Conception de dispositifs de contrôle asynchrones et distribués pour la gestion de l'énergie

Chadi Al Khatib

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES
Spécialité : **Nano-électronique et Nano-technologie**
Arrêté ministériel : 7 août 2006

Présentée par

« Chadi Al Khatib »

Thèse dirigée par **« Laurent Fesquet »** et
codirigée par **« Gilles Sicard »**

préparée au sein du **Laboratoire TIMA**
dans **l'École Doctorale EEATS**

Conception de dispositifs de contrôle asynchrones et distribués pour la gestion de l'énergie

Thèse soutenue publiquement le **01/03/2016**
devant le jury composé de :

M. Patrick Girard
Directeur de recherche CNRS, Université de Montpelier,Président
Mme. Nathalie Julien
Professeur, Université Bretagne-Sud, Rapporteur
M. Bruno Allard
Professeur, INSA de Lyon, Rapporteur
M. Laurent Fesquet
MdC-HDR, Grenoble INP, Directeur de thèse
M. Gilles Sicard
Ingénieur de recherche, CEA Letti, Co-directeur de thèse
M. Cyril Chevalier
Ingénieur, STMicroelectronics, Invité

Remerciements

Les travaux présentés dans cette thèse ont été effectués au sein du laboratoire TIMA (*Technique de l'Informatique et de la Microélectronique pour l'Architecture des systèmes intégrés*). Je commence par remercier la directrice de l'époque Dominique Borrione qui m'accueilli dans son établissement.

Je remercie mon directeur de thèse, M. Laurent Fesquet, maitre de conférences à l'Institut Polytechnique de Grenoble et chef du groupe CIS (*Concurrent Integrated Systems*), pour m'avoir accueilli pour cette thèse. Je le remercie pour sa confiance, son support technique, ses efforts et pour son soutien. Toute ma gratitude pour tout ce que tu as fait pour moi au niveau professionnel et personnel. Ce fût un honneur de travailler avec toi.

Egalement, je remercie mon co-directeur M. Gilles Sicard, maitre de conférences à l'université Joseph Fourier de Grenoble. Je te remercie de m'avoir encadré dans cette thèse, merci pour tes conseils conviviaux et pour tes encouragements.

Je tiens à remercier Alexandre Chagoya pour sa disponibilité et sa gentillesse. Excuse-moi Alexandre, je sais que je t'ai retenu plusieurs fois.

Toute ma gratitude pour M. Cyril Chevalier ingénieur à STMicroelectronics pour sa disponibilité et ses efforts.

Je ne veux oublier ma belle stagiaire Claire Aupetit. Un grand merci à toi et pour les efforts que tu as fait pendant ton stage parce que, pour moi, c'était comme le point de départ de la réussite dans cette thèse.

Je remercie vivement nos partenaires dans ce projet, Sylvian et Chouki. Merci de m'avoir fait confiance. C'était un grand honneur d'être avec vous dans ce projet.

Un grand merci pour mes collègues de TIMA et surtout de notre groupe CIS. Hassan, Amani, Abdelkarim, Jean, Leonel, c'était un grand plaisir de travailler à vos côtés. Je tiens à remercier aussi Alice De Bignicourt pour ses encouragements et son soutien pendant toute cette

période. Je remercie encore mon équipe de foot de TIMA, Marcos, Amin, Kays, Ismail, Martin, Francois, Rushdi… .

Je remercie maintenant mes collègues et mes amis libanais. Vous êtes nombreux et je ne peux pas tous vous compter ici. Un « grand merci » pour tous les membres de notre équipe de foot et pour les bons moments qu'on a passés ensemble. Merci pour votre aide et vos conseils.

Pour terminer, j'envoie des grands mercis à mes parents pour m'avoir toujours soutenu. Je remercie particulièrement ma mère et mon père. C'est grâce à eux je suis arrivé au bout de mon rêve de doctorat.

Enfin, je tiens à renouveler mes remerciements à tous et à présenter mes excuses aux personnes que j'aurais par hasard oubliées ici.

Merci à tous

Chadi Al Khatib

A ceux qui m'ont donné la vie, qui m'ont montré le chemin,

A mes parents,

A ceux à qui je dois tant,

Mes frères et mes sœurs

Résumé

Les systèmes intégrés sont aujourd'hui de plus en plus fréquemment confrontés à des contraintes de faible consommation ou d'efficacité énergétique. Ces problématiques se doivent d'être intégrées le plus en amont possible dans le flot de conception afin de réduire les temps de design et d'éviter de nombreuses itérations dans le flot. Dans ce contexte, le projet collaboratif HiCool, partenariat entre les laboratoires LIRMM et TIMA, les sociétés Defacto, Docea et ST Microelectronics, a mis en place une stratégie et un flot de conception pour concevoir des systèmes intégrés faible consommation tout en facilitant la réutilisation de blocs matériels (IPs) existants. L'approche proposée dans cette thèse s'intègre dans cette stratégie en apportant une petite dose d'asynchronisme dans des systèmes complètement synchrones. En effet, la réduction de la consommation est basée sur le constat que l'activation permanente de la totalité du circuit est inutile dans bien des cas. Néanmoins, contrôler l'activité avec des techniques de « *clock gating* » ou de « *power gating* » nécessitent usuellement d'effectuer un re-design du système et d'ajouter un organe de commande pour contrôler l'activation des zones effectuant un traitement. Le travail présenté dans ce manuscrit définit une stratégie basée sur des contrôleurs d'horloge et de domaine d'alimentation, asynchrones, distribués et facilement insérables dans un circuit avec un coût de re-design des plus réduit.

Mots clés: faible consommation, clock gating, power gating, contrôleurs asynchrones, contrôleurs distribués, flot de conception pour la faible consommation, systèmes synchrones.

Abstract

Today integrated system requirements are more and more dealing with low power and energy efficiency constraints. These issues have to be early addressed in the design flow in order to reduce design time and avoid as much as possible iterations. In the framework of the collaborative HiCool project - with the laboratories LIRMM and TIMA, and the companies, Defacto, Docea and ST Microelectronics - a design strategy and its associated design flow has been developed in order to enhance low power characteristics of integrated circuits and ease the reuse of existing hardware blocks (IPs). The proposed approach in this PhD thesis fits into this strategy by integrating a small amount of asynchrony in completely synchronous systems. Indeed, the power reduction is based on the observation that permanently stimulating the entire circuit is unnecessary in many cases. However, controlling the activity with techniques such as "*clock gating"* or "*power gating*" usually needs to redesign the system and to add circuitry for activating the blocks that really processing the data. The work, presented in this manuscript, provides a new strategy based on asynchronous distributed clock and power domain controllers that can easily be inserted at a very low redesign cost.

Keywords: low power, clock gating, power gating, asynchronous controllers, distributed controllers, low-power design flow, synchronous systems

TABLE DES MATIERES

INTRODUCTION GENERALE…….. 2

CHAPITRE 1…….. 8

ETAT DE L'ART DES CIRCUITS ASYNCHRONES…….. 8

1.1. Introduction....8
1.2. Les circuits synchrones....9
1.3. Le mode de fonctionnement asynchrone....9
1.4. Le principe fondamental : contrôle local....10
1.5. Caractéristiques d'un opérateur asynchrone....10
1.6. Protocoles de communication....11
1.7. Propriétés des circuits asynchrones....13
1.7.1. Calcul en temps minimum....13
1.7.2. Un pipeline élastique....13
1.7.3. Absence d'horloge....14
1.7.4. Modularité....14
1.7.5. Migration....15
1.8. La porte Muller....15
1.9. Classification des circuits asynchrones....16
1.9.1. Circuits insensibles aux délais (Delay Insensitive)....17
1.9.2. Circuits quasi insensibles aux délais (Quasi Delay Insensitive)....17
1.9.3. Circuits indépendants de la vitesse (Speed Independent)....18
1.9.4. Micropipeline....18
1.9.5. Les circuits de Huffman....21
1.10. Conclusion....21

CHAPITRE 2…….. 25

CARACTERISTIQUES DE LA STRUCTURE CIBLE……25

2.1. Introduction....25
2.2. Pipeline synchrone....25

2.3. Pipeline asynchrone et circuits micropipelines 26
2.4. Performances et contraintes d'un circuit micropipeline 27
2.5. Pipeline linéaire et non-linéaire 28
2.5.1. Pipeline linéaire 28
2.5.2. Pipeline non-linéaire 28
2.6. Etudes antérieures 31
2.7. Structure cible 32
2.8. La porte de Muller dissymétrique 33
2.8.1. La porte de Muller dissymétrique positive 34
2.8.2. La porte de Muller dissymétrique négative 34
2.8.3. La porte de Muller double dissymétrique (positive et négative) 35
2.9. Micropipelines linéaires 35
2.9.1. Protocole séquentiel 35
2.9.2. Structure WCHB (Weak Condition Half Buffer) 36
2.10. Micropipelines non-linéaires 38
2.10.1. Structure avec un protocole séquentiel 39
2.10.2. Structure WCHB 42
2.11. Conclusion 45

CHAPITRE 3 48

DES CONTROLEURS DISTRIBUES ASYNCHRONES POUR UNE NOUVELLE APPROCHE DU CLOCK GATING 48

3.1. Introduction 48
3.2. Sources de la consommation dynamique 49
3.2.1. Puissance de commutation (switching power) 50
3.2.2. Puissance due au court-circuit (Short-Circuit power) 50
3.3. Techniques de base de réduction de la consommation dynamique 51
3.3.1. Clock gating 52
3.3.2. DVFS (Dynamic Voltage and Frequency Scaling) 54
3.4. Contrôleur asynchrone d'horloge 55
3.4.1. Approche 55
3.4.2. Structure du contrôleur linéaire d'horloge 56
3.4.3. Structure du contrôleur non-linéaire d'horloge *(Split)* 57
3.5. Etude de l'approche sur un micro-processeur 58

3.5.1. Présentation 58
3.5.2. Simulation et estimation de la consommation 59
3.6. Conclusion 63

CHAPITRE 4 68

REDUIRE LA CONSOMMATION DYNAMIQUE AUTOUR D'UN SYSTEME DE COMMUNICATION 68

4.1. Introduction 68
4.2. Les bus AXI 69
4.2.1. Définition et caractéristiques 69
4.2.2. Communication dans un bus AXI 71
4.3. Insertion du contrôleur d'horloge 72
4.3.1. Insertion manuelle 72
4.3.2. Insertion automatique 74
4.4. Test de mémoires : SPRAM_0 et SPRAM_1 76
4.5. Test STMicroelectronics (circuit en technologie 28 nm FDSOI) 79
4.5.1. Présentation du circuit 79
4.5.2. Vérification de fonctionnement et estimation de la consommation 81
4.6. Conclusion 82

CHAPITRE 5 87

CONSOMMATION STATIQUE ET TECHNIQUES DE REDUCTION 87

5.1. Introduction 87
5.2. Source de leakage 87
5.2.1. Reverse-biased junction leakage current (IREV) 89
5.2.2. Gate induced drain leakage 89
5.2.3. Gate direct-tunneling leakage (IG) 89
5.2.4. Sub-threshold (weak inversion) leakage (ISUB) 90
5.3. La technologie FDSOI 28 nm (Fully Depleted Silicon On Insulator) 91
5.4. Techniques traditionnelles de réduction de la consommation statique 92
5.4.1. Multi-threshold 93
5.4.2. Power gating 94

5.4.3. Body biasing....95
5.5. Application des techniques de *power gating* et de *body biasing* pour réduire les courants de fuite....96
5.5.1. Structure d'isolation EP28SOI_VDDI_VDDISWITCH....96
5.5.2. Structure d'isolation EP28SOI_VDDI_VDDISWITCH_H....96
5.5.3. Structure d'isolation EP28SOI_SVDDI_VDDISWITCH....97
5.5.4. Dispositif intégré de distribution de l'énergie (Embedded Power Distribution)....97
5.5.5. Le contrôleur EPOD....98
5.6. .Structure de gestion de power gating....100
5.7. Structure de gestion de body biasing....101
5.7.1. BBMux (*Body Bias Multiplexer*)....103
5.7.2. BBGen (*Body Bias Generator*)....104
5.8. Circuit de test STMicroelectronics (FDSOI 28 nm)....104
5.9. Contrôle de la tension de polarisation V_{bb}....109
5.10. Conclusion....111

CONCLUSIONS ET PERSPECTIVES..... 114

Tables des figures

Figure 0-1: Flot du projet 3

Figure 1-1:Structure de base d'un circuit asynchrone 10
Figure 1-2:Principe de protocole deux phases 11
Figure 1-3: Principe de protocole 4 phases 12
Figure 1-4: La Porte Muller, implémentation et spécification 15
Figure 1-5: la porte Muller asymétrique 16
Figure 1-6: Terminologie des différentes classes des circuits asynchrones 16
Figure 1-7: Fourche isochrone si d2 et d3 sont identiques 17
Figure 1-8: Le modèle de pipeline 18
Figure 1-9: Structure micro-pipeline avec traitement et registres 19
Figure 1-10: Structure micro-pipeline indépendante de la vitesse 21

Figure 2-1: Pipeline Synchrone 26
Figure 2-2: Circuit micropipeline 26
Figure 2-3: Pipeline linéaire 28
Figure 2-4: La fourche 29
Figure 2-5: La jonction 29
Figure 2-6: Le multiplexage 30
Figure 2-7: Le démultiplexage 31
Figure 2-8: Modèle cible de micropipeline 32
Figure 2-9: Symbole d'une porte de Muller dissymétrique avec sa table de vérité 34
Figure 2-10: Symbole d'une porte de Muller dissymétrique négative avec sa table de vérité 34
Figure 2-11: Symbole d'une porte de Muller double dissymétrique avec sa table de vérité .. 35
Figure 2-12: Micro pipeline séquentiel d'une structure linéaire 36
Figure 2-13: Protocole WCHB d'une structure linéaire 37
Figure 2-14: Micropipeline séquentiel d'une fourche 39
Figure 2-15: Micropipeline séquentiel de la convergence 40

Figure 2-16: Micropipeline séquentiel d'un démultiplexeur 40
Figure 2-17: Micropipeline séquentiel en multiplexeur..... 41
Figure 2-18: Micropipeline de fourche en protocole WCHB 42
Figure 2-19: Micropipeline de convergence en protocole WCHB 43
Figure 2-20: Micropipeline d'un démultiplexeur en protocole WCHB..... 44
Figure 2-21: Micropipeline de multiplexeur en protocole WCHB 45

Figure 3-1: Consommations statique et dynamique en fonction de la technologie 49
Figure 3-2: Les différents niveaux d'abstraction utiles pour l'optimisation de la consommation d'énergie 51
Figure 3-3: Clock gating 53
Figure 3-4: Horloges distribuées pour un système synchrone 56
Figure 3-5: Structure du contrôleur d'horloge 57
Figure 3-6: Contrôleur d'horloge non-linéaire (Split)..... 58
Figure 3-7: Chaine de démodulation..... 58
Figure 3-8: Système processeur-périphériques 59
Figure 3-9: Système avec contrôleurs d'horloge linéaire et non-linéaire..... 60
Figure 3-10: Chronogrammes de sortie 60
Figure 3-11: Résultat de la consommation quand le système est complètement synchrone: 167 mA..... 61
Figure 3-12: Résultat de la consommation asynchrone avec le scénario 2: 48 mA 62
Figure 3-13: Résultat de la consommation asynchrone avec le scénario 1: 1.67 mA 62

Figure 4-1: Architecture des bus AXI..... 69
Figure 4-2: Vue général de la couche « Canal » dans le bus AXI 70
Figure 4-3: Protocole handshake dans un bus AXI 71
Figure 4-4: Transactions Lecture/Ecriture 72
Figure 4-5: Wrapper..... 72
Figure 4-6: Méthode d'insertion manuelle du contrôleur d'horloge 73

Figure 4-7: flot d'insertion automatique 74
Figure 4-8: Insertion automatique du contrôleur d'horloge dans un système à base du bus AXI 75
Figure 4-9: Circuit de test avec le bus AXI et deux mémoires 76
Figure 4-10: Le circuit de test après insertion des contrôleurs asynchrones 77
Figure 4- 11: Résultats de simulation sous Modeslim 78
Figure 4-12: Circuit de test avec 3 mémoires sur un bus AXI 80
Figure 4-13: Résultats de simulation du circuit de test 81

Figure 5-1: Sources de leakage 88
Figure 5-2: Vue schématique en coupe de transistors NMos et PMos fabriqués en technologie FDSOI 91
Figure 5-3: Original MTCMOS 93
Figure 5-4: Polarisation du substrat 95
Figure 5-5: Diagramme fonctionnel d'un EPOD 97
Figure 5-6: Contrôleur EPOD 98
Figure 5-7: Structure générique de gestion du power gating 100
Figure 5-8: Logique de génération des signaux Req-Ack 101
Figure 5-9: Structure de gestion de body biasing 102
Figure 5-10: Schéma de principe du block BBMux 103
Figure 5-11: Structure BBGEN 104
Figure 5-12: Circuit de test 105
Figure 5-13: Résultat de simulation 105
Figure 5-14: Flot d'insertion automatique 106
Figure 5-15: Circuit après insertion de la structure power gating 107
Figure 5-16: Résultats de simulation après l'insertion automatique 107
Figure 5-17: Exemple de lecture de données 108
Figure 5-18: Exemple d'écriture de données 108
Figure 5-19: Courant de fuite avant et après insertion de la structure de gestion du power gating 108

Figure 5-20: Circuit de test 109
Figure 5-21: Résultat de simulation (zoom) 110
Figure 5-22: Résultat de simulation 110

Introduction générale

Contexte

Le marché de la microélectronique est en perpétuelle évolution à la recherche de meilleures performances pour les circuits intégrés notamment en termes de consommation et d'efficacité énergétique. Les secteurs clés de l'industrie microélectronique comme l'appareillage nomade, le calcul intensif et les infrastructures de télécommunication sont soumis à des exigences grandissantes concernant la consommation d'énergie. La consommation de puissance est devenue en effet une contrainte et une spécification clé de la conception des systèmes électroniques. L'évolution de la technologie et l'augmentation de la fréquence d'horloge dans les systèmes embarqués induisent une consommation d'énergie croissante dans les systèmes numériques depuis quelques années. Les circuits industriels sont tous synchrones. Outre les inconvénients usuels, tels qu'émissions électromagnétiques dues aux harmoniques de l'horloge, bruit sur l'alimentation, hypothèses temporelles pire cas et consommation dynamique importante, le mode de synchronisation global ne permet pas de détecter la présence des données dans le circuit. En revanche, une synchronisation locale basée sur un protocole « à poignée de mains » ou « *Handshake* » le permet. Cette solution a été adoptée par les concepteurs de circuits dits « asynchrones » d'où l'idée de profiter de ces mécanismes pour contrôler la présence des données et donc l'activité « fonctionnelle » de telle ou telle autre bloc. L'objectif de cette thèse est donc de trouver des solutions architecturales simples, efficaces et industriellement compatibles qui, basées sur l'expertise en conception de circuit asynchrone du laboratoire, permettra d'améliorer les procédés usuels actuels de minimisation de la consommation électrique dans les circuits numériques complexes.

La consommation se divisant en deux contributions, dynamique et statique, il est normal que notre stratégie de détection de l'activité s'applique à ces deux composantes. La consommation dynamique est liée aux périodes d'activité ou de fonctionnement du circuit (charge et décharge des différentes capacités durant les phases de commutation) et la consommation statique est associée aux périodes d'inactivité ou de veille (principalement due aux courants de fuite des

transistors). L'optimisation de la puissance consommée peut se faire à différents niveaux dans le flot de conception des systèmes intégrés numériques en partant du niveau architectural et en allant jusqu'au niveau circuit. Des techniques, maintenant largement répandues, peuvent être appliqués au niveau circuit pour diminuer la consommation dynamique telle que l'ajustement de la tension d'alimentation (*Voltage Scaling*) ainsi que la technique de *clock gating*. En ce qui concerne la consommation statique, on pourra parler des techniques de *power gating* et *back biasing*.

Pour répondre à ces exigences nouvelles, les sociétés STMicroelectronics Grenoble, DOCEA et DEFACTO et les laboratoires TIMA et LIRMM se sont associées dans le cadre d'un projet commun nommé « HICOOL ». L'objectif de ce projet est d'améliorer le flot de conception des systèmes sur puce exigeant une forte maîtrise de leur consommation électrique. La Figure 0-1 suivante illustre le flot de conception des circuits intégrés développé dans le cadre de ce projet.

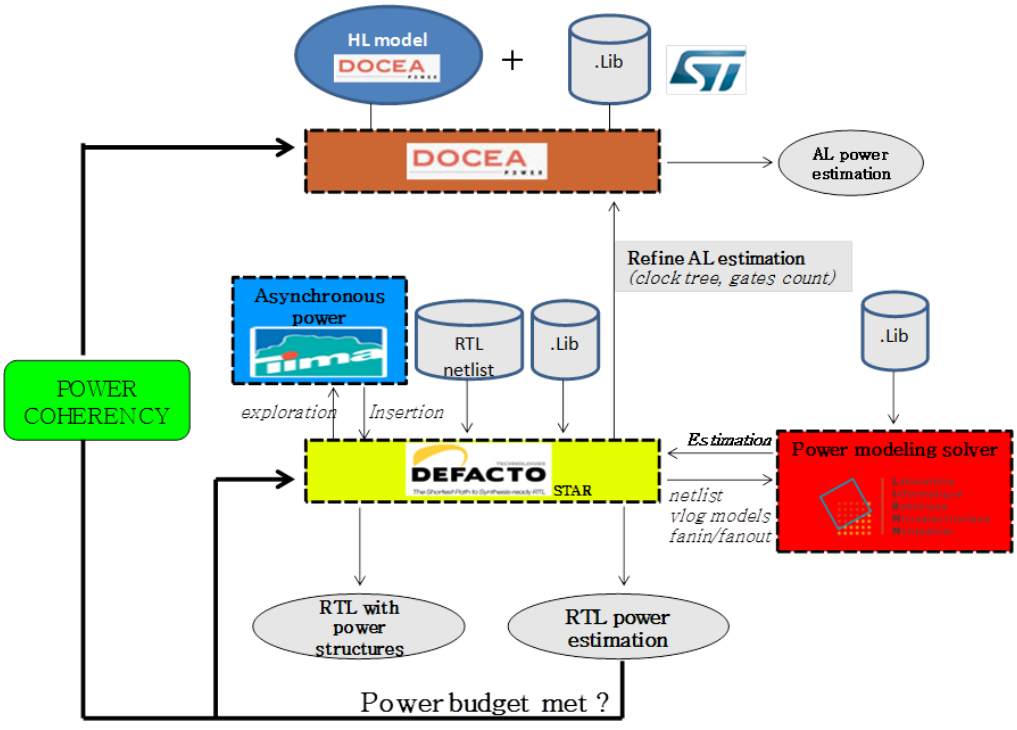


Figure 0-1: Flot du projet

Cette figure montre que ce flot de conception inclut de nouvelles approches pour modéliser et estimer la consommation d'énergie à différentes niveaux d'abstraction (AL, RTL et GL).

Au niveau architectural AL (*Architectural Level*), il est possible d'estimer directement la consommation d'énergie grâce aux bibliothèques (Lib) de STMicroelectronics et aux outils de DOCEA Power (devenu Intel depuis le 1er Janvier 2015). Il est également possible d'évaluer, à ce niveau, différents types d'architectures afin de définir les meilleures stratégies de gestion de la puissance. L'outil d'estimation de puissance exploite des fichiers UPF (*Unified Power Format*) et des *netlists* hiérarchiques.

Dans ce contexte, ce travail de thèse a permis de développer une bibliothèque de structures de contrôle asynchrones pour gérer les modes *low power* au niveau RTL (*Register Transfer Level*) ou GL (*Gate Level*). Ces structures asynchrones sont ensuite distribuées dans le circuit afin de piloter l'extinction de l'horloge sur certains blocs (*clock gating*) et des organes de gestion de la puissance comme des isolateurs (*power gating*) ou des dispositifs de *body biasing*. Des cellules asynchrones spécifiques ont été développées pour effectuer ce contrôle distribué exploitant des techniques comme le *clock gating*, le *power gating* et le *body biaising)* d'une part et à mettre en place un flot et une stratégie d'insertion automatique de ces cellules d'autres part.. La stratégie mise en place, basée sur une connaissance *a priori* du système de communication de la puce (bus ou NoC), nous permet d'insérer facilement et automatiquement les cellules asynchrones de contrôle mais aussi simplifie le travail du concepteur en évitant de longues phases de re-design du système. En effet, un bloc (IP) existant peut, avec la méthode développée, être activé que lorsqu'il a réellement un travail à effectuer sans qu'il soit nécessaire d'intervenir ni sur sa conception, ni sur la conception globale du système. Ce n'est typiquement pas le cas lorsqu'on décide de faire *clock gating* sur un bloc existant car il doit être resynthétisé avec les options permettant la génération du *clock gating*.

A partir des fichiers RTL et UPF, il est possible d'analyser l'arbre d'horloge pour déterminer où l'on peut insérer nos dispositifs de *clock gating,* de *power gating* et de *body biaising* mais aussi les canaux de communication intra-puces que nous allons utiliser pour mettre en œuvre notre stratégie *low power*. L'outil STAR de DeFacto Technologies permet d'insérer

automatiquement les structures « *low power* » aux endroits appropriés. Il analyse chemins d'horloge et réseaux de communication de la puce afin d'insérer selon une stratégie prédéfinie nos dispositifs basse consommation dans un code RTL. Une fois, les structures de contrôle asynchrones insérées, il est possible d'estimer la consommation avec les fichiers produits par l'outil STAR au niveau RTL ou GL (LIRMM).

Plusieurs familles d'applications pourront tirer profit des résultats de ce projet comme, par exemple, la téléphonie mobile. Les téléphones comptent aujourd'hui parmi les applications les plus exigeantes, à la fois en consommation et en performances (500 MHz-2GHz), tout en intégrant toujours plus de fonctionnalités.

Plan du manuscrit

Le chapitre 1 introduit un état de l'art sur les circuits asynchrones. Dans ce chapitre, on définit les principes de fonctionnement de ces circuits, qui sont basés sur une synchronisation locale, plutôt que globale comme c'est le cas en synchrone, afin de mettre en évidence les bonnes propriétés de cette logique (latence, temps de cycle et protocole de communication). Enfin, Une classification des circuits asynchrones (il y a, contrairement en synchrone, plusieurs classes de circuits asynchrone) en fonction des hypothèses temporelles est présentée à la fin de ce chapitre.

Le deuxième chapitre de ce manuscrit illustre les différents modèles de circuits « pipelinés » synchrones et asynchrones. Le modèle de circuits asynchrones que nous avons étudié plus en profondeur s'appelle micropipeline et nous avons analysé différentes configurations du micropipeline (linéaires et non-linéaires) avec plusieurs protocoles (séquentiel, WCHB, PCHB et PCFB). Les chapitres 3 et 4 présentent l'approche que nous avons retenue et démontrent l'impact positif des structures que nous générons en termes de consommation dynamique. Nous réduisons donc la consommation dynamique grâce au principe du *clock gating*. Dans ce modèle, on insère un bloc purement asynchrone au bloc de *clock gating*. L'ajout de cette interface asynchrone permet de gérer le dispositif de *clock gating* en fonction des données arrivant sur les blocs dont on souhaite minimiser la consommation dynamique. Des tests ont été effectués pour démontrer l'efficacité de cette structure d'abord sur un petit processeur et ensuite sur un véhicule

de test industriel implémenté en technologie FDSOI 28 nm et architecturé autour d'un bus AXI. Pour étendre notre étude, nous avons développé conjointement avec la société DEFACTO un algorithme qui permet d'insérer automatiquement nos structures dans des circuits architecturés autours d'un bus ou d'un NoC.

Enfin, le dernier chapitre de ce manuscrit se penche sur l'étude de la consommation statique et les techniques pour la minimiser. La technologie avancée rend les circuits de plus en plus complexes et consommant. Un certain nombre de techniques existent pour réduire ce type de consommation comme le *power gating* et le *back biasing*. L'approche, héritée de celle déployée pour la consommation dynamique, a été adaptée pour gérer et piloter des dispositifs de *power gating* et de *body biasing* sur un circuit de test implémenté en en technologie FDSOI 28 nm de STMicroelectronics.

Chapitre 1

Etat de l'art des circuits asynchrones

1.1. Introduction

Dans le cadre de la conception de circuits à basse consommation, le recours aux circuits asynchrones constituent une alternative aux circuits synchrones pour minimiser la consommation. En effet, les circuits synchrones ont des contraintes fortes de fonctionnement aux fréquences élevées liées à la distribution et à la propagation de l'horloge produisant du même coup une surconsommation [DAL 98].

Les circuits asynchrones sont apparus dans les années cinquante et ont été étudiés par D .E. Muller et D. A. Huffman. Muller a été le premier à proposer un protocole de communication spécifique à ce type de circuit [MUL 59]. Quant à Huffman, il fût le premier à concevoir une machine à état asynchrone [HUF 54].

En effet, l'évolution de la technologie rend les circuits intégrés de plus en plus complexes et l'augmentation du degré d'intégration rend le paradigme synchrone plus difficile à mettre en œuvre. L'approche asynchrone permet d'assouplir ces difficultés grâce, notamment, à sa modularité quasi-parfaite. Cette dernière, avec le contrôle local des synchronisations, permet une réutilisabilité des blocs asynchrones très faciles dans les systèmes complexes.

Ce chapitre décrit l'état de l'art des systèmes asynchrones, leurs principes de base, leurs caractéristiques et leurs propriétés ainsi que les principales différences entre les architectures synchrones et asynchrones en présentant les avantages et les inconvénients. L'idée de base des

opérateurs asynchrones est fondée sur des synchronisations locales. Un opérateur peut communiquer avec son environnement au travers de canaux de communications. L'échange d'information entre les blocs se fait par l'implémentation de protocoles de communication de type "poignée de main" dans les canaux.

1.2. Les circuits synchrones

Les systèmes microélectroniques sont usuellement des circuits synchrones. Ils sont adoptés depuis longtemps par l'industrie pour résoudre le problème de la synchronisation des communications dans une puce. A l'aide d'un signal global, l'horloge, les données sont transférées et échantillonnées périodiquement et en cadence dans la totalité du circuit. Grâce à la digitalisation des signaux et à la discrétisation du temps avec l'horloge, l'implémentation des circuits numériques est finalement simple à mettre en œuvre. C'est pour ces raisons qu'ils sont aujourd'hui quasiment tous synchrones et qu'ils ont été adoptés par l'industrie dans les années soixante au détriment de leurs homologues asynchrones.

Dans la suite de ce chapitre, on s'intéressera à la description des circuits asynchrones, à leurs caractéristiques et à leur fonctionnement.

1.3. Le mode de fonctionnement asynchrone

Le mode d'exécution des circuits synchrones où l'horloge joue le rôle d'un actionneur global et où toutes les données se propagent dans les parties combinatoires à chaque front montant, implique que tous les éléments doivent respecter un temps maximum d'exécution imposé par la fréquence d'horloge. Avec les circuits asynchrones, les relations temporelles entre les événements peuvent être relâchées, voire supprimées. Cela permet donc d'éliminer, au besoin, toute contrainte temporelle.

Dans les circuits asynchrones, chaque élément sera actif si et seulement si les données en ses entrées sont présentes. Ce type de fonctionnement est similaire à celui des systèmes dits "flot de données". Les blocs sont interconnectés entre eux par un canal de communication comme indique la Figure 1-1.

1.4. Le principe fondamental : contrôle local

Le principe de base des circuits asynchrones est que le transfert des informations se fait localement. Ce type de contrôle doit remplir les fonctions suivantes: être à l'écoute des communications entrantes, déclencher le traitement local si toutes les informations sont disponibles (rendez-vous) et produire les valeurs des sorties. Après un certain temps, un signal d'acquittement indique à l'émetteur que les données sont bien consommées. La Figure 1-1 illustre la structure de base d'un circuit asynchrone. Sur cette figure, on remarque la signalisation bidirectionnelle, qui permet un fonctionnement correct et indépendant du temps. Chaque évènement doit être acquitté par le récepteur pour que l'émetteur puisse envoyer à nouveau une donnée. Ce type de communication est appelée "à poignée de mains" ou requête-acquittement.

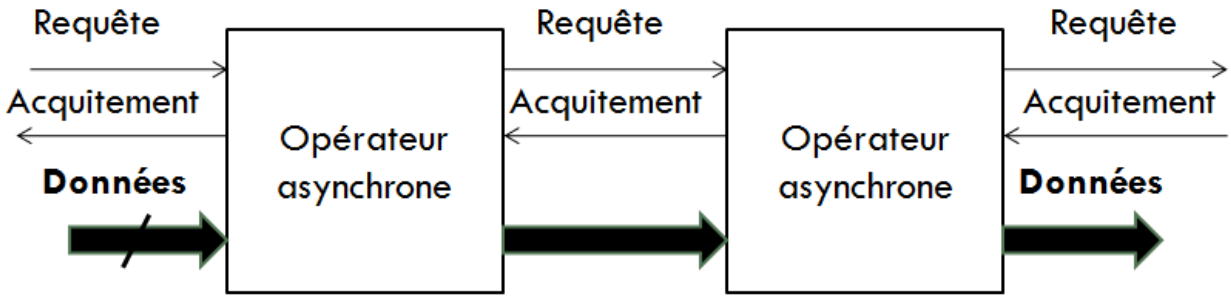


Figure 1-1:Structure de base d'un circuit asynchrone

1.5. Caractéristiques d'un opérateur asynchrone

Comme on peut l'observer sur la Figure 1-1, la communication entre les opérateurs se fait à partir des canaux de communications. Ces canaux peuvent échanger des données avec l'environnement ainsi que des informations de synchronisations.

En effet, un opérateur asynchrone peut se caractériser au moins par quatre paramètres fondamentaux.

-**Temps de latence:** C'est le temps nécessaire à une donnée pour traverser le chemin combinatoire la plus long avant d'être présentée en sortie. Notons ici que le temps de latence

dépend du chemin combinatoire suivi dans l'opérateur et qu'il peut donc être variable en fonction des données en entrées.

-**Temps de cycle:** C'est le temps minimal qui sépare l'acceptation de deux informations en entrée. En général, c'est le temps nécessaire pour échanger une donnée entre deux ressources de mémorisation connectées.

-**Profondeur du pipeline:** C'est le nombre maximum d'informations que l'opérateur peut mémoriser.

-**Protocole de communication:** La caractéristique fondamentale d'un opérateur asynchrone est le protocole de communication qui est utilisé pour échanger des données avec son environnement. On décrira ces protocoles ultérieurement. Il est donc important de spécifier le protocole utilisé.

1.6. Protocoles de communication

Pour que l'échange entre deux opérateurs ait lieu, deux classes principales de protocoles de communication sont utilisés : le protocole 2 phases (ou NRZ pour Non-Retour à Zéro ou encore "Half-Handshake") et le protocole 4 phases (ou RZ pour Retour à Zéro ou "Full-Handshake").

La Figure 1-2 décrit le fonctionnement du protocole 2 phases. Ce protocole est donc composé de deux phases de fonctionnement :

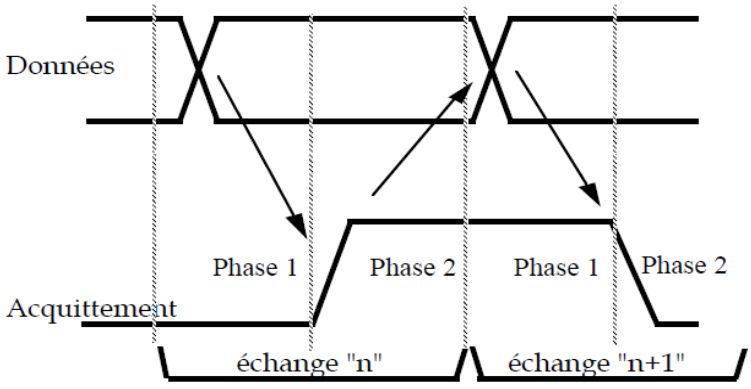


Figure 1-2:Principe de protocole deux phases

Phase 1: c'est la phase active du récepteur qui détecte la présence de nouvelles données, il effectue le traitement et génère le signal d'acquittement.

Phase 2: c'est la phase active de l'émetteur qui détecte le signal d'acquittement et émet les nouvelles données si elles sont disponibles.

La Figure 1-3 décrit le principe du protocole 4 phases. On utilise donc ici quatre phases de fonctionnement :

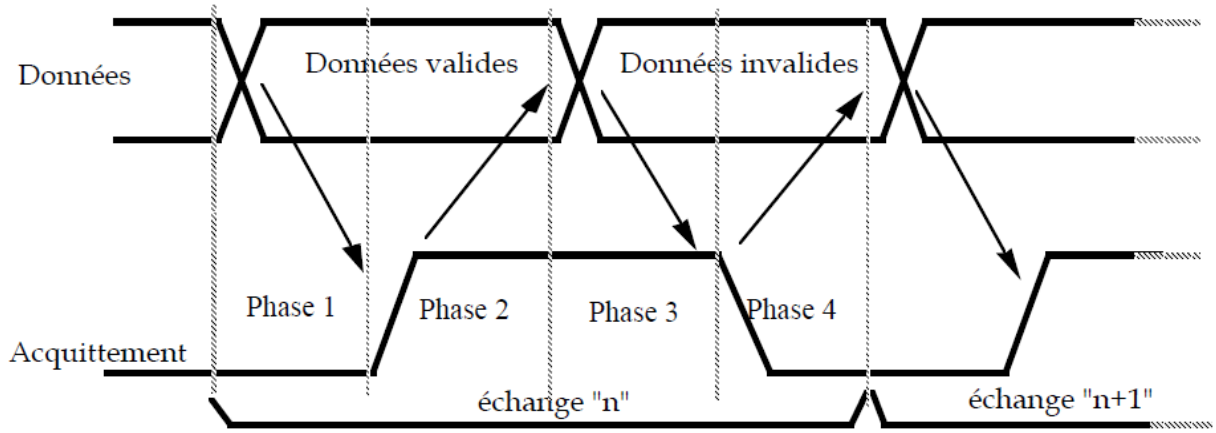


Figure 1-3: Principe de protocole 4 phases

Phase 1: c'est la première phase active du récepteur qui détecte la présence de nouvelles données, il effectue le traitement et génère le signal d'acquittement.

Phase 2: c'est la première phase active de l'émetteur qui détecte le signal d'acquittement et émet des données invalides (retour à zéro du signal requête).

Phase 3: c'est la deuxième phase active de récepteur qui détecte le passage des données dans l'état invalide et remet le signal d'acquittement à sa valeur initiale.

Phase 4: c'est la deuxième phase active de l'émetteur qui détecte le retour de signal acquittement et remet des nouvelles données.

Par comparaison entre les deux protocoles, on remarque que le protocole 4 phases nécessite deux fois plus de transitions que le protocole 2 phases. Il est, a *priori,* plus lent. Le protocole 2 phases, quant à lui, requiert un matériel plus complexe parce qu'il détecte des transitions et non

des niveaux. Le gain de consommation, que l'on pourrait espérer du fait de la réduction du nombre des transitions avec un protocole deux phases, est en fait largement annihilé par la complexité du matériel nécessaire à ce protocole. En pratique, le protocole 4 phases est souvent adopté car l'implémentation d'un protocole deux phases est plus coûteuse en surface.

En conclusion, le protocole 4 phases est le plus utilisé pour implémenter les parties internes d'un circuit intégré. Par contre, lorsque les signaux doivent traverser des éléments possédant une latence plus élevée, comme les plots par exemple, le recours à un protocole deux phases est adopté [REN 98].

1.7. Propriétés des circuits asynchrones

1.7.1. Calcul en temps minimum

La première conséquence du fonctionnement flot de données permis par les circuits asynchrones, est qu'un opérateur peut évaluer une fonction en un temps variable qui correspond au passage des données des entrées vers les sorties. Le temps et le chemin empruntés par les données peuvent varier en fonction des données elles-mêmes.

En ce qui concerne les caractéristiques de vitesse d'un circuit, elles varient en fonction des paramètres qui influencent le fonctionnement des dispositifs élémentaires, telles que la variation de la température, les paramètres technologiques ou la tension d'alimentation. En revanche, le fonctionnement flot de données des circuits asynchrones les rend très robustes vis-à-vis de ces variations.

1.7.2. Un pipeline élastique

Un pipeline élastique signifie que le nombre de données présentes dans la structure est variable contrairement à un pipeline inélastique où ce nombre est fixe. En synchrone, c'est l'arrivée d'un front montant de l'horloge qui provoque la progression des données dans les registres. Dans un pipeline asynchrone, les registres se comportent comme une file "FIFO". Les

données se propagent tant qu'elles ne trouvent pas de ressources occupées. Cette propriété peut être exploitée au profit de la vitesse et/ou de la consommation.

1.7.3. Absence d'horloge

L'absence d'horloge est l'une des principales propriétés des circuits asynchrones. Tous les problèmes liés à l'usage de l'horloge sont alors supprimés. En effet, les limitations des circuits à horloge ont redonné ces dernières années un regain d'intérêt pour les circuits asynchrones. Comme cela a déjà été expliqué, le principe de base des circuits asynchrones est le contrôle local. On relâche donc l'ensemble des contraintes liées à une synchronisation globale, telle que la détermination du chemin critique qui contraint la fréquence de fonctionnement du circuit.

Les blocs calculent uniquement s'il apparaît des données à ses entrées et les éléments de synchronisation sont distribués dans l'ensemble du circuit. Leur conception est ainsi beaucoup plus aisée.

Un avantage de ce type de communications est que les pics de consommation sont distribués temporellement. En effet, l'activité électrique des circuits asynchrones est mieux répartie dans le temps que celle d'un circuit synchrone car l'avancement des données n'est plus lié aux fronts d'une horloge.

1.7.4. Modularité

Un système asynchrone possède une excellente modularité due au contrôle local. Cette modularité permet d'augmenter le degré d'intégration des circuits complexes. Elle simplifie en outre la conception de parties séparées d'un circuit par différentes équipes de concepteur. Dans les systèmes synchrones, la modularité est plus faible car il est difficile d'ajouter ou de remplacer un composant sans que cela ait une incidence sur le fonctionnement global.

Ces propriétés sont donc particulièrement intéressantes lorsqu'on souhaite favoriser la réutilisation de blocs au sein d'une entreprise ou d'un point de vue plus général, lors de l'utilisation de blocs de propriétés intellectuelles (IPs).

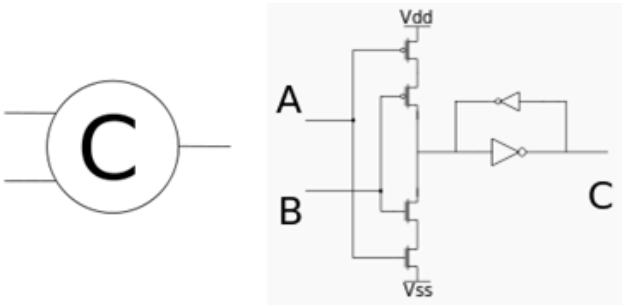
1.7.5. Migration

Parmi les atouts des circuits asynchrones, on peut lister : la faible consommation, le faible rayonnement électromagnétique [BOU 04], et la robustesse aux variations PVT.

Les circuits asynchrones se prêtent également assez facilement aux migrations technologiques, les rendant aussi plus attractifs. En effet, le comportement fonctionnel d'un circuit asynchrone peut être indépendant de la réalisation des blocs qui le constituent ou du nœud technologique pourvu que le protocole de communication soit respecté. Ainsi, il est possible de modifier l'implémentation ou la technologie des cellules de base sans modifier la fonctionnalité du circuit.

1.8. La porte Muller

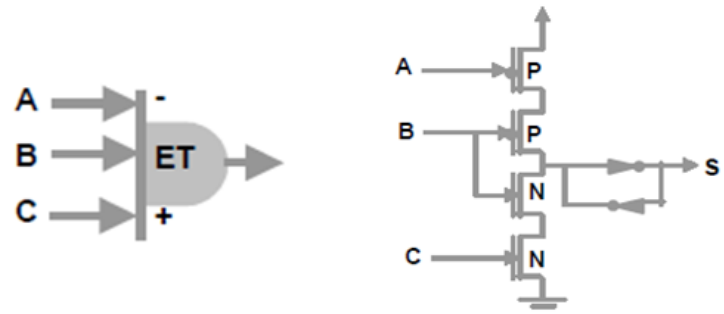
La porte Muller ou "C-element" permet d'implémenter le protocole de communication des circuits asynchrones. Elle a été introduite par D.E. Muller [MUL 59]. Cette porte est formée de deux entrées A et B et d'une sortie C. Elle assure un rendez-vous entre les signaux d'entrées ; elle copie la valeur de ses entrées lorsqu'elles sont identiques, sinon elle mémorise la valeur précédemment acquise. La Figure 1-4 montre la porte Muller avec sa spécification.



A	B	C
0	0	0
0	1	C^{-1}
1	0	C^{-1}
1	1	1

Figure 1-4: La Porte Muller, implémentation et spécification

Une version étendue de la porte Muller est la porte de Muller asymétrique généralisée [BAP 02]. Par exemple pour une porte à trois entrées, elle est formée de trois entrées A, B et C et d'une sortie S. La Figure 1-5 décrit cette porte et son comportement.



A	B	C	S
0	0	S^{-1}	0
S^{-1}	1	1	1

Figure 1-5: la porte Muller asymétrique

1.9. Classification des circuits asynchrones

Il existe différentes classes de circuits asynchrones différenciés en fonction de leurs hypothèses temporelles. La Figure 1-6 illustre ces types de circuits en notant que le fonctionnement respecte la notion d'asynchronisme telle que nous l'avons définie.

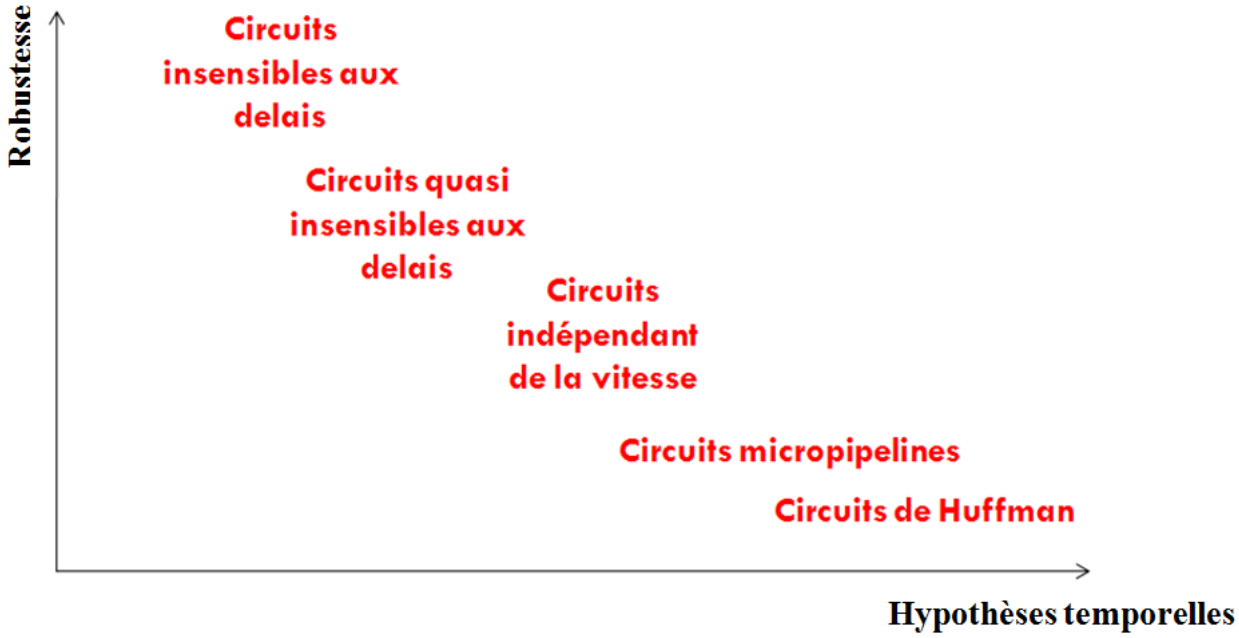


Figure 1-6: Terminologie des différentes classes des circuits asynchrones

1.9.1. Circuits insensibles aux délais (Delay Insensitive)

Les circuits insensibles aux délais sont des circuits qui ne font aucune hypothèse temporelle. Ils sont purement asynchrones et utilisent un modèle de délai non-borné. Ils fonctionnent indépendamment des délais introduits par les fils ou les éléments logiques. Ce type de circuit est très robuste vis-à-vis des variations de température, de tension ou des procédés de fabrication. Le point négatif est que ces circuits ne permettent pas d'utiliser les portes logiques standards mais imposent des portes complexes plusieurs sorties [REN 02].

1.9.2. Circuits quasi insensibles aux délais (Quasi Delay Insensitive)

Cette classe de circuits utilise aussi un modèle de délai non-borné ainsi qu'une hypothèse dite de "fourche isochrone". Une fourche est un fil qui connecte un émetteur à deux récepteurs. Elle est qualifiée d'isochrone si les délais entre l'émetteur et les deux récepteurs sont identiques, c.à.d qu'un signal issus de l'émetteur doit arriver à l'entrée de deux récepteurs en même temps.

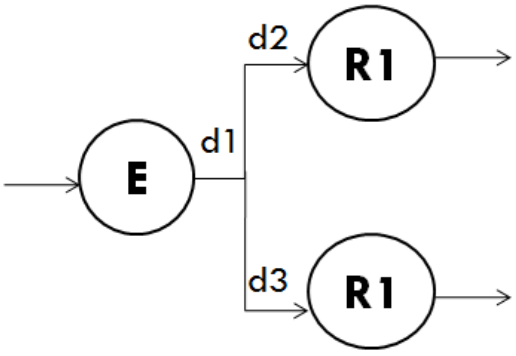


Figure 1-7: Fourche isochrone si d2 et d3 sont identiques

La Figure 1-7 présente la structure d'une fourche isochrone. Cette hypothèse permet d'utiliser des portes logiques à une seule sortie, ce qui permet d'utiliser un flot de conception plus standard. L'utilisation d'un modèle de délais non-bornés avec des portes logiques plus simples à une seule sortie associé à l'hypothèse de fourches isochrones rendent les circuits quasi-insensibles aux délais implémentables avec des cellules standard telles qu'on les utilise pour la conception de circuits synchrones. Par ailleurs, cette hypothèse de fourches isochrones peut

sembler très forte mais ce n'est absolument pas le cas en pratique. Enfin, les fourches isochrones (toutes les fourches n'imposent pas l'isochronisme) peuvent être marquées afin de vérifier leur comportement post-placement et routage.

1.9.3. Circuits indépendants de la vitesse (Speed Independent)

Les circuits indépendants de la vitesse ne tiennent pas compte des délais dans les fils. Ils les considèrent comme négligeables et conservent un modèle de délais non-bornés dans les portes ce qui implique que toutes les fourches dans les circuits sont isochrones [HAU 95].

1.9.4. Micropipeline

Ivan Sutherland [SUT 89] fut le premier à introduire les circuits Micropipeline. Les circuits de cette classe sont composés de parties de contrôle insensibles aux délais qui commandent des chemins de données conçus en utilisant un chemin de donnés identique à un circuit synchrone. La structure de base de cette classe de circuits est le contrôle d'une file (FIFO). Elle se compose d'éléments identiques connectés tête-bêche. Les opérateurs notés « C » sont des portes de Muller dont la sortie est une copie des niveaux d'entrée lorsqu'ils sont identiques, et une copie du niveau de sortie précédent lorsque les entrées sont différentes.

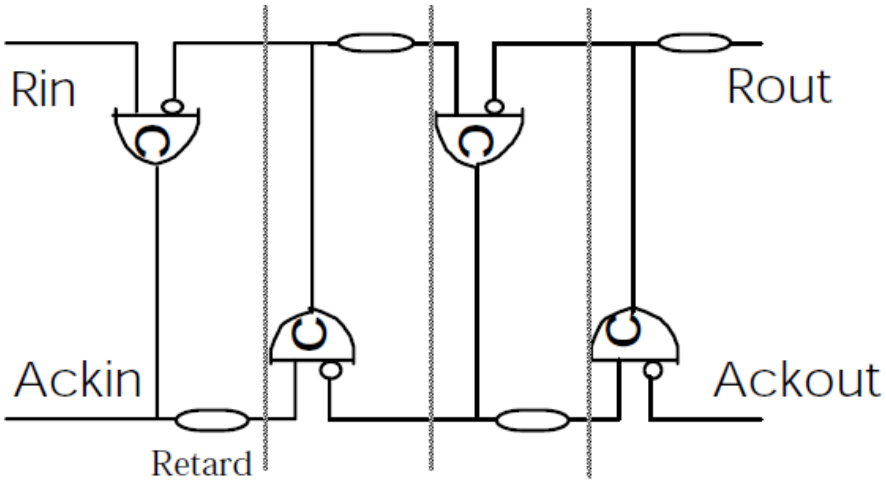


Figure 1-8: Le modèle de pipeline

Le circuit réagit à des transitions de signaux et non pas à des états (protocole deux phases). On parle également de logique à évènements. Chaque transition étant associée à un évènement. Ainsi, tous les signaux sont supposés à '0' initialement. Une transition positive sur Rin provoque une transition positive sur Ackin qui se propage également à l'étage suivant. Le deuxième étage produit une transition positive qui d'une part, se propage à l'étage suivant et qui d'autre part, revient au premier étage, l'autorisant à traiter une transition négative cette fois. Les transitions de signaux se propagent donc dans la structure tant qu'elles ne rencontrent pas une cellule « occupée ». C'est un fonctionnement de type FIFO.

Cette structure peut être enrichie d'opérateurs de mémorisation et d'opérateurs de traitement combinatoires [SUTH 89]. La Figure 1-9 illustre un micro pipeline avec traitement.

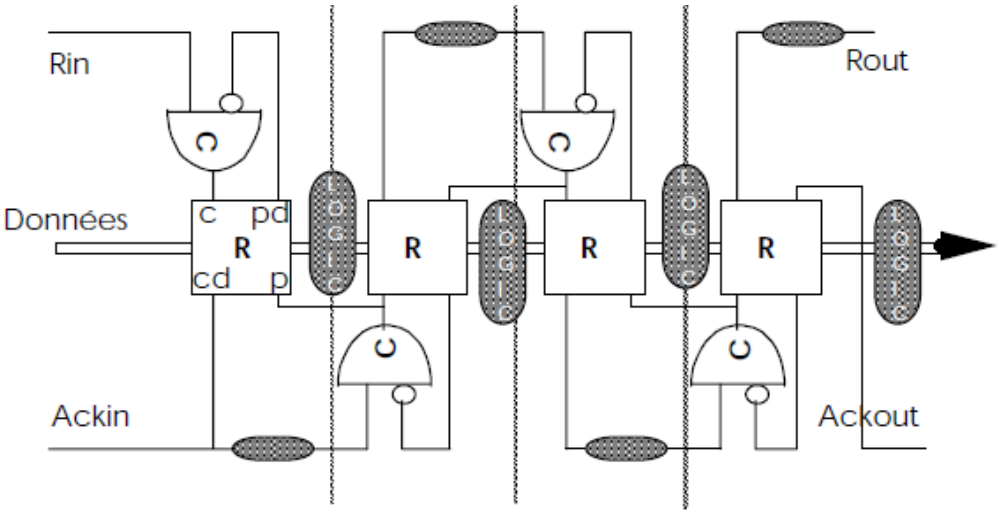


Figure 1-9: Structure micro-pipeline avec traitement et registres

Dans cette structure, les opérateurs dénotés « R » sont des registres qui capturent la donnée entrante sur l'occurrence du signal C. Ils produisent le signal « CD » lorsque la donnée est mémorisée. En pratique, CD est une version retardée du signal C. Durant cette phase, la donnée précédemment mémorisée dans le registre est maintenue en sortie. Lorsque P est actif, le registre laisse passer la donnée d'entrée à la sortie. Le signal « Pd » signale que le registre est bien transparent. La structure de la figure ci-dessus, dépouillée de la logique combinatoire, réalise une

FIFO (initialement les registres laissent passer les données). Avec la logique, la structure est celle d'un circuit asynchrone « pipeliné ».

Le protocole de communication utilisé ici, est de type « données groupées » ([SUTH 89]). Lors d'une occurrence de Rin, les données d'entrée sont stockées dans le premier étage. Rin est propagé vers l'étage suivant qui conserve le résultat transmis par la logique du premier étage. La propagation de Rin est retardée de façon à s'assurer que la logique combinatoire a bien convergé avant la capture du résultat dans le deuxième étage. La capture du deuxième étage étant effectuée, le premier registre est rendu passant ce qui permet le traitement de la donnée présente dans le premier étage et autorise la prise en compte d'un nouvel évènement sur Rin du premier étage.

La motivation première pour le développement de cette classe de circuits était de permettre un pipeline élastique. En effet, le nombre de données présentes dans le circuit peut être variable, les données progressant dans le circuit aussi loin que possible en fonction du nombre d'étages disponibles ou vides. Cependant, ce type de circuits révèle un certain nombre d'inconvénients.

Tout d'abord, il faut remarquer que les problèmes d'aléas ont été écartés en ajoutant des retards sur les signaux de contrôle. Cela permet en fait de se ramener à un fonctionnement en temps discret dans lequel il est autorisé la mémorisation des données seulement lorsqu'elles sont stables (à la sortie des portes combinatoires). Les délais étant de durée fixe dans la proposition initiale de Sutherland, ces circuits effectuent les traitements en pire cas. Il n'est donc pas possible de tirer parti de la variation dynamique de la chaîne critique des opérateurs de traitement.

Il est possible cependant de s'affranchir de cette contrainte en utilisant des registres et des opérateurs combinatoires capables de générer leur propre signal de fin de calcul sans avoir recours à des délais fixes. Alors il est obtenu des structures du type de celles implémentées sur la Figure 1-10 dans laquelle les délais fixes sont remplacés par des délais variables implémentés dans les registres et la logique combinatoire. Ici, le temps de calcul peut varier en fonction de données.

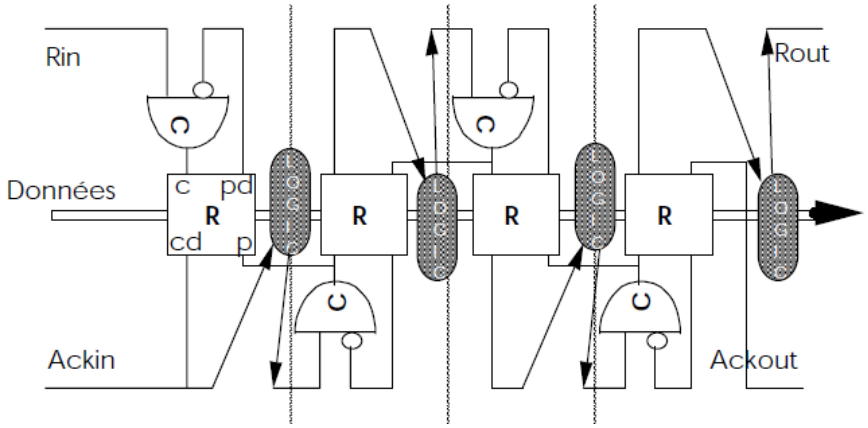


Figure 1-10: Structure micro-pipeline indépendante de la vitesse

1.9.5. Les circuits de Huffman

En se référant à la Figure 1-6, les circuits de Huffman sont les moins robustes. Ils supposent que les délais dans toutes les connexions et éléments du circuit sont bornés et de valeurs connues. Il y a donc de très nombreuses hypothèses temporelles locales. En conclusion, ce type des circuits sont les circuits asynchrones les plus difficiles à tester. L'ajout de redondance et de retards rend certains nœuds du circuit difficilement observables ou pas observables, et oblige à tester les fautes de délais [BEE 91] [LAV 93a] [KIS 98] [LAV 94] [NOW 97]

1.10. Conclusion

Dans ce chapitre, on a présenté les systèmes numériques synchrones et asynchrones ainsi que les concepts de base de la conception des circuits asynchrones. On a également décrit les différentes caractéristiques des systèmes asynchrones ainsi que leur principe de fonctionnement. Les différentes classes de circuits asynchrones ont été présentées. Parmi ces classes de circuits, il est à noter que les circuits quasi insensibles aux délais sont des circuits facilement implémentables car ils font un minimum d'hypothèses temporelles. Les circuits micropipelines sont eux aussi très attractifs car ils ressemblent fortement aux circuits synchrones auxquels on aurait substitué l'arbre d'horloge par un contrôleur asynchrone.

En résumé, les circuits asynchrones calculent leurs sorties uniquement en fonction de la présence de données sur ses entrées. Le temps de calcul des blocs combinatoires dépend des données à traiter. Enfin, dans un circuit asynchrone, la synchronisation des blocs est faite localement à l'aide de canaux de communication dédiés.

Chapitre 2

Caractéristiques de la structure cible

2.1. Introduction

Dans le cadre de cette thèse nous nous intéressons aux structures de type micropipeline. Ce sont les circuits asynchrones les plus proches des circuits synchrones. Ils sont composés d'un ensemble des blocs combinatoires interconnectés par des registres. Il apparaît que les réalisations basées sur ce concept ont conduit à des circuits fonctionnels [FURB94]. Les travaux effectués à TIMA dans le cadre de cette thèse, portent sur l'étude des circuits asynchrones micropipelines et de leur synthèse à partir d'une description de haut niveau basée sur le langage C. Ce chapitre porte sur la définition d'une structure micropipeline utilisant le protocole quatre phases ainsi que la présentation des divers modèles de structures micropipelines.

2.2. Pipeline synchrone

Un pipeline est une structure composée de plusieurs étages concurrents dont le but est d'augmenter le débit de sortie du système avec un faible coût en surface. Ce type de circuit est réalisé en alternant en une partie combinatoire qui effectue le traitement des données et une partie séquentielle qui sert à stocker les données. Dans ce chapitre, on s'intéresse aux pipelines synchrone et asynchrone.

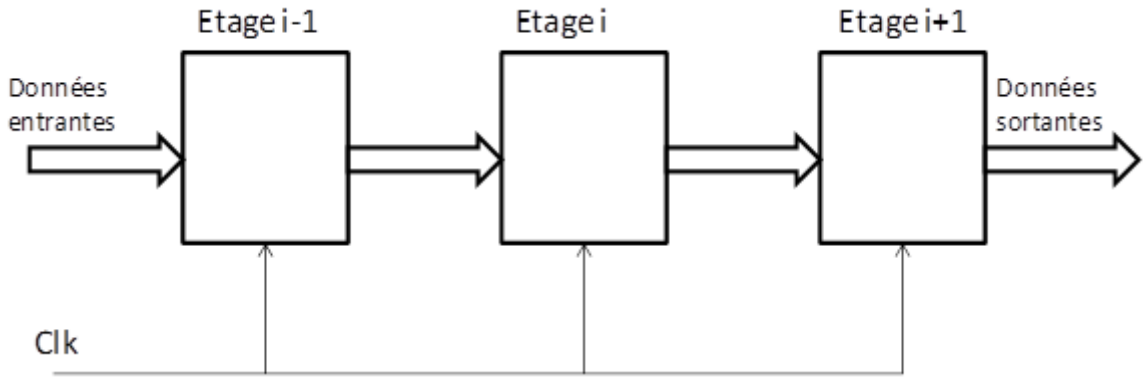


Figure 2-1: Pipeline Synchrone

La Figure 2-1 montre la structure d'un pipeline synchrone dont le signal d'horloge joue le rôle d'un actionneur global du circuit. Ce signal assure le contrôle global du système. A chaque front, tous les blocs capturent de nouvelles données.

Le calcul d'un étage doit s'effectuer dans un temps inférieur à celui qui s'écoule entre deux fronts montants d'horloge (la période de l'horloge).

2.3. Pipeline asynchrone et circuits micropipelines

Le principe de fonctionnement d'un pipeline asynchrone est basé sur le contrôle local des étages au contraire d'un pipeline synchrone. La Figure 2-2 suivante illustre le principe du contrôle local au travers de deux étages d'un pipeline asynchrone.

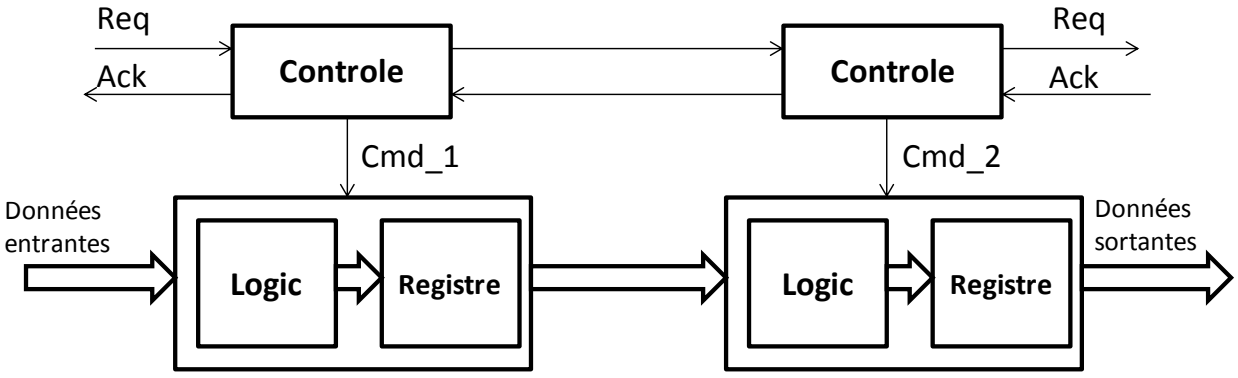


Figure 2-2: Circuit micropipeline

Dans un pipeline asynchrone, la synchronisation s'effectue localement entre chaque étage. La partie contrôle sera explicitée ultérieurement. Les requêtes sont actives à l'état haut et les acquittements sont actifs à l'état bas. L'apparition d'un front sur le signal Req en entrée indique que l'on a une donnée entrante prête à être traitée. Par conséquent un signal « cmd » est généré sur le bloc correspondant. Un des avantages de ces circuits est que si le récepteur n'acquitte pas la donnée reçue pour signaler la fin de traitement, on n'émet pas une nouvelle requête en aval. On assiste alors à un blocage des données en amont. Une bulle peut alors se créer entre les étages sans que cela perturbe le flot de données.

En conclusion, dans un pipeline asynchrone, les données progressent dans le circuit tant que les étages sont disponibles. En plus de cette propriété, les circuits asynchrones présentent de nombreuses autres caractéristiques telles que de faibles appels de courant car les étages ne capturent pas les données simultanément contrairement aux circuits synchrones (qui échantillonnent aux fronts d'horloge). Enfin, il est à noter que la première réalisation a été faite avec des verrous [SUTH 89].

2.4. Performances et contraintes d'un circuit micropipeline

Les quatre paramètres principaux qui servent à mesurer la performance d'un pipeline sont la latence, le temps de cycle, le débit et la capacité mémoire. La latence, c'est le temps nécessaire pour qu'une donnée traverse les étages d'un pipeline. Le temps de cycle est le temps maximum qui sépare la prise en compte de deux données successives dans un étage. Le débit c'est le nombre de données par unité de temps qui passe dans un étage pipeline. La capacité mémoire c'est le nombre de données que peut contenir un pipeline sans bloquer l'entrée du pipeline. Dans le cas d'un protocole quatre phases, il faut bien prendre en compte le temps de remise à zéro.

Les circuits micropipelines sont des circuits avec des hypothèses temporelles qui, à l'instar de leurs homologues synchrones, sont locales. En effet, cette contrainte se traduit par un retard introduit sur le signal de requête (d'acquittement) entre les étages. Ce retard permet de couvrir l'hypothèse faite sur le chemin critique de la partie combinatoire se trouvant dans les étages.

2.5. Pipeline linéaire et non-linéaire

La majorité des structures pipelines, qui sont présentées dans la littérature probablement par souci de simplicité, sont des structures linéaires [REZ 04]. Or l'évolution et la complexité des nouveaux systèmes numériques synchrones ou asynchrones nécessitent des structures non-linéaires.

2.5.1. Pipeline linéaire

Un étage pipeline linéaire est simplement un étage avec un seul canal d'entrée et un seul canal de sortie.

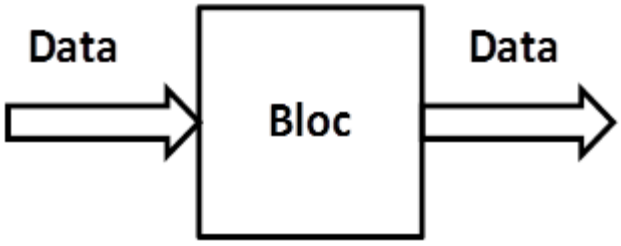


Figure 2-3: Pipeline linéaire

Cette figure montre le principe d'un pipeline linéaire. Cette structure montre une succession régulière d'étages avec ou sans traitement entre chaque étage.

2.5.2. Pipeline non-linéaire

Un pipeline non-linéaire est une structure ayant plusieurs canaux d'entrées et plusieurs canaux de sorties. Il est possible de définir quatre structures de base permettant de couvrir la totalité des situations : la fourche, la jonction, le multiplexage et le démultiplexage.

2.5.2.1. La fourche

La fourche est comme son nom indique est une structure avec un canal d'entrée et plusieurs canaux de sorties comme indiqué sur la Figure 2-4.

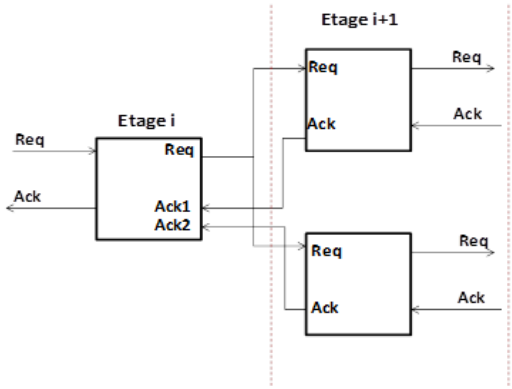


Figure 2-4: La fourche

Pour déclencher le traitement de la donnée dans une fourche, il faut attendre l'apparition d'une requête d'entrée. Après avoir diffusé cette requête aux canaux de sorties, le canal d'entrée attend l'acquittement pour l'exécution d'une nouvelle donnée.

2.5.2.2. La jonction

La jonction est la structure inverse de la fourche. C'est une structure avec plusieurs canaux en entrée et un unique canal de sortie. Les entrées convergent vers une seule sortie.

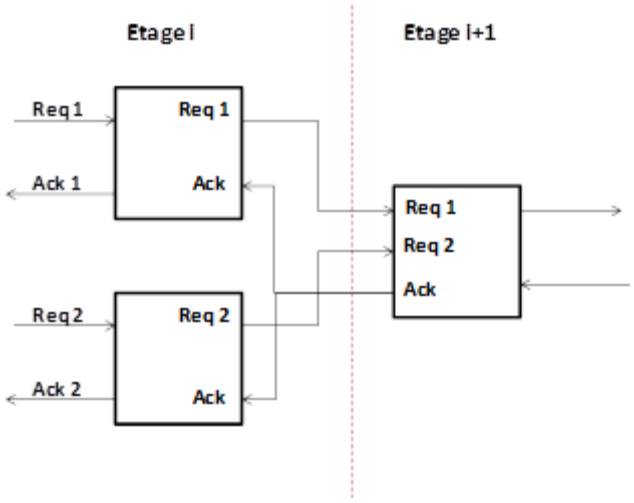


Figure 2-5: La jonction

La jonction est un opérateur de rendez-vous au niveau canal. Il réalise une synchronisation entre plusieurs canaux. Dans le cas d'un protocole quatre phases, la convergence qui reçoit les données des canaux d'entrée doit émettre un signal d'acquittement sur la totalité des canaux d'entrée.

2.5.2.3. Le multiplexage

Le multiplexage est une structure de choix dans un pipeline non-linéaire. Il propage une de ses entrées vers la sortie. La Figure 2-6 illustre le schéma du multiplexeur.

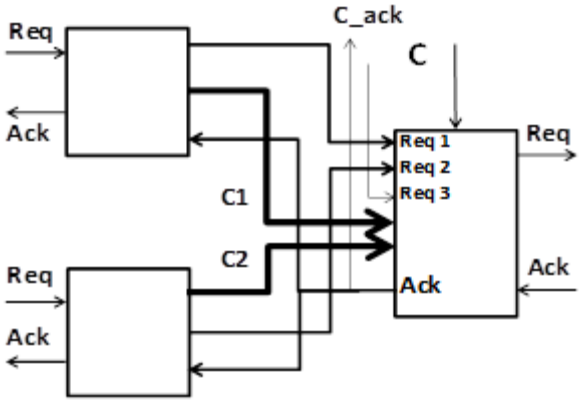


Figure 2-6: Le multiplexage

Le bloc de sortie a besoin du canal exprimant le choix pour sélectionner la requête d'entrée afin qu'elle soit envoyée vers la sortie.

2.5.2.4. Le démultiplexage

Au contraire du multiplexage, le démultiplexage est une structure aiguillant une donnée en fonction d'une condition. C'est donc un dispositif avec un seul canal d'entrée et plusieurs canaux de sortie comme l'indique la Figure 2-7. Le canal C en entrée sert à sélectionner le canal de sortie où la requête d'entrée sera envoyée.

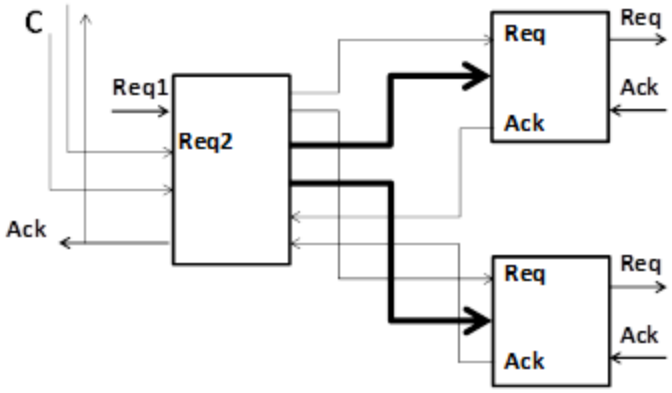


Figure 2-7: Le démultiplexage

Le canal C est dit canal de choix. Il assure le contrôle des canaux de sortie du circuit.

2.6. Etudes antérieures

Les différents modèles de pipelines asynchrones se caractérisent par leur latence, leur temps de cycle et leur robustesse. A. Martin et A. Lines [LINE 98] ont étudié les pipelines asynchrones réalisés avec une implémentation QDI (*Quasi Delay Insensitive*) qui présente une grande robustesse aux variations de tension, de température mais aussi vis-à-vis des procédés de fabrication. Dans la suite de cette thèse, nous nous intéressons plutôt aux circuits micropipelines. Dans ce type de circuits, les signaux de synchronisation (requête-acquittement) sont utilisés conjointement avec les chemins de données. Ces circuits présentent des avantages par rapport aux autres circuits asynchrones parmi lesquels une faible surface et une facilité de construction pour un concepteur habitué à travailler en synchrone. A cela, il faut ajouter les avantages usuels des circuits asynchrones comme une faible consommation, l'absence de contraintes dues à l'arbre d'horloge et moins de rayonnement électromagnétique.

P. Beerel a proposé une approche qui est une extension des travaux de K. van Berkel [BER 96]. Cette approche étudie plusieurs modèles de pipelines qui ciblent divers codages : double rails, 1 parmi N ou encore l'encodage de type données groupées [FER 02], [OZD 02], [OZD 02b] et [TUG 02]. Ces modèles offrent des pipelines à grain fin et à haute vitesse ainsi que des structures de pipelines linéaires et non-linéaires. Leur principal inconvénient est que, lors de la synthèse, ils demandent une bibliothèque de cellules complexes. [FER 02] propose une

approche dite « single-track » dans laquelle un canal encodé par un codage 1 parmi N est utilisé pour porter des données tout en servant également pour le signal d'acquittement.

2.7. Structure cible

Comme on a vu précédemment (Para. 2.3) que les systèmes asynchrones sont implémentés en séparant les parties contrôle et chemin de données. Plusieurs techniques ont été proposées pour l'implémentation des circuits pipelines QDI [LIN 98]. Ces techniques sont identifiés par leur protocole de communication : séquentiel, WCHB (*Weak Condition Half Buffer*), PCHB (*PreCharge Half Buffer*) ou PCFB (*PreCharge Full Buffer*) [LIN 98].

Dans la suite de cette thèse, on s'intéresse plus particulièrement aux circuits micropipelines [SUTH 89] utilisant le protocole 4 phases qui offre de bonnes performances ainsi qu'un faible coût. Ces circuits sont composés de parties combinatoires qui sont comparables à celles des circuits synchrones et qui sont connectés entre eux par des registres. Cette classe de circuits est donc formée de deux parties principales: la partie contrôle qui est insensible aux délais et la partie chemin de donnée qui comporte une partie combinatoire et un registre comme indiqué sur la Figure 2-8 ci-dessous**.**

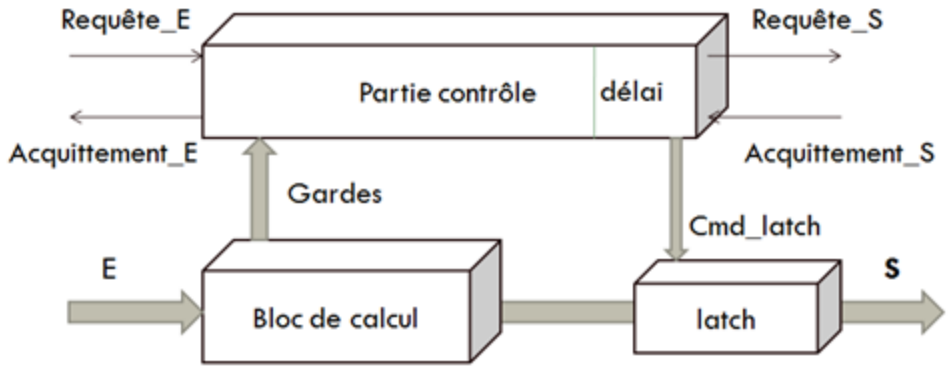


Figure 2-8: Modèle cible de micropipeline

La Figure 2-8 donne un aperçu du modèle d'un circuit micropipeline. Le but de cette étude est de concevoir un système moins consommant. De ce fait, cette représentation asynchrone nous offre plusieurs avantages en termes de consommation. Elle permet d'éliminer l'énergie dissipée

par le système de distribution des horloges qui peut atteindre 50% de la consommation globale du circuit. Cette structure assure la mise en veille de l'horloge quand le circuit ne reçoit pas de données. Il ne consomme donc pas en l'absence de données. La robustesse et l'adaptation aux conditions de fonctionnement permet de réduire facilement la consommation de circuit par la réduction de la tension d'alimentation. En conclusion, dans un circuit asynchrone, seulement les parties effectuant le traitement consomment.

Au contraire des circuits synchrones où la donnée avance d'un étage à un autre à chaque front d'horloge, dans un circuit asynchrone la donnée ne considère pas l'horloge mais avance dans le circuit au fur et à mesure qu'une place se libère. Cette représentation permet également que la partie chemin de données, qui contient les *latches/flip-flops* ainsi que les fonctions combinatoires, soit synthétisée selon la méthode synchrone classique et que la partie contrôle est réalisée selon une stratégie propre aux circuits asynchrones [SPA 02].

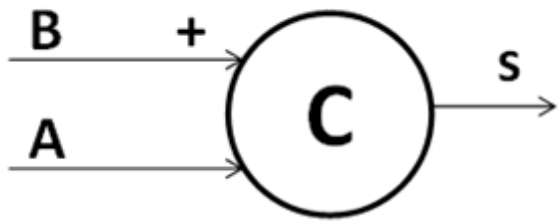
Les hypothèses temporelles (délais) de la partie contrôle permettent de nous donner le signal de fin de calcul. Ces hypothèses dépendent du bloc de calcul combinatoire et servent à adapter les signaux de synchronisation (requête et acquittement) en fonction des éléments de calcul. Par ailleurs, dans le cas d'un micropipeline non-linéaire, en plus de ses données en sortie, la fonction combinatoire d'un micropipeline peut aussi calculer les gardes nécessaires à une structure de choix qui sont transmises au contrôleur. Dans ce qui se suit, nous présentons les différentes structures micropipeline linéaires et non-linéaires.

2.8. La porte de Muller dissymétrique

La porte de Muller est une porte qui assure le rendez-vous entre deux ou plusieurs signaux. Cette porte constitue un élément clé de la conception des circuits asynchrones [BAP 02] [MB 59]. Bien qu'il existe de nombreuses variantes de cette porte, dans ce manuscrit, on détaille uniquement les variantes qui nous intéressent [FBF 07]. Dans le cas où toutes les entrées de la porte n'interviennent pas pour faire basculer l'état de sortie, la porte est dite « dissymétrique ».

2.8.1. La porte de Muller dissymétrique positive

Avec une porte de Muller, les entrées, lorsqu'elles sont identiques, forcent la valeur de sortie. Toutefois, cette propriété ne s'applique pas à toutes les portes de Muller. La Figure 2-9 suivante montre le symbole ainsi que la table de vérité d'une porte de Muller dissymétrique positive.



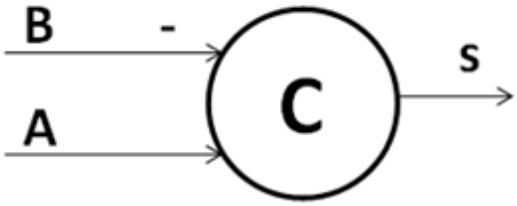
A	B	S
0	0	0
0	1	0
1	0	S^{-1}
1	1	1

Figure 2-9: Symbole d'une porte de Muller dissymétrique avec sa table de vérité

Sur cette figure, on remarque que l'entrée A seule fait commuter la sortie à '0' quand elle vaut '0', mais il faut les entrées A et B à 1 pour faire basculer la sortie à '1'.

2.8.2. La porte de Muller dissymétrique négative

La Figure 2-10 ci-dessous illustre le symbole avec la table de vérité de la porte de Muller dissymétrique négative.



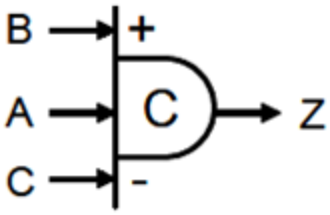
A	B	S
0	0	0
0	1	S^{-1}
1	0	1
1	1	1

Figure 2-10: Symbole d'une porte de Muller dissymétrique négative avec sa table de vérité

Dans ce cas, seule l'entrée 'A' fait monter la sortie 'S' à '1'. Cette sortie ne vaudra '0' que quand les deux entrées vaudront '0'.

2.8.3. La porte de Muller double dissymétrique (positive et négative)

Cette porte présente une combinaison entre les deux portes précédentes. Les signaux qui assure la montée de la sortie sont différents que ceux qui assure la descente. La Figure 2-11 suivante représente le symbole ainsi que la table de vérité de cette porte.



C	A	B	Z
0	0	X	0
X	1	1	1

Figure 2-11: Symbole d'une porte de Muller double dissymétrique avec sa table de vérité

2.9. Micropipelines linéaires

Les structures micropipelines peuvent adoptées différents protocoles tels que les protocoles séquentiels, WCHB (*Weak Condition Half Buffer*), PCHB (*PreCharge Half Buffer*) et PCFB (*PreCharge Full Buffer*). Dans cette thèse, on s'intéressera principalement au protocole WCHB.

2.9.1. Protocole séquentiel

La synchronisation entre les entrées et les sorties d'un bloc fonctionnel peut être effectuée sans aucun contrôle direct sur le bloc. On parle ici de protocole séquentiel. .La communication entre l'entrée et la sortie n'est pas contrainte par le contrôleur, qui est ici rudimentaire. En revanche, il est à noter que le contrôleur prend bien en charge les hypothèses temporelles faites sur le bloc combinatoire. La Figure 2-12 montre l'implémentation du protocole séquentiel.

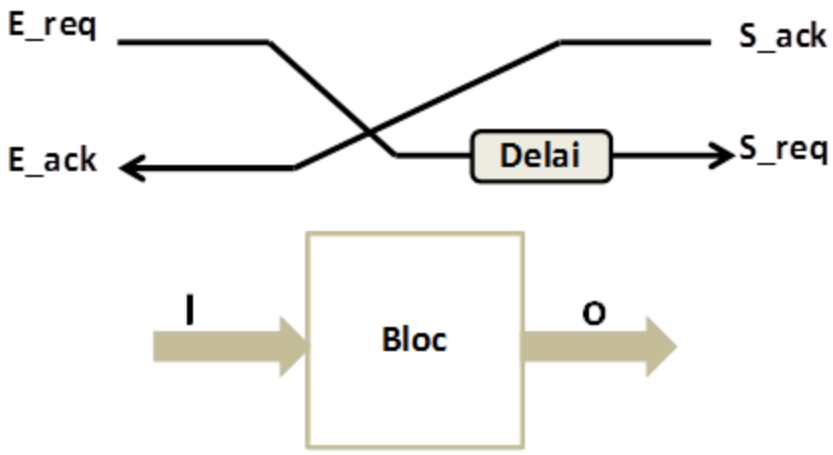


Figure 2-12: Micro pipeline séquentiel d'une structure linéaire

Cette illustration montre la liaison entre les entrées et les sorties de la structure en notant que « bloc » est le bloc de calcul avec la sortie « O ». L'entrée E_req et la sortie S_req sont reliées via un délai qui représente le temps de calcul du bloc tandis que les signaux d'acquittements en entrée et en sortie sont reliés directement. En conclusion, ce protocole est le plus lent car le temps de cycle correspond au chemin critique correspondant à la succession de blocs interconnectés. Une fois la donnée produite, l'acquittement est donné et rétro propagé. De ce fait, tout est bien séquentiel ici.

2.9.2. Structure WCHB (Weak Condition Half Buffer)

Le mot « *Half Buffer* » vient du fait que cette structure ne peut pas avoir deux données distinctes simultanément en entrée et en sortie. Ce protocole est le plus proche du protocole séquentiel. Il assure la synchronisation des phases montantes et des phases descendantes entre les canaux d'entrées et de sorties. L'évolution des phases des canaux d'entrée et de sortie sont fortement liées [REZ 04]. La Figure 2-13 illustre la structure du contrôleur correspondant à ce protocole.

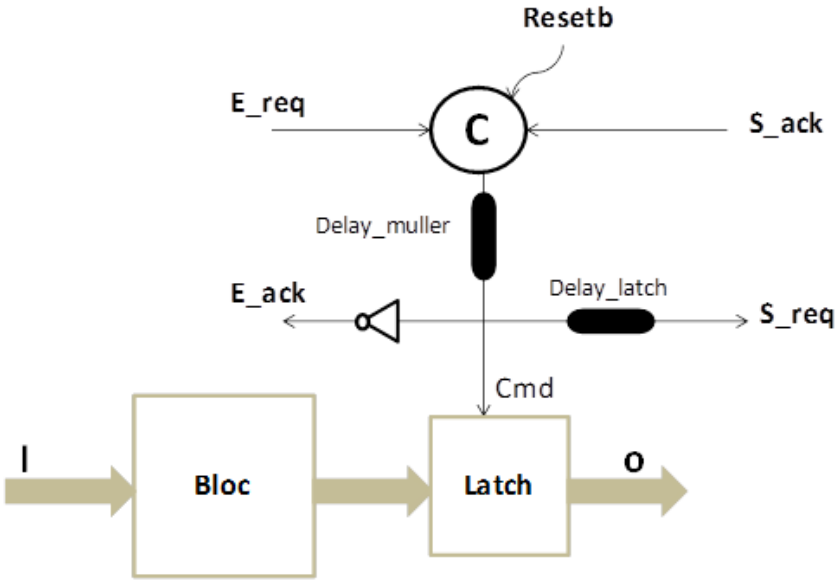


Figure 2-13: Protocole WCHB d'une structure linéaire

Les signaux de requêtes sont actifs à l'état haut et les signaux d'acquittement sont actifs à l'état bas. Après le signal de reset (Resetb) chaque signal est dans son état initial. Au moment où la donnée arrive, le signal E_req passe à '1'. Puisque les signaux d'acquittement sont initialisés à '1', la sortie de la porte de Muller passe à '1'. Le signal E_ack de sortie descend à '0'. Le signal S_req monte à '1' après un délai équivalent au temps de calcul des sorties et après que le signal Cmd ait déclenché la capture des calculs effectués dans le bloc combinatoire. Le signal S_ack peut alors acquitter les données en descendant à '0'. L'entrée E_req est en mesure de réagir à cette descente en descendant à '0' également. Le signal S_req repasse à '0', provoquant le passage à '1' du signal S_ack et par conséquent du signal E_ack. Tous les signaux sont alors revenus dans leur état initial. On remarque que ce protocole assure la symétrie des signaux E_req et S_ack sur les phases montantes et descendantes. Ceci permet d'avoir une implémentation régulière de la partie contrôle du circuit asynchrone.

Un des principaux avantages des pipelines est leur capacité à augmenter le débit. Enfin d'amplifier ce gain, il est possible de recourir à des protocoles encore plus efficaces comme le PCHB *(Pre-Charged Half Buffer)* ou le PCFB *(Pre-Charged Full Buffer)*. En effet, dans un

protocole PCHB, la désynchronisation entre les phases descendantes des signaux de requêtes et des acquittements autorise une communication indépendante entre les entrées et les sorties et par conséquent assure un gain en vitesse. Avec le protocole PCFB, il y a un découplage total des canaux d'entrée et de sortie. Ce découplage offre des meilleures performances au niveau de la vitesse mais l'implémentation devient alors plus coûteuse.

En conclusion, les protocoles les plus performants et les plus représentatifs utilisés lors de la conception de circuits asynchrones sont :

- Le protocole WCHB : c'est le protocole le plus simple. Il assure une symétrie d'utilisation des signaux de synchronisations des communications.
- Le protocole PCHB : il assure une désynchronisation des communications favorable à l'augmentation des performances en débit au détriment de la complexité du contrôleur.

Le protocole PCFB : c'est le protocole le plus rapide et le plus concurrent mais aussi le plus complexe.

2.10. Micropipelines non-linéaires

Vue la complexité des systèmes intégrés synchrones, le choix des circuits asynchrones est devenue une option à réétudier. Ils avaient en effet été délaissés à cause de leur complexité de mise en œuvre à l'époque ! La complexité des circuits synchrones se traduit au travers des hypothèses temporelles toujours plus difficiles à tenir mais aussi au travers d'architectures faisant apparaître des structures non linéaires par exemple. Une solution qui permet de simplifier et de renforcer l'intégration de ces structures non-linéaires est l'utilisation de circuits asynchrones qui vont relâcher certaines contraintes dans la gestion de ces structures. Dans ce paragraphe, on abordera les différents types des circuits micropipelines non-linéaires tels que fourches, convergences, multiplexages et démultiplexages.

2.10.1. Structure avec un protocole séquentiel

La Figure 2-14 ci-dessous illustre une fourche implémentée avec le protocole séquentiel.

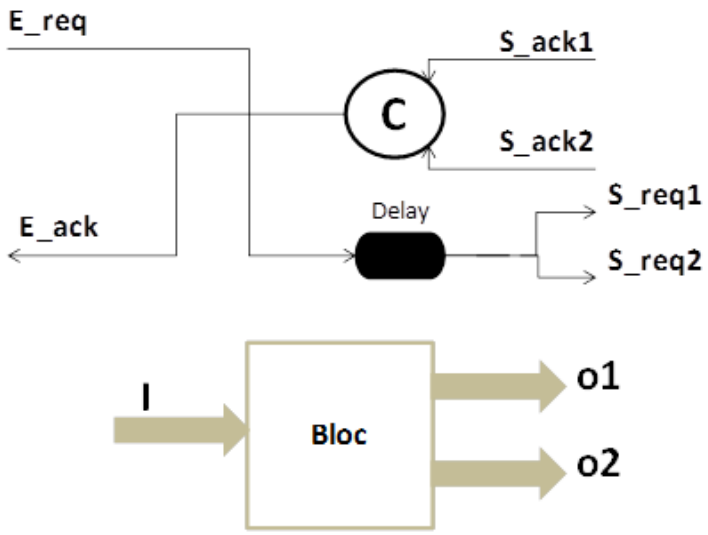


Figure 2-14: Micropipeline séquentiel d'une fourche

Une fourche est une structure a un canal d'entrée et à plusieurs canaux de sorties. La porte de Muller permet de synchroniser les deux acquittements en sortie S_ack1 et S_ack2. La synchronisation a lieu grâce à cette porte de Muller. Ce problème vient du fait que le canal de sortie ne s'acquitte que quand les deux sorties ont reçu les données. Le délai présenté sur la ligne de requête représente le temps maximal nécessaire au calcul des deux sorties O1 et O2.

La convergence est la structure duale de la fourche. C'est une structure à plusieurs canaux d'entrée et à un unique canal de sortie. La Figure 2-15 suivante représente la structure de convergence en protocole séquentiel.

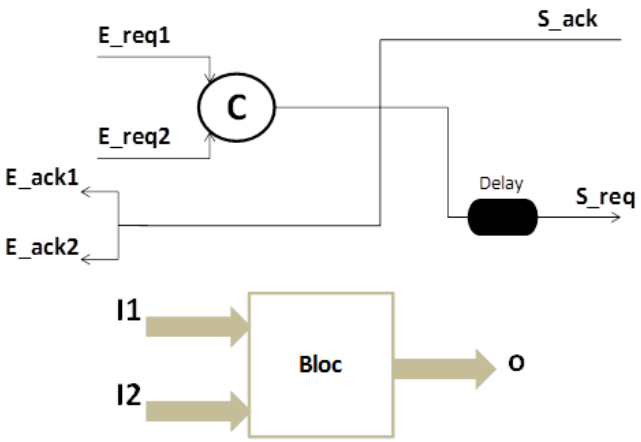


Figure 2-15: Micropipeline séquentiel de la convergence

Cette structure permet de synchroniser les canaux d'entrées. La donnée ne peut pas passer en sortie sans avoir des données sur les deux canaux d'entrée I1 et I2. Cette exigence se fait au travers de la porte de Muller qui synchronise les deux signaux de requêtes en entrée. Par ailleurs, les deux acquittements E_ack1 et E_ack2 en entrée proviennent du canal de sortie O.

Les structures de choix sont divisées en deux types : le multiplexage et le démultiplexage. Ces structures sont plus complexes que les structures de convergence et de fourche. La principale différence entre ces structures est la présence d'un canal de sélection. Ce canal permet de décider quelle entrée doit être transmise et sur quelle sortie. La Figure 2-16 suivante illustre la structure d'un démultiplexeur.

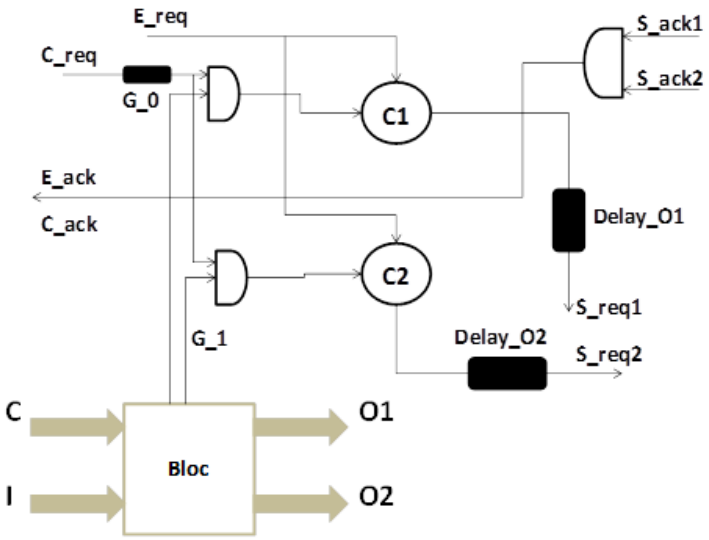


Figure 2-16: Micropipeline séquentiel d'un démultiplexeur

Le démultiplexage est une structure à un canal d'entrée et à plusieurs canaux de sortie. Les portes de Muller C1 et C2 servent à synchroniser ses signaux d'entrée. G_0 et G_1 sont les gardes. Elles sont calculées dans la partie combinatoire avant d'être envoyées à la partie contrôle. Le calcul de la donnée de sélection peut être complexe et il nécessite un bloc de calcul. Ce bloc de calcul peut exister dans la partie chemin de donnée qui est synthétisée de façon synchrone.

La mutuelle exclusion des canaux garantit qu'un seul signal d'acquittement en sortie sera donné. Comme ces signaux sont actifs à l'état bas, une simple porte « AND » pourra jouer le rôle du « OR » pour transmettre le signal d'acquittement C_ack du canal de sélection.

La structure du multiplexeur est la structure duale du démultiplexeur : plusieurs canaux d'entrée pour un seul canal de sortie. La Figure 2-17 ci-dessous illustre la structure micropipeline séquentiel en multiplexeur.

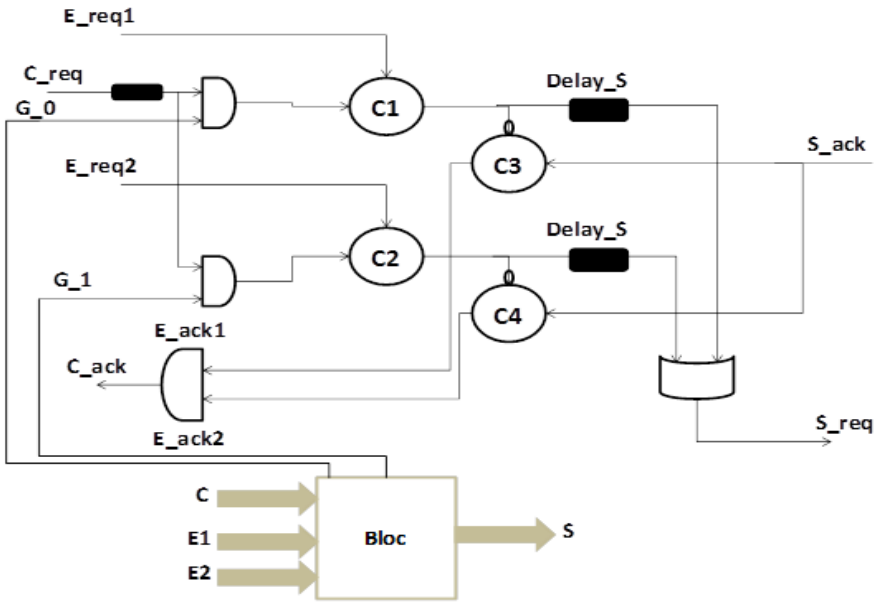


Figure 2-17: Micropipeline séquentiel en multiplexeur

Cette structure comporte deux entrées et une seule sortie S_req. Les portes de Muller C3 et C4 permettent de faire la sélection dans laquelle une des deux entrées va être transmise en sortie (transmission des requêtes E_req1 ou E_req2). Par conséquent, elles n'acquittent que le canal

d'entrée sélectionné. Les inverseurs présents à ses entrées viennent du fait que l'acquittement est actif à l'état bas. Une porte OR existe en sortie pour générer le signal de requête choisi.

2.10.2. Structure WCHB

La structure WCHB est une structure simple utilisant une porte de Muller et quelques portes logiques. Dans ce paragraphe, on va décrire les différents contrôleurs que l'on rencontre avec un circuit micropipeline utilisant un protocole WCHB. La Figure 2-18 illustre une fourche implémentée avec le protocole WCHB.

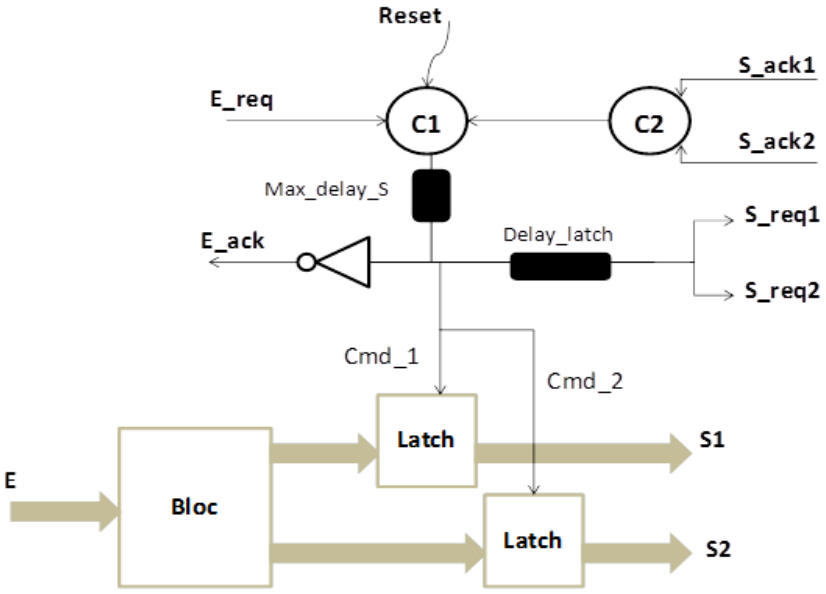


Figure 2-18: Micropipeline de fourche en protocole WCHB

En WCHB, les données ne sont pas transmises aux sorties tant qu'elles ne sont pas disponibles. Une porte de Muller C2 assure la synchronisation des signaux d'acquittement S_ack1 et S_ack2. L'acquittement en entrée est généré à l'aide d'un inverseur. Les acquittements sont initialisés à '1', donc un événement sur le signal E_req permet de générer les signaux de commandes des *latches* cmd_1 et cmd_2 après un temps couvrant les hypothèses temporelles du bloc combinatoire. Les requêtes de sortie S_req1 et S_req2 montent à '1' après un délai qui est équivalent au passage des données via les *latches*.

La convergence représente la structure duale de la fourche. Elle propose deux canaux d'entrée et un seul en sortie. Cette structure peut être considérée comme un rendez-vous au niveau du canal. La Figure 2-19 suivante illustre la structure de la convergence.

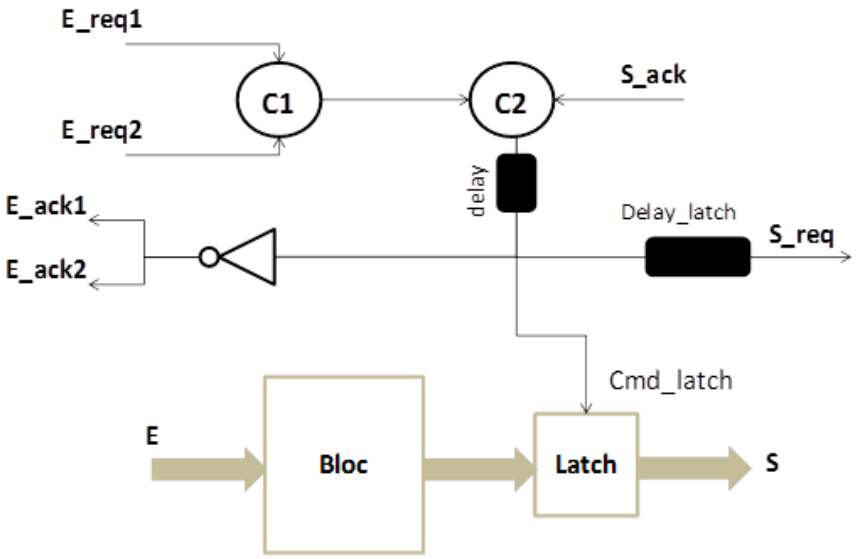


Figure 2-19: Micropipeline de convergence en protocole WCHB

Pour que la donnée passe de l'entrée vers la sortie, la structure doit recevoir en entrée les deux requêtes E_req1 et E_req2. La porte de Muller C2 assure la synchronisation entre ces requêtes et le signal d'acquittement afin de générer le signal S_req en sortie. Cette structure permet d'assurer la synchronisation entre plusieurs canaux ou entre plusieurs processus concurrents.

L'implémentation des structures de choix en protocole WCHB est illustrée à la Figure 2-20. Les signaux de sélection (les gardes G_0 et G_1) produits par le bloc fonctionnel sont envoyés au bloc de contrôle. Une porte « AND » est utilisée pour générer les acquittements dans le cas du démultiplexeur. Cette porte réalise la fonction d'un « OR » exclusif parce que les acquittements sont actifs au niveau bas. La Figure 2-20 illustre l'architecture d'un démultiplexeur en circuit micropipeline en protocole WCHB.

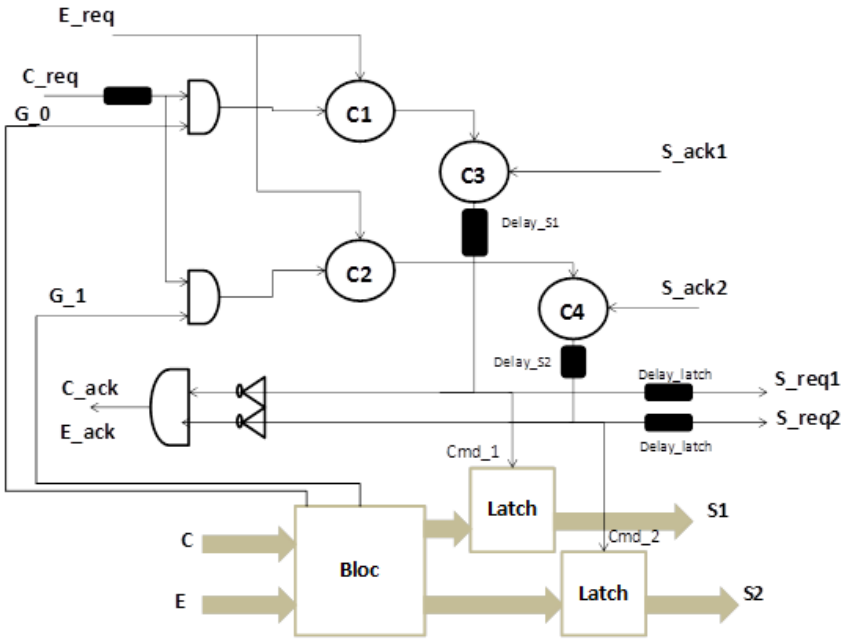


Figure 2-20: Micropipeline d'un démultiplexeur en protocole WCHB

Le canal 'C' est le canal de sélection. Une fois, ce canal actif, la valeur qu'il transporte sélectionne le canal de sortie autorisé à communiquer et envoyer une donnée.

La Figure 2-21 montre la structure d'un multiplexeur d'un circuit micropipeline de utilisant le protocole WCHB. Le multiplexeur propage sur sa sortie la valeur reçue sur l'une de ces entrées. Un canal de sélection choisit l'entrée dans laquelle les données seront prélevées. Ainsi, pour transmettre une donnée vers la sortie, on doit attendre deux événements : une requête sur l'une des entrées et une deuxième requête sur le canal de sélection. En conclusion, un multiplexage est une « convergence contrôlée ».

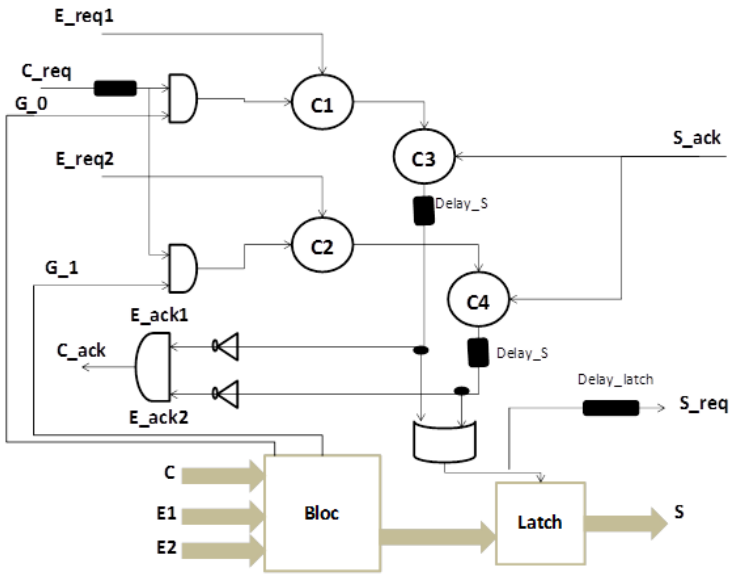


Figure 2-21: Micropipeline de multiplexeur en protocole WCHB

2.11. Conclusion

Différents types de structures micropipelines linéaires et non-linéaires ont été présentés dans ce chapitre ainsi que les différents protocoles qui gèrent les communications dans les canaux. Les circuits de contrôle asynchrones permettant de gérer les différents protocoles ont été détaillés. La structure micropipeline cible, basée sur une séparation de la partie contrôle et de la partie chemin de donnée, a été également explicitée. On a notamment identifié des structures pipelines asynchrones non-linéaires favorables à une réduction de la complexité des circuits synchrones. Les problèmes liés au temps de propagation et aux synchronisations dans les branches sont réglés grâce à l'utilisation de portes de Muller et l'implémentation de délais couvrant obligatoirement les hypothèses temporelles locales faites dans ce type de circuits.

Enfin, nous avons montré notre structure cible de circuits micropipelines, composée d'une partie contrôle et d'un chemin de données, que nous pourrons exploiter dans la suite de cette thèse pour minimiser la consommation d'énergie. Le chapitre suivant illustrera, grâce à quelques résultats de simulation, l'efficacité de cette structure associée à un dispositif de contrôle de l'horloge en termes de consommation et de surface.

Chapitre 3

Des contrôleurs distribués asynchrones pour une nouvelle approche du *clock gating*

3.1. Introduction

La consommation est devenue un paramètre clé pour la conception des systèmes électroniques. Elle est considérée souvent comme la contrainte la plus forte. Cette consommation se divise en consommation dynamique et en consommation statique. La première est due à l'activité du circuit et la deuxième est due aux courants de fuite, devenus non négligeable dans les circuits utilisant une technologie fortement submicronique. Dans ce chapitre, on s'intéresse à la consommation dynamique, à ses origines et aux techniques de réduction associées...

Dans un système synchrone, l'arbre d'horloge peut représenter jusqu'à 45% de l'ensemble de la consommation dynamique [ZAK 01]. Plusieurs techniques existent pour limiter ce genre de consommation parmi lesquelles les techniques de *clock gating* et de DVFS (*Dynamic Voltage and Frequency Scaling*).

Dans un circuit asynchrone, l'arbre d'horloge n'existe pas, le contrôle se faisant localement. En d'autre termes, le système sera actif si et seulement si il y a une donnée prête à être traitée. Cela provoque une réduction de la consommation dynamique en général car seule la partie du circuit exécutant un traitement est activée. Le gain peut être important surtout lorsque l'arbre d'horloge est très consommant. Dans ce chapitre, une nouvelle méthodologie de réduction de consommation basée sur le principe du *clock gating* est introduite. Cette technique de *clock gating* est certainement celle qui est la plus connue. Néanmoins, elle nécessite un calcul supplémentaire pour déterminer quelle partie de circuit doit être éteinte.

3.2. Sources de la consommation dynamique

Comme cela vient d'être mentionné, la consommation est divisée en consommation dynamique et en consommation statique. L'équation (i) ci-dessous définit les sources de consommation dans un circuit :

$$P = P_{leakage} + P_{switching} + P_{short\ circuit} \qquad \text{(i)}$$

La puissance statique est donnée par $P_{leakage}$ alors que la puissance dynamique est donnée par $P_{switching}$ et $P_{short\ circuit}$. $P_{switching}$ est due à l'activité de commutation d'un état logique à un autre ce qui provoque un courant de commutation (charge et décharge des capacités). $P_{short\ circuit}$ est due au court-circuit quand les transistors NMOS et PMOS sont actifs simultanément, cela provoque le passage direct de courant de VDD jusqu'à GND. $P_{switching}$ et $P_{short\ circuit}$ sont détaillés dans le paragraphe suivant.

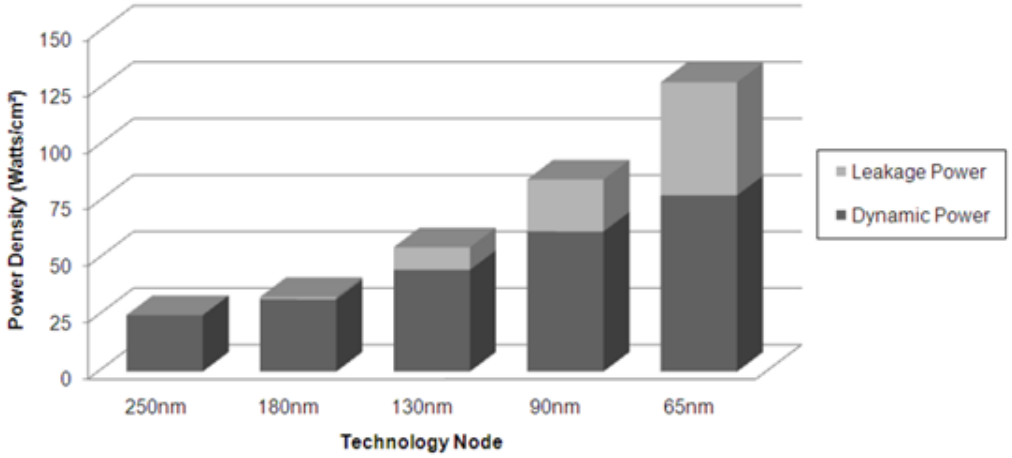


Figure 3-1: Consommations statique et dynamique en fonction de la technologie

La Figure 3-1 ci-dessus [PAN 05] montre l'évolution de la consommation en fonction de la technologie. On remarque sur cette figure que la consommation statique augmente avec l'évolution de la technologie mais que la consommation dynamique reste prépondérante. Dans ce chapitre, notre étude sera concentrée sur l'activité de commutation afin de pouvoir réduire cette activité et par conséquent réduire la puissance de commutation $P_{switching}$.

3.2.1. Puissance de commutation (switching power)

La puissance due à la commutation est donnée par la formule (ii) suivante :

$$P_{switching} = \alpha . C_l . f . V_{dd}^2 \qquad \text{(ii)}$$

Avec α est la probabilité de commutation d'un transistor pendant un cycle d'horloge à la fréquence f. C_L est la capacité de charge dite « *load capacitance* » et Vdd est la tension d'alimentation. D'après cette équation, on remarque bien que la puissance de commutation dépend de l'activité de commutation ainsi que des capacités qui sont liées à la géométrie.

Pour réduire la consommation, et principalement $P_{switching}$ on doit diminuer ces quatre facteurs :

- La tension d'alimentation V_{dd} : son intervention quadratique permet d'envisager un gain important d'optimisation par sa réduction.
- La capacité physique C_L, décomposée en une partie venant des cellules CMOS et une autre provenant des interconnexions.
- Le paramètre α qui décrit l'activité de commutation. Il montre que même un circuit très complexe ne consomme que s'il est en activité.
- La fréquence f d'horloge : la puissance instantanée diminue (et augmente) avec la fréquence.

Les techniques de réduction de la consommation seront détaillées ultérieurement.

3.2.2. Puissance due au court-circuit (Short-Circuit power)

Effectivement ce type de puissance apparait pendant une période très petite. Elle apparaît quand le réseau « P » et le réseau « N » sont actifs simultanément. Cette puissance est donnée par l'équation (iii) suivante :

$$p_{short-circuit} = \frac{\beta}{12} (V_{dd} - 2.V_{th})^3 \frac{\tau}{T} \qquad \text{(iii)}$$

Avec V_{th} la tension de seuil, T le temps de montée/descente des entrées, τ le délai d'une porte et β le facteur de conductivité de transistor par unité de tension.

3.3. Techniques de base de réduction de la consommation dynamique

L'évolution actuelle de la technologie et la complexité des circuits font de la consommation dynamique un facteur essentiel pour un nombre croissant d'applications industrielles. Le concepteur d'un système électronique doit maîtriser tout au long de sa démarche, les optimisations architecturales et technologiques tout en faisant un compromis entre vitesse, coût et consommation. En outre, les applications actuelles telles que les ordinateurs portables, ainsi que les télécommunications sans fil (GSM, radio-mobile…) intègrent des fonctionnalités complexes (codage parole ou vidéo, multimédia…) qui demandent une réelle capacité de traitement avec une forte contrainte sur la consommation.

La Figure 3-2 illustre le rapport de consommation dynamique/statique. Il est clair que la consommation dynamique est le facteur dominant. Les techniques de réduction de consommation peuvent s'appliquer à différents niveaux d'abstraction comme l'indique la Figure 3-2 ci-dessous.

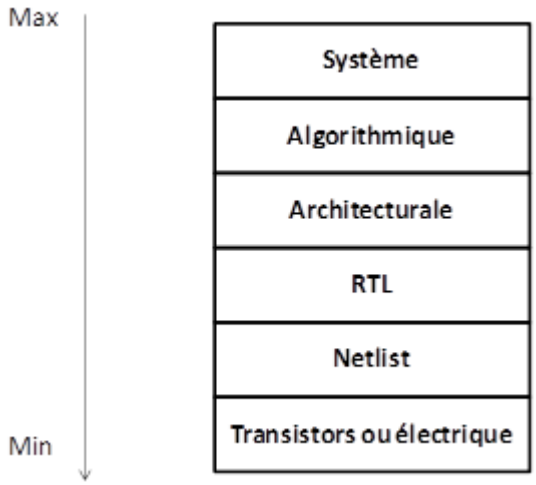


Figure 3-2: Les différents niveaux d'abstraction utiles pour l'optimisation de la consommation d'énergie

Au niveau système, l'architecture matérielle n'est pas définie et les algorithmes sont décrits par des langages de haut niveau comme Matlab et C++. Quand on passe au niveau architectural, l'architecture peut être décrite en SystemC par exmple.

La modélisation au niveau RTL correspond à la description de l'implémentation physique du système à l'aide d'une sémantique permettant la description des bascules, multiplexeurs, registres, UAL, portes, etc. Cette description se fait usuellement avec les langages VHDL ou Verilog.

On remarque sur la Figure 1-1 que le gain optimal en consommation est atteint au niveau le plus haut, c'est-à-dire le niveau système. A ce niveau, la technique *DVFS* peut amener un gain très substantiel mais elle nécessite généralement de développer un matériel spécifique et un logiciel de contrôle de la tension et de la fréquence [FLA 04]. Cependant, les techniques de réduction de la consommation sont souvent implémentées au niveau RTL (*Register Transfer Level*). Ainsi, la technique de *clock gating,* est largement répandue à ce niveau d'abstraction pour la consommation dynamique. De même, les techniques de *power gating* ou de *back biasing* sont très connues pour réduire la consommation statique (détaillées dans le chapitre 5).

3.3.1. Clock gating

Comme cela a déjà été mentionné, la consommation due à l'arbre d'horloge peut aller jusqu'à 45% de la consommation totale du circuit [ZAK 01]. Dans un système synchrone l'horloge est toujours active, ce qui explique la contrainte sur la consommation dans ces circuits.

Une des techniques largement répandue pour réduire la consommation dynamique dans un circuit est la technique dite de « *Clock gating* ». Cette technique d'optimisation de puissance peut être employée dans un ASIC ou un FPGA pour diminuer ou éliminer l'activité inutile [CAR 01] [AMI 01]. Par exemple, dans les circuits VLSI tels que les processeurs, qui peuvent contenir plusieurs unités arithmétiques et logiques avec des registres, etc. Les registres n'ont pas besoin d'être utilisés en permanence. L'idée est donc de bloquer l'horloge sur ces registres et, par conséquent, de supprimer l'activité de commutation correspondante. En effet, dans le modèle de conception synchrone utilisé dans la plupart des outils de synthèse, le système véhiculant le

signal d'horloge est connecté à chaque pin d'horloge de chaque bascule. Il en résulte trois principales sources de consommation d'énergie:

1- la puissance consommée par les bascules

2- la puissance consommée par la logique combinatoire dont les valeurs changent suite au changement des valeurs des bascules à chaque front d'horloge

3- la puissance consommée par l'arbre d'horloge lui-même (*clock tree*).

La technique de *clock gating* consiste à couper l'horloge dans les zones de circuit qui ne produisent pas des données pertinentes [POK 07]. Cette fonctionnalité s'obtient relativement aisément en modifiant l'arbre d'horloge existant. La Figure 3-3 ci-dessous illustre le schéma de principe d'un bloc de *clock gating*.

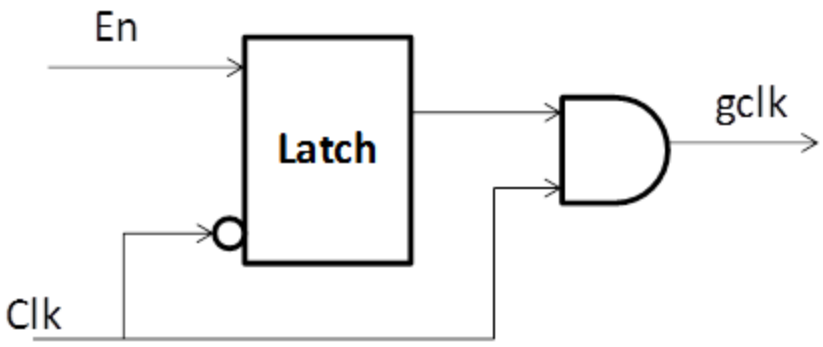


Figure 3-3: Clock gating

La Figure 3-3 ci-dessus formé par une bascule et une porte AND. Sur cette figure, on remarque que l'horloge est active sur front descendant. L'entrée «En» (*Enable*) indique la validité des données pour le chemin de données qui utilise un sous-ensemble de l'arbre d'horloge qui sera donc activé si cela est nécessaire. A partir de cette structure on peut éteindre l'horloge en fonction des besoins. Deux phases existent, la première est quand un front apparait sur l'entrée «En» qui nous informe qu'il y a une donnée prête pour le traitement. La deuxième phase est la phase d'échantillonnage, comme indiqué sur la Figure 3-3 qui se fait au front descendant de l'horloge afin d'envoyer l'horloge en sortie.

Dans [POK 07], Pokhrel a commencé son expérience sur une petite puce implémentée avec et sans *clock gating*. Une puce en technologie 180 nm a été ré- implémentée dans la même technologie avec le bloc de *clock gating*. Des petits changements ont été effectués dans la logique (suppression des petits blocs et ajouts d'autres). Pokhrel rapporte une réduction de 20% en surface ainsi qu'un gain entre 34% et 43% en consommation en fonction du mode de fonctionnement.

En se référant à l'équation (ii), on remarque que la puissance est proportionnelle à la tension d'alimentation et la fréquence d'horloge. La fréquence varie linéairement alors que la tension d'alimentation varie quadratiquement. [FLA 04] [POU 01] [DHA 01] et [DHA 02] ont travaillé sur la technique dite *DVFS (Dynamic Voltage and Frequency Scaling)* qui permet de réduire significativement la puissance dynamique dans la plupart des systèmes SoC complexes par le contrôle de leur tension d'alimentation et de leur fréquence d'horloge.

3.3.2. DVFS (Dynamic Voltage and Frequency Scaling)

L'équation (ii) montre la proportionnalité entre la puissance dynamique consommée, la fréquence d'horloge et la tension d'alimentation. La puissance varie linéairement avec la fréquence alors qu'elle varie quadratiquement avec la tension d'alimentation. Le *Dynamic Voltage et frequency Scaling* "DVFS" est une technique efficace de réduction de la puissance dynamique applicable dans la plupart des systèmes SoC complexes par le contrôle de la tension d'alimentation et de la fréquence d'horloge [FLA 04]. Si T_d est le délai de propagation dans une porte, l'équation (iv) montre la relation entre T_d et la tension d'alimentation :

$$T_d \, \alpha \, \frac{V_{dd}}{(V_{dd} - V_{th})^2} \qquad \text{(iv)}$$

Le délai diffère d'une librairie à une autre comme, par exemple, pour les librairies LVT (Low V_t) et HVT (High V_t) d'un même fondeur. Pour la librairie LVT, ce délai est minimal alors qu'il est plus grand pour HVT. Si la fréquence d'horloge est trop grande, l'échantillonnage des données peut devenir incorrect. Il faut donc prendre en compte l'hypothèse faite surle chemin critique qui doit être inférieure à $1/f_{clk}$ [POU 01].

$$T_{d(chemin\ critique)} < \frac{1}{f_{clk}} \qquad (v)$$

Les équations (iv) et (v) montrent la dépendance entre la tension d'alimentation et la fréquence d'horloge. Ces deux paramètres ne doivent pas être séparés. Il est nécessaire de diminuer la fréquence d'horloge avant la diminution de la tension d'alimentation et, respectivement, augmenter la tension d'alimentation avant d'augmenter la fréquence d'horloge. Ce principe est nécessaire quel que soit le système afin de garantir l'hypothèse faite sur le chemin critique. Cette approche couplant tension d'alimentation et fréquence peut offrir un gain substantiel sur la consommation [DHA 01] [DHA 02].

3.4. Contrôleur asynchrone d'horloge

Les circuits micro-pipelines sont assez semblables à des circuits synchrones, mais leur mécanisme de synchronisation est basé sur des contrôleurs asynchrones qui remplacent l'arbre d'horloge, comme les contrôleurs WCHB présentés au chapitre précédent. Dans ce paragraphe, nous allons détailler une nouvelle structure de contrôleur asynchrone susceptible d'être exploité sur un dispositif de *clock gating*.

3.4.1. Approche

Dans les systèmes synchrones, l'arbre d'horloge est toujours activé. A chaque front d'horloge, les données sont transmises entre registres au travers d'une partie combinatoire. C'est exactement ce nous décrivons au niveau RTL [REN 02]. Cela peut aussi s'interpréter comme une synchronisation globale et aveugle qui ne possède aucune connaissance spécifique des zones où les données sont réellement traitées. En effet, l'horloge globale synchronise l'ensemble de la puce sans éteindre les parties de circuits inactifs. Pour cette raison, la mise en œuvre d'une structure exploitant des *gated clo*cks est un bon moyen pour limiter la consommation d'énergie dynamique. Néanmoins, cela nécessite un calcul supplémentaire pour déterminer quelles parties du circuit doivent être éteintes.

Une autre approche consiste à synchroniser les données localement à l'interface des blocs synchrones. Cela rend le circuit globalement asynchrone et localement synchrone (GALS) [SUH

08]. Pour la mise en œuvre d'une telle approche, il est nécessaire de concevoir des contrôleurs asynchrones, semblables aux contrôleurs WCHB, capables de fournir une signalisation aux données. Cela donne aussi la possibilité d'exploiter ces signaux de synchronisation locaux pour commander localement l'horloge. La Figure 3-4 ci-dessous illustre le principe d'un tel système.

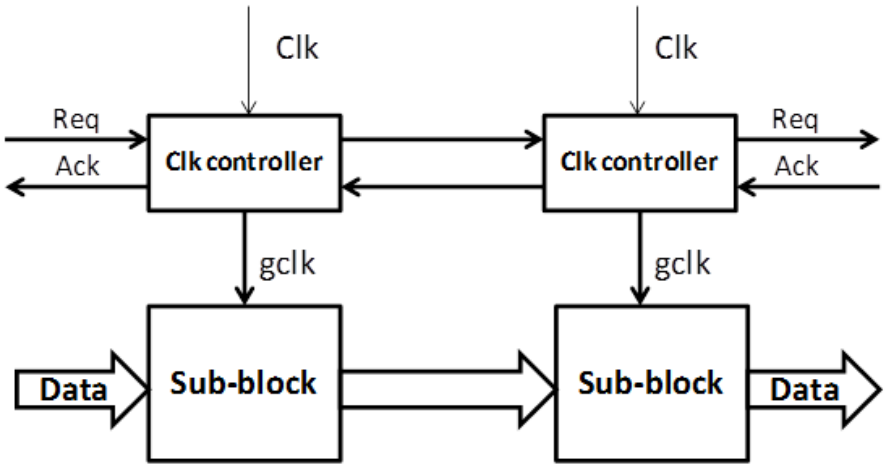


Figure 3-4: Horloges distribuées pour un système synchrone

Les signaux de requêtes sont actifs à l'état haut alors que les signaux d'acquittements sont actifs à l'état bas. L'apparition d'un front montant sur la requête indique que les données sont prêtes à être traitées et, par conséquent le signal d'horloge "gclk" doit être fournie pour le sous-block de réception synchrone (Sub-block). Le signal d'acquittement envoyé par le récepteur indique qu'il a terminé le traitement et qu'il est de nouveau prêt à recevoir de nouvelles données.

3.4.2. Structure du contrôleur linéaire d'horloge

Comme cela a été mentionné, la principale différence entre les approches synchrones et asynchrones est que, dans un système asynchrone, le contrôle est local et distribué. L'idée d'appliquer ce principe asynchrone à une granularité plus grossière pourrait nous permettre de :

- Distribuer les contrôleurs d'horloge dans le système de gestion des *gated clocks*.
- Activer localement les blocs/IPs synchrones grâce à nos contrôleurs d'horloge

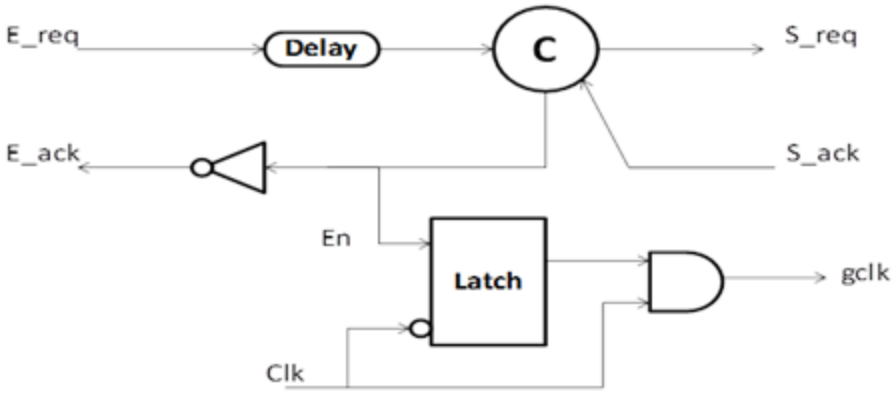


Figure 3-5: Structure du contrôleur d'horloge

La Figure 3-5 illustre le contrôleur asynchrone conçu pendant cette thèse qui permet d'assurer la synchronisation locale pour chaque bloc du circuit. Cette structure correspond au cas le plus simple, et est utilisé pour contrôler un pipeline synchrone. Comme S_ack est initialisé à '1', lorsque E_req passe au niveau haut, le signal de sortie de la porte de Muller et le signal En du latch montent à '1'. Ceci indique que les données sont prêtes à être traitées et que le signal d'horloge doit être fourni au bloc. Le signal de fin de calcul est déterminé par les délais, mais de manière plus appropriée peut être aussi fourni comme un résultat du traitement du sous-bloc. Ces techniques sont capables de fournir au bon moment le signal d'acquittement. Si des hypothèses temporelles sont utilisées, le retard doit couvrir la contrainte temporelle imposée par le sous-bloc synchrone. Le bloc de *clock gating* échantillonne le signal 'En' au front descendant. Quand ce signal passe à '1', la valeur est capturée dans le latch et l'horloge peut se propager au travers de la porte AND (voir la Figure 3-3).

3.4.3. Structure du contrôleur non-linéaire d'horloge *(Split)*

Dans le chapitre précédent, nous avons détaillé les différentes structures de types micropipeline linéaires et non-linéaires. Dans ce chapitre on va aborder la structure de choix *(split)*. Cette structure est de type non-linéaire. Comme précédemment, afin de gérer localement le signal d'horloge, un bloc *clock gating* a été ajouté en sortie de cette structure pour générer des horloges quand on a besoin. La Figure 3-6 ci-dessous illustre le contrôleur d'horloge non-linéaire.

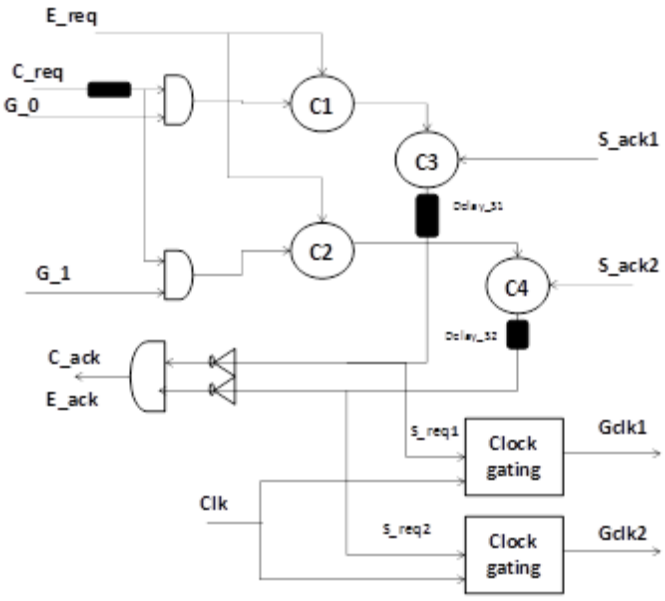


Figure 3-6: Contrôleur d'horloge non-linéaire (Split)

Comme montre la Figure 3-6 , cette structure contient des signaux supplémentaires par rapport aux contrôleurs linéaires. Le signal C_req est le signal de choix. Une porte AND existe entre ce signal et les signaux de garde (G_0 ou G_1) et sa sortie décide à quel signal de sortie le signal E_req doit être envoyé (S_req1 ou S_req2) afin de générer le signal d'horloge local.

3.5. Etude de l'approche sur un micro-processeur

3.5.1. Présentation

Pour montrer l'efficacité de notre structure, nous l'avons appliqué sur un petit micro-processeur implémenté dans un composant programmable présent dans un dispositif de démodulation d'un signal numérique comprenant un filtre analogique comme on peut le voir sur la Figure 3-7.

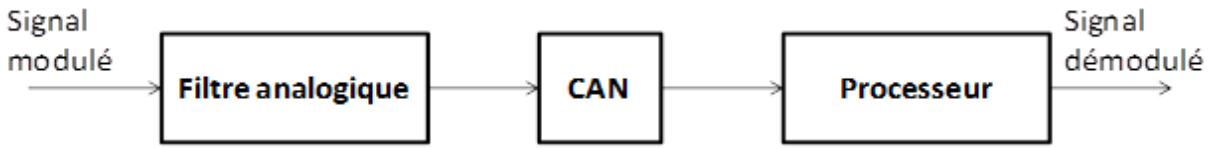


Figure 3-7: Chaine de démodulation

La première étape de la démodulation consiste à utiliser un filtre passe-bas pour créer une modulation d'amplitude sur le signal. En sortie du filtre, le signal est modulé en fréquence, mais aussi en amplitude. Le signal est ensuite numérisé par un CAN (convertisseur analogique numérique). Le but de notre étude est ici de vérifier la validité fonctionnelle de l'approche et d'estimer les gains en consommation au niveau du système.

Le processeur est doté d'une mémoire ROM pour stocker le programme à exécuter, d'une mémoire RAM pour stocker les paramètres et variables et, de ports d'entrées/sorties afin de dialoguer avec son environnement. La Figure 3-8 présente l'architecture du système.

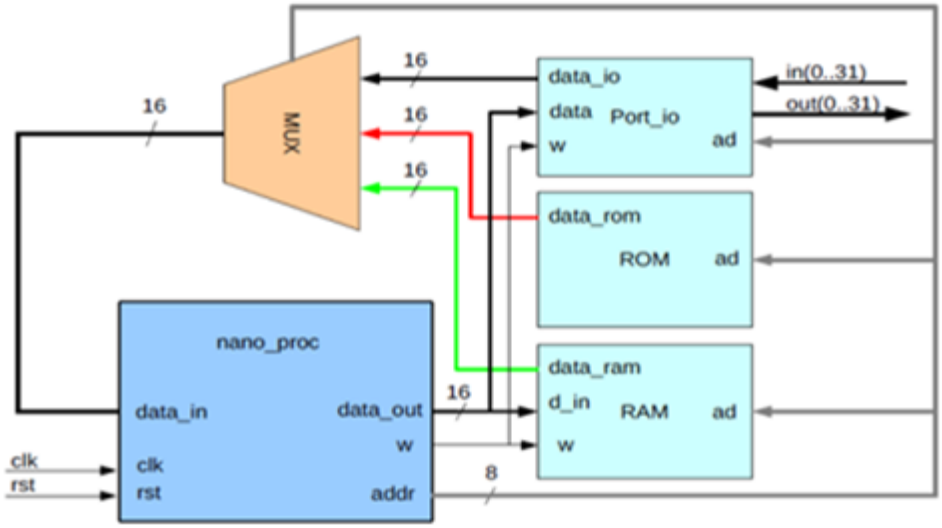


Figure 3-8: Système processeur-périphériques

3.5.2. Simulation et estimation de la consommation

L'ensemble du système est décrit en langage VHDL. Après avoir vérifié le fonctionnement de ce système synchrone, un simulateur analogique (*Nanosim*) a été utilisé pour estimer la consommation. La deuxième étape a consisté à insérer les structures de contrôle de l'horloge en plusieurs endroits. Pour ce faire, on a utilisé deux types de structures (voir chapitre 2) : une structure linéaire au niveau du processeur et une structure de choix au niveau de la RAM et des ports d'entrées/sorties.

.

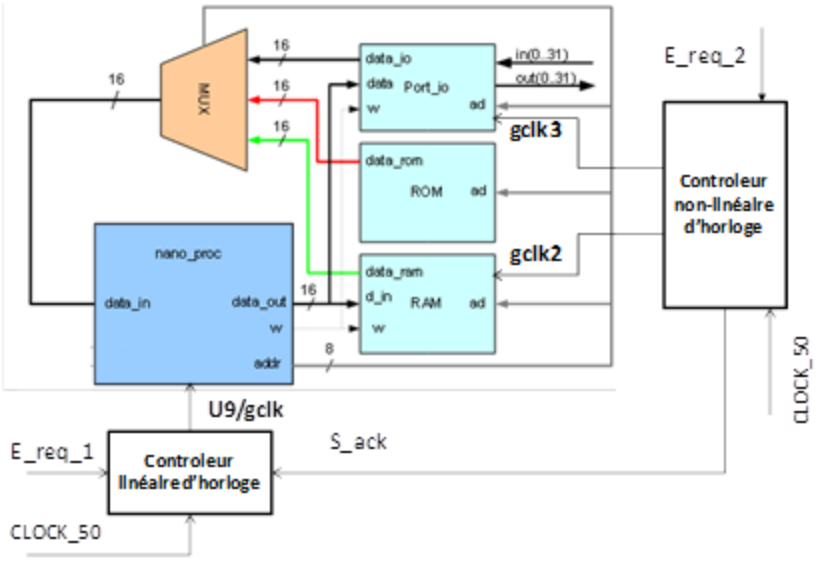


Figure 3-9: Système avec contrôleurs d'horloge linéaire et non-linéaire

La Figure 3-9 illustre l'architecture du système après avoir insérer les structures de contrôle d'horloge au niveau de chaque périphérique. La structure non-linéaire utilisée est un *split* qui permet de propager ou de stopper le signal d'horloge vers la mémoire RAM ou les ports d'entrées/sorties. Notons que les signaux de contrôle E_req et de gardes G_0 et G_1 associés au signal C_req (Figure 3-6) sont issus d'un ADC dans notre système. La première étape a été de vérifier le bon fonctionnement du circuit. Les requêtes d'entrées sont générées au travers du bloc CAN (qui génère l'entrée Filter_In du processeur) placé en amont du processeur. La Figure 3-10 montre que le comportement du système est correct.

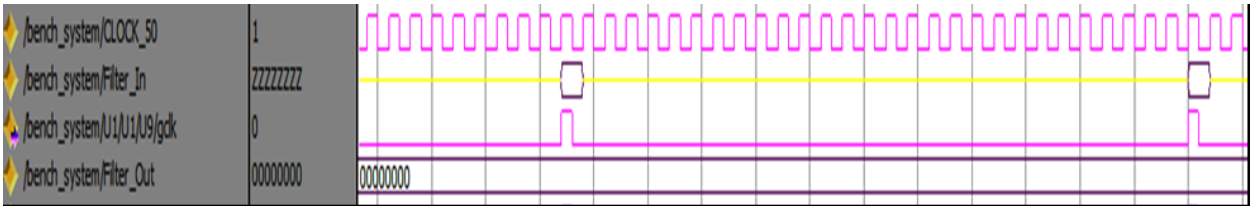


Figure 3-10: Chronogrammes de sortie

Ces chronogrammes affichent les résultats obtenus avec l'outil *Modelsim* (Le signal Filrer_In est le signal d'entrée du registre à décalage en entrée du filtre et le signal Filter_Out représente le signal de sortie du buffer en sortie du filtre). Ils traduisent le bon fonctionnement du système. On remarque qu'à chaque fois qu'on a une donnée prête à être traitée le système propage un cycle

d'horloge vers les périphériques (le signal d'horloge reste au repos sinon). Puisque le fonctionnement du système dépend seulement du processeur et de la mémoire ROM (le processeur lit et exécute le programme stocké dans la ROM), le processeur doit conserver son horloge active

Afin d'évaluer les avantages de la technique, le système a été synthétisé en technologie 0.35AMS (Vdd = 3.3V). La consommation d'énergie du système a été ensuite évaluée avec deux scénarios différents mis en œuvre par le processeur. Ces scénarios ont été conçus en vue d'exploiter plus ou moins la RAM et le port IO. Ces deux périphériques se comportent très différemment en termes de puissance. La RAM consomme plus d'énergie que le port IO qui, dans notre cas, est un périphérique très léger.

Les figures : Figure 3-11, Figure 3-12 et Figure 3-13 montrent les courbes de consommation obtenue avec *Nanosim*. Deux scénarios ont donc été définis pour effectuer les simulations et avoir des éléments de comparaison par rapport à une simulation de référence complètement synchrone. Le premier scénario a utilisé principalement l'horloge du processeur (seules le processeur et la ROM sont actifs) alors que le second scénario a utilisé de façon plus intensive la RAM.

Figure 3-11: Résultat de la consommation quand le système est complètement synchrone: 167 mA

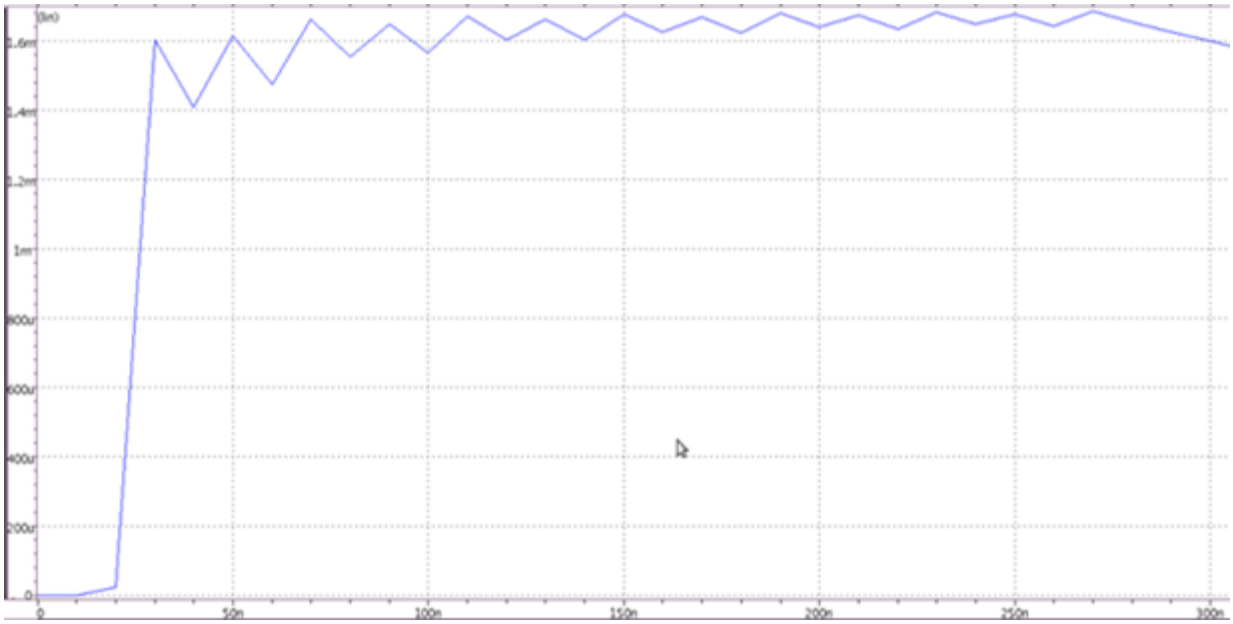


Figure 3-13: Résultat de la consommation asynchrone avec le scénario 1: 1.67 mA

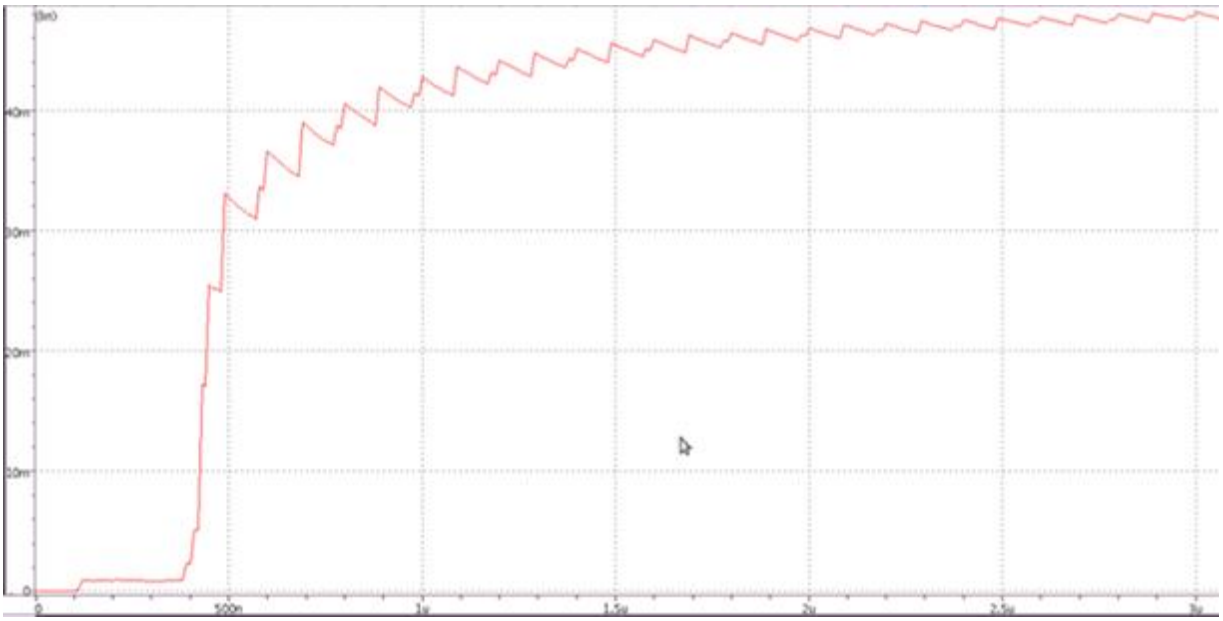


Figure 3-12: Résultat de la consommation asynchrone avec le scénario 2: 48 mA

Le tableau ci-dessous récapitule les résultats de simulation obtenus avec Nanosim.

	Synchrone	Asynchrone	
Courant moyen (mA)	167	Scenario 1	Scenario 2
		1.7	48

Tableau 1: Résultats de simulation

Afin d'obtenir des informations pertinentes, nous avons pris la consommation de courant moyenne obtenue en simulant notre système avec le simulateur électrique rapide Synopsys NanoSim. La consommation pour le système synchrone était de 167 mA sous une tension de 3.3V. Le premier scénario nous a donné une valeur de 1.7 mA tandis que le scénario 2 donnait une consommation de 48 mA. Même si les scénarii de simulation impactent les résultats, l'étude a montré la pertinence et l'efficacité du contrôleur d'horloge utilisé. Avec l'outil *Design Vision*, on a calculé le rapport de surface entre les deux systèmes. La structure de contrôle représente seulement 1.1% du circuit sur un circuit qui est lui-même de taille modeste. Cette très faible augmentation de surface peut pourtant permettre de substantielles économies de consommation d'énergie.

3.6. Conclusion

Dans ce chapitre, une nouvelle méthode pour réduire la consommation dynamique a été définie. Cette nouvelle technique est basée sur le principe du *clock gating* mais associé à un contrôleur asynchrone et distribué dans le système. Ce contrôleur asynchrone dit « WCHB » a été ajouté afin de gérer automatiquement la commande d'extinction de l'horloge. Le principe développé s'appuie sur la signalisation des données. En effet, la signalisation de la présence des données permet d'activer ou non l'arbre d'horloge du bloc(les blocs reçoivent le signal d'horloge si ils ont des données prêtes à être traitées). Cette nouvelle technique présente deux avantages : le premier est un gain en consommation dynamique (seulement les blocs qui traitent consomment) et le deuxième est la simplicité d'insertion de ces contrôleurs asynchrones dans le circuit qui de

surcroît sont de petites tailles. On pourra donc aisément réutiliser des blocs sans nécessité de les re-synthétiser pour y ajouter le dispositif de *clock gating*. Cette technique a été essayée sur un petit processeur avec ses périphériques. Les résultats obtenus montrent l'efficacité de notre technique en termes de consommation dynamique avec un faible coût additionnel en surface. Dans le chapitre suivant, cette technique sera appliquée sur un système utilisant un réseau de communication (NoC) ou un bus.

Chapitre 4

Réduire la consommation dynamique autour d'un système de communication

4.1. Introduction

L'objectif global du projet « HICOOL » est de permettre aux industriels de relever le défi relatif à l'élaboration des circuits intégrés de nouvelle génération dont la consommation doit être faible pour répondre aux attentes du marché et aux contraintes environnementales. Plusieurs études ont été faites aux différents niveaux d'abstraction (AL (*Architectural Level*), RTL (*Register Transfer Level*) et GL (*Gate Level*)) dans le but d'avoir des résultats cohérents entre ces trois niveaux et ainsi d'aider à une prise de décision efficace et optimale dans le flot de conception.

Les systèmes sur puce sont des systèmes qui possèdent nativement un système de communication (Bus/NoC). Ces dispositifs de communication utilisent des protocoles qui leurs sont propres. L'idée est donc d'exploiter ces systèmes afin d'agir sur la consommation de la puce. Dans ce chapitre, nous allons adapter la structure définie au chapitre précédent à un système de communication basé sur le bus AXI en guise d'exemple afin de réduire la consommation. L'approche présentée, même si elle est illustrée par le bus AXI, est reproductible avec la plupart des systèmes de communication. En effet, l'adaptation se fait par le choix de signaux issus du système de communication gérant le protocole et capables de piloter les signaux de contrôle (Req-Ack) de notre structure. Ainsi, il est possible d'insérer nos structures de contrôle d'horloge dans un design existant sans recourir à des modifications des blocs existants. Cette approche est réalisable « à la main » mais se prête bien à une insertion via un flot

automatique. Une structure générique basée sur ce principe a été définie afin qu'elle soit adaptable à la plupart des situations où l'on cherche à réduire la consommation dynamique.

4.2. Les bus AXI

4.2.1. Définition et caractéristiques

Les bus AXI sont depuis longtemps devenus un des standards les plus usités dans l'industrie. Ils sont formés de trois bus principaux : un bus de donnée, un bus d'adresse, un bus de contrôle et de cinq canaux de communications : un canal de lecture d'adresse, un canal d'écriture d'adresse, un canal de lecture de donnée, un canal de lecture de donnée et un canal de réponse [AMB] [ARM 10]. Le bus AXI est un système performant capable de fonctionner à une fréquence élevée et d'utiliser des modes de fonctionnement optionnels favorisant des transactions peu gourmandes en énergie [ZHL 01] [CAO 06]. En outre, les bus AXI peuvent être utilisés pour vérifier la connectivité et la fonctionnalité des processeurs maîtres et des périphériques esclaves. La Figure 4-1 suivante illustre l'architecture générale d'un bus AXI.

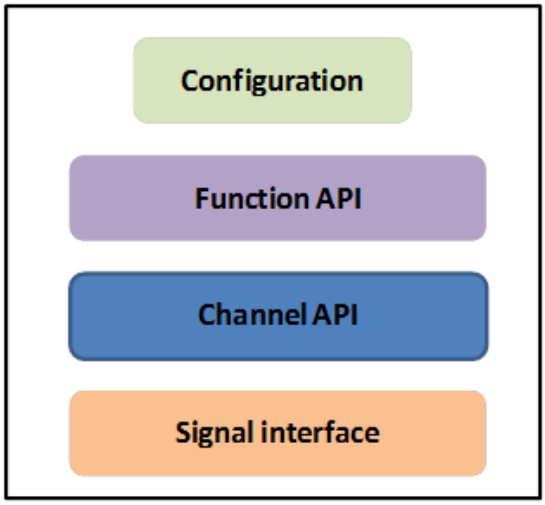


Figure 4-1: Architecture des bus AXI

Chaque bus AXI possède 3 couches : une couche signal, une couche canal et une couche fonctions. La couche signal renferme les ports d'entrées/sorties qui sont équivalents aux signaux d'entrées/sorties. La couche canal est une abstraction des bus de données, d'adresses et de

contrôle qui sera explicitée dans le paragraphe suivant. Le niveau fonction explicite la liste des tâches au niveau contrôle. Enfin, le niveau configuration définit la largeur des bus d'adresse et de données ainsi que d'autres paramètres [ARM 10].

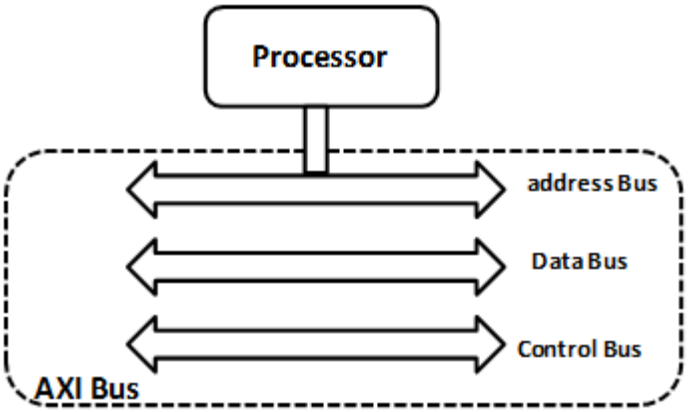


Figure 4-2: Vue général de la couche « Canal » dans le bus AXI

La Figure 4-2 ci-dessus illustre une vue générale de la couche « Canal » présentée dans l'architecture des bus AXI. Le canal d'adresse comporte des informations de commande décrivant la nature des données à transférer. Les données sont transférées entre maître et esclave en utilisant soit :

- « Ecriture des données » pour transférer la donnée du maître vers l'esclave
- « Lecture des données » pour transférer la donnée de l'esclave au maître

Dans un bus AXI les canaux de communication sont indépendants [ALK 15]. Chaque canal utilise un ensemble de signaux d'information ainsi que des signaux de contrôle « Valid » et « Ready » qui fournissent le mécanisme de « handshake ». Ce mécanisme assure le transfert entre deux périphériques. En effet, la source utilise le signal « Valid » pour signaler qu'une adresse est valide et qu'une donnée ou une information de contrôle est disponible. Quant au bloc destination, il utilise le signal « Ready » pour indiquer sa disponibilité. Le transfert a lieu quand ces deux signaux sont actifs. Il est à noter qu'en mode « burst », les canaux de lecture et d'écriture utilisent le signal « Last » pour indiquer la fin du transfert des données [XIA 09]. Nous avons ici aussi un signal qui permet de faire du « handshake ».

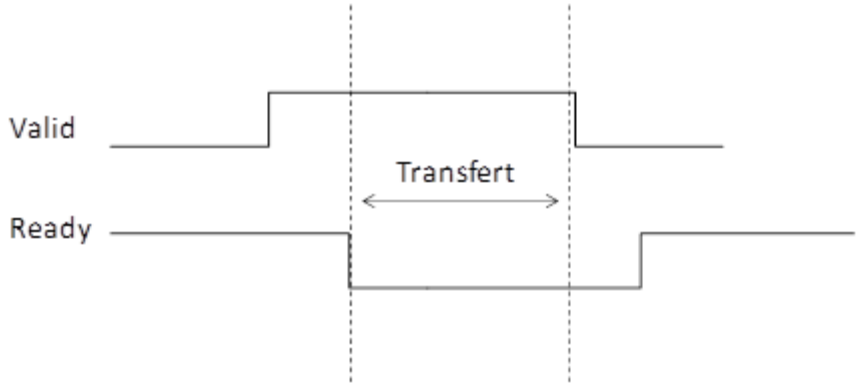


Figure 4-3: Protocole handshake dans un bus AXI

Finalement, comme on peut le voir sur la Figure 4-3, les bus AXI utilisent un protocole de communication qui est similaire à celui des systèmes asynchrones (protocole 4-phases) [REN 98]. Dans notre étude, on s'est intéressé aux signaux de contrôle sortant du bus et plus particulièrement les signaux Wvalid, Wready, Rvalid et Rready pour insérer notre structure de contrôle dans le circuit afin de générer ses signaux d'interface (requêtes et acquittements).

4.2.2. Communication dans un bus AXI

Dans le bus AXI, nous nous intéresserons aux deux modes de communication : la communication en lecture (Read communication) et la communication en écriture (Write communication) (cf. Figure 4-4). La première sert à transférer les données de l'esclave au maître et la deuxième l'inverse. Dans une communication en lecture, le bus AXI utilise les signaux Rvalid et Rready pour déclencher le transfert. En effet, l'esclave génère un signal Rvalid pour indiquer la validité des données à transférer. Ensuite, le maître génère le signal Rready pour signaler qu'il est prêt à recevoir ces données. En mode burst, un signal supplémentaire « Rlast » apparaît et passe à '1' pour indiquer la fin du transfert. De la même façon, une communication en écriture se passe de façon similaire mais avec un transfert des données du maître vers l'esclave. Le signal Wvalid généré par le maître passe à '1' pour indiquer qu'il y a des données prêtes à être traitées. L'esclave, de son côté, génère un signal Wready pour dire qu'il est prêt à les recevoir. C'est le signal « Wlast » qui indique la fin de transfert. Il est à noter que le transfert aura lieu si et seulement si les deux signaux « Valid » et « Ready » sont actifs.

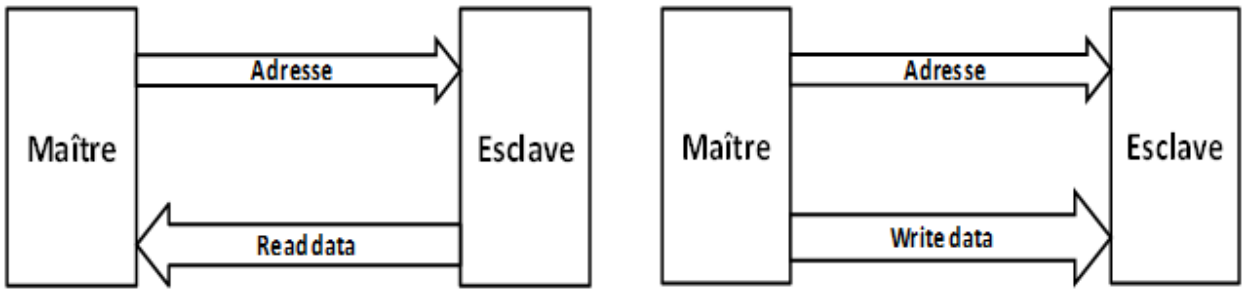


Figure 4-4: Transactions Lecture/Ecriture

4.3. Insertion du contrôleur d'horloge

Afin d'essayer la structure proposée au chapitre précédent, on doit suivre un flot rigoureux pour l'insérer dans un circuit car l'objectif final est de parvenir à automatiser autant que possible l'opération. On a donc défini deux méthodes d'insertion : une première manuelle pour mettre les choses en place et une deuxième automatique qui est réalisable avec l'outil STAR de DEFACTO technologie.

4.3.1. Insertion manuelle

Une fois le contrôleur conçu, la structure doit être insérée dans le système. Le procédé d'insertion consiste à détecter les signaux de commande spécifiques du protocole du bus AXI, de les couper et d'insérer le contrôleur d'horloge. Comme on l'a déjà indiqué, seule la partie contrôle du bus AXI nous intéresse. En effet, certains signaux de commande du bus sont utilisés pour générer les signaux de synchronisation du contrôleur (requêtes/acquittements). Pour adapter notre structure avec le système basé sur un bus AXI (ou tout autre système de communication, NoC), un *wrapper* doit être conçu afin d'adapter les signaux du bus à ceux de notre contrôleur.

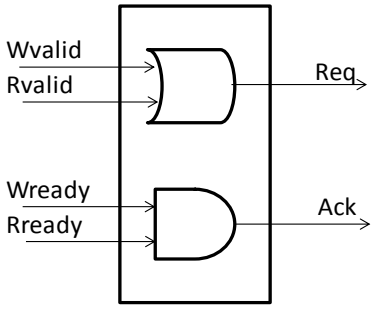


Figure 4-5: Wrapper

La Figure 4-5 montre le *wrapper* qui permet d'adapter le contrôleur d'horloge au protocole du bus AXI. Il est formé de deux portes logiques : une porte AND et une porte OR qui permettent de générer les signaux de contrôle de notre contrôleur (Req-Ack). L'objectif est de faire correspondre les signaux de contrôle du bus AXI avec ceux du contrôleur. En se référant au paragraphe précédent, on remarque que le bus AXI utilise le même principe que les circuits asynchrones (protocole Handshake). Donc pour faciliter l'insertion, il nous faut juste identifier des signaux qui doivent être exploités avec le contrôleur (Wvalid, Wready pour le mode en écriture et Rvalid, Rready pour le mode en lecture). La Figure 4-6 illustre la technique d'insertion manuelle entre un bus AXI et un périphérique.

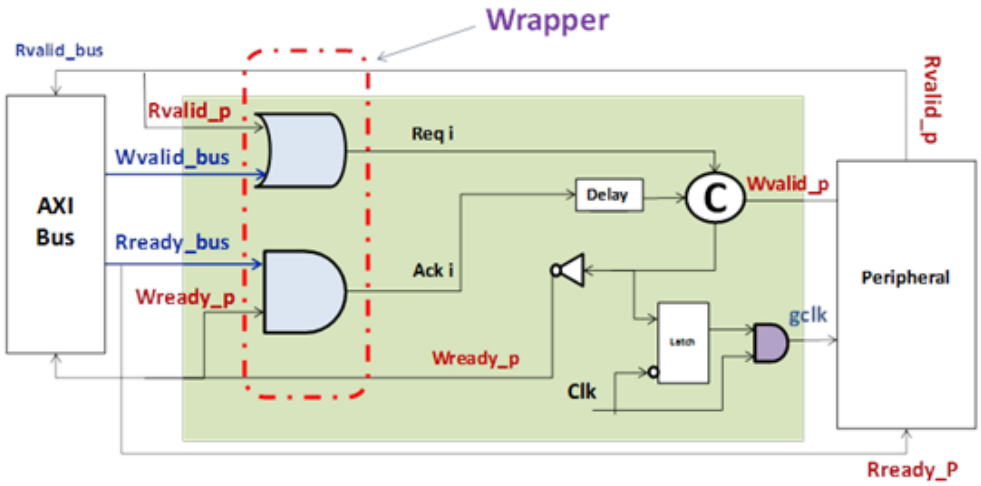


Figure 4-6: Méthode d'insertion manuelle du contrôleur d'horloge

Cette méthode consiste donc à couper les signaux de contrôle entre le bus et le périphérique afin d'insérer le contrôleur. Après insertion, on reconnecte les signaux afin de générer l'horloge du périphérique. Le *wrapper* sert à générer les signaux de synchronisation (requête-acquittement du contrôleur asynchrone). Sur la Figure 4-6, les signaux qui ont le suffixe « _P » correspondent au périphérique. Cette figure illustre le cas où on veut implémenter le contrôle de l'horloge avec des transactions en lecture et en écriture sur le périphérique. Dans le cas de l'écriture, il suffit d'avoir une transition sur le signal Wvalid_bus (Wvalid_bus passe à '1' et par conséquent le signal Req passe aussi à '1') pour signaler qu'il y a une donnée prête à être écrite dans le périphérique. Dans ce cas, le bus AXI doit attendre le signal Wready_P (Ack passe à '0') avant

de commencer la transaction. De même, pour la lecture de données, le périphérique génère un signal Rvalid_P et attend le retour du bus AXI. Donc dès qu'il reçoit le signal Rready_bus, la transaction commence. On remarque que notre technique d'insertion est basée sur l'usage de signaux bien spécifiques qui existent sur quasiment tous les systèmes de communication. Il est donc tout à fait envisageable de déployer cette approche sur un autre système de communication moyennant le développement d'un nouveau *wrapper*.

4.3.2. Insertion automatique

Les travaux présentés dans cette partie ont été faits en collaboration avec nos partenaires : STMicroelectronics Grenoble et DeFacto Technologies. DeFacto technologies a développé un outil nommé « STAR » dans lequel l'insertion de structures additionnelles dans un circuit peut se faire automatiquement. Dans notre système, les contrôleurs doivent être insérer au niveau des blocs qui sont connectés au bus AXI. La Figure 4-7 suivante montre le flot d'insertion automatique.

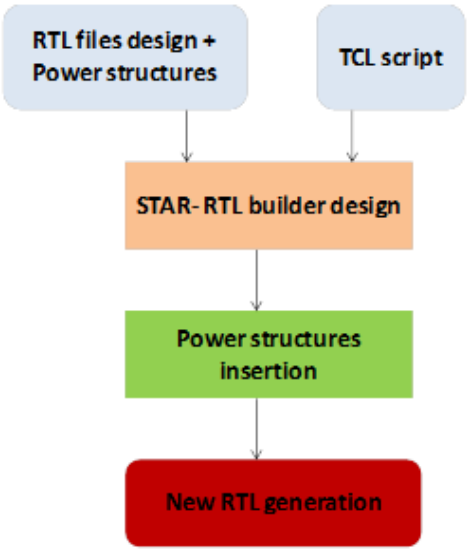


Figure 4-7: flot d'insertion automatique

Comme on l'a déjà précisé, pour un système utilisant le bus AXI, chaque périphérique utilise une interface bus (bus slave) qui assure la communication entre le bus AXI et le périphérique. Le flot d'insertion peut se traduire par l'algorithme suivant :

1. Parcourir la hiérarchie

2. Détecter les interfaces d'interconnexion et les signaux de contrôle
3. Couper les signaux de contrôle et insérer les *wrappers*
4. Insérer les contrôleurs
5. Ajouter les connexions entre : le bus AXI, les *wrappers*, les contrôleurs et les périphériques
6. Générer la nouvelle *netlist* du système

La Figure 4-8 résulte de la méthode d'insertion automatique. On voit clairement qu'au début le périphérique (Sram) est directement lié au bus esclave (AXIMemCtrl). Après insertion, l'interconnexion entre ces deux blocs (AXIMemCtrl et Sram) se fait via le *wrapper* et le contrôleur d'horloge. La Figure 4-8 ci-dessous a été générée avec l'outil STAR de DeFacto technologies.

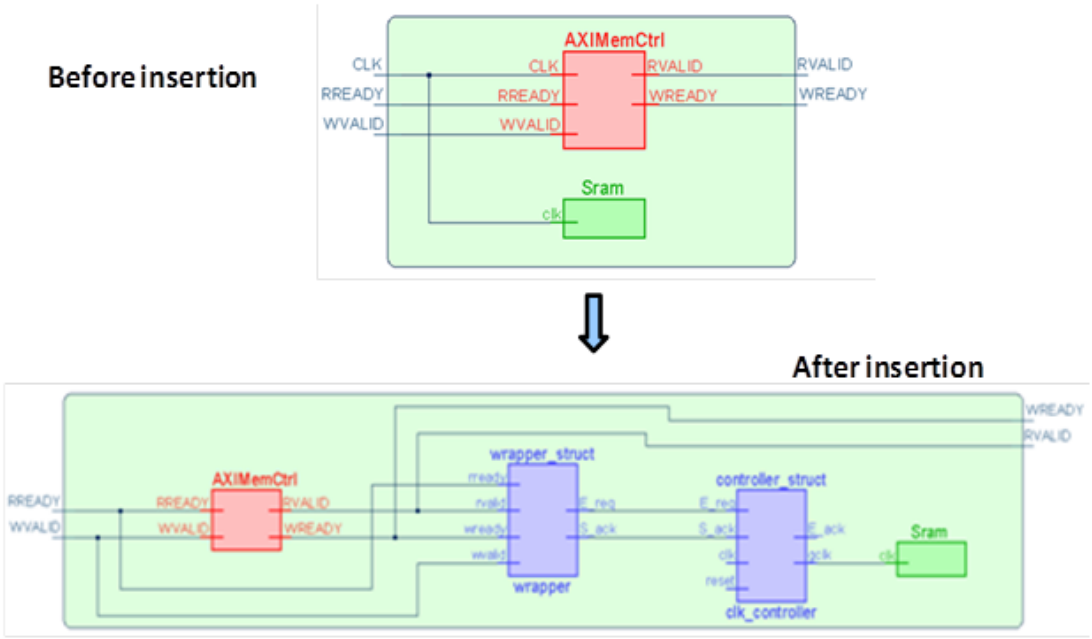


Figure 4-8: Insertion automatique du contrôleur d'horloge dans un système à base du bus AXI

Sur cette figure, le *wrapper* (Wrapper_struct) récupère les signaux de contrôle de l'interface bus esclave (AXIMemCtrl) et du périphérique (Sram) afin de générer les signaux de synchronisation qui servent à générer le signal « gclk ». On notera que l'horloge au niveau du bloc AXIMemCtrl reste inchangée (c'est l'horloge du système).

4.4. Test de mémoires : SPRAM_0 et SPRAM_1

Notre étude sur le bus AXI a commencé sur un petit circuit implémenté en technologie AMS 0.35 µm contenant deux mémoires SPRAM0 et SPRAM1. Dans cet exemple, le bus prend en charge les différents modes de communication (simple, burst, …) fournissant ici un service similaire au bus sans contrôleur asynchrone. Nous partons d'un circuit de test fourni par ST Microelectronics complètement synchrone. Seul le signal global CLK gère l'avancement des données. Le but est ici de valider notre approche afin d'estimer le gain en consommation après avoir appliqué notre structure sur ce système. La Figure 4-9 ci-dessous illustre le système qui sert de test.

L'étude qui suit est divisée en trois étapes : insertion de la structure, validation du fonctionnement et estimation de la consommation.

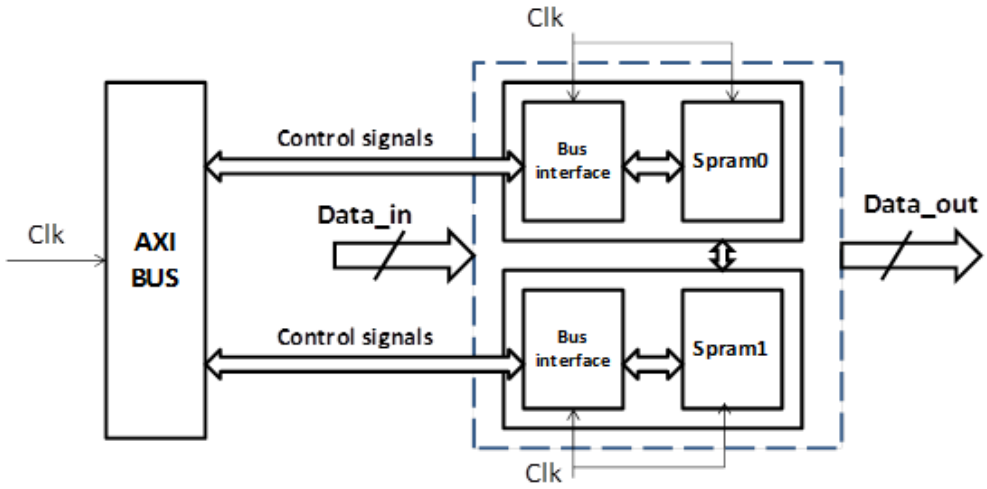


Figure 4-9: Circuit de test avec le bus AXI et deux mémoires

Nous nous concentrons ici que sur le bus de contrôle car il est le seul à rendre compte des échanges d'informations entre le périphérique et le bus AXI. Ses signaux sont utilisés pour générer les signaux de synchronisation (REQ et ACK) de la structure asynchrone comme nous l'avons mentionné auparavant.

Le but de cette expérience est d'insérer notre structure de façon systématique et automatique dans un système utilisant le bus AXI afin d'en réduire la consommation. Pour cela,

on contrôle localement chaque bloc dont le but de l'activer si et seulement si une donnée est prête en entrée. La Figure 4-10 montre le circuit après l'insertion du contrôleur d'horloge.

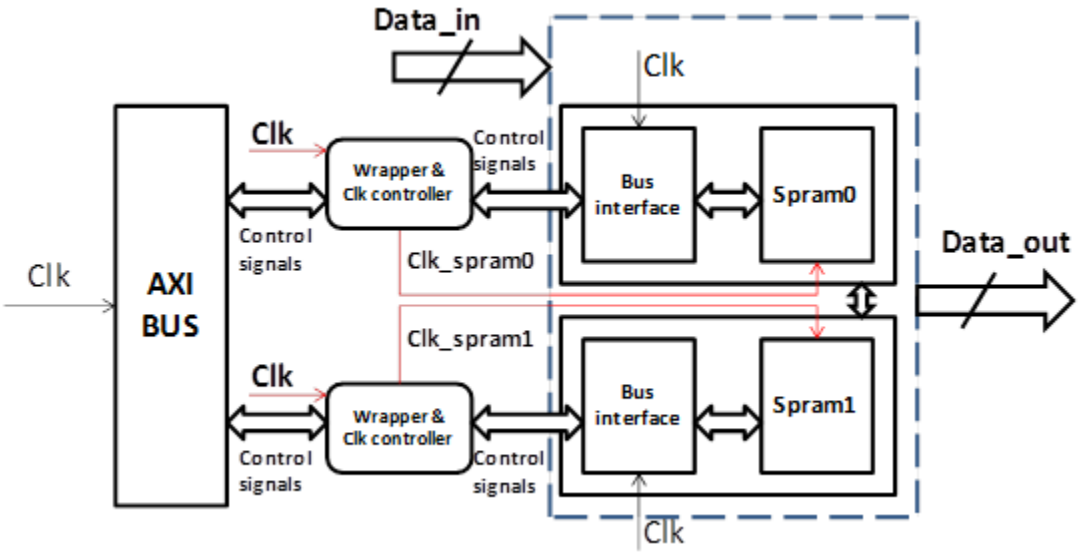


Figure 4-10: Le circuit de test après insertion des contrôleurs asynchrones

Comme cela a déjà été mentionné, les bus AXI sont formés de trois bus principaux : le bus de données, le bus d'adresse et le bus de contrôle. Les signaux de contrôle qui existent à l'entrée du *Wrapper* servent à gérer les deux types de communication du bus AXI (communication Lecture/Ecriture). Le signal Wvalid indique une donnée valide à écrire et Rvalid indique une donnée valide à lire. Les signaux Wready et Rready acquittent respectivement ces deux signaux. On notera que, dans le cas où les acquittements seraient actifs à l'état haut, la porte AND est à remplacer par une porte OR. Le circuit maître génère un signal « valid » pour indiquer la validité des données (REQ passe à '1'), le périphérique esclave renvoie le signal « ready » d'acquittement pour dire qu''il est prêt de recevoir la donnée. Le transfert aura lieu quand requête et acquittement seront actifs simultanément. Sur la Figure 4-10, le *wrapper* reçoit les signaux de contrôle puis génère les signaux de synchronisation pour les contrôleurs d'horloge fournissent les deux horloges pilotées (*gated clocks*) de SPRAM0 et celle de SPRAM1. Avec l'outil *Modelsim,* le fonctionnement du circuit est vérifié et on s'assure que le comportement est identique à celui que nous avions avant insertion. Les chronogrammes sont affichés sur la Figure 4-11. Le mode

de fonctionnement du circuit est présenté en mode burst. Les signaux de contrôle passent à '1' quand on a une donnée valide et les signaux d'acquittement descendent à '0' à la fin du traitement. On observe sur les chronogrammes que les deux blocs SPRAM0 et SPRAM1 ont une horloge active quand ils sont utilisés et sans activité ni signal d'horloge sinon. Ils restent donc en mode d'attente d'une éventuelle donnée. En mode burst, un signal supplémentaire apparait en entrée et en sortie, c'est le signal « last » (Wlast en mode écriture et Rlast en mode lecture). Ce signal passe à '1' pour signaler la dernière donnée dans les deux modes (lecture ou écriture).

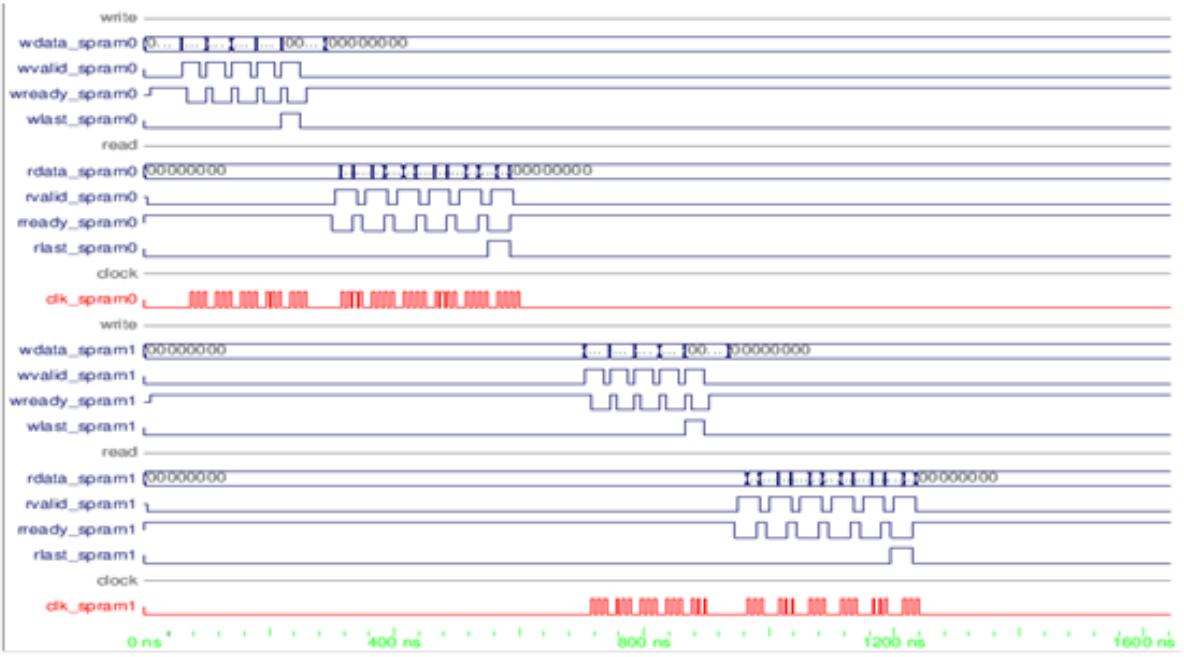


Figure 4-11: Résultats de simulation sous Modeslim

L'état de chaque bloc est conditionné par les données qui se trouvent sur ses entrées. On notera que dans un système utilisant les bus AXI, l'accès à plusieurs mémoires ne peut pas se faire simultanément. Cela induira nécessairement un gain sur la consommation. Pour estimer la consommation, on a eu recours à l'outil *Design vision*. Le Tableau 2 ci-dessous affiche les résultats obtenus en mode synchrone et en mode asynchrone ainsi que l'estimation en surface.

	Synchrone	Asynchrone
Surface(%)	-	3
Consommation (mW)	11	7

Tableau 2: Consommation et surface estimés pour chaque bloc

Ces résultats sont donnés pour la nouvelle *netlist* générée par l'outil STAR. En regardant les rapports de surface, on remarque que la structure de contrôle représente un faible pourcentage du circuit global (3%). Il est à noter que ce pourcentage serait probablement bien plus faible sur un circuit plus conséquent. Notre structure nous a donc permis de gagner presque 40% d'énergie grâce à l'efficacité de la structure. Comme seule une des deux mémoires est active à la fois, il est donc facile d'expliquer ce gain. On voit ici l'avantage principal du dispositif : un gain substantiel en consommation avec un faible coût en surface.

4.5. Test STMicroelectronics (circuit en technologie 28 nm FDSOI)

STMicroelectronics Grenoble est un de nos partenaires dans ce projet. Il s'occupe notamment du développement des cellules à contrôler. Dans un premier temps, les travaux menés dans ce projet visent à proposer des techniques d'estimation de la consommation aux niveaux RTL (Register Transfer Level) et GL (Gate Level) afin de valider notre approche. Par la suite, on remontera en abstraction jusqu'à un niveau AL (Architecture Level) afin de décider au plus tôt dans le flot des choix d'implantation de nos contrôleurs asynchrones. Le circuit présenté dans ce paragraphe a été livré par STMicroelectronics afin de valider la technique proposée.

4.5.1. Présentation du circuit

Afin de valider l'efficacité de notre contrôleur, on l'a appliqué à un circuit industriel conçu par STMicroelectronics en technologie FDSOI 28 nm. Ce test industriel, plus riche et plus complexe, est similaire fonctionnellement à l'exemple précédent. Il nous permettra de valider le fonctionnement de notre structure de contrôle après avoir l'insérer automatiquement. Le système est formé de trois mémoires (*1024 words stack and 32 bit width*) qui sont connectées au bus AXI

comme indiqué sur la Figure 4-12. Pour chaque banque de mémoire, l'interface avec le bus se fait par un contrôleur qui permet de gérer les lectures / écritures avec le protocole AXI.

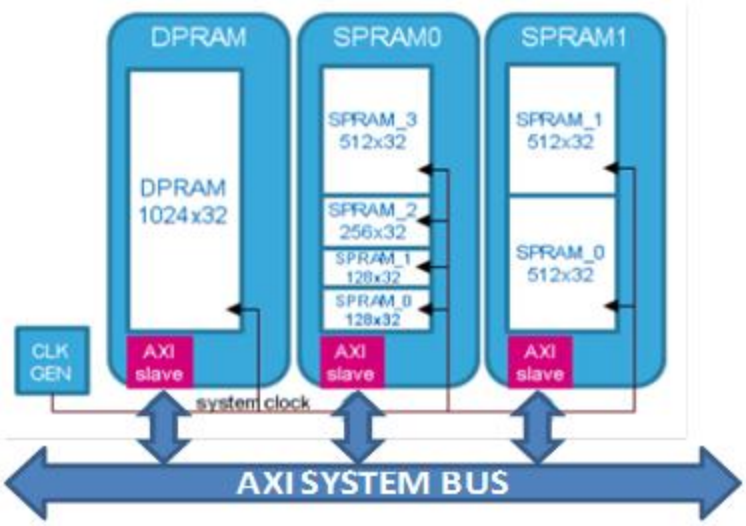


Figure 4-12: Circuit de test avec 3 mémoires sur un bus AXI

Ce système est implémenté en technologie FDSOI 28nm. On remarque la présence des interfaces AXI slave pour chaque périphérique afin d'assurer la communication entre le bus AXI et les mémoires RAM. Pour valider la réduction de puissance apportée par les contrôleurs, il a été choisi de les insérer dans les mémoires SPRAM0, SPRAM1 et DPRAM, de sorte que l'horloge de ces IPs ne soit active que si et seulement si le bus AXI envoie les signaux assurant la génération des requêtes aux contrôleurs (et donc les données). Le reste du temps, l'horloge sera inactive dans chacun de ces modules. Le test bench émule donc un maître qui accède successivement aux blocs DPRAM, SPRAM0 et SPRAM1 pour y effectuer des opérations d'écriture et de lecture. Pour toute opération W / R (Ecriture/Lecture), le bus fournit des signaux de commande au bloc esclave et attend, en retour, le signal d'acquittement du bloc esclave indiquant la fin de la transaction et l'extinction du signal d'horloge. Cela ne prend en général que quelques cycles d'horloges pour effectuer une opération de lecture ou d'écriture simple.

4.5.2. Vérification de fonctionnement et estimation de la consommation

Pour vérifier le fonctionnement de notre système et avant d'estimer la consommation, on a eu recours à l'outil VCSMX qui est capable d'analyser, compiler et simuler les descriptions *Verilog* d'un système. On notera qu'on garde le même test *bench* avant et après l'insertion des contrôleurs asynchrones. La Figure 4-13 illustre les résultats de simulation de notre système.

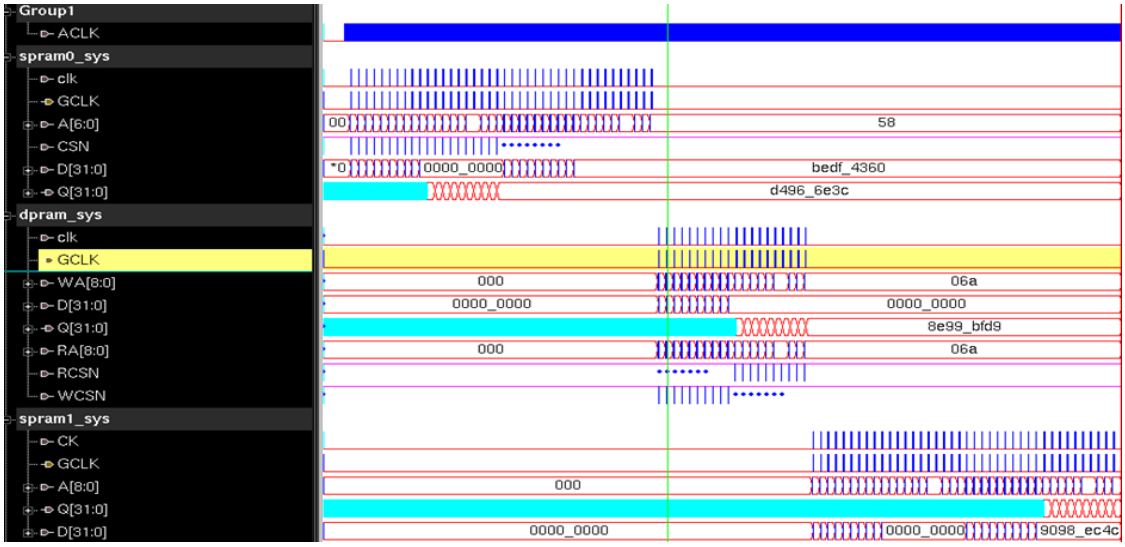


Figure 4-13: Résultats de simulation du circuit de test

Les chronogrammes présentés sur cette figure montrent une simulation du système. Les mémoires sont synchronisés au travers du signal GCLK. La génération des pics d'activité des horloges au niveau de chaque mémoire dépend de son utilisation. Il apparaît clairement que l'horloge de chaque IP est éteinte lorsqu'elle n'est pas utilisée. Le tableau 2 montre les résultats de consommation obtenus avant et après l'insertion des contrôleurs asynchrones. Ces résultats ont été obtenus avec l'outil *PrimeTimePX*.

	SPRAM_0	SPRAM_1	DPRAM
Synchrone (µW)	24	60	13
Asynchrone (µW)	2	12.8	0.6
Surface (%)	0.0008	0.28	0.05

Tableau 2 : Consommation et surface pour chaque bloc

Les résultats présentés dans ce tableau reflètent l'efficacité de ces contrôleurs d'horloge. Il est à noter que la tension d'alimentation est de 0.9V. Après avoir inséré les contrôleurs d'horloge, la décroissance de la consommation de chaque bloc est remarquable. Cette réduction de puissance est corrélée à l'activité de circuit. En effet, l'activité de bus est partagée entre les différents blocs mémoire. En termes de surface, ces contrôleurs sont négligeables par rapport à la taille des blocs mémoire dans lesquels ils sont insérés. L'intérêt d'utiliser des blocs mémoires comme IPs dans cette étude vient du fait que dans les technologies avancées (comme dans notre cas le FDSOI 28 nm), la consommation de ces blocs par rapport aux blocs de calcul est particulièrement élevée.

Finalement, comme cela est relaté dans ce chapitre, nous sommes arrivés à définir une méthodologie complète, systématique, automatisable et susceptible d'aider les concepteurs à insérer des blocs de contrôle de l'horloge. Au-delà de ce travail, dans le contexte plus vaste du projet HiCool, ce travail ouvre aussi des perspectives aux concepteurs pour prendre des décisions pertinentes suffisamment en amont dans le flot pour obtenir, de façon efficace, un circuit à faible consommation, du moins, à ce stade, en ne considérant que l'horloge et la consommation dynamique.

4.6. Conclusion

Vue la complexité des SoCs et l'évolution des technologies, la dissipation de puissance dans un circuit CMOS devient une préoccupation de plus en plus importante. Les techniques d'estimation de la consommation de puissance et l'activité de commutation au niveau transfert de registre (RTL) ont été largement explorées par la recherche industrielle et académique ces dernières années afin d'aider les concepteurs à concevoir des circuits moins consommant. Dans ce chapitre, on a mis en oeuvre une nouvelle technique de réduction de la consommation dynamique. Cette technique basée sur le principe bien connu du *clock gating* a été revisité en exploitant la connaissance des circuits asynchrones et des systèmes de communication sur puce qui eux aussi, bien souvent, exploitent des protocoles de communication de type Requête-Acquittement.

En effet, la technique de *clock gating* est une manière efficace pour réduire la consommation dynamique. En revanche, cette technique a habituellement besoin d'un calcul supplémentaire

pour indiquer quelle partie de circuit doit suspendre l'activité de l'horloge. L'ajout d'une interface asynchrone, a permis d'aller plus loin avec ce principe de *clock gating*. En effet, l'ajout de contrôleurs asynchrones sur chaque bloc synchrone connecté à un système de communication permet de tirer parti de signaux de contrôle existants et inexploités pour ce besoin. Par ailleurs, l'approche proposée constitue une simplification dans la conception d'un système sur puce dans la mesure où il est possible de réutiliser les blocs (IPs) directement sans nécessité de les reconcevoir. Cette nouvelle méthodologie doit être utilisée comme un complément aux techniques de *clock gating* car elle évite d'intervenir sur les blocs déjà conçus. Son seul impact dans le flot est une modification du back-end global de la puce et plus spécifiquement au niveau du routage du système de communication. Elle ouvre donc des perspectives pour contrôler l'horloge de circuits ou de blocs synchrones afin d'en réduire la consommation dynamique. Cette approche a été implémentée avec succès sur des systèmes utilisant le bus AXI. Les résultats obtenus traduisent l'efficacité de notre approche en termes de consommation avec un surcoût en surface négligeable par rapport à la surface globale de la puce. Le chapitre suivant se consacrera aux techniques de réduction de la consommation statique qui est aussi un paramètre important pour les circuits qui sont alimentés sur de longues durées par des batteries.

Chapitre 5

Consommation statique et techniques de réduction

5.1. Introduction

Dans les systèmes embarqués, la consommation et l'efficacité énergétique sont des paramètres qui ont très souvent une très haute priorité. L'efficacité énergétique est importante pour améliorer les performances des systèmes sur puce et augmenter le temps de fonctionnement des dispositifs alimentés par une batterie.

L'évolution de la technologie des circuits intégrés a induit une diminution de la tension d'alimentation ainsi que de la tension de seuil [GOU 01]. La consommation statique, due à cette faible tension de seuil, peut constituer une part considérable de la consommation globale d'un circuit. Ce problème est dû au courant de fuite dans les transistors, susceptible de représenter une fraction importante de la consommation globale (parfois jusqu'à 50%). Dans ce cadre, le laboratoire TIMA s'est penché sur une structure limitant les pertes d'énergie issues des courants de fuites (leakage). Dans ce chapitre, on rappellera les techniques usuelles de réduction de la consommation statique comme le *Mutli-threshold*, le *power gating* et le *body biasing* [BOR 05] [PFI 07]. Une nouvelle structure basée sur notre stratégie asynchrone est ensuite présentée dans le chapitre. Elle assure la gestion automatique des deux techniques suivantes : le *power gating* et le *body biasing* afin de réduire la consommation statique dans le circuit.

5.2. Source de leakage

Comme cela a déjà été mentionné, la consommation dans un circuit est divisée en deux parties : la consommation statique d'une part et la consommation dynamique (Chap. 3) d'autre part. Dans ce chapitre, on va illustrer les différentes sources produisant la partie statique de la

consommation ainsi que les techniques de réduction. La consommation statique est donnée par l'équation (i) suivante :

$$P_{Leak} = I_{Leak} \, x \, V_{DD} \qquad \text{(i)}$$

En particulier, avec la réduction de la tension de seuil (transistor haute performance), la puissance liée aux fuites est devenue un élément important de la consommation totale d'énergie. La réduction des courants de fuite doit être réalisée en agissant à deux niveaux différents. Le premier est le niveau technologique et le second est le niveau conception. Au niveau technologique, la réduction des fuites peut être obtenue en contrôlant les dimensions (longueur effective, épaisseur d'oxyde, profondeur de jonction, etc.) et le dopage dans les transistors. Au niveau du concepteur, la tension de seuil et les courants de fuite des transistors peuvent être efficacement contrôlés en agissant sur les tensions aux bornes du drain, de la source, de la grille et du substrat.

Quatre sources de consommation statique sont identifiables dans les transistors CMOS :

1. Reverse-biased junction leakage current (I_{REV})
2. Gate induced drain leakage (I_{GIDL})
3. Gate direct-tunneling leakage (I_G)
4. Sub-threshold (weak inversion) leakage (I_{SUB})

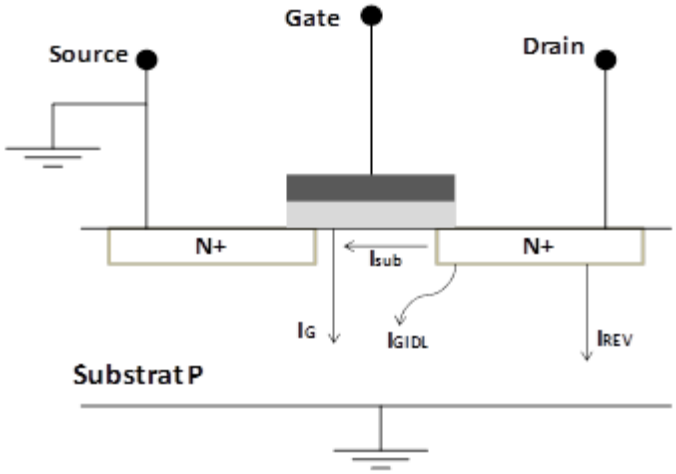


Figure 5-1: Sources de leakage

La Figure 5-1 montre les différentes sources de fuite dans un transistor. Dans les nouvelles technologies CMOS (FDSOI), Isub est le facteur dominant dans la consommation statique [SEM 02]. Dans la suite, on présente succinctement chaque source de *leakage*.

5.2.1. Reverse-biased junction leakage current (IREV)

Le courant de polarisation inverse (Irev) est courant s'écoulant du drain vers le substrat quand le transistor est à l'état OFF. En effet, une jonction polarisée en inverse a deux composantes principales : la première est la diffusion de porteurs minoritaires et la deuxième est l'apparition de paires électron-trou dans la zone de déplétion de la jonction [ZEG 01]. Lorsque les deux zones N et P sont fortement dopés, comme dans le cas des MOSFET avancés, il y a une fuite due à l'effet tunnel qui apparaît (Band to band tunneling BTBT) [TAU 02]. Ce type de fuite peut être considéré comme négligeable par rapport aux autres composantes.

5.2.2. Gate induced drain leakage

Ces fuites (I_{GIDL}) sont dues à l'effet de champ élevé au niveau du drain dans les transistors MOS. Il est constitué par une haute tension drain-substrat avec une haute tension drain-grille. Lorsque la grille est polarisée pour former une couche d'accumulation à la surface du silicium, la surface qui se trouve au-dessous de la grille a presque le même potentiel que le substrat de type P. En raison de la présence des trous accumulés, elle se comporte comme une région P presque dopée comme le substrat. Cela provoque donc une zone de déplétion plus étroite qu'ailleurs. Le rétrécissement de cette zone de déplétion au niveau de la surface provoque une augmentation des courants de fuite du fait du champ électrique local dans cette région [KES 03] [ROY 01].

5.2.3. Gate direct-tunneling leakage (IG)

Les dimensions des transistors diminuant avec l'évolution de la technologie, les épaisseurs d'oxyde de grille ont également diminué favorisant ainsi une augmentation du champ électrique dans l'oxyde [SEM 02]. Ce champ électrique élevé associé à cette faible épaisseur d'oxyde de grille induit des courants par effet tunnel entre substrat et grille.

5.2.4. Sub-threshold (weak inversion) leakage (ISUB)

La fuite liée à la tension de seuil correspond au courant drain-source d'un transistor fonctionnant en régime de faible inversion. Elle apparaît quand la tension de grille V_{GS} est inférieure à la tension de seuil Vth [TAU 01]. Dans les technologies actuelles, ce paramètre est dominant comparativement aux autres sources de fuite présentées ci-dessus [SEM 01]. Ceci est principalement dû au fait que la tension de seuil V_{th} est relativement faible dans les dispositifs CMOS modernes. I_{SUB} est calculée en utilisant la formule (ii) suivante:

$$I_{sub} = \frac{W}{L}\,\mu U_t^{\,2} C_{sth} e^{\frac{V_{GS}-V_{Th}+\eta V_{DS}}{nU_t}} \left(1 - e^{\frac{-V_{DS}}{U_{th}}}\right) \qquad \text{(ii)}$$

Cette équation montre la dépendance entre le courant de fuite et la tension de seuil. W et L représentent la géométrie du transistor (longueur et largeur), µ désigne la mobilité des porteurs, $U_t = KT/q$ la tension thermique à une température T (q la charge de l'électron et k la constante de Boltzmann), $C_{sth} = C_{dep} + C_{it}$ est la somme des capacités de la zone de déplétion, ŋ est le coefficient d'abaissement de la barrière induite par le drain [SHE 13] et n le coefficient de l'effet substrat. Il est donné par :

$$n = 1 + \frac{C_{sth}}{C_{ox}} \qquad \text{(iii)}$$

Dans le cas où source et drain sont suffisamment espacés, l'effet de la zone de déplétion sur le potentiel est presque négligeable. Par conséquent, pour ces dispositifs, la tension de seuil est pratiquement indépendante de la longueur du canal et de la polarisation du drain. Par contre, quand source et drain sont proches nous sommes dans le cadre de ce qu'il est communément appelé les effets canaux courts. La tension de seuil varie en fonction de la polarisation du drain.

L'équation (ii) ci-dessus illustre l'influence de la tension de seuil et de la longueur sur la fuite. La diminution de la tension de seuil fait augmenter la fuite exponentiellement. De plus, la longueur, lorsqu'elle se raccourcit, agît également sur l'augmentation des fuites.

Dans le paragraphe suivant, nous présentons la technologie FDSOI 28 nm de STMicroelectronics qui sera utilisé pour notre véhicule de test.

5.3. La technologie FDSOI 28 nm (Fully Depleted Silicon On Insulator)

La technologie FD-SOI permet de contrôler le comportement des transistors non seulement au travers de sa grille, mais aussi via la polarisation de son substrat, de manière similaire à la polarisation du substrat dans une technologie « *Bulk* ». Dans cette dernière, l'effet de la polarisation est très limité car elle est limitée par les courants de fuite.

En technologie FDSOI, l'architecture des transistors rend cette technique (polarisation de substrat) bien plus efficace. Elle utilise une fine couche dopée et enterrée sous une couche d'oxyde appelée BOX (Buried Oxide) qui isole un film de silicium non-dopé du reste du substrat. Cette couche dopée appelée back-plane (BP) a pour but de contrôler les champs électrostatiques. Ce contrôle est d'autant plus important que le BP à un dopage de type opposé à celui de la source et du drain (Figure 5-2).

L'utilisation d'un oxyde enterré mince présente également un intérêt pour l'ajustement de la tension de seuil des transistors FDSOI. C'est un point clé de cette technologie, permettant de contrôler les fuites et/ou d'augmenter la vitesse d'un même dispositif avec une grande flexibilité.

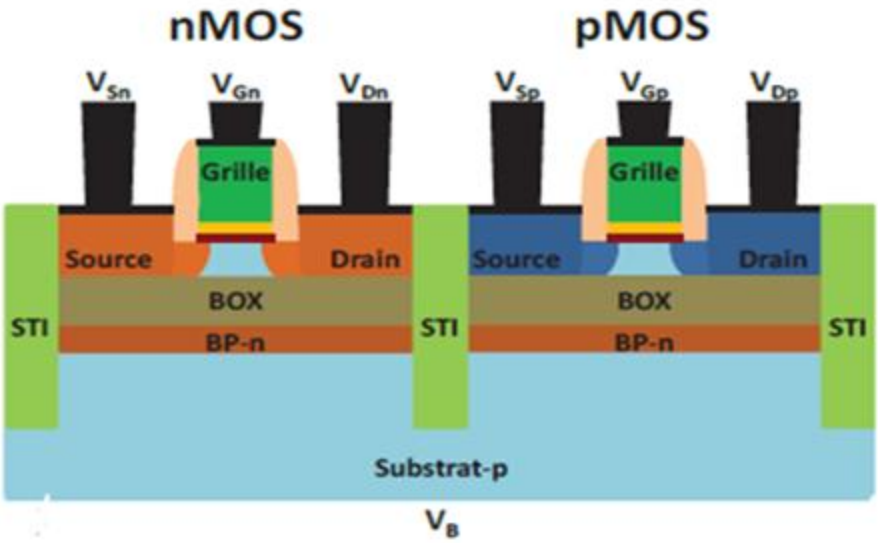


Figure 5-2: Vue schématique en coupe de transistors NMos et PMos fabriqués en technologie FDSOI

L'intégration d'un BP implanté sous le BOX peut être utilisée comme une seconde grille, dont la polarisation VB agit sur les caractéristiques électriques du dispositif. En appliquant une différence de potentiel entre le BP et la source du transistor (négatif pour les NMOS et positive pour les PMOS), la tension de seuil augmente permettant ainsi de réduire le courant I_{OFF} (courants de fuite). Cette technique est communément appelée RBB (*Reverse Back Biasing*). Par opposition, quand la différence de potentiel est positive pour les NMOS et négative pour les PMOS, le courant de drain en régime de forte inversion augmente ce qui permet d'accroître la vitesse du dispositif. Cette technique s'appelle FBB (*Forward Back Biasing*).

La structure (Figure 5-9) conçue dans le cadre de cette thèse permet de polariser le BP de façon à utiliser la technique de RBB durant l'inactivité d'un bloc pour réduire les pertes par courants de fuite et la technique de FBB durant l'activité du bloc pour augmenter sa vitesse de fonctionnement. Ce système permettra d'augmenter de manière significative les performances des circuits FDSOI tout en diminuant les pertes d'énergie. Il offrira alors un avantage certain à la technologie FDSOI développée par STMicroelectronics sur un marché toujours très compétitif.

Enfin, la technologie FD-SOI permet d'atteindre des performances élevées sous une faible tension avec une efficacité énergétique supérieure à celle du CMOS massif.

5.4. Techniques traditionnelles de réduction de la consommation statique

La conception des circuits intégrés passe par plusieurs étapes avec différents niveaux d'abstraction : niveau fonctionnel, niveau architectural (AL), niveau RTL, niveau logique (GL) et, enfin, niveau électrique et physique. Les niveaux RTL et GL sont les niveaux du flot qui nous intéresse dans le cadre de ce travail de thèse. Dans le cadre du projet HiCool, le niveau AL a également été étudié par notre partenaire Docea Power. Dans cette thèse, notre étude se concentre sur le seul niveau RTL mais est également applicable au niveau GL. En effet, le niveau RTL est le niveau qui permet de conserver le plus haut niveau d'abstraction qui permet d'insérer notre structure asynchrone. Par ailleurs, on assiste au développement d'outils qui sont aujourd'hui capable de faire l'estimation de la consommation à ce niveau (comme l'outil Sypglass Power de Synopsys par exemple). Il est donc tout à fait cohérent de travailler à ce

niveau d'abstraction. Le niveau GL est aussi intéressant car il permet d'avoir une estimation plus fine de la consommation. En revanche, la perte de nommage liée à la synthèse logique peut constituer un frein à l'usage de notre technique à ce niveau d'abstraction. Enfin, l'analyse des solutions usuelles de réduction de la consommation statique montre que les niveaux RTL et GL sont partiellement compatibles avec des stratégies telles que le *Multi-threshold*, le *power gating* et le *body biasing* [BOR 05] [PFI 07] mais qu'ils sont des niveaux d'abstraction indispensables à l'analyse de ces techniques.

5.4.1. Multi-threshold

L'équation (ii) illustre la dépendance du courant de fuite aux différents paramètres. L'augmentation de la tension de seuil permet par exemple de réduire la fuite de courant. La méthode de *Multi-threshold-Voltage CMOS (MTCMOS)* permet d'introduire une tension de seuil Vth élevée [MUT 95]. Dans ce cas, on insère des transistors avec des tensions de seuil différentes dans le circuit. Les transistors à basse tension de seuil (LVT ou Low V_T) sont utilisés pour implémenter la logique nécessitant de la performance (chemin critique) afin d'en augmenter les performances alors que ceux qui ont une tension de seuil élevée (HVT ou High V_T) pour l'implémentation de la logique non critique et des « *sleep* transistors » [WEI 01] comme cela est présenté sur la Figure 5-3. Cette figure nous montre une stratégie facilement implémentable qui permet de réduire les fuites combinant des transistors LVT et HVT. La technique est connue sous le nom de MTCMOS.

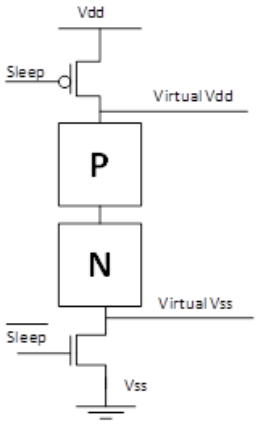


Figure 5-3: Original MTCMOS

En mode actif, l'entrée du sleep transistor PMOS (resp. NMOS) est connectée à 0V (resp. VDD) et l'alimentation virtuelle (resp. la masse virtuelle) est reliée à Vdd (rep. Vss). En mode veille, les entrées sont reliées à Vdd + ΔV (resp. Vss - ΔV) pour couper complètement le courant de fuite. Par ailleurs, cette technique peut être utilisée pour faire du « *power gating* » [MUT 95] [KAO 01].

5.4.2. Power gating

Pour réduire la consommation dynamique, il est souhaitable de couper l'horloge quand il n'y a pas de données à traiter, il est tout aussi souhaitable d'éteindre les blocs (les mettre en mode veille) quand on n'en a pas besoin. Cette technique est connue sous le nom de « *Power Gating* ».

La technique de *power gating* est basée sur l'insertion de deux transistors PMOS et NMOS dans les deux réseaux P et N comme cela est représenté sur la Figure 5-3. Ces transistors peuvent diviser la structure en deux parties : la première est le réseau d'alimentation permanent relié à la tension d'alimentation VDD et la seconde est le réseau d'alimentation virtuel qui anime les cellules et en assure la mise en veille. En mode veille, le *sleep* transistor est en mode OFF. On notera que la diminution des fuites a lieu si ce transistor a une tension de seuil élevée. Dans le cas contraire, la technique n'aura pas l'efficacité attendue. Pour garantir le bon fonctionnement du circuit, le *sleep* transistor doit être bien dimensionné pour minimiser sa chute de tension tant qu'il est en mode ON [ANI 02].

En conclusion, la technique de *power gating* est très efficace pour réduire les courants de fuite dans un circuit CMOS, mais, par contre, elle souffre de quelques inconvénients :

1. Le maintien d'un jeu de transistors HVT et d'un jeu de transistors LVT dans le circuit nécessite des adaptations des procédés de fabrication de la technologie CMOS (par rapport au cas standard avec un seul V_T.
2. L'évolution de la technologie induit une diminution de la tension de seuil des transistors à chaque nouvelle génération. Cela signifie que les fuites vont continuer à croître (malheureusement exponentiellement).
3. Le dimensionnement des sleep transistors n'est pas une tâche triviale pour le concepteur.

5.4.3. Body biasing

Une des méthodes qui permet d'augmenter artificiellement la tension de seuil des transistors afin de réduire les fuites dans un circuit est la méthode dite « *Reverse Body Bias RBB*» [SET 03]. Cette méthode est appliquée en mode veille. La tension de seuil dans un transistor est donnée par la formule (v) suivante :

$$V_t = V_{t0} + \gamma\left(\sqrt{|-2\varphi_f + V_{SB}|} - \sqrt{|2\varphi_f|}\right) \qquad (V)$$

Avec V_{t0} est la tension de seuil quand $V_{SB} = 0$, ϕ_f est le potentiel de Fermi et γ est le coefficient de l'effet substrat [KAN 04]. A partir de cette formule on peut augmenter la tension de seuil. Pour un NMOS (resp. PMOS), la méthode de la polarisation en inverse (RBB) sert à augmenter la tension de la région p-well (resp. n-well).

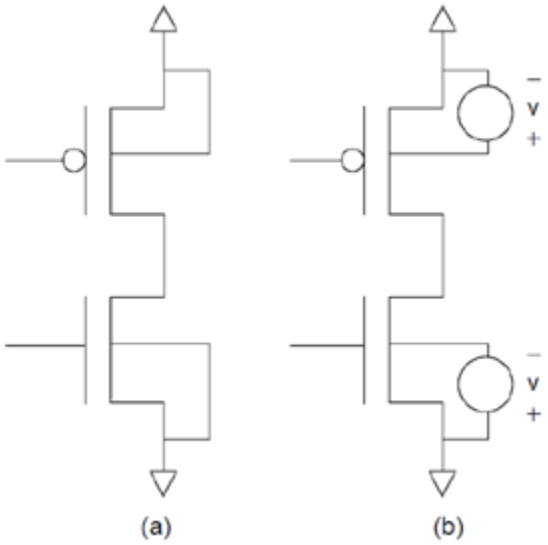


Figure 5-4: Polarisation du substrat

Un avantage important de cette technique c'est que la tension de seuil peut se réduire pour les lourds traitements afin d'augmenter la vitesse des cellules.

La Figure 5-4 montre la technique RBB. Normalement les NMOS (resp. PMOS) sont reliés directement à GND (resp. VDD) (Figure 5-4 (a)). Mais après polarisation, une source de tension est insérée entre le substrat et la source (Figure 5-4 (b)), Si la tension appliquée au substrat est

positive, la largeur de la zone de déplétion diminue et par conséquent la tension de seuil V_{th} diminue, ce principe est connu sous le nom de FBB (*Forward Back Biasing*). Par contre, si la tension appliquée est négative, on est dans le cas du RBB (*Reverse Back Biasing* ou polarisation en inverse), la largeur de la zone augmente et la tension de grille devient donc plus élevée, ce qui entraine une augmentation de la tension de seuil.

5.5. Application des techniques de *power gating* et de *body biasing* pour réduire les courants de fuite

Enfin de parvenir à implémenter ces deux techniques, il est nécessaire de développer des structures d'isolation (ou *switch*) pour réaliser du power gating ainsi que des générateurs capables de réaliser les tensions nécessaires à la polarisation d'un substrat. Dans le cadre du projet HiCool, nous avons pu travailler sur la technologie FDSOI 28 nm de STMicroelectronics qui dispose de structures polarisantes et d'isolation. En revanche, le contrôleur ou le processeur en charge de les piloter est à la discrétion du concepteur. Afin de comprendre les expérimentations, voici une courte description des structures que l'on peut trouver dans la technologie FDSOI 28 nm de STMicroelectronics.

5.5.1. Structure d'isolation EP28SOI_VDDI_VDDISWITCH

Ce switch est un commutateur élémentaire de hauteur 20.4 µm et de largeur 6.8 µm pour séparer la tension locale d'alimentation (à un bloc) VDDI de la tension d'alimentation globale du circuit VDDO. Cette cellule est entièrement analogique. Elle est commandée par une cellule appelée contrôleur EPOD, via des signaux analogiques.

5.5.2. Structure d'isolation EP28SOI_VDDI_VDDISWITCH_H

Cette structure est identique à la précédente mais avec une hauteur de 40.8 µm et une tension VDDI plus élevée (le « H » attaché à son nom). Elle sépare les deux tensions VDDI et VDDO. Cette cellule est compatible avec un anneau de type VDDI_H et est commandée par le contrôleur EPOD.

5.5.3. Structure d'isolation EP28SOI_SVDDI_VDDISWITCH

Cette cellule possède une hauteur de 10.2 µm avec une largeur de 6.8 µm. Sa fonction est la même que les deux premières structures présentées. Le « S » dans SVDDI signifie « Small ». Elle est compatible pour les SVDDI de type « ring ». Elle est entièrement analogique et est commandée par la cellule EPOD.

5.5.4. Dispositif intégré de distribution de l'énergie (Embedded Power Distribution)

Notre étude ayant pour but la gestion de plusieurs domaines d'alimentation dans un même circuit, notre design intègre donc une ou plusieurs alimentations commutables. Le concepteur se doit d'utiliser des cellules fournies par la bibliothèque EPOD de STMicroelectronics (Annexe I). Ces cellules doivent être en accord avec des règles fixes et des règles ajustables données dans l'Annexe II. Le diagramme fonctionnel d'un contrôleur EPOD est donné par la Figure 5-5 suivante :

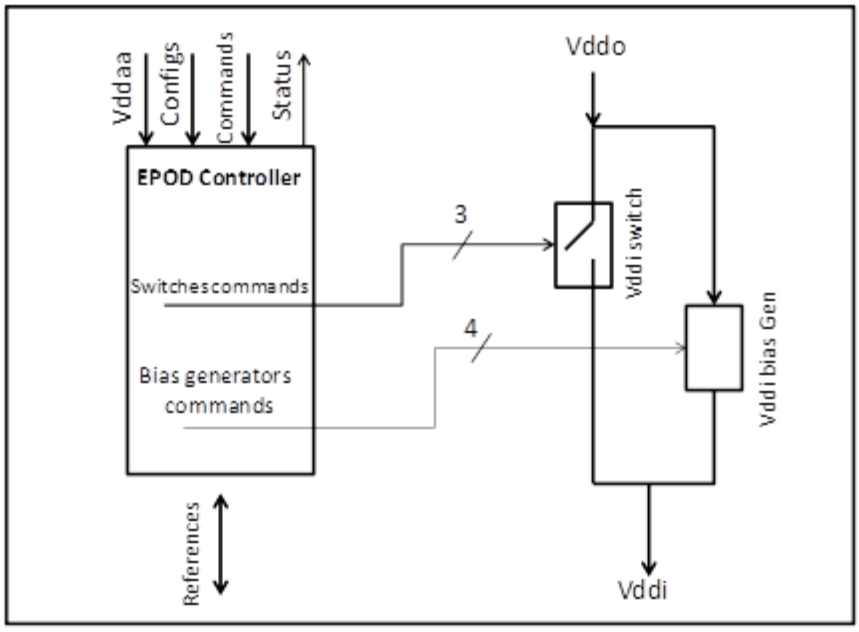


Figure 5-5: Diagramme fonctionnel d'un EPOD

On constate que cette structure est formée de trois blocs: le contrôleur EPOD lui-même qui génère les signaux de commandes, le « Vddi switch » qui implémente la fonction de *power gating* et le générateur de polarisation « Vddi bias Gen » qui gère la polarisation des substrats.

5.5.5. Le contrôleur EPOD

Cette cellule est utilisée pour fournir tous les signaux de commande qui coupe l'alimentation externe en fonction de la non-activité des blocs. La illustre la structure de ce contrôleur.

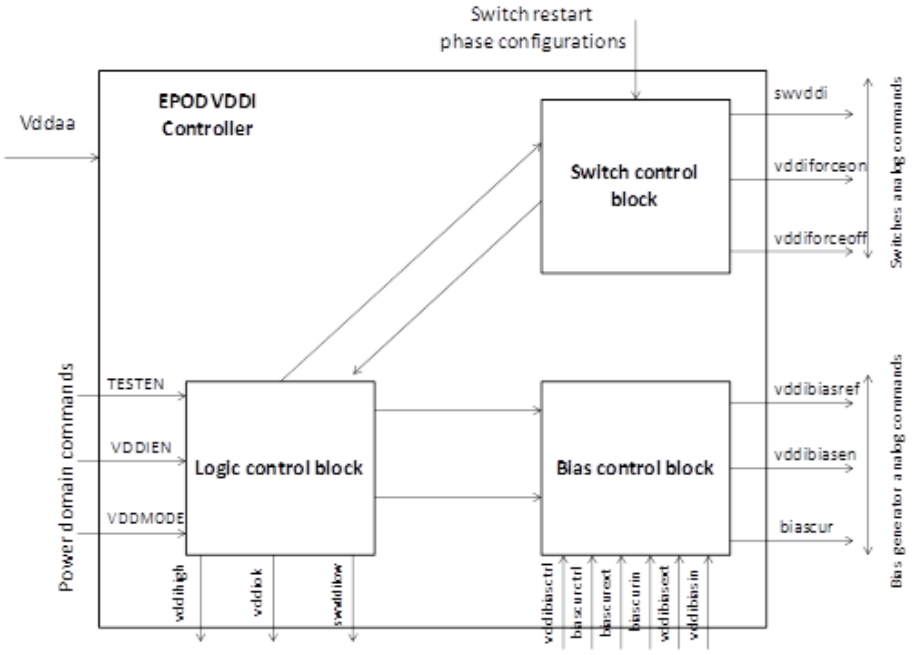


Figure 5-6: Contrôleur EPOD

Le Tableau 3 ci-dessous décrit la fonction de quelques pins. Dans l'annexe III, tous les plots sont décrits avec leur fonction. Les exemples donnés ci-dessous permettent de comprendre le principe de l'approche proposée, et de mesurer la complexité de ces dispositifs de contrôle de l'énergie qui sont embarqués dans les puces numériques aujourd'hui.

Nom du pin	Type	Fonction
VDDIEN	Digital In	Pour commuter en VDDI
VDDIMODE	Digital In	Mode inactive
VDDIOK	Digital Out	Etat de sortie de VDDI
TESTEN	Digital In	Activation des sorties
VDDIHIGH	Digital Out	Etat de sortie du niveau de VDDI
VDDBIASIN	Analog in	Entrée pour une source de tension de polarisation de référence sur VDDI
VDDAA	Tension externe	Alimentation de toutes les fonctions numériques de commandes
VDDO	Tension externe	Tension d'alimentation connectée à VDDI lorsque le switch est ON
VDDI	Tension interne	Alimentation commutable / polarisables dédiée à la fourniture des cellules standard
GNDO	Masse externe	Masse

Tableau 3: Description des entrées/sorties

L'état de la tension d'alimentation est donné par l'entrée VDDIEN :

- Tension d'alimentation active : VDDIEN est relié à VDDAA => VDDI est connecté à VDDO.
- Tension d'alimentation inactive : VDDIEN est relié à GNDO => VDDI est OFF.

L'EPOD est en mode « Debug » si l'entrée TESTEN est connectée à VDDAA, sinon elle est connectée à GNDO.

La source de polarisation n'est pas implémentée directement dans la cellule du contrôleur EPOD. Elle est en revanche contrôlée par l'EPOD. Le signal VDDMODE contrôle l'activation de cette source :

- L'entrée VDDIEN est connectée à VDDAA donc les switches sont actifs.
- VDDMODE se relie à VDDAA, la source de polarisation est initialisée à VDDI.
- VDDIEN se connecte à GNDO, VDDI sera en mode « bias ».
- VDDMODE se connecte à GNDO, la source devient inactive.
- VDDI devient OFF quand VDDIEN sera connecté à GNDO.

5.6. .Structure de gestion de power gating

Le but de cette thèse est de concevoir de nouvelles structures permettant de contrôler la consommation en agissant aux niveaux RTL et GL. Les modèles concernant la gestion de la consommation statique que nous avons développés exploitent les techniques existantes comme le *power gating* et le *body biasing*.

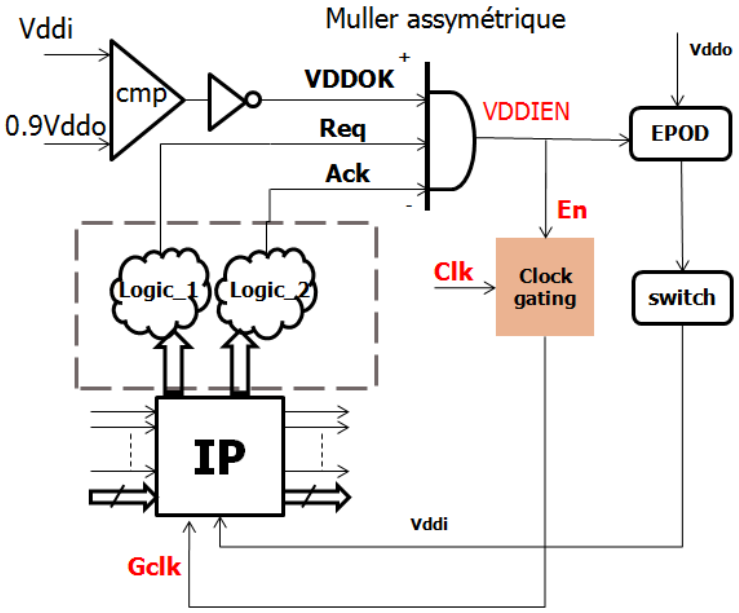


Figure 5-7: Structure générique de gestion du power gating

Dans ce paragraphe, on va décrire une nouvelle structure de contrôle asynchrone et distribuée exploitant la technique du *power gating*. Cette structure va nous permettre de gérer automatiquement la tension d'alimentation d'un circuit avec un contrôle distribué de sa tension.

La Figure 5-7 illustre la structure générique de notre dispositif de *power gating*. Cette structure renferme une porte de Muller asymétrique à 3 entrées (voir chapitre 2). Le but est de piloter l'entrée « VDDEN » du switch (entrée du bloc *logic control* dans le contrôleur EPOD de la Figure 5-6). Le dispositif compare en premier lieu la tension d'alimentation Vddi à un seuil donné (0.9Vddo par exemple). Dans la pratique, cette information peut être récupérée à la sortie de l'EPOD par le biais du signal VDDOK (le signal est actif niveau bas). De petits blocs logiques sont utilisés, tout comme nous l'avions déjà fait avec la structure de *clock gating*, pour générer les entrées Requêtes-Acquittements de la structure de contrôle de l'alimentation à partir de signaux issus du réseau de communication de la puce. La Figure 5-8 montre la logique que nous avons implémentée pour générer les signaux de requêtes et d'acquittements.

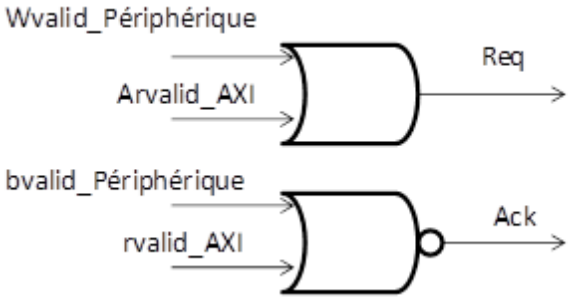


Figure 5-8: Logique de génération des signaux Req-Ack

Cette figure représente les blocs logiques 1 et 2 de la Figure 5-7 utilisés pour générer les signaux de contrôle en entrée de la porte de Muller asymétrique.

5.7. Structure de gestion de body biasing

Cette structure est identique à celle de *power gating* mais il s'agit cette fois de piloter l'entrée VDDMODE du contrôleur EPOD.

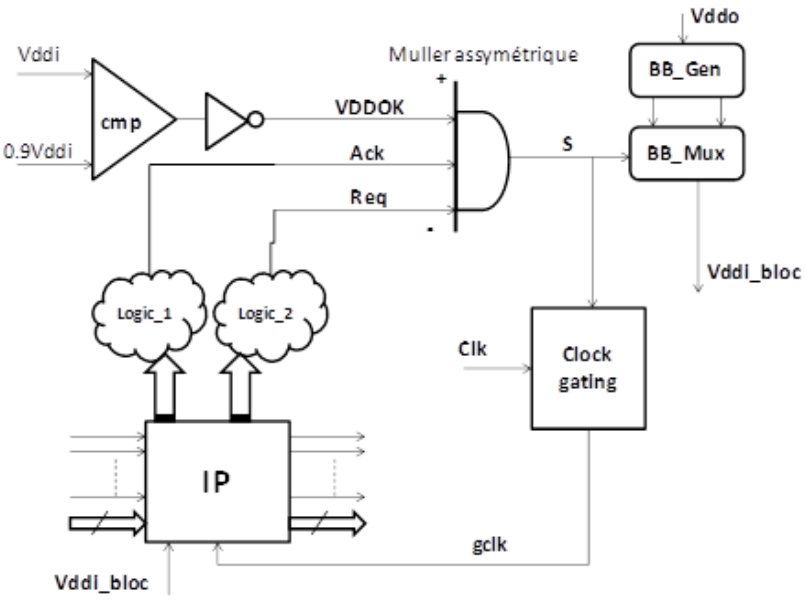


Figure 5-9: Structure de gestion de body biasing

L'entrée VDDMODE de l'EPOD sert à gérer la commande du *back biasing* qui pilote des blocs BB_Mux. Le fonctionnement de ces blocs sont associés à un bloc générateur de polarisation BB_Gen. BB_Gen et BB_Mux seront explicitées dans le paragraphe suivant. Dans cette architecture, on remarque l'existence d'un bloc de *clock gating*. En effet, en mode FBB (*Forward back Biasing*), l'horloge doit être active. Nous rappelons que le mode FBB permet d'augmenter les performances des transistors (fonctionnement plus rapide). Il est à noter que le mode FBB est souvent associé à une alimentation avec une tension réduite ce qui permet de réduire la consommation dynamique. Or une tension d'alimentation réduite décroit également les performances (en vitesse) du circuit. Afin de palier à cette perte de performance, le mode FBB permet de compenser cette réduction de vitesse. Enfin, en mode RBB (*Reverse Back Biasing*), l'horloge doit s'arrêter car le circuit est en mode veille. On limite donc les courants de fuite d'une part et on supprime la consommation dynamique d'autre part. La porte de Muller asymétrique sur la Figure 5-9 assure la génération du signal VDDMODE du contrôleur EPOD afin de faire du *back biasing,* ainsi que le signal de commande du bloc de *clock gating*.

5.7.1. BBMux (*Body Bias Multiplexer*)

Un BBMUX ou *Body Bias Multiplexer* est utilisé pour activer ou désactiver dynamiquement les transistors NMOS, PMOS de polarisation. Cette structure permet à l'utilisateur de basculer la tension entre deux domaines d'alimentation différents. Elle est généralement utilisée pour basculer la tension du *bulk* sur une tension extérieure ou une tension interne fournie par la structure BBGEN (paragraphe suivant). Le multiplexeur (l'entrée SEL [1 :0]) est utilisé en mode FBB (*Forward Body Biasing*) pour augmenter les performances et en mode RBB pour limiter les fuites. La Figure 5-10 montre le bloc diagramme de cette structure.

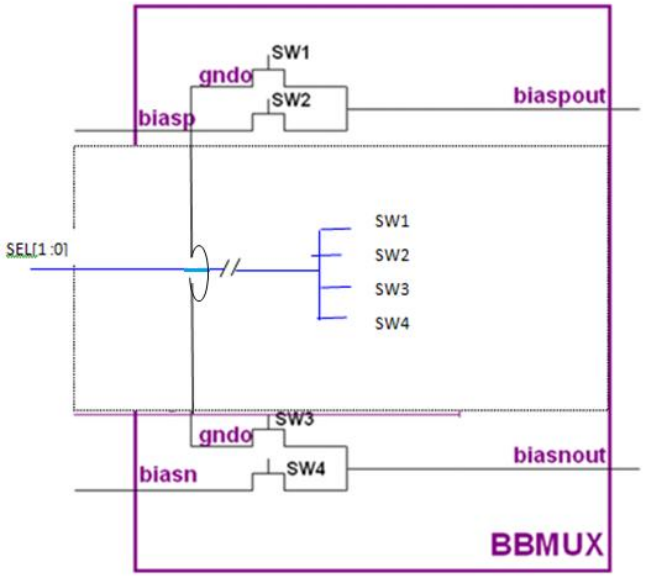


Figure 5-10: Schéma de principe du block BBMux

Les entrées Biasp et Biasn sont générées par le bloc BBGEN (paragraphe suivant) afin de contrôler la structure BBMUX.

5.7.2. BBGen (*Body Bias Generator*)

Cette structure est seulement utilisée pour faire du FBB (*Forward Body Biasing*) afin d'augmenter la vitesse en diminuant la tension du seuil. La Figure 5-11 montre le diagramme de cette structure.

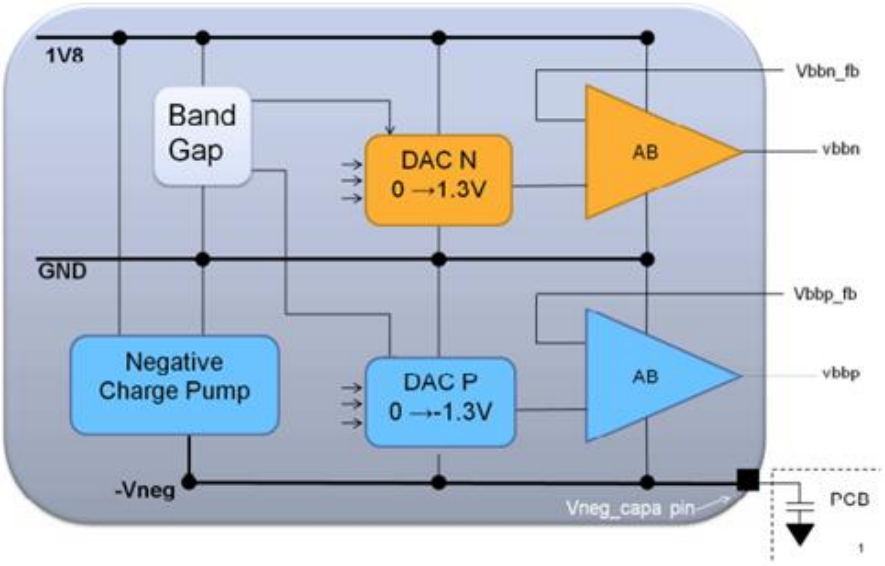


Figure 5-11: Structure BBGEN

Le bloc « *Band Gap* » permet de fournir une tension de référence pour les DACs. Les tensions de sortie VBBN et VBBP sont générées à travers deux amplificateurs « suiveur ». Un condensateur de découplage PCB est utilisé en sortie afin d'assurer une tension négative suffisante pendant la transition de VBBN. L'annexe 4 détaille le fonctionnement de cette structure.

5.8. Circuit de test STMicroelectronics (FDSOI 28 nm)

Afin de valider le fonctionnement de notre structure, un test a été effectué avec un véhicule de test de STMicroelectronics en technologie FDSOI 28 nm. Le but est ici de réduire au minimum le courant de fuite. Ce circuit contient donc des contrôleurs EPOD, des mémoires connectées à un bus AXI avec des IPs qui fonctionnent à des tensions d'alimentation différentes.

Nous commençons par étudier la technique du *power gating* afin d'estimer le courant de fuite avant et après l'insertion de la structure de gestion. Donc dans un premier temps, on valide le fonctionnement du circuit après avoir piloté l'entrée VDDIEN du contrôleur EPOD. La illustre la structure du circuit de test utilisé.

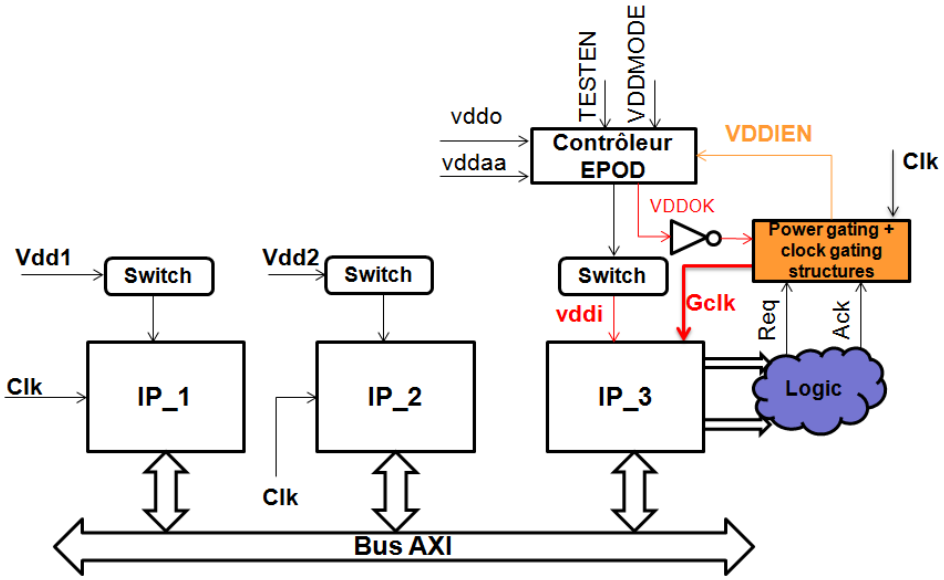


Figure 5-12: Circuit de test

Ce circuit possède trois mémoires qui communiquent (Lecture/Ecriture) via un bus AXI. Un contrôleur EPOD est appliqué au niveau de la mémoire 3 afin de contrôler la tension d'alimentation. On notera que la tension Vddok en sortie de l'EPOD représente le résultat de comparaison entre la tension interne Vddi et un seuil donné (90%Vddo par exemple). Les résultats de simulation sont donnés par la Figure 5-13 suivante :

Figure 5-13: Résultat de simulation

Sur cette figure, le signal VDDIEN est toujours actif. Pour limiter le courant de fuite, on doit limiter l'activation de ce signal de façon à l'activer seulement quand on en a besoin.

Pour insérer notre structure dans le circuit on a recours à l'outil STAR de DeFacto Technologies. Comme nous l'avons vu précédemment, cet outil permet d'écrire un algorithme qui nous permet d'insérer automatiquement une structure au bon endroit dans un circuit. Le flot d'insertion est donné par l'algorithme de la Figure 5-14.

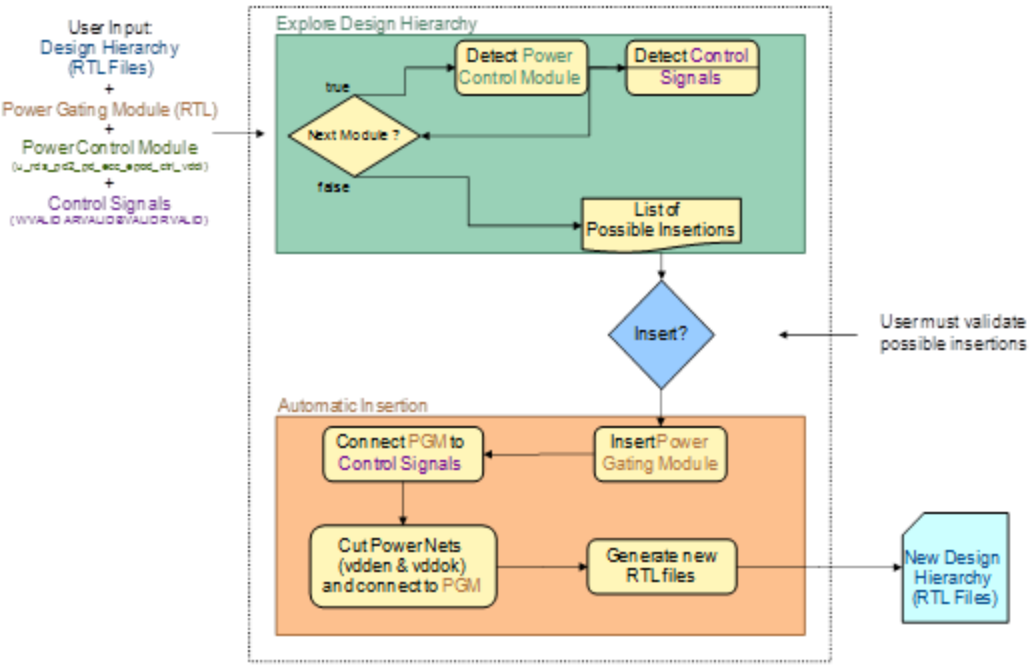


Figure 5-14: Flot d'insertion automatique

Le block inséré au niveau de la mémoire 3 (SPRAM_1), est représenté sur la Figure 5-15.

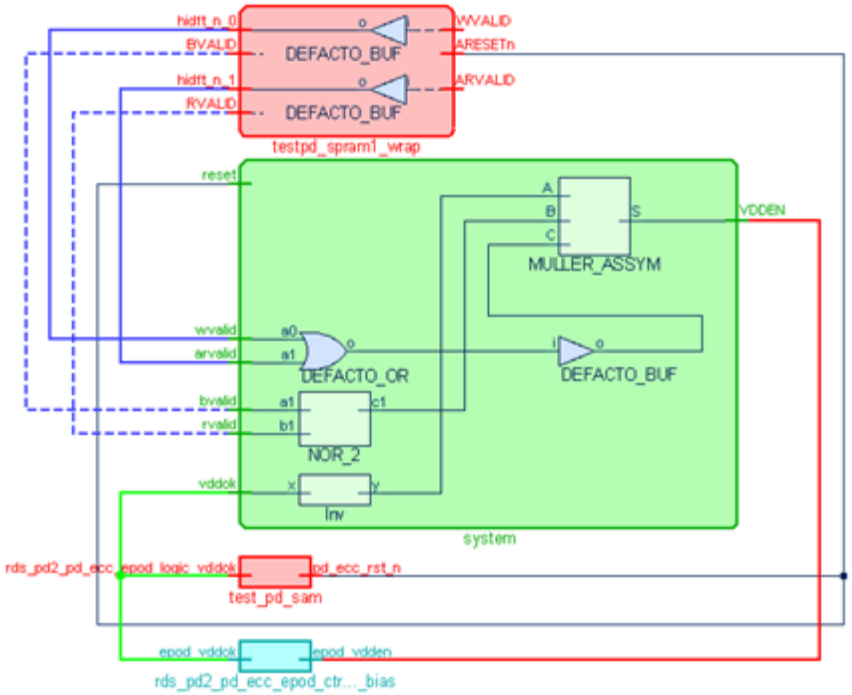


Figure 5-15: Circuit après insertion de la structure power gating

Cette nouvelle structure a donc été insérée entre la mémoire SPRAM_1 et le contrôleur EPOD dans le but d'activer l'entrée VDDEN du contrôleur afin de réduire la consommation statique quand le bloc mémoire n'est pas utilisé. Des signaux de contrôle de la mémoire sont utilisés en retour afin de générer les signaux de requêtes et d'acquittements que l'on trouve également en entrée de la structure. Après insertion, nous obtenons les résultats illustrés par les figures : Figure 5-16, Figure 5-17 et Figure 5-18 .

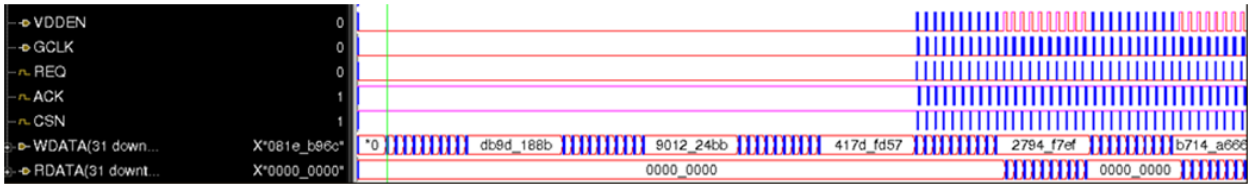


Figure 5-16: Résultats de simulation après l'insertion automatique

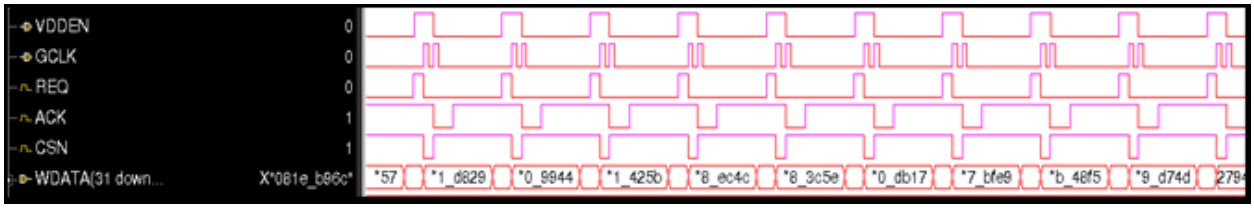


Figure 5-18: Exemple d'écriture de données

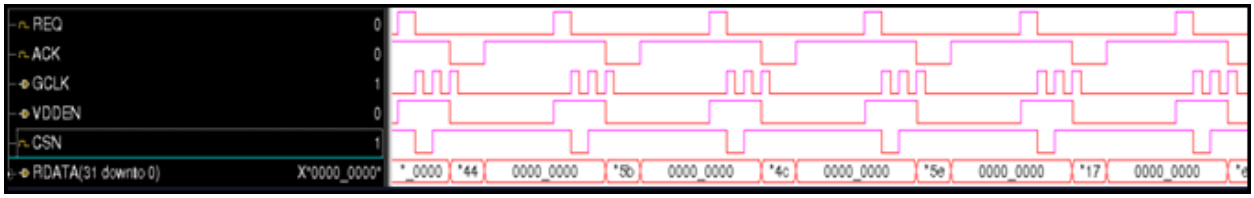


Figure 5-17: Exemple de lecture de données

On observe qu'après insertion le signal VDDEN est devenu actif seulement quand il y a un traitement à faire (lecture ou écriture des données), ce qui permet de réduire les courants de fuite. Cette commutation de '0' à '1' fait que le bloc passe de l'état actif à l'état inactif et vice-versa. Ce chronogramme montre aussi les signaux de contrôles de la structure insérée (Req-Ack).

Les signaux de requêtes passent à '1' seulement quand il y a une nouvelle donnée. Ces deux figures montrent les modes de lecture et d'écriture. On voit clairement que le signal VDDEN est actif si et seulement si un traitement a lieu. Le signal CSN correspond au signal *chip select* de la mémoire.

Pour estimer les courants de fuite dans le système avant et après insertion de notre structure, nous avons effectué nos simulations avec le logiciel ACEPLORER de notre partenaire DOCEA (A ce moment INTEL). Ces résultats donnent la consommation statique au niveau architectural.

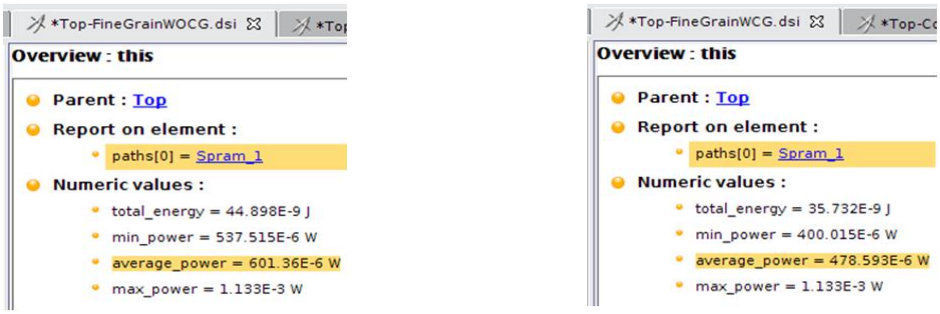


Figure 5-19: Courant de fuite avant et après insertion de la structure de gestion du power gating

Sur ces deux figures, le courant de fuite au niveau architectural a été estimé. On voit que la consommation moyenne passe de 0.6 mA à 0.47 mA. Ce qui valide l'efficacité de notre approche en termes de réduction de la consommation statique.

5.9. Contrôle de la tension de polarisation V_{bb}

La deuxième technique à valider dans cette étude est le contrôle de polarisation. Dans ce contexte la structure de contrôle de *body biasing* a été insérée dans un circuit afin de gérer automatiquement la tension Vbb. La Figure 5-20 ci-dessous illustre le circuit de test.

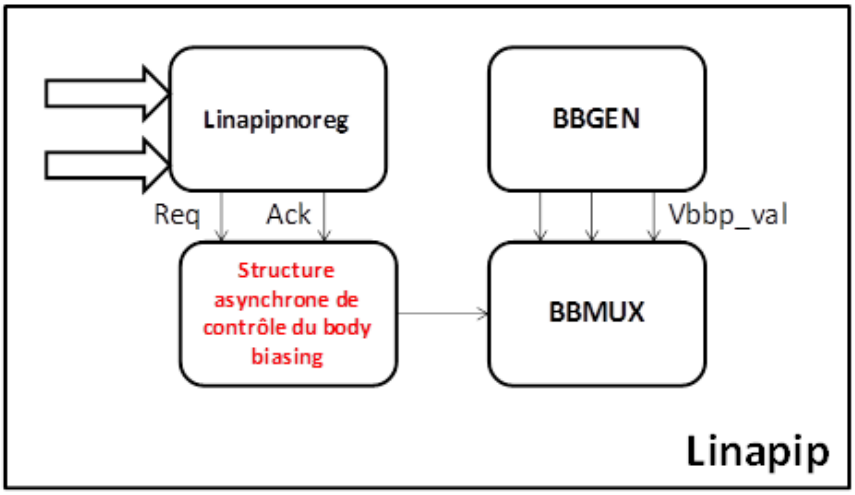


Figure 5-20: Circuit de test

Le but de cette étude est de contrôler la tension Vbb en entrée du bloc BBMUX. Les signaux de requête et d'acquittement de la structure asynchrone ont été générés par le block linapipnoreg afin de contrôler le signal Sel [1:0] à l'entrée du bloc BBMUX. Le signal Sel est le signal de choix qui sélectionne le signal SW1 ou bien SW2 en entrée du bloc BBMUX (SW3 et SW4 ne sont pas utilisés ici la valeur de SEL est limitée entre 00 et 01). La Figure 5-22 montre les résultats de simulation sous VCSMX.

Figure 5-22: Résultat de simulation

Figure 5-21: Résultat de simulation (zoom)

Sur la le signal PACECOUNTER_RELOAD représente le signal de requête. Quand le signal FREEZE passe à '0' (pas d'activité), le BBGEN sera prêt à polariser le body à 900 mV. La valeur de Sel passe à '1' pour activer le mode FBB (Forward Body Biasing) pour optimiser les performances.

Ce qui importe c'est :

- Le traitement démarre quand le block linapip est disponible (FREEZE =0) et que le signal PACECOUTER_RELOAD = 0 (d'où REQ == PACECOUTER_RELOAD ==0)
- Le traitement s'arrête quand le block linapip est indisponible (FREEZE ==1) et que la requête REQ == PACECOUTER_RELOAD ==1 »

En conclusion, la structure asynchrone insérée permettra d'activer ou inactiver le contrôle quand on a besoin.

5.10. Conclusion

L'échelle nanométrique des transistors a fait apparaître une contrainte importante du point de vue de la consommation statique. Cette consommation est problématique durant les périodes d'inactivité ou de veille, notamment pour des circuits alimentés sur batterie pour de longues durées prévues. La diminution de la tension de seuil des transistors a également contribué à cette augmentation des courants de fuite. Les concepteurs ont donc développé des techniques de réduction des courants de fuites (en sus des travaux effectués par les technologues) afin de rendre ces circuits plus attrayants pour des applications nomades. Plusieurs techniques permettent de réduire ce type de consommation telles que le *power gating* et le *body biasing*.

Dans ce chapitre, une nouvelle méthode de gestion de la consommation statique a été introduite. L'idée exploite des techniques usuelles comme le *power gating* et le *body biasing*. La structure asynchrone et distribuée définie dans ce chapitre permet de gérer les diverses tensions d'alimentation d'un même circuit. Le principe de la détection de l'activité par la présence de données en entrées des blocks a été retenu avec une logique asynchrone assurant une synchronisation locale par requête et acquittement. L'évaluation de cette stratégie a été validée

sur un circuit implémenté en technologie FDSOI 28 nm de STMicroelectronics. Les résultats obtenus traduisent l'efficacité de notre principe en termes de gestion des techniques de réduction de la consommation statique afin de la réduire. Au final, les structures proposées rendent le circuit plus efficace vis-à-vis des fuites de courant avec un faible coût en surface et l'absence d'un système de gestion centralisé des domaines d'alimentation. Par ailleurs, leur insertion peut se faire de façon automatique, de façon similaire aux techniques de *clock gating* présentés dans le chapitre précédent, en s'appuyant sur les outils de Defacto Technologies.

Conclusions et perspectives

L'évolution des technologies a permis, depuis presque un demi-siècle, d'augmenter sans discontinuité les performances et les fonctionnalités des produits utilisant des circuits intégrés. La consommation représente un paramètre essentiel de ces performances. Réduire la consommation d'énergie des systèmes embarqués est donc devenu la principale priorité pour un bon nombre de concepteurs.

La réduction des dimensions des transistors jusqu'à une échelle nanométrique a contribué au développement de l'électronique à forte intégration, à faible coût et aussi à faible consommation. Toutefois, cette technologie avancée a conduit à un accroissement relatif de la consommation statique par rapport à la consommation dynamique. Il faut donc, en fonction des applications, tenir compte de ces deux composantes de la consommation. L'objectif de cette thèse a été de modéliser et de développer de nouvelles structures de contrôle distribuées et asynchrones afin d'améliorer les techniques usuelles de réduction de la consommation que l'on rencontre dans les circuits intégrés. Ce manuscrit, divisé en deux parties, a fait dans une première partie (Chap. 1 et Chap. 2) un état des lieux des circuits asynchrones et de leurs principes de fonctionnement, tandis que la deuxième partie (Chap. 3, Chap. 4 et Chap. 5) explicite les nouvelles méthodes qui permettent de réduire la consommation dynamique puis la consommation statique en mode veille.

Les études comparatives menées au sein du groupe CIS du laboratoire TIMA, consacrées aux techniques de réduction de l'énergie des systèmes intégrés (mise en veille et adaptation de la vitesse), montre clairement qu'une stratégie synchrone, est souvent synonyme de surcoût temporels et/ou énergétiques. En effet, la gestion est souvent moins efficace qu'avec une approche asynchrone. Les circuits asynchrones ne consomment pas lorsqu'ils n'ont pas de données à traiter. Ils se mettent naturellement en mode veille et reprennent immédiatement leur activité si nécessaire. C'est une des bonnes propriétés natives des circuits asynchrone. L'étude présentée dans cette thèse se base sur une stratégie asynchrone afin de disposer d'un contrôle local des blocs d'un circuit intégré dans un but de réduction de la consommation. Les principes développés dans cette thèse constituent une alternative aux stratégies tout synchrone. Deux

techniques, basées sur la logique asynchrone, ont été développées pour réduire la consommation, l'une pour la consommation dynamique, l'autre pour la consommation statique.

Afin de valider nos approches, nous avons réalisé en collaboration avec STMicroelectronics une série de tests. Les circuits de test proposés dans ce manuscrit ont été implémentés en technologie FDSOI 28 nm. Ces circuits sont tous basés sur un bus spécifique, le bus AXI dans notre cas. Il est à noter que dans leur immense majorité les systèmes de communication bus ou NoC exploitent un protocole de type requête-acquittement mais avec des signaux de contrôle synchronisés avec l'horloge. Afin d'aboutir à une réalisation facilement implémentable et reproductible sur un grand nombre de systèmes, nous avons décidé de tirer parti de ces signaux de contrôle déjà disponibles. L'insertion des structures de contrôle asynchrones permettant de réduire la consommation a donc largement exploitée les signaux de contrôle du bus AXI. L'outil STAR de DEFACTO technologie nous a fourni une aide précieuse pour automatiser notre approche et valider les algorithmes d'insertion que nous avons développés. Par ailleurs, le travail a toujours cherché à maintenir une approche suffisamment générique pour qu'elle soit reproductible avec d'autres systèmes de communications, bus ou NoC. Le tableau ci-dessous représente un bilan des différents modes possibles de réduction de la consommation avec une technologie FDSOI en analysant les paramètres suivants : la fréquence f, la tension de polarisation Vbb, le courant de fuite (*Leakage*), la tension d'alimentation Vdd et l'activité de l'horloge.

Clk « OFF »	P_{dyn} ↘		
V_{dd} =0 et Clk « OFF »	P_{dyn} ↘	P_{leak} ↘	
f= cte	V_{bb} ↗	V_{dd} ↘	P_{dyn} ↘
Clk « OFF »	V_{bb} ↘	V_{dd} = cte	P_{leak} ↘

Tableau : Modes de réduction des puissances dynamique P_{dyn} et statique P_{leak}

Comme on peut le voir sur le tableau, il est possible d'avoir un gain immédiat sur la consommation dynamique en appliquant une technique de *clock gating*. Si, de plus, on coupe l'alimentation avec un dispositif de *power gating*, on élimine toute consommation et, *a fortiori,* les fuites de courant résiduelles qui subsistaient après extinction de l'horloge. Cependant, avec la technologie FDSOI, il est possible d'aller plus loin dans la gestion de la consommation. En effet, il est possible, tout en conservant la fréquence d'horloge à une valeur fixe, d'abaisser la consommation dynamique. Pour ce faire, il suffit de baisser la tension d'alimentation Vdd et d'augmenter la tension de polarisation Vbb afin de conserver la vitesse de fonctionnement du circuit. Notre approche permet de mettre à profit ce genre de technique en pilotant un dispositif de polarisation du substrat. De même, une polarisation inverse peut être mise à profit pour limiter les fuites de courant quand un bloc n'est pas utilisé et que son horloge n'est pas activée. Cette stratégie peut s'avérer payante pour les systèmes alimentés sur batterie. Nous avons vu dans ce manuscrit que la logique asynchrone s'impose peu à peu comme une solution d'évolution ou d'accompagnement de la logique synchrone. L'apparition de systèmes GALS *(Globally Asynchronous Locally synchronous)* a constitué sans doute une première étape qui a montré la voie. Cette logique est devenue au fil du temps un moyen d'apporter des solutions de simplification à l'approche tout synchrone, qui mène à une complexité de la conception des circuits du fait de leurs hypothèses temporelles, de l'augmentation de leur taille et de leur fréquence de fonctionnement. Se libérer de cette contrainte de synchronisation globale permettra à l'avenir de concevoir des architectures plus complexes et plus fiables. Enfin, cette étude a abordé les problématiques de conception pour la faible consommation aux niveaux RTL et portes. Les structures « *low power* », qui ont été proposées, s'insèrent dans les systèmes intégrés en limitant l'effort de *redesign* et en suivant une méthodologie qui sera reproductible dans de nombreux circuits. De plus, un effort d'automatisation et d'intégration dans les flots existants a été systématiquement mené. Enfin, l'approche est modélisable au niveau architectural, ce qui permet d'évaluer la pertinence de l'approche très en amont dans le flot de conception.

Pour conclure, cette thèse a été effectuée dans le cadre d'un projet nommé « HICOOL » incluant plusieurs partenaires : le laboratoire TIMA qui a porté cette thèse et la conception des contrôleurs asynchrones de gestion de la consommation, STMicroelectronics pour l'évaluation

des résultats et la livraison d'un environnement de test industriel, DEFACTO technologies et DOCEA pour leur apport dans la réflexion sur le flot de conception et leurs outils de conception nécessaires et, enfin, le laboratoire LIRRM qui a développé des techniques d'estimation de la consommation aux niveaux GL et RTL. Le transfert des connaissances et la bonne collaboration entre partenaires a permis de concevoir une approche innovante pour modéliser et réduire la consommation dans les systèmes intégrés.

Références

[ALK 15] Al Khatib, Chadi; Aupetit, Claire; Chevalier, Cyril; Aktouf, Chouki; Sicard, Gilles; Fesquet, Laurent, "A generic clock controller for low power systems: Experimentation on an AXI bus," *in Very Large Scale Integration (VLSI-SoC), 2015 IFIP/IEEE International Conference on , vol.*, no., pp.307-312, 5-7 Oct. 2015

[AMB] AMBA AXI Protocol specification. www.arm.com/armtech/AXI.

[AMI 01] Amini, E.; Najibi, M.; Pedram, H., "Globally asynchronous locally synchronous wrapper circuit based on clock gating," *Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on , vol., no.*, pp.6 pp.,, 2-3 March 2006.

[ANI 02] M. Anis, et al. "Dynamic and Leakage Power Reduction in MTCMOS Circuits Using an Automated Efficient Gate Clustering Technique", *Proc. Design Automation Conference,* June 2002.

[ARM 10] ARM AMBA 4.0 AXI4 Protocol Specification, version 1.0, March 2010 http://www.arm.com/products/system-ip/amba/amba-open specifications.php

[BAP 02] Baptiste J. "Spécification de bibliothèque pour la synthèse des circuits asynchrones", PhD thesis, December 2002.

[BEE 91] Peter A. Beerel and Teresa H.-Y. Meng, "Testability of asynchronous self-timed control circuits with delay assumptions", *In Proc. ACM/IEEE Design Automation Conference,* pp.446-451, IEEE Computer Society Press, June 1991.

[BER 96] Kees Van Berkel, Arjin Bink; Single-track Handshaking signaling with application to micropipelines and handshake circuits; *Proc. International*

Symposium on advanced research on asynchronous circuits and systems, pp. 122-133, 1996-03-01.

[BOR 05] Borinski E., Arnim V. and Seegebrecht P.: 'Efficiency of body biasing in 90-nm CMOS for low-power digital circuits'. IEEE Journal of Solid-state Circuits, 40(7), pp. 1549-1556, 2005.

[BOU 04] Bouesse G.F., Robisson B., Liardet P.-Y., Renaudin M., Beigné E., Prevosto S., DPA on quasi delay insensitive asynchronous circuits: concrete results, *XIX Conference on Design of Circuits and Integrated Systems (DCIS'04)*, Bordeaux, France, November 24-26, 2004

[CAR 01] Carlsson, J.; Palmkvist, K.; Wanhammar, L., "A Clock Gating Circuit for Globally Asynchronous Locally Synchronous Systems," *Norchip Conference, 2006. 24th , vol., no.*, pp.15,18, Nov. 2006.

[CAO 02] Cao K., Lee W.-C, Liu W., Jin X., Su P., Fung S., An J., Yu B., and Hu C., "BSIM4 gate leakage model including source drain partition," *in Tech. Dig. Int. Electron Devices Meeting*, 2000, pp. 815–818.

[CAO 06] Y.cao et al., "research and application of AMBA 3.0 AXI Bus Interface Protocol," CCIC2006.

[DAL 98] W. Dally and J. Poulton, *Digital System Enginneering.* Combridge University Press, 1998.

[DHA 01] Dhar S., Dhar E. and Maksimovic D.: "Switching regulator with dynamically adjustable supply voltage for low power VLSI", *In Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 103-107, 2001.

[DHA 02] Dhar S., Maksimovic D. and Kranzen B.: 'Closed-loop adaptive voltage scaling controller for standard-cell ASICs', *In ISLPED '02: Proceedings*

of the 2002 international symposium on Low power electronics and design, pp. 103-107, New York, NY, USA, ACM, 2002.

[FBF 07] 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.

[FER 02] Marcos Ferretti and Peter A. Beerel. Single-track asynchronous pipeline templates using 1-of-n encoding. *In Proc. Design Automation and test in Europe (DATE)*, pages 1008-1015, March 2002.

[FLA 04] Flautner K., Flynn D., Roberts D. and Patel D.: 'Iem926: An energy efficient SoC with dynamic voltage scaling', *Design, Automation and Test in Europe Conference and Exhibition*, 3, pp. 324-327, 2004.

[FURB 94] S.B. Furber, P. Day, J.D, Garside, N.C, Paver, S. Temple et J.V. Woods, "the design and evaluation of an asynchronous microprocessor", *International Conf. Computer Design, (ICCD), IEEE Computer Society Press,* Oct. 1994.

[GOU 01] Goulding-Hotta N., Sampson J., Venkatesh G., Garcia S., Auricchio J., Huang P., et al., "The GreenDroid mobile application processor: An architecture for silicon's dark future," IEEE Micro, vol. 31, no. 2, pp. 86–95, Mar. 2011.

[HAU 95] S. Hauck, "Asynchronous Design Methodologies : An Overview Proceeding of the IEEE, Vol. 83, N˚ 1, pp. 69-93, January, 1995."

[HUF 54] HUFFMAN, D. A., The Synthesis of Sequential Switching Circuits, *Journal of the Franklin Institute 257, nos, 3 and 4* March and Apr. 1954, 294—295.

[KAN 04] S-M. Kang and Y. Lelebici, *CMOS Digital Integrated Circuits, Mc Graw Hill*, second edition, 1999.

[KAO 01] J. T. Kao, A. P. Chandrakasan, ``Dual-threshold voltage techniques for low-power digital circuits," *IEEE Journal of Solid-State Circuits, Vol. 35*, July 2000, pp. 1009-1018.

[KES 03] A., Roy K., and Hawkins C. F., "Intrinsic Leakage In Low Power Deep Submicron CMOS Ics," *in Proceedings of International Test Conference*, pp. 146–155, 1997.

[KIS 98] Michael Kishnevsky, Alex Kondratyev, Luciano Lavagno, Alex Saldanha and Alexander Taubin, "Partial-scan delay fault testing of asynchronous circuits", *IEEE Transactions on Computer-Aided Design, Vol*. 17(11), pp. 1184-1199, November 1998.

[LAV 93a] L. Lavagno et A. Sangiovanni-Vincentelli, "Algorithms for Synthesis ans Testing of Asynchronous Circuits", Kluwer Academic Publishers, 1993.

[LAV 94] Luciano Lavagno, Michael Kishinevsky, and Antonio Lioy, "Testing redundant asynchronous circuits by variable phase splitting", *In Proc. European Design Automation Conference (EURO-DAC), IEEE Computer Society Press*, pp. 328-333, September 1994.

[LIN 98] A. M. Lines. Pipelined Asynchronous circuits. M. Sc. Thesis, California Institute of technology, June 1995, revised 1998.

[MB 59] D. E. Muller et W. S. Bartky: A theory of asynchronous circuits. Dans Proceedings of the International Symposium on the Theory of Switching, 1959.

[MUL 59] D. E. Muller and W. S. Bartky. A theory of asynchronous circuits. Annals f the Computation Laboratory of Harvard University. Volume *XXIX:*

Proceedings of an International Symposium on the Theory of Switching, Part1, Pages 204—243, 1959.

[MUT 95] S. Mutoh, et. al., "1-V Power Supply High-Speed Digital Circuit Technology with Multi-threshold Voltage CMOS," *IEEE Journal of Solid-State Circuits, vol.30*, pp. 847-854, August 1995.

[NOW 97] S. M. Nowick, N. K. Jha, and F.-C. Cheng, "Synthesis of asynchronous circuits for stuck-at and robust path delay fault testability", *IEEE Transactions on Computer-Aided Design, Vol.* 16(12), pp. 1514-1521, December 1997.

[OUC 11] Ouchet F, "Analyse et amelioration de la robustesse des circuits asynchronies QDI", PhD thesis, December 2011.

[OZD 02] Recep O. Ozdag, Peter A. Beerel. High-Speed QDI Asynchronous Pipelines, *Proceedings on the English International Symposium on Asynchronous Circuits and Systems*, p. 13, April 08-11, 2002.

[OZD 02b] Recep O. Ozdag, Montek Singh, Peter A. Beerel, Steven M. Nowick, "High-Speed Non-Linear Asynchronous Pipelines", *Proceedings of Design, Automation and Test in Europe* (DATE-02), Paris, France, March 2002.

[PAN 05] Pangrle B. and Kapoor S.: 'Leakage power at 90nm and below', Technical report, Synopsis Inc, 2005.

[PFI 07] Pfitzner A., Kuzmicz W., Piwowarska E. and Kasprowicz D.: 'Static power consumption in nano-CMOS circuits: Physics and modeling'. *In Proceeding of the 14th International Conference Mixed Design of Integrated Circuits and Systems*, pp. 163-168, 2007.

[POK 07] Pokhrel K.: 'Physical and Silicon Measures of Low Power Clock Gating Success: An Apple to Apple Case Study', SNUG, http://www.snuguniversal. org, 2007.

[POU 01] Pouwelse J., Langendoen K. and Sips H.: 'Dynamic voltage scaling on a low-power microprocessor', *In MobiCom'01 Proceedings of the 7th annual international conference on Mobile computing and networking*, New York, NY, USA, ACM, pp. 251-259, 2001.

[REN 02] Renaudin M., Rigaud J.B., Etude de l'art sur la conception des circuits asynchrones, perspectives pour l'intégration des systèmes complexes, ISRN: TIMA-RR--02/12-02--FR, 1 janvier 2002

[REN 98] RENAUDIN M. VIVET P. ROBIN F., << ASPRO-216: a standart-cell Q.D.I. 16-bit RISC asynchronous microprocesseur >>, *Proc. Of the Fourth International Symposium on Advanced Research in Asynchronous Circuits and Systems*, 1998, IEEE, p.22—31.

[REZ 04] Rezzag A. "Logical synthesis of micropipeline asynchronous circuits", PHD thesis, Dec. 2004.

[ROY 01] Roy K. and Prasad S. C., Low-Power CMOS VLSI Circuit Design, New York, USA: Wiley Interscience Publications, 2000, ch. 2, pp. 28-29.

[SEM 02] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2010 edition, http://public.itrs.n

[SEM 03] Y. Semiat et R. Ginosar, "Timing Measurements of Synchronization Circuits", *Proceedings of the Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC'03*, Vancouver, Canada..

[SET 03] K. Seta, H. Hara, T. Kuroda, et al., "50% active-power saving without speed degradation using standby power reduction (SPR) circuit," *IEEE International. Solid-State Circuits Conf.*, February 1995, pp. 318-319.

[SHE 13] B. Sheu, D. Scharfetter, P. Ko, and M. Jeng, "BSIM: Berkeley short-channel IGFET model for MOS transistors," *IEEE Journal of Solid State Circuits, Vol. 22*, August 1987, pp. 558-566.

[SPA 01] Jens Sparso, Steeve Furber, Eds. "Principles of Asynchronous Circuit Design, A Systems Perspective", Springer, SBN: 978-1-4419-4936-3 (Print) 978-1-4757-3385-3

[SPA 02] SPARSO Jens, FURBER Steeve, "Principle of Asynchronous Circuit Design: A System Perspectives", *European Low Power Initiative for Electronic System Design*, pp.18-19, 2001.

[SUH 08] Suhaib, S., Mathaikutty, D., Shukla, S.: Dataflow Architectures for GALS. ACM Journal. Electronic Notes in Theoretical Computer Science (ENTCS), Vol. 200, No. 1, 33–50 (2008)

[SUT 89] SUTHERLAND I. E., <<Micropipelines>> , Communication of the ACM, Volume 32, N˚6, June 1989.

[TAU 01] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, New York, USA: Cambridge University Press, 1998, ch. 3, pp. 120-128.

[TAU 02] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*. New York: Cambridge Univ. Press, 1998, Ch. 2, pp. 94–95.

[TUG 02] Sunan Tugsinavisut, Peter A. Beerel, Control Circuits Templates for Asynchronous Bundled-Data Pipelines. *In Proc. Design, Automation and Test in Europe* (DATE), page 1098, March 2002.

[XIA 09] Fu-ming Xiao; Dong-sheng Li; Gao-Ming Du; Yu-kun Song; Duo-li Zhang; Ming-Lun Gao, "Design of AXI bus based MPSoC on FPGA," *in Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009. 3rd International Conference on , vol., no.*, pp.560-564, 20-22 Aug. 2009

[WEI 01] L. Wei, et. al., "Design and Optimization of Dual Threshold Circuits for Low Voltage Low Power Applications", *IEEE Transactions on VLSI Systems*, pp. 16-24, March 1999.

[ZAK 01] H. Zakaria, L. Fesquet, "Designing Process Variability Robust Energy-Efficient Control for Complex SoCs" *IEEE Trans. On Emerging and Selected Topics in Circuits and Systems* (JETCAS), Vol. 1, No. 2, June 2011.

[ZHL 01] Jiang Zhou-liang, Quan Jin-guo, Lin Xiao-kang, "Analysis and Application of New Generation AMBA 3 AXI Protocol."

[ZEG 01] B. Van Zeghbroeck, *Principles of Semiconductor Devices, http://ecewww.colorado.edu/~bart/book/book/title.htm, ch.4.*

Annexe I

Cell	Ring-type	Function
EP28SOI_VDDI_CTRL	VDDI	Controls all the power distribution mechanisms with only one switchable supply structure.
EP28SOI_VDDI_CTRL_NOBIAS	VDDI	Controls all the power distribution mechanisms with only one switchable supply structure and no source biasing capability.
EP28SOI_VDDI_DECAPGNDOVDDO50	VDDI	Can replace a filler of the same size and same ring-type. Provides a capacitance between GNDO and VDDO.
EP28SOI_VDDI_DECAPGNDOVDDI50	VDDI	Can replace a filler of the same size and same ring-type. Provides a capacitance between GNDO and VDDI.
EP28SOI_VDDI_VDDISWITCH	VDDI	Switch on VDDI (PMOS switch between VDDO and VDDI).
EP28SOI_SVDDI_VDDISWITCH	SVDDI	10.2 µm height switch on VDDI (PMOS switch between VDDO and VDDI).
EP28SOI_VDDI_VDDISWITCH_H	VDDI_H	Switch on VDDI (PMOS switch between VDDO and VDDI). To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_VDDI_VDDOVDDISBIAS	VDDI	Source biasing generator for VDDI from VDDO.
EP28SOI_VDDI_ANTICORNER	VDDI	Corner to 'open' the ring. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_ANTICORNER_H	VDDI_H	Corner to 'open' the ring. Propagates metal ring signals relative to its ring-type. To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_ANTICORNER	SVDDI	10.2 µm height corner to 'open' the ring. Propagates metal ring signals relative to its ring-type.
EP28SOI_SVDDI_ANTICORNER_R2S	VDDI	20.4 µm to 10.2 µm height corner to 'open' the ring. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_CORNER	VDDI	Corner to 'close' the ring. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_CORNER_H	VDDI_H	Corner to 'close' the ring. Propagates metal ring signals relative to its ring-type. To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_CORNER	SVDDI	10.2 µm height corner to 'close' the ring. Propagates metal ring signals relative to its ring-type.
EP28SOI_SVDDI_CORNER_R2S	VDDI	20.4 µm to 10.2 µm height corner to 'close' the ring. Propagates metal ring signals relative to its ring-type.

5

Cell	Ring-type	Function
EP28SOI_VDDI_TCORNER	VDDI	20.4 µm height T-corner to add an intrusive branch in the Power domain. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_TANTICORNER	VDDI	20.4 µm height T-anticorner to add a branch. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER5	VDDI	Filler cell, propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER5_H	VDDI_H	Filler cell, propagates metal ring signals relative to its ring-type. To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_FILLER5	SVDDI	10.2 µm height filler cell. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER25	VDDI	Filler cell, propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER25_H	VDDI_H	Filler cell, propagates metal ring signals relative to its ring-type. To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_FILLER25	SVDDI	10.2 µm height filler cell. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER50	VDDI	Filler cell, propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER50_H	VDDI_H	Filler cell, propagates metal ring signals relative to its ring-type. To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_FILLER50	SVDDI	10.2 µm height filler cell. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER100	VDDI	Filler cell, propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FILLER100_H	VDDI_H	Filler cell, propagates metal ring signals relative to its ring-type. To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_FILLER100	SVDDI	10.2 µm height filler cell. Propagates metal ring signals relative to its ring-type.
EP28SOI_VDDI_FORCE	VDDI	Distributed cell that correctly ties some control nets along the EPOD structure.

Cell	Ring-type	Function
EP28SOI_VDDI_FORCE_H	VDDI_H	Distributed cell that correctly ties some control nets along the EPOD structure. To be placed between modular precharge cells or source biasing generators and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_FORCE	SVDDI	10.2 µm height distributed cell that correctly ties some control nets along the EPOD structure.
EP28SOI_VDDI_ANTENNA	VDDI	To avoid DRC antenna error on some control nets along the EPOD structure (due to long metal lines).
EP28SOI_VDDI_ANTENNA_H	VDDI_H	To avoid DRC antenna error on some control nets along the EPOD structure (due to long metal lines). To be placed between modular precharge cells or source biasing generators and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_ANTENNA	SVDDI	10.2 µm height distributed cell to avoid DRC antenna error on some control nets along the EPOD structure (due to long metal lines).
EP28SOI_VDDI_PRECHOPP0_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP0
EP28SOI_VDDI_PRECHOPP1_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP1
EP28SOI_VDDI_PRECHOPP2_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP2
EP28SOI_VDDI_PRECHOPP3_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP3
EP28SOI_VDDI_PRECHOPP4_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP4
EP28SOI_VDDI_PRECHOPP5_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP5
EP28SOI_VDDI_PRECHOPP6_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP6
EP28SOI_VDDI_PRECHOPP7_H	VDDI_H	Modular precharge cell dedicated to wake-up for OPP7
EP28SOI_VDDI_FILLER_CDM	VDDI	Filler cell, diodes and GGMOS for CDM protection.
EP28SOI_VDDI_FILLER_CDM_H	VDDI_H	Filler cell, diodes and GGMOS for CDM protection. To be placed between source bias or modular precharge and ctrl. (Refer to Section 5.2.2.2, "EP28SOI_VDDI_VDDISWITCH_H")
EP28SOI_SVDDI_FILLER_CDM	SVDDI	10.2 µm height filler cell. Diodes and GGMOS for CDM protection.
EP28SOI_VDDI_FILLER_CDM_PROPP_H	VDDI_H	Filler cell, diodes and GGMOS for CDM protection on PROPPxN To be placed between modular precharge and ctrl. (Refer to Section 5.2.8, "Modular precharge cells")
EP28SOI_VDDI_FILLER_PRCUT_H	VDDI_H	Filler cell, propagates metal ring signals relative to its ring-type. To be placed between modular precharge and other cells. (Refer to Section 5.2.8, "Modular precharge cells")
EP28SOI_SVDDI_WRAPPER_R2S	VDDI	20.4 µm to 10.2 µm height wrapper cell. Propagates metal ring signals relative to its ring-type.

Cell	Ring-type	Function
EP28SOI_SVDDI_BRANCH_END	SVDDI	End ring cell , propagates metal ring signals relative to its ring-type and vddiforceon/off signals back to the CTRL.. To be placed at each ring terminaison , and/or in the middle of a close ring. (Refer to Section 5.2.4, "END cells")
EP28SOI_VDDI_BRANCH_END	VDDI	End ring cell , propagates metal ring signals relative to its ring-type and vddiforceon/off signals back to the CTRL.. To be placed at each ring terminaison , and/or in the middle of a close ring. (Refer to Section 5.2.4, "END cells")
EP28SOI_VDDI_BRANCH_END_H	VDDI_H	End ring cell , propagates metal ring signals relative to its ring-type and vddiforceon/off signals back to the CTRL.. To be placed at each ring terminaison , and/or in the middle of a close ring. (Refer to Section 5.2.4, "END cells")
EP28SOI_VDDI_BRANCH_CUT	VDDI	Cell to cut feedback of vddiforceon/off signals to the CTRL. To be abutted to T-Corner cells on the shortest ring branch (Refer to Section 5.2.3.26, "EP28SOI_VDDI_BRANCH_CUT")

Annexe II

1. Fixed Functional Rules

Each embedded power distribution structure must embed the following:

- With source biasing feature:
  - One controller cell EP28SOI_VDDI_CTRL.
  - At least, one source biasing generator.
  - One or more switching cell EP28SOI_VDDI_VDDISWITCH.
  - EP28SOI_VDDI_FILLER_CDM, EP28SOI_VDDI_FORCE, EP28SOI_VDDI_ANTENNA cells, depending on switch number
  - At least two EP28SOI_VDDI_BRANCH_END cells.

- With no source biasing:
  - One controller cell EP28SOI_VDDI_CTRL_NOBIAS.
  - One or more switching cell EP28SOI_VDDI_VDDISWITCH.
  - EP28SOI_VDDI_FILLER_CDM, EP28SOI_VDDI_FORCE, EP28SOI_VDDI_ANTENNA cells
  - Depending on switch number
  - At least two EP28SOI_VDDI_BRANCH_END cells

- With modular precharge wake-up feature: VDDISWCURCTRL<1:0>=01
  - One controller cell EP28SOI_VDDI_CTRL or EP28SOI_VDDI_CTRL_NOBIAS.
  - One or more switching cell EP28SOI_VDDI_VDDISWITCH.
  - A number of modular precharge cells controlled by OPP condition
  - EP28SOI_VDDI_FILLER_CDM, EP28SOI_VDDI_FORCE and EP28SOI_VDDI_ANTENNA cells depending on switch number or ring
  - At least two EP28SOI_VDDI_BRANCH_END cells

2. **Variable Rules**
    - Number of switches must be in accordance with electrical rules and characteristics relative to switching cells.
    - Number of each source biasing generator must be in accordance with electrical characteristics given for these cells.
    - Number of modular precharge cells must be in accordance with the OPP conditions.

Annexe III

Pin	Pin Name	Type	Function
1	VDDIEN	Digital IN	To command switch on VDDI
2	VDDIMODE	Digital IN	To set Inactive mode (OFF or BIAS) of VDDI
3	VDDIOK	Digital OUT	Status output on VDDI (supply available for activity)
4	VDDISWCURCTRL1	Digital IN	To set current maximum peak at re-powering VDDI supply
5	VDDISWCURCTRL0	Digital IN	To set current maximum peak at re-powering VDDI supply
6	TESTEN	Digital IN	To enable all status outputs
7	VDDIHIGH	Digital OUT	Status output on VDDI level
8	SWVDDILOW	Digital OUT	Status output on swvddi (VDDI power switch command) level
9	VDDIBIASCTRL2	Digital IN	To set the level of internally generated bias voltage references for VDDI
10	VDDIBIASCTRL1	Digital IN	To set the level of internally generated bias voltage references for VDDI
11	VDDIBIASCTRL0	Digital IN	To set the level of internally generated bias voltage references for VDDI
12	BIASCURCTRL1	Digital IN	To set the internally generated current (used in bias voltage generators)
13	BIASCURCTRL0	Digital IN	To set the internally generated current (used in bias voltage generators)
14	VDDIBIASEXT	Digital IN	To use an external voltage reference for biasing level on VDDI
15	VDDIBIASIN	Analog IN	Input for source biasing reference voltage on VDDI (if external usage activated by VDDIBIASEXT) (between gndo and vref or floating)
16	BIASCUREXT	Digital IN	To use an external current reference (used in bias voltage generators)
17	BIASCURIN	Analog IN	Input for external current reference (used in bias voltage generators) (Refer to Section 7.4.4, "Internal Current Reference")
18	BOOST	Digital IN	Not used.For Future improvement.
19	INPROPP0	Digital IN	To select the modular precharge of the dedicated OPP
20	INPROPP1	Digital IN	To select the modular precharge of the dedicated OPP
21	INPROPP2	Digital IN	To select the modular precharge of the dedicated OPP
22	INPROPP3	Digital IN	To select the modular precharge of the dedicated OPP
23	INPROPP4	Digital IN	To select the modular precharge of the dedicated OPP
24	INPROPP5	Digital IN	To select the modular precharge of the dedicated OPP
25	INPROPP6	Digital IN	To select the modular precharge of the dedicated OPP
26	INPROPP7	Digital IN	To select the modular precharge of the dedicated OPP

Pin	Pin Name	Type	Function
27	vref	External power	Reference power supply of bridge dedicated to generation of internal bias reference.
28	vddaa	External power	Power supply used to supply all digital features of controller.
29	vddo	External power	Power supply connected to VDDI, when relative power switch is ON.
30	gndo	External ground	Product ground.
31	vdd1V8	External power	Power supply at 1.8 V (IO-ring VDDE level-like).
32	vddi	Internal power supply	Switchable/Biasable supply dedicated to supply standard cells and S-RAM periphery in the power domain.
33	gnds	Body polarization	Polarization of substrate for switches and in Ctrl and Force. Must be a SG-compatible level.Reconnected to gndo.

Annexe IV

INTERFACE DESCRIPTION

POWER PINS

Signal	Type	Voltage range	Voltage domain (SoC)	Function	Comment
vddcore	power	[0 to 1.3V]	Varm		NOT USED
gnd	ground	0V	gnd Soc	IP ground	
vana	power	[1.62;1.98]	vdd_IO	IP Main supply	1.8V (like VIO supply) (+/- 10% at transistor level)
vddaa	power	[0.8V;1.15V]	Vsafe	Always-on supply	Main digital supply voltage of the IP connected to Vsafe. Min. value is 0.80V

Tab. 2: Power pins table

ANALOG INPUT PINS

Signal	Type	Voltage range	Voltage domain (SoC)	Function	Comment
vbbp_fb	input	[-1.3V ; 0.3V]*	Vdds	Feedback input from vdds grid	
vbbn_fb	input	[0V ; 1.3V]	Gnds	Feedback input from gnds grid	
vref	input	[0 to 1.3V]*	vddcore		NOT USED

Tab. 3: Analog input pins table

ANALOG OUTPUT PINS

Signal	Type	Voltage range	Voltage domain (SoC)	Function	Comment
vbbp	output	[-1.3V ; 0.3]	Vdds	body bias for Pmos	Step of 100mV
vbbn	output	[0V ; 1.3]	Gnds	body bias for Nmos	Step of 100mV
vneg_capa	output	[-1.8; 0]	Negative voltage	Pin used for decoupling internal negative voltage generated by charge pump	To be connected to a tank capacitor on PCB with the lowest resistance possible<1Ohm

DIGITAL INPUT PINS

Signal	Type	Active	Voltage domain (IP)	Function	Comment
spare_bit[3:0]	in			Bit 0 : use to activate CP clock output Bit [1:3]: used to trim internal voltage reference.	Name cannot be changed because of top RTL frozen NOT USED

Signal	Type	Active	Voltage domain (IP)	Function	Comment
spare_bit[3:0]	in			Bit 0 : use to activate CP clock output Bit [1:3]: used to trim internal voltage reference.	Name cannot be changed because of top RTL frozen NOT USED
vbbp_val[3:0]	in	High	vddaa /gndcore	sets vbbp voltage	Programmable step 100mV
vbb_val_update	in	rise edge	vddaa /gndcore	vbbp_val[3:0] and vbbn_val[3:0] are valid	active pos edge
vbb_enable	in	High	vddaa /gndcore	Sets the IP core BB generator ON/OFF	1: IP core BB gen is ON 0: IP is OFF, with minimum leakage
vbbp_compens[3:0]	in	High	vddaa /gndcore	internal loop stability compen-sation adjustment	default=0000
vbbn_val[3:0]	in	High	vddaa /gndcore	sets vbbn voltage	Programmable step 100mV
vbbn_compens[3:0]	in	High	vddaa /gndcore	internal loop stability compen-sation adjustment	default=0000
vbb_cp_compens[2:0]	in	High	vddaa /gndcore	Used to compensate internal CP clk	default=000

vbb_cp_ext_clk	in	High	vddaa /gndcore	Used to select external clock to drive Charge Pump	
vbb_cp_enable	in	High	vddaa /gndcore	Used to enable Charge Pump	
rbb_enable	in	High	vddaa /gndcore	Activate RBB mode	NOT USED
vbb_ext_clk_in	in	high	vddaa /gndcore	External clock in	F must around 200MHz

Tab. 5: Digital input pins table

DIGITAL OUTPUT PINS

Signal	Type	Active	Voltage domain (IP)	Function	Comment
vbb_cp_clk_out	out	High	vddaa /gndcore	Could be used to caliberate internal CP clock	
vbbn_ok	out	High	vddaa /gndcore	output flag high when output voltage has settled. Valid in ON mode (FBB) only	

Résumé

Les systèmes intégrés sont aujourd'hui de plus en plus fréquemment confrontés à des contraintes de faible consommation ou d'efficacité énergétique. Ces problématiques se doivent d'être intégrées le plus en amont possible dans le flot de conception afin de réduire les temps de design et d'éviter de nombreuses itérations dans le flot. Dans ce contexte, le projet collaboratif HiCool, partenariat entre les laboratoires LIRMM et TIMA, les sociétés Defacto, Docea et ST Microelectronics, a mis en place une stratégie et un flot de conception pour concevoir des systèmes intégrés faible consommation tout en facilitant la réutilisation de blocs matériels (IPs) existants. L'approche proposée dans cette thèse s'intègre dans cette stratégie en apportant une petite dose d'asynchronisme dans des systèmes complètement synchrones. En effet, la réduction de la consommation est basée sur le constat que l'activation permanente de la totalité du circuit est inutile dans bien des cas. Néanmoins, contrôler l'activité avec des techniques de « *clock gating* » ou de « *power gating* » nécessitent usuellement d'effectuer un re-design du système et d'ajouter un organe de commande pour contrôler l'activation des zones effectuant un traitement. Le travail présenté dans ce manuscrit définit une stratégie basée sur des contrôleurs d'horloge et de domaine d'alimentation, asynchrones, distribués et facilement insérables dans un circuit avec un coût de re-design des plus réduit.

Mots clés: faible consommation, clock gating, power gating, contrôleurs asynchrones, contrôleurs distribués, flot de conception pour la faible consommation, systèmes synchrones.

Abstract

Today integrated system requirements are more and more dealing with low power and energy efficiency constraints. These issues have to be early addressed in the design flow in order to reduce design time and avoid as much as possible iterations. In the framework of the collaborative HiCool project - with the laboratories LIRMM and TIMA, and the companies, Defacto, Docea and ST Microelectronics - a design strategy and its associated design flow has been developed in order to enhance low power characteristics of integrated circuits and ease the reuse of existing hardware blocks (IPs). The proposed approach in this PhD thesis fits into this strategy by integrating a small amount of asynchrony in completely synchronous systems. Indeed, the power reduction is based on the observation that permanently stimulating the entire circuit is unnecessary in many cases. However, controlling the activity with techniques such as "*clock gating"* or "*power gating*" usually needs to redesign the system and to add circuitry for activating the blocks that really processing the data. The work, presented in this manuscript, provides a new strategy based on asynchronous distributed clock and power domain controllers that can easily be inserted at a very low redesign cost.

Keywords: low power, *clock gating, power gating*, asynchronous controllers, distributed controllers, low-power design flow, synchronous systems.

ISBN : 978-2-11-129209-3