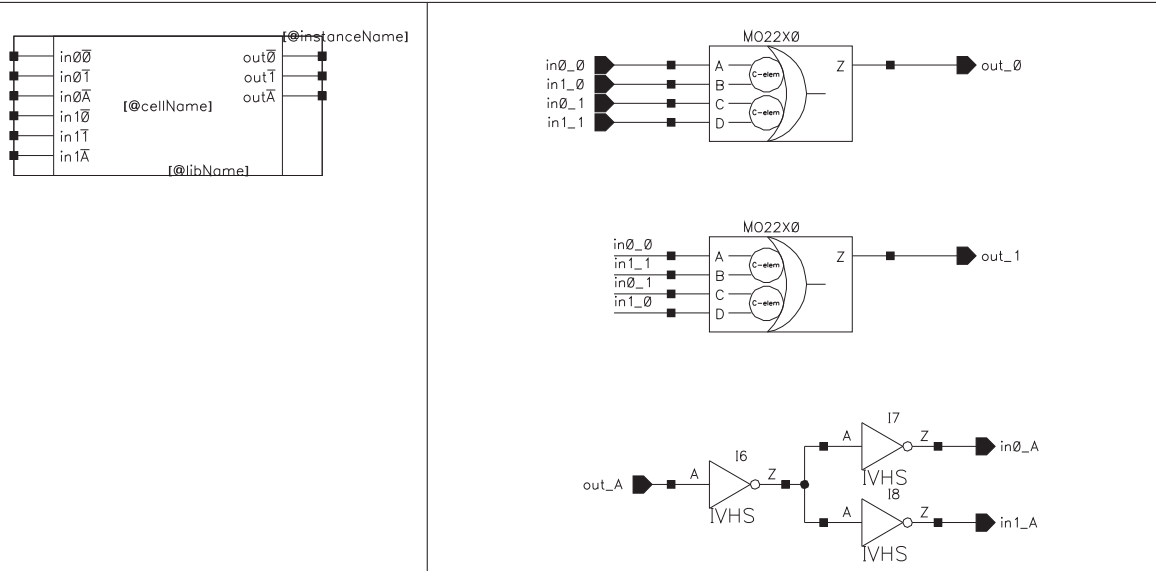
Half-buffer WCHB initialement à 1

Chaque jeton du canal d'entrée (*in*) est mémorisé et envoyé au canal de sortie (*out*).
Le buffer émet initialement la donnée 1.

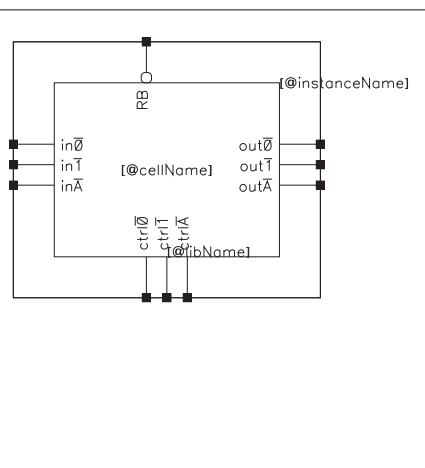


XOR dual-rail à deux entrées

Un jeton de chaque entrée ($in0$ et $in1$) sont utilisés pour déterminer la valeur du canal de sortie (out).

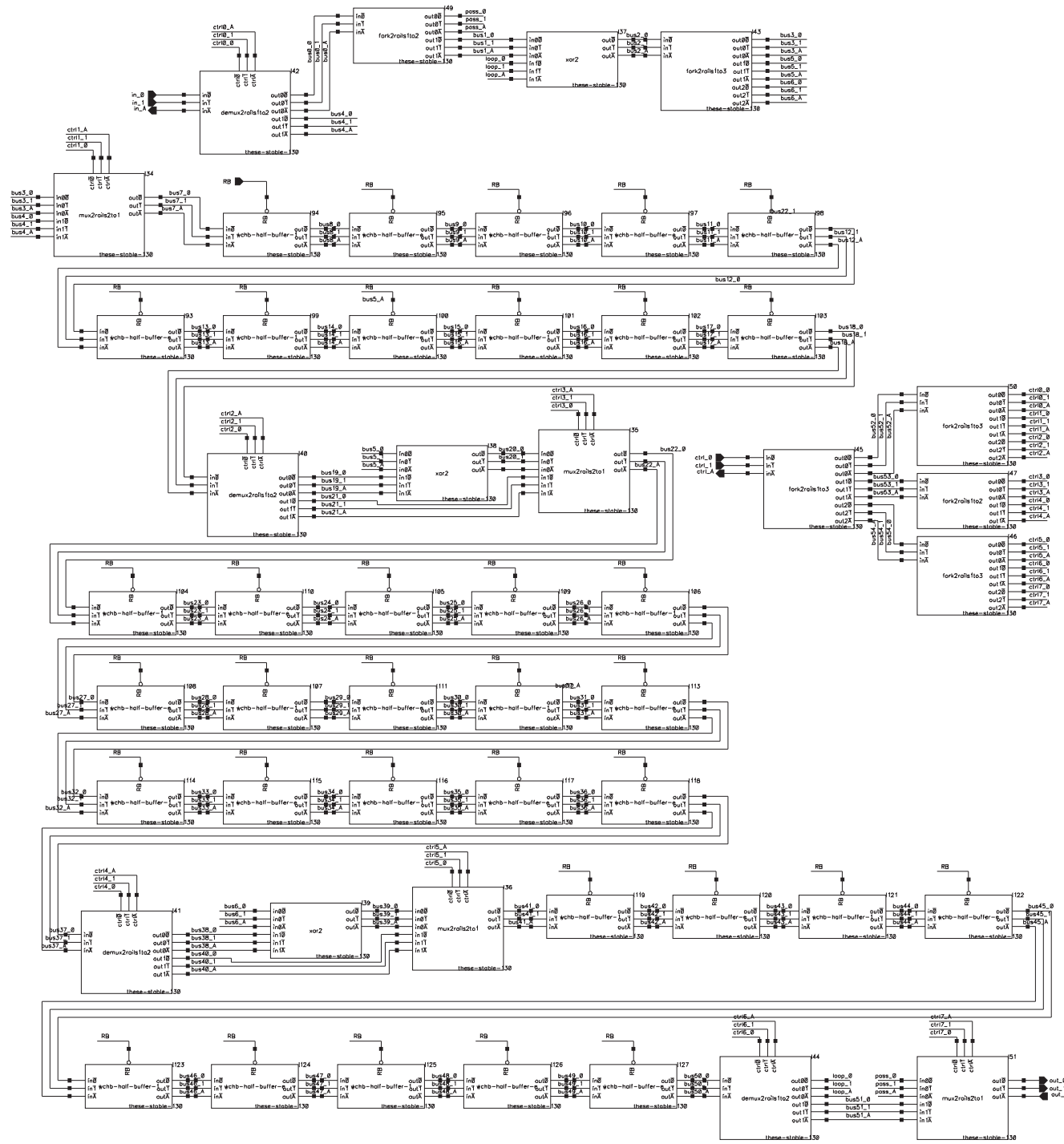


CRC16 CCITT



Pour chaque jeton de contrôle (canal $ctrl$) et de donnée (canal in), un jeton est émis en sortie (canal out). Tant que les jetons de contrôle sont égaux à 0, ils sont réémis sur le canal de sortie et dans le même temps le code CRC correspondant est calculé en interne. A la fin des données, il faut émettre 16 jetons de contrôle à 1 et 16 jetons en entrée (leurs valeurs fixent la valeur initiale du prochaine code CRC) pour purger le registre à décalage et émettre en sortie les 16 bits de CRC correspondant. Le *schematic* asynchrone est affiché en page suivante.

A.2 Schematic principal



Fichiers de configuration de Coho

B.1 Spécifications du circuit et des entrées d'une porte WM2

```
1 function c = circuit_create(csize,inputFn)
2
3 % structure de dimensionnement si absente
4 if(nargin<1||isempty(csize))
5     r = 2.2; nm = 1e-9; c = 2;
6     ip = 2*c*r*nm; in = 2*c*nm;
7     op = c*r*nm; on = c*nm;
8     kp = r*nm; kn = 1*nm;
9     csize = struct('ip',ip,'in',in,'op',op,'on',on,'kp',kp,'kn',kn);
10 end
11
12 % specification de l'entree si absente
13 if(nargin<2||isempty(inputFn))
14     input specification
15     radius = [1.5,2]*1e10;
16     vbnds = [0,0.2,1.6,1.8];
17     minT = 1e-9;
18     sbnds = [0,0.2,1.6,1.8];
19     inputFn = brockett(radius,vbnds,minT,sbnds);
20 end
21
22 ip = csize.ip; in = csize.in;
23 op = csize.op; on = csize.on;
24 kp = csize.kp; kn = csize.kn;
25
26 % create a empty circuit
27 c = circuit('celement_wf');
28
29 % add inputs and nodes
30 [c,i1] = addInput(c,inputFn,'i1');
31 [c,i2] = addInput(c,inputFn,'i2');
```

```

32 [c,p] = addNode(c,'p');
33 [c,n] = addNode(c,'n');
34 [c,y] = addNode(c,'y');
35 [c,z] = addNode(c,'z');
36
37 % add transistors
38 c = addpfet(c,vdd(c),i1,p, ip);
39 c = addpfet(c,p,i2,y, ip);
40 c = addnfet(c,gnd(c),i1,n, in);
41 c = addnfet(c,n,i2,y, in);
42
43 % add inverter
44 c = addpfet(c,vdd(c),y,z, op);
45 c = addnfet(c,gnd(c),y,z, on);
46
47 % add keeper
48 c = addpfet(c,vdd(c),z,y, kp);
49 c = addnfet(c,gnd(c),z,y, kn);
50
51 % add output capacitance
52 c = addcap(c,z,circuit_get(c,'cap',i1));
53 c = addcap(c,p,circuit_get(c,'cap',p)*-0.75);
54 c = addcap(c,n,circuit_get(c,'cap',n)*-0.75);

```

B.2 Spécifications de l'automate de simulation

```

1 function ha = circuit_ha(varargin)
2 % This function create hybrid automaton for the celement circuit.
3 % In the
4 % micro-pipeline, x1/x2 can not rise/fall until z is clear low/high.
5 % The
6 % output is low/high iff x1 and x2 are both low/high. That is, inputs can
7 % not change until the output has the same value with inputs. Therefore,
8 % if the initial state is (0,0,0), the only path to B4 is go to <B3,B3>
9 % first, and go to B4 until output is high. And from (1,1,1), the only
10 % path to B2 is go to <B1,B1> first, and can not continue until output
11 % is low. Therefore, we can split the computation into two phases
12 % low2high <0,0,0> -> <1,1,1>
13 % high2low <1,1,1> -> <0,0,0>
14
15 opt = struct('ha','two','hid',0,'splitNum',5,'fast',false);
16 opt = utils_struct(opt,'set',varargin{:});
17 splitNum = opt.splitNum;
18
19 % circuit name
20 circuit = circuit_config('get','circuit');
21 name = circuit_get(circuit,'name');
22 % faces from <0,0,0> and <1,1,1>
23 [inits,tphs] = circuit_initRegion;
24
25 % fwdOpt and planes
26 if (opt.fast) % run fast
27     fwdOpt = ph_getOpt('fast');

```

```

26  ha_config('set','maxstep',50);
27  else % regular
28    fwdOpt = ph_getOpt;
29  end
30
31  % you can use different phinfo for each state
32  phinfo = [];
33
34  % stable states
35  maxsteps = zeros(2,2,3);
36  maxsteps(2,1,[2,3]) = 100; % quite stable
37  maxsteps(1,2,[2,3]) = 100;
38  maxsteps(1,1,1) = 1000; % output switch
39  maxsteps(2,2,1) = 1000;
40
41  switch(lower(opt.ha))
42  case 'one'
43    region = true(4,4);
44    region(4,2) = false; region(2,4) = false;
45    region = repmat(region,[1,1,3]);
46    region(1,4,1) = false; region(3,2,1) = false;
47    region(4,1,1) = false; region(2,3,1) = false;
48    region(1,3,1) = false; region(3,1,1) = false;
49    region(1,[1,2],2) = false; region(3,[3,4],2) = false;
50    region([1,2],1,3) = false; region([3,4],3,3) = false;
51    sources = [2.1,1.0;1.0,2.1];
52    init = inits{1};
53    ha = circuit_ha_create_2d('name',name,'splitNum',splitNum,...
54      'useRegion',region,'sources',sources,'initials',init,...
55      'phinfo',phinfo,'fwdOpt',fwdOpt,'maxsteps',maxsteps,'tphs',tphs);
56  case 'two'
57    name1 = [name,'_l2h'];
58    region = true(4,4); region(:,4) = false;
59    region(4,:) = false; region(1,1) = false;
60    region = repmat(region,[1,1,3]);
61    region(1,:,2) = false; % no input, dangling states
62    region(:,1,3) = false; % no input, dangling states
63    sources = [2.1,1.0;1.0,2.1];
64    init = inits{1};
65    ha(1) = circuit_ha_create_2d('name',name1,'splitNum',splitNum,...
66      'useRegion',region,'sources',sources,'initials',init,...
67      'phinfo',phinfo,'fwdOpt',fwdOpt,'maxsteps',maxsteps,'tphs',tphs);
68
69    name2 = [name,'_h2l'];
70    region = true(4,4); region(:,2) = false;
71    region(2,:) = false; region(3,3) = false;
72    region = repmat(region,[1,1,3]);
73    region(3,:,2) = false; % no input, dangling states
74    region(:,3,3) = false; % no input, dangling states
75    sources = [4.1,3.0;3.0,4.1];
76    init = inits{2};
77    ha(2) = circuit_ha_create_2d('name',name2,'splitNum',splitNum,...
78      'useRegion',region,'sources',sources,'initials',init,...
79      'phinfo',phinfo,'fwdOpt',fwdOpt,'maxsteps',maxsteps,'tphs',tphs);

```

```

80 case 'l2h' %test <0,0,0>-><1,1,1>
81     name1 = [name, '_l2h_r1r2']; % r1 raises then r2 raises
82     region = false(4,4,3);
83     region(2,1,1) = true; region(3,3,1) = true; region(3,[1,2],2) = true;
84     sources = [2.1,1.0];
85     init = inits{1}{1};
86     ha(1) = circuit_ha_create_2d('name',name1,'splitNum',splitNum,...
87         'useRegion',region,'sources',sources,'initials',init,...
88         'phinfo',phinfo,'fwdOpt',fwdOpt,'maxsteps',maxsteps,'tphs',tphs);
89
90     name2 = [name, '_l2h_r2r1']; % r2 raises then r1 raises
91     region = false(4,4,3);
92     region(1,2,1) = true; region(3,3,1) = true; region([1,2],3,3) = true;
93     sources = [1.0,2.1];
94     init = inits{1}{2};
95     ha(2) = circuit_ha_create_2d('name',name2,'splitNum',splitNum,...
96         'useRegion',region,'sources',sources,'initials',init,...
97         'phinfo',phinfo,'fwdOpt',fwdOpt,'maxsteps',maxsteps,'tphs',tphs);
98
99     otherwise
100         error('do not support now');
101 end
102
103 if(opt.hid>0)
104     ha = ha(opt.hid);
105 end

```

B.3 Spécifications du circuit et des entrées d'une porte CM2

```

1 function m = circuit_create
2
3 % input specification
4 radius = [1.5,2]*1e10;
5 vbnds = [0,0.2,1.6,1.8];
6 minT = 1e-9;
7 sbnds = [0,0.2,1.6,1.8];
8 inputFn = brockett(radius,vbnds,minT,sbnds);
9
10
11 % 130nm process
12 nm = 1e-9;
13 %u = 130*nm
14 ip = 920*nm; in = 200*nm;
15 op = 345*nm; on = 150*nm;
16 kp = 150*nm; kn = 150*nm;
17
18 % create a empty circuit
19 m = circuit;
20
21 % add inputs and nodes
22 [m,i1] = addInput(m,inputFn,'i1');

```

```
23 [m,i2] = addInput(m,inputFn,'i2');
24 [m,p] = addNode(m,'p');
25 [m,n] = addNode(m,'n');
26 [m,pk] = addNode(m,'pk');
27 [m,nk] = addNode(m,'nk');
28 [m,y] = addNode(m,'y');
29 [m,z] = addNode(m,'z');
30
31 % add transistors
32 m = addpfet(m,vdd(m),i1,p, ip);
33 m = addpfet(m,p,i2,y, ip);
34 m = addnfet(m,gnd(m),i1,n, in);
35 m = addnfet(m,n,i2,y, in);
36
37 % add inverter
38 m = addpfet(m,vdd(m),y,z, op);
39 m = addnfet(m,gnd(m),y,z, on);
40
41 % add keeper
42 m = addpfet(m,pk,z,y, kp);
43 m = addpfet(m,vdd(m),i1,pk,kp);
44 m = addpfet(m,vdd(m),i2,pk,kp);
45 m = addnfet(m,nk,z,y, kn);
46 m = addnfet(m,gnd(m),i1,nk,kn);
47 m = addnfet(m,gnd(m),i2,nk,kn);
48
49 % add output capacitance (equal to one input capacitance)
50 m = addcap(m,z,circuit_get(m,'cap',i1));
51 % reduce the capacitance of internal node
52 % keep only 1/4 of the calculated capacitance since it is over-approximated.
53 m = addcap(m,p,circuit_get(m,'cap',p)*-0.75);
54 m = addcap(m,n,circuit_get(m,'cap',n)*-0.75);
55 m = addcap(m,pk,circuit_get(m,'cap',pk)*-0.75);
56 m = addcap(m,nk,circuit_get(m,'cap',nk)*-0.75);
```


Bibliographie

- [ABR⁺01] André Abrial, Jacky Bouvier, Marc Renaudin, Patrice Senn et Pascal Vivet: *A new contactless smart card IC using an on-chip antenna and an asynchronous microcontroller*. Dans *IEEE Journal of Solid-State Circuits July*; 36(7), 2001.
- [ADM02] Eugene Asarin, Thao Dang et Oded Maler: *Lecture Notes in Computer Science*, chapitre The d/dt Tool for Verification of Hybrid Systems, pages 746–770. Springer, 2002.
- [AN10] Melinda Y. Agyekum et Steven M. Nowick: *An Error-Correcting Unordered Code and Hardware Support for Robust Asynchronous Global Communication*. Dans *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2010.
- [ATRY10] Z. Al Tarawneh, G. Russell et Alex Yakovlev: *An analysis of SEU robustness of C-element structures implemented in bulk CMOS and SOI technologies*. Dans *Proceedings of the International Conference on Microelectronics (ICM)*, 2010.
- [BBD⁺03] Dominique Borrione, Menouer Boubekour, Emil Dumitrescu, Marc Renaudin, Jean Baptiste Rigaud et Antoine Sirianni: *An Approach to the Introduction of Formal Validation in an Asynchronous Circuit Design Flow*. Dans *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)*, 2003.
- [Bre07] Vivian Bregier: *Synthèse automatisée de circuits asynchrones optimisés prouvés Quasi Insensibles aux Délais*. Thèse de doctorat, Institut National Polytechnique de Grenoble, 2007.
- [CAD] *Synopsys Cadabra*. <http://www.synopsys.com/TOOLS/IMPLEMENTATION/CUSTOMIMPLEMENTATION/Pages/Cadabra-ds.aspx>.
- [CKK⁺97] Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno et Alexandre Yakovlev: *Petrify : A Tool for Manipulating Concurrent Specifications and Synthesis of Asynchronous Controllers*. IEICE TRANSACTIONS on Information and Systems, E80-D :315–325, 1997.
- [Cla67] Wesley A. Clark: *Macromodular computer systems*. Dans *Proceedings of the AFIPS spring joint computer conference*, 1967.

-
- [ESFR02] Mohammed Es Salhiene, Laurent Fesquet et Marc Renaudin: *Dynamic Voltage Scheduling for Real Time Asynchronous Systems*. Dans *Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation*, tome 2451 de *Lecture Notes in Computer Science*, pages 155–171. Springer, 2002.
- [FBFR07] Bertrand Folco, Vivian Bregier, Laurent Fesquet et Marc Renaudin: *Technology Mapping for Area Optimized Quasi Delay Insensitive Circuits*. Vlsi-Soc : From Systems To Silicon, Volume 240/2007(978-0-387-73660-0) :55–69, 2007.
- [FFRP04] Krisztián Flautner, David Flynn, David Roberts et Dipesh I. Patel: *IEM926 : An Energy Efficient SoC with Dynamic Voltage Scaling*. Dans *Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2004.
- [Fod88] György Fodor: *Nodal analysis of electrical networks*. Elsevier, Amsterdam, 1988.
- [Fre05] Goran Frehse: *PHAVer : Algorithmic verification of hybrid systems past HyTech*. 2005.
- [GGD00] W. Steven Gray, Oscar R. González et Mustafa Doğan: *Stability Analysis of Digital inear Flight Controllers Subject to Electromagnetic Disturbances*. IEEE Transactions on Aerospace and Electronic Systems, 36 :1204–1218, 2000.
- [GMD⁺97] Mohan Guruswamy, Robert L. Maziasz, Daniel Dulitz, Srilata Raman, Venkat Chiluvuri, Andrea Fernandez et Larry G. Jones: *CELLERITY : A Fully Automatic Layout Synthesis System for Standard Cell Libraries*. Dans *Proceedings of 34th Design Automation Conference (DAC)*, 1997.
- [Hau95] Scott Hauck: *Asynchronous Design Methodologies : An Overview*. Dans *Proceedings of the IEEE, Vol. 83, No. 1*, 1995.
- [Hoa85] C. A. R. Hoare: *Communicating Sequential Processes*. Prentice Hall International, 1985.
- [IC09] TIEMPO IC: *Smart Card Trends : CONTACTLESS/SECURITY 'THE' BREAKTHROUGH*, 2009. http://www.tiempo-ic.com/uploads/Press/Tiempo%5FSmartCardsTrends_N39-Issue4-VolVI.pdf.
- [IEEa] *IEEE Standard for Verilog® Hardware Description Language*.
- [IEEb] *IEEE Standard VHDL Language Reference Manual*.
- [JMG00] Hans Jacobson, Chris Myers et Ganesh Gopalakrishnan: *Achieving Fast and Exact Hazard-Free Logic Minimization of Extended Burst-Mode gC Finite State Machines*. Dans *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD)*., 2000.
- [KH09] Cédric Koch-Hofer: *Modélisation, Validation et Présynthèse de Circuits Asynchrones en SystemC*. Thèse de doctorat, INP-Grenoble, 2009.
-

- [LPMC09] Régis Leveugle, Laurence Pierre, Paolo Maistri et Renaud Clavel: *Soft Error Effect and Register Criticality Evaluations : Past, Present and Future*. Dans *Proceedings of the IEEE Workshop on Silicon Errors in Logic - System Effects (SELSE)*, 2009.
- [Mar89] Alain J. Martin: *Programming in VLSI : From Communicating Processes to Delay-Insensitive Circuits*. rapport technique, California Institute of Technology, 1989.
- [Mar90a] Alain J. Martin: *Formal program transformations for VLSI circuit synthesis*. Addison-Wesley Longman Publishing Co., Inc., 1990.
- [Mar90b] Alain J. Martin: *The limitations to delay-insensitivity in asynchronous circuits*. Dans *Proceedings of the sixth MIT conference on Advanced research in VLSI*, 1990.
- [Mar91] Alain J. Martin: *Synthesis of Asynchronous VLSI Circuits*. rapport technique, California Institute of Technology, 1991.
- [MB59] D. E. Muller et W. S. Bartky: *A theory of asynchronous circuits*. Dans *Proceedings of the International Symposium on the Theory of Switching*, 1959.
- [Mil65] Raymond E. Miller: *Switching Theory Vol 2 : Sequential Circuits and Machines*. Wiley, 1965.
- [MJ01] Chris Myers et Hans Jacobson: *Efficient Exact Two-Level Hazard-Free Logic Minimization*. Dans *Proceedings of the 7th International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, 2001.
- [MLM⁺97] Alain J. Martin, Andrew Lines, Rajit Manohar, Mika Nystroem, Paul Penzes, Robert Southworth et Uri Cummings: *The Design of an Asynchronous MIPS R3000 Microprocessor*. Dans *Proceedings of the 17th Conference on Advanced Research in VLSI (ARVLSI)*, 1997.
- [MND91] David Messerschmitt, Steven Nowick et David Dill: *Synchronization design for digital systems*. Kluwer Academic, 1991. <http://books.google.fr/books?id=MAeAIdQsnUkC&printsec=frontcover>.
- [MNP⁺03] Alain J. Martin, Mika Nyström, Karl Papadantonakis, Paul I. Pénez, Piyush Prakash, Catherine G. Wong, Jonathan Chang, Kevin S. Ko, Benjamin Lee, Elaine Ou, James Pugh, Eino Ville Talvala, James T. Tong et Ahmet Tura: *The Lutonium : A Sub-Nanojoule Asynchronous 8051 Microcontroller*. Dans *Proceedings of the 9th IEEE International Symposium on Asynchronous Systems & Circuits (ASYNC)*, 2003.
- [Mon07] Yannick Monnet: *Etude et Modélisation de Circuits Résistants aux Attaques non Intrusives par Injection de Fautes*. Thèse de doctorat, INP Grenoble, 2007.
- [Mye01] Chris J. Myers: *Asynchronous circuit design*. Wiley, 2001.
- [ND95] Steven M. Nowick et David L. Dill: *Exact Two-Level Minimization of Hazard-Free Logic with Multiple-Input Changes*. Dans *IEEE Transactions*

-
- on Computer-Aided Design of Integrated Circuits and Systems. Vol. 14 (8)*, 1995.
- [NNSvB94] Lars S. Nielsen, Cees Niessen, Jens Sparso et Kees van Berkel: *Low-Power Operation Using Self-Timed Circuits and Adaptive Scaling of the Supply Voltage*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems - Special issue on low-power design, 2 :391–397, 1994.
- [OFMA11] Florent Ouchet, Laurent Fesquet et Katell Morin-Allory: *C-elements for hardened self-timed circuits*. Dans *Proceedings of the 21th International Workshop on Power And Timing Modeling, Optimization and Simulation (PATMOS)*, 2011.
- [OMAF10] Florent Ouchet, Katell Morin-Allory et Laurent Fesquet: *Delay insensitivity does not mean slope insensitivity!* Dans *Proceedings of the 16th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2010.
- [Pap05] Karl Papadantonakis: *Design Rules for Non-Atomic Implementations of Production Rule Set*. rapport technique, California Institute of Technology, 2005. http://www.async.caltech.edu/Pubs/PDF/2005_karl_tr.pdf.
- [PB61] W. W. Peterson et D. T. Brown: *Cyclic Codes for Error Detection*. Dans *Proceedings of the IRE*, 1961.
- [Pig91] Christian Piguet: *Logic synthesis of race-free asynchronous CMOS circuits*. IEEE Journal of Solid-State Circuits, 26 :371–380, 1991.
- [Pig98] Christian Piguet: *Synthesis of Asynchronous CMOS Basic Sequential Cells with Negative Gates*. Dans *Proceedings of the ACiD Workshop, Torino, January 26-27*, 1998.
- [PN95] Stanislaw J. Piestrak et Takashi Nanya: *Towards totally self-checking delay-insensitive systems*. Dans *Digest of Papers of the 25th International Symposium on Fault-Tolerant Computing (FTCS)*, 1995.
- [PSR04] Dhanistha Panyasak, Gilles Sicard et Marc Renaudin: *A current shaping methodology for lowering EM disturbances in asynchronous circuits*. Microelectronics Journal, June : 35(6) :531–540, 2004.
- [Rez04] Amine Rezzag: *Synthèse Logique de Circuits Asynchrones Micropipelines*. Thèse de doctorat, INP-Grenoble, 2004.
- [Rig02] Jean Baptiste Rigaud: *Spécification de bibliothèques pour la synthèse de circuits asynchrones*. Thèse de doctorat, Institut National Polytechnique de Grenoble, 2002.
- [RSMF05] Abbas Rahimi, Mostafa E. Salehi, Siamak Mohammadi et Sied Mehdi Fakhraie: *Dynamic Voltage Scaling for Fully Asynchronous NoCs Using FIFO Threshold Levels*. Dans *Proceedings of the 15th CSI International Symposium on Computer Architecture and Digital Systems (CADSD)*, 2005.
- [RVR98] Marc Renaudin, Pascal Vivet et Frédéric Robin: *ASPRO-216 : A Standard-Cell Q.D.I. 16-Bit RISC Asynchronous Microprocessor*. Dans *Proceedings*
-

- of the 4th International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC), 1998.
- [SB05] Nikolai Starodoubtsev et Sergei Bystrov: *Behavior and Synthesis of Two-Input-Gate Asynchronous Circuits*. Dans *Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2005.
- [SDM94] Polly Siegel et Giovanni De Micheli: *Decomposition Methods for Library Binding of Speed-Independent Asynchronous Designs*. Dans *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 1994.
- [SEE97] Maitham Shams, Jo C. Ebergen et Mohamed I. Elmasry: *Optimizing CMOS Implementations of the C-element*. Dans *Proceedings of the IEEE International Conference on Computer Design : VLSI in Computers and Processors (ICCD)*, 1997.
- [Shi05] Tatsuya Shimoda: *Introduction of TFT R&D Activities in Seiko Epson Corporation*, 2005. http://www.holtronic.ch/White_papers/SE2005_1.pdf.
- [Shi09] Suhas V. Shinde: *Muller-c element*, 2009.
- [Sho94] Martin L. Shooman: *A Study of Occurrence Rates of Electromagnetic Interference (EMI) to Aircraft With a Focus on HIRF (External) High Intensity Radiated Fields*. rapport technique, NASA, 1994.
- [SRH05] David Snowdon, Sergio Ruocco et Gernot Heiser: *Power Management and Dynamic Voltage Scaling : Myths and Facts*. Dans *Proceedings of the 2nd International Workshop on Power-aware Real-time Computing (PARC)*, 2005.
- [Sut89] Ivan Edward Sutherland: *Micropipelines*. Communications of the ACM, 32 :720–738, 1989.
- [Syn10] Synopsys: *DesignWare GTECH Library Databook*. rapport technique, Synopsys Inc., 2010.
- [TEC] *Technology Threshold Voltages*. http://www.interfacebus.com/voltage_threshold.html.
- [TIE] *TIEMPO : Asynchronous Circuit Compiler (ACC)*. <http://www.tiempo-ic.com/products/acc.html>.
- [Tie95] Jose Andres Tierno: *An energy-complexity model for VLSI computations*. Thèse de doctorat, California Institute of Technology, 1995.
- [TNW96] Michael Theobald, Steven M. Nowick et Tao Wu: *Espresso-HF : A Heuristic Hazard-Free Minimizer for Two-Level Logic*. Dans *Proceedings of the 33rd Design Automation Conference*, 1996.
- [Ung83] Stephen H. Unger: *Asynchronous Sequential Switching Circuit*. Wiley, 1983.
- [vB92] Kees van Berkel: *Beware the isochronic fork*. Integration, the VLSI Journal, 13 :103–128, 1992.

- [VSB10] Santosh N. Varanasi, Kenneth S. Stevens et Graham Birtwistle: *Concurrency Reduction of Untimed Latch Protocols – Theory and Practice*. Dans *Proceedings of the 16th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2010.
- [WV93] Tzyh Yung Wu et Sarma B. K. Vrudhula: *A Design of a Fast and Area Efficient Multi-Input Muller C-element*. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1 :215–219, 1993.
- [Yah09] Eslam Yahya: *Modélisation, Analyse et Optimisation des Performances des Circuits Asynchrones Multi-Protocoles*. Thèse de doctorat, Institut polytechnique de Grenoble, 2009.
- [Yan] Chao Yan: *openCoho : reachability analysis platform for high-dimensional, nonlinear hybrid systems*. <http://sourceforge.net/projects/coho/>.
- [Yan06] Chao Yan: *Coho : A Verification Tool for Circuit Verification by Reachability Analysis*. rapport technique, The University of British Columbia, 2006.
- [YD92] Kenneth Y. Yun et David L. Dill: *Automatic synthesis of 3D asynchronous state machines*. Dans *Proceedings of the IEEE/ACM international conference on Computer-aided design (ICCAD)*, 1992.
- [YG07] Chao Yan et Mark Greenstreet: *Circuit Level Verification of a High-Speed Toggle*. Dans *Proceedings of Formal Methods in Computer Aided Design, FMCAD '07*, 2007.
- [YG08a] Chao Yan et Mark Greenstreet: *Faster projection based methods for circuit level verification*. Dans *Proceedings of Asia and South Pacific Design Automation Conference, ASPDAC 2008*, 2008.
- [YG08b] Chao Yan et Mark Greenstreet: *Verifying an Arbiter Circuit*. Dans *Proceedings of Formal Methods in Computer-Aided Design, FMCAD '08*, 2008.
- [YOFMA11] Chao Yan, Florent Ouchet, Laurent Fesquet et Katell Morin-Allory: *Formal Verification of C-element Circuits*. Dans *Proceedings of the 17th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2011.
- [YRL08] Eslam Yahya, Marc Renaudin et Gregory Lopin: *Standard-Logic Quasi Delay Insensitive Registers*. Dans *Proceedings of the 16th IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2008.
- [YS10] Steven Yancey et Scott C. Smith: *A Differential Design for C-elements and NCL Gates*. Dans *Proceedings of the 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2010.

Résumé – La conception de circuits intégrés asynchrones, notamment de circuits QDI (Quasi-Delay Insensitive), offrent la possibilité de disposer de circuits très robustes aux conditions environnementales (tension, température) ainsi qu’aux variations des procédés de fabrication. Ces bonnes propriétés sont dues à une conception ne comportant pas d’hypothèses temporelles à l’exception de la fourche isochrone –hypothèse finalement très faible. Ainsi, une variation de la tension se traduit par une réduction de la vitesse de fonctionnement sans pour autant altérer la fonctionnalité du circuit.

Cette thèse étudie la robustesse des circuits asynchrones dans des environnements de fonctionnement très sévères susceptibles de mettre en défaut la correction fonctionnelle des circuits asynchrones QDI. Cette situation se présente par exemple quand les transitions des signaux sur les portes deviennent très lentes. Cette situation exceptionnelle peut-être directement provoquée par un environnement agressif (émission électromagnétique, particules à haute énergie, ...) ou par les effets du vieillissement du circuit intégré. Dans un contexte où le circuit est employé à des fins sécuritaires telles que les applications aéronautiques, spatiales ou médicales, il s’avère nécessaire de quantifier les limites de fonctionnement des circuits asynchrones et de trouver des moyens pour améliorer leur robustesse. Ce manuscrit propose une étude complète du comportement des circuits asynchrones et propose des techniques de conception pour en améliorer la robustesse. Les résultats obtenus ont été validés sur des technologies CMOS avancées de ST Microelectronics par des simulations analogiques d’une part, et avec l’aide d’un outil de preuve formelle développé à l’Université British Columbia au Canada d’autre part.

Mots Clefs : Asynchrone, Robustesse, Portes Logiques.

ROBUSTNESS ANALYSIS AND IMPROVEMENT OF QDI SELF-TIMED CIRCUITS

Abstract – The design of self-timed integrated circuits, including QDI (Quasi-Delay Insensitive) circuits, lead to robust circuits against variabilities in manufacturing processes and in running conditions (voltage, temperature). These qualities are consequences of the synthesis flow that does not create timing assumptions excepted a weak one related to isochronic forks. In self-timed circuits, the running speed automatically adjusts to the available supply voltage with no behavioral changes.

This work focuses on the self-timed circuit robustness in the context of environments where running conditions can make QDI self-timed circuits failing. For instance, this happens when transition speeds at gate entrances become very slow. This uncommonly encountered situation can be triggered in harsh environments (with electromagnetic disturbances, high-energy particles...) or because of age effects on manufactured chips. If the integrated circuit is designed for critical operations such as in aeronautical, spatial or medical applications, the self-timed circuit limits have to be carefully evaluated and eventually shifted in order to improve the circuit robustness. This publication includes a complete study of the self-timed circuit behaviors and some design proposals in order to enhance the circuit robustness. Experimental results were obtained firstly, during analog simulations targeting advanced CMOS technologies from STMicroelectronics and secondly, using formal methods implemented in a tool from the University of British Columbia.

Keywords : Asynchronous, Self-Timed, Robustness, Standard-Cells.

Adresse : Laboratoire TIMA, 46 Avenue Félix Viallet, 38031 Grenoble Cedex, France.

ISBN : 978-2-84813-182-5