



HAL
open science

REDUCTION DE L'EMISSION ELECTROMAGNETIQUE DES CIRCUITS INTEGRES : L'ALTERNATIVE ASYNCHRONE

D. Panyasak

► **To cite this version:**

D. Panyasak. REDUCTION DE L'EMISSION ELECTROMAGNETIQUE DES CIRCUITS INTEGRES : L'ALTERNATIVE ASYNCHRONE. Micro et nanotechnologies/Microélectronique. Institut National Polytechnique de Grenoble - INPG, 2004. Français. NNT: . tel-00007775

HAL Id: tel-00007775

<https://theses.hal.science/tel-00007775>

Submitted on 16 Dec 2004

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

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

THESE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : Microélectronique

préparée au laboratoire **TIMA** dans le cadre de l'Ecole Doctorale d'
« **Electronique, Electrotechnique, Automatique, Télécommunications, Signal** »

présentée et soutenue publiquement

par

Dhanistha PANYASAK

Le 14 Juin 2004

Titre :

***REDUCTION DE L'EMISSION ELECTROMAGNETIQUE DES
CIRCUITS INTEGRES : L'ALTERNATIVE ASYNCHRONE***

Directeur de Thèse : Marc RENAUDIN

Codirecteur : Gilles SICARD

JURY

M. Christian PIGUET	, Rapporteur
M. Etienne SICARD	, Rapporteur
M. Jean-Pierre SHOELLKOPF	, Examineur
M. Olivier SENTIEYS	, Président du Jury
M. Marc RENAUDIN	, Directeur
M. Gilles SICARD	, Codirecteur

REMERCIEMENTS

Cette thèse a été effectuée au sein du laboratoire TIMA (Techniques de l'Informatique et de la Microélectronique pour l'Architecture des ordinateurs) sur le site Viallet de l'Institut National Polytechnique de Grenoble. Je tiens à remercier son directeur, Mr. Bernard Courtois, pour m'avoir permis d'effectuer ma thèse dans son laboratoire, ainsi que pour son écoute et ses encouragements.

Je remercie vivement les membres de mon Jury de thèse :

Mr. Christian Piguet, professeur à l'Ecole Polytechnique Fédérale de Lausanne, pour avoir accepté de corriger mon rapport de thèse, et pour m'avoir fait l'honneur de participer à mon jury.

Mr. Etienne Sicard, professeur à Institut National des Sciences Appliquées de Toulouse pour avoir apporté des corrections à mon manuscrit et pour avoir accepté de participer à mon jury.

Mr. Olivier Sentieys, professeur à Ecole Nationale Supérieure des Sciences Appliquées et de Technologie de Lannion, pour m'avoir fait l'honneur de présider mon jury de thèse.

Mr Jean-Pierre Schoellkopf, chef du département ADT de STMicroelectronics pour avoir fait parti de mon jury de thèse. Entant que responsable industriel, il a toujours été à l'écoute et a su porter un réel intérêt à mon travail.

Je tiens à exprimer toute ma gratitude à mon directeur de thèse, Mr. Marc Renaudin, professeur à l'Ecole Nationale Supérieure de Radioélectricité de Grenoble. Il m'a guidée dans les dédales de la conception asynchrone et m'a fourni de précieux conseils pour que je puisse accomplir ce travail de thèse.

Je tiens à exprimer ma profonde reconnaissance à Mr. Gilles Sicard (ou plus sympathiquement Gillou), Maître de Conférence à Université Joseph Fourier, mon co-encadrant de thèse, pour son soutien, ses corrections, sa compagnie dans les conférences, son écoute et sa bonne humeur.

Je tiens aussi à remercier Mr. Laurent Fesquet, Maître de Conférence à l'Ecole Nationale Supérieure de Radioélectricité de Grenoble, pour ses encouragements et ses nombreux conseils.

Je remercie tout particulièrement Mr. Pierre Gentil, Directeur de l'Ecole Doctorale EEATS de Grenoble, qui a accepté que j'effectue mon DEA Microélectronique et ma thèse au sein de son école.

Je tiens aussi à avoir une attention toute particulière pour Mr. Gilbert Vincent responsable du DEA Microélectronique, qui m'a encouragée dans cette voie. Ses cours de licence et de maîtrise m'ont fait découvrir et aimer la microélectronique.

Je souhaite remercier sincèrement Mr. Jean-François Hugues de la société STMicroelectronics pour m'avoir accueillie et encouragée durant mon stage de DEA et au début de ma thèse.

Je remercie également Mr. Ernesto Perea de la société STMicroelectronics pour m'avoir conseillée et orientée vers mon stage de DEA et ma thèse.

Je voudrais remercier également les administrateurs du CIME : Alexandre Chagoya pour avoir géré mon compte avec soin et Bernard Bosc pour les impressions de mes posters.

Je souhaite ici remercier chaleureusement les personnes du TIMA qui m'ont apporté du réconfort par leur gentillesse, leurs sourires, leurs encouragements d'autres par leurs conseils techniques. Elles ont fait de mon environnement de travail, un environnement très agréable :

Seb F. et Fred C. au cœur du système, Salvador, Libor, Hubert, et sa grande sympathie, Françoise et sa mine inépuisable d'articles scientifiques, Gérard et ses histoires sur le triathlon, Emmanuel S., Benoît et ses photos truquées, Kholdoun et son aide technique très précieuse, Skandar et sa passion pour les différences, Paul et sa sacoche, Patricia, Ahmed et leur bonne humeur, Jean-François et son humour, Sylvaine et son sourire, Joëlle et son accueil à la compta, Marie-Christine et ses récits d'aventure, Chantal, Isabelle A. et leur gentillesse, Anne-Laure qui apporte un nouveau souffle au labo, Corinne et sa fille Sabine, Lucie et son sourire aux fournitures, Julie qui vient aider de temps en temps. Iris, Raoul et ses énigmes, Alain et sa sympathie, Antoine, Dominique la plus dynamique, Claude et ses conseils en VHDL, Pierre et son soutien, Eric, Bénédicte, Nacer et ses conseils, Fred R, son aide technique et ses histoires de kangourous, Sonya et son accent sympathique, Amel, Nicolas G., Christian (Couzi), Guillaume (James Bond), Yasser, Fabien et sa motivation, Lorena et son écoute, Karim, Susie, Paul et Gilles qui ne s'arrêtent jamais, Faiza. Ghiath, Julien, Emile, Diana, Nicolas, les jolies Lobna et Iuliana, Fred H. et ses boulettes de papier.

Merci aussi à ceux qui m'ont laissé un souvenir de leur sympathie : Rosario d'Espagne, Ong, Eng, Eugene, Yii Rui, Andrea, Sonia, Robin, Thibault, Arnaud (la piplette), Anthony (petit canard), Emilie, Pierre, Thai et ses OuiOuiOui, Gwenole l'aventurier, Fred L. (découvert à mon bureau), Julien futur thésard du CIS, Pierrick, Damien C. thésard oublié.

Comment ne pas remercier aussi Gégé du LYS et les 3 mousquetaires de l'ancien groupe Asynchrone d'ADT? : Pascal (très attentif à mes travaux), Fred R. et Alexandre.

Durant ces années de thèse j'ai rencontré des personnes qui ont égayé ma vie de thésarde à l'intérieur et à l'extérieur du labo. J'ai appris à les connaître et à les apprécier. Je voudrais leur dire un grand MERCI pour tous les bons moments que nous avons passés ensemble :

Le comité d'organisation des JNRDM, toujours motivé et motivant pour organiser de nouveaux évènements ! : Carlo, Delphine, Caroline, François D., François L, Fred D., Bogdan, Jérémy, Kruno.

Ceux qui m'ont beaucoup appris sur la différence et la richesse de leur culture : Yen-Min, Jacky, Jackie, James, Kian, Donna.

Ceux qui sont partis trop vite : Elizabeth la cantatrice, Philippe (Phi) et Erwan (Wanou), qui m'apportaient, chaque jour, un rayon de soleil lorsque je travaillais seule à la cave.

Puis un groupe à part entière, rythmant activement ma thèse par de franches rigolades, des soirées animées... :

- Les "tous nouveaux" : Aurélien et son humour glauque, la petite Livière, David le plus motivé de tous et sa femme Katia.

- Les "un peu plus anciens" : Joao et Marianna et leur succulente recette aux haricots, Fabien le rugbyman du groupe (meilleur conteur de blagues que je connaisse) et Juliette, Beber et ses soirées kart, Sophie championne de taekwondo et organisatrice de laser-game, Vivi, Yannick le top model du calendrier, Fraidy futur président du Congo, Arnaud le roi de la salsa et des pompes, Isabelle et ses fous rires, les motards : Yanick et Emeline, Greg et sa bergère, Sébastien C.; Guillaume de l'ICP, Cyrille, Marie-Ange et leurs soirées Barbecue, Salim le plus sportif, Yann (l'homme à la banane), Estelle (qui prend le relais pour mes fiches légendaires) et son fiancé Christophe.

- Les "anciens", la plupart étant déjà partis : Damien (l'autre D.), Lovic (Lov'), Jérôme et ses 400 coups, Mohamed, Kamel le plus musclé, Amine le plus généreux. Manu (Brad Pitt), JB (la peste qu'on adore), Anh-Vu (plus connu sous le nom de Frère Bob).

Je remercie énormément ma famille qui m'a encouragée durant ces années de thèse: Khambao et Kheuakham mes parents, Thnimphorn et Atsavinn mes deux sœurs et mon frère Seriseng.

Enfin MERCI à Fred pour m'avoir écoutée et encouragée durant toutes ces années.

Table des matières

REMERCIEMENTS	5
TABLE DES ILLUSTRATIONS	XVIII
INTRODUCTION	1
CHAPITRE 1 INTERFERENCES ELECTROMAGNETIQUES	1
1.1 INTRODUCTION.....	1
1.2 GENERALITES.....	1
1.2.1 Champ électromagnétique	2
1.2.1a Champ magnétique	2
1.2.1b Champ électrique	2
1.2.1c Champ électromagnétique	2
1.2.2 Interférences électromagnétiques	5
1.2.2a Définition	5
1.2.2b Emission.....	7
1.2.2c Susceptibilité.....	8
1.2.2d Bruit de mode commun et de mode différentiel	9
1.2.3 Compatibilité Electromagnétique.....	10
1.2.3a Motivations.....	10
1.2.3b Définition de la compatibilité électromagnétique	10
1.2.3c Normes CEM.....	11
1.3 CEM DES CIRCUITS INTEGRES	11
1.3.1 Interférences électromagnétiques	11
1.3.1a Principaux acteurs des interférences électromagnétiques	11
1.3.1b Conséquences du phénomène d'interférence électromagnétique	14
1.3.1c Circuits numériques et analogiques	15
1.3.1d Réduction des interférences électromagnétiques	15
1.3.2 Normes IEC	16
1.3.2a Auto-compatibilité électromagnétique	16
1.3.2b Standards de modélisation.....	17
1.3.2c IBIS (I/O Buffer Information Specification)[MAIO99]	17
1.3.2d ICEM (Integrated Circuit Electromagnetic compatibility Model)	18
1.3.3 Le générateur Ib dans le modèle ICEM : l'activité du coeur du circuit.....	19
1.4 CONCLUSION.....	19

CHAPITRE 2 METHODES DE REPARTITION DE L'ACTIVITE DU COURANT	21
.....	
2.1 INTRODUCTION	21
2.2 ETAPES DE LA CONCEPTION CONCERNEES	22
2.2.1 Au niveau portes.....	22
2.2.2 Au niveau registres.....	22
2.2.3 A la description comportementale.....	23
2.3 BUT DES METHODES DE REPARTITION	23
2.3.1 Spectre de l'activité du courant.....	23
2.3.1a Spectre en fréquence d'un signal périodique	24
2.3.1b Spectre en fréquence d'un signal aléatoire	26
2.3.2 Principe des méthodes de répartition de l'activité du courant	26
2.4 ETALEMENT DU SPECTRE DE L'HORLOGE.....	27
2.4.1 Principe.....	27
2.4.2 Applications et résultats	29
2.5 CONTROLE DU DEPHASAGE DE L'HORLOGE (<i>CLOCK SKEW OPTIMIZATION</i>)	30
2.5.1 Déphasage d'horloge (<i>clock skew</i>).....	30
2.5.2 Arbre d'horloge.....	30
2.5.3 Contraintes de <i>zero clocking</i> et <i>double clocking</i>	31
2.5.4 Principe de la méthode	31
2.5.5 Description de différents travaux	31
2.5.5a Travaux de L. Benini et al. [<i>BENI97</i>]	31
2.5.5b Travaux de I. Blunno et al. [<i>BLUN02</i>]	33
2.5.5c Travaux de E. Mussol et J. Cortadella [<i>MUSS96</i>]	35
2.6 ORDONNANCEMENT DE LA PUISSANCE DYNAMIQUE	36
2.6.1 Ordonnancement	36
2.6.2 Principe.....	36
2.6.3 Description de différents travaux	36
2.6.3a Travaux de A. Azevedo et al. [<i>AZEV01</i>]	36
2.6.3b Travaux de R.S. Martin et J.P. Knight [<i>MART96</i>]	37
2.7 CONCLUSION.....	37
CHAPITRE 3 CIRCUITS ASYNCHRONES	39
3.1 INTRODUCTION	39
3.2 CONCEPTS DE BASE	40
3.2.1 Principe de fonctionnement.....	40
3.2.1a Fonctionnement d'un circuit synchrone	40

3.2.1b	<i>Fonctionnement d'un circuit asynchrone, notion de canal</i>	40
3.2.2	Caractéristiques d'un opérateur asynchrone	41
3.2.2a	<i>Latence</i>	42
3.2.2b	<i>Cycle</i>	42
3.2.2c	<i>Profondeur du pipeline</i>	42
3.3	IMPLEMENTATION DES CIRCUITS ASYNCHRONES	42
3.3.1	Aléas	42
3.3.2	Gestion des aléas	43
3.3.3	Catégories de circuits asynchrones	44
3.3.3a	<i>Circuits Insensibles aux Délais (DI)</i>	44
3.3.3b	<i>Circuits Quasi Insensibles au Délais (QDI)</i>	45
3.3.3c	<i>Circuits indépendants de la vitesse (SI)</i>	45
3.3.3d	<i>Circuits micropipelines</i>	45
3.3.3e	<i>Circuits de Huffman</i>	46
3.3.4	Protocoles de communication	46
3.3.4a	<i>Protocole 2 phases</i>	46
3.3.4b	<i>Protocole 4 phases</i>	47
3.3.5	Codage des données	50
3.3.5a	<i>Codage données groupées</i>	50
3.3.5b	<i>Codage quatre états</i>	50
3.3.5c	<i>Codage 3 états</i>	51
3.3.6	Porte de Muller	52
3.3.7	Duplicateur et Rendez-vous	53
3.3.7a	<i>Duplicateur</i>	53
3.3.7b	<i>Rendez-vous</i>	54
3.3.8	Latches vs Flip-flop	54
3.3.9	Délais	55
3.3.9a	<i>Le délai classique</i>	55
3.3.9b	<i>Le délai asymétrique</i>	56
3.4	EMERGENCE DES CIRCUITS ASYNCHRONES	56
3.4.1	Performances des circuits asynchrones	57
3.4.1a	<i>Emission électromagnétique</i>	57
3.4.1b	<i>Signature électrique</i>	58
3.4.1c	<i>Consommation</i>	59
3.4.1d	<i>Surface</i>	59
3.4.1e	<i>Dissipation thermique</i>	59
3.4.1f	<i>Vitesse</i>	60

3.4.2 Concevoir et produire des circuits asynchrones	60
3.4.2a Méthodes et outils de conception dédiés aux circuits asynchrones	60
3.4.2b GALS, GSLA	62
3.4.2c Circuits Désynchronisés	63
3.5 CONCLUSION.....	64
CHAPITRE 4 DE LA DESCRIPTION RTL AU CIRCUIT ASYNCHRONE :	
ASYNCHRONISATION OU DE-SYNCHRONISATION.....	65
4.1 INTRODUCTION	65
4.2 CONTEXTE	66
4.2.1 Principe.....	66
4.2.2 Contraintes.....	66
4.2.3 Choix de la méthode.....	66
4.2.3a Description du circuit synchrone	67
4.2.3b Approche par graphe des dépendances	67
4.2.3c Utilisation d'un langage standard (VHDL ou Verilog)	68
4.2.4 Introduction de la méthode dans le flot de conception.....	68
4.2.5 Elaboration de la méthode.....	69
4.3 CONCEPTS DU FONCTIONNEMENT SYNCHRONE	69
4.3.1 Briques de bases : les processus	70
4.3.1a Processus combinatoire	70
4.3.1b Processus à mémoire	70
4.3.2 Signaux d'horloge	72
4.3.3 Signaux d'initialisation	72
4.3.4 Compositions.....	72
4.3.4a Dépendance	73
4.3.4b Interdépendance	74
4.4 ARCHITECTURE GENERALE POUR L'ASYNCHRONISATION	75
4.5 ASYNCHRONISATION	76
4.5.1 Séquence des événements	76
4.5.1a Dépendances et canaux	76
4.5.1b Jetons et bulles	77
4.5.1c Protocole découplé, protocole non découplé	78
4.5.1d Validité des données	79
4.5.2 Initialisation.....	79
4.5.2a Initialisation des signaux de contrôle.....	79
4.5.2b Initialisation des signaux de donnée	79

4.5.2c Convention pour l'initialisation	79
4.5.3 Conditions de vivacité et bon fonctionnement	80
4.5.3a L'inter-blocage.....	80
4.5.3b La correction fonctionnelle.....	83
4.5.3c Conditions de vivacité et de bon fonctionnement	85
4.5.4 Compatibilité de cycle	86
4.6 COMPOSANTS POUR L'ASYNCHRONISATION	87
4.6.1 Eléments pour les dépendances multiples	87
4.6.2 Tampons (<i>buffer</i> , <i>half buffer</i>).....	87
4.6.2a Tampon sans jeton d'initialisation	88
4.6.2b Tampon avec jeton d'initialisation.....	88
4.6.2c Interfaces d'entrée et de sortie.....	89
4.7 CHOIX D'UN PROTOCOLE	90
4.7.1 Latence du circuit	90
4.7.2 Surface	91
4.8 METHODE.....	92
4.8.1 Analyse du code RTL	92
4.8.2 Description et définition de l'interface.....	93
4.8.3 Passage du synchrone à l'asynchrone	93
4.8.4 La synthèse : utilisation des outils standards.....	93
4.9 IMPLEMENTATION	94
4.9.1 Filtre à Réponse impulsionnelle finie 4 coefficients	94
4.9.2 Application de la méthode	94
4.9.2a Analyse du code RTL	94
4.9.2b Description et définition de l'interface entrées/sorties	96
4.9.2c Passage du synchrone à l'asynchrone	96
4.9.2d Implémentations et Résultats	97
4.10 CONCLUSION.....	98
CHAPITRE 5 METHODE "MISE FORME DU COURANT" (<i>CURRENT SHAPING</i>).....	101
5.1 INTRODUCTION.....	101
5.2 CONTEXTE.....	102
5.2.1 Contraintes.....	102
5.2.2 Choix	102
5.2.2a La méthode s'inspire de l'ordonnancement de la puissance dynamique..	102
5.2.2b La méthode s'applique au niveau architecture	102

5.2.3 Principe.....	103
5.2.4 Fonction de coût.....	103
5.2.4a Activité du courant.....	104
5.2.4b Activité globale du courant.....	104
5.3 APPROCHE.....	106
5.3.1 Modélisation structurelle de l'architecture.....	107
5.3.1a DFG/CDFG.....	107
5.3.1b PN/DFG.....	107
5.3.1c Choix du graphe de représentation de l'architecture.....	108
5.3.1d Compositions.....	108
5.3.2 Estimation du temps de calcul des blocs de l'architecture.....	108
5.3.3 Modélisation des profils de courants des opérateurs.....	109
5.3.3a Analyse du protocole : Phases de l'activité en courant.....	109
5.3.3b Raffinement du CDFG en phases de courant.....	109
5.3.3c Modèle de courant dans une phase.....	110
5.3.3d Allure du courant global.....	110
5.3.3e Influence de la modélisation des profils de courant sur la fonction de coût.....	110
5.3.4 Optimisation du profil du courant global par ordonnancement.....	111
5.3.4a Définition des pas en temps.....	111
5.3.4b Choix de l'algorithme.....	112
5.3.4c Force Directed Scheduling.....	115
5.4 METHODE.....	121
5.4.1 Définition de la méthode.....	121
5.4.2 Illustration de la méthode.....	122
5.4.2a Analyse du modèle de l'architecture.....	122
5.4.2b Utilisation des modèles de courant.....	122
5.4.2c Définition de pas en temps (ou pas de contrôle) et discrétisation des modèles.....	123
5.4.2d Application de l'algorithme.....	124
5.5 IMPLEMENTATION.....	124
5.5.1 Filtre à Réponse Impulsionnelle 4 coefficients.....	124
5.5.2 Application de la méthode.....	125
5.5.3 Résultats.....	126
5.6 CONCLUSION.....	126

CHAPITRE 6 ACTIVITE DU COURANT DANS LES PROTOCOLES 4 PHASES

.....	129
6.1 INTRODUCTION.....	129
6.2 PROTOCOLES 4 PHASES.....	130
6.2.1 Décomposition des protocoles en phases	130
6.2.2 Principales phases de l'activité du courant dans un contrôleur.....	131
6.2.3 Activité du courant dans un multiplieur utilisant un contrôleur WCHB.....	132
6.2.4 Nature des dépendances entre les phases de l'activité du courant.....	135
6.3 MODELISATIONS UTILISEES	137
6.3.1a <i>HandShaking Expansion (HSE)</i>	137
6.3.1b <i>HSE Temporel</i>	138
6.3.1c <i>STG (Signal Transition Graph)</i>	139
6.3.1d <i>CDFG (Control Data Flow Graph) et DFG (Data Flow Graph)</i>	139
6.3.1e <i>CDFG déroulé dans le temps</i>	140
6.3.1f <i>Modélisation structurelle</i>	141
6.3.1g <i>Choix d'une modélisation</i>	141
6.4 DELAIS.....	142
6.4.1a <i>Délais intrinsèques (délais fonctionnels)</i>	142
6.4.1b <i>Délais pour la faible émission électromagnétique</i>	143
6.5 DESCRIPTION DE L'ETUDE.....	144
6.6 LE PROTOCOLE WCHB	145
6.6.1 Formalisme	145
6.6.2 Architecture du système étudié.....	145
6.6.3 Fonctionnement du protocole	146
6.6.3a <i>A l'initialisation</i>	146
6.6.3b <i>Phase 1 de l'opérateur B</i>	146
6.6.3c <i>Phase 2 de l'opérateur B</i>	147
6.6.3d <i>Phase 3 de l'opérateur B</i>	147
6.6.4 Phases de l'activité du courant.....	148
6.6.4a <i>Modélisation par CDFG</i>	148
6.6.4b <i>Temps de cycle d'un opérateur</i>	149
6.6.5 Dépendances.....	149
6.6.5a <i>Influence des évènements</i>	149
6.6.5b <i>Les types de dépendances</i>	150
6.6.5c <i>Dépendances lâches et fixes</i>	151
6.6.6 Contrôle des dépendances	152

6.6.6a	<i>Dépendances lâches</i>	152
6.6.6b	<i>Dépendance fixe</i>	153
6.6.6c	<i>Contrôleur générique</i>	154
6.6.7	Autres structures.....	155
6.6.7a	<i>Le rendez-vous (Join)</i>	156
6.6.7b	<i>Le duplicateur</i>	157
6.6.7c	<i>Multiplexeur</i>	157
6.6.7d	<i>Démultiplexeur</i>	158
6.7	PROTOCOLE PCHB.....	159
6.7.1	Formalisme.....	159
6.7.2	Architecture du système étudié.....	160
6.7.3	Fonctionnement du protocole.....	161
6.7.3a	<i>A l'initialisation</i>	161
6.7.3b	<i>Phase 1 de l'opérateur B</i>	161
6.7.3c	<i>Phase 2 de l'opérateur B</i>	162
6.7.3d	<i>Phase 3 de l'opérateur B</i>	162
6.7.4	Phases de l'activité du courant.....	163
6.7.4a	<i>Modélisation par CDFG</i>	163
6.7.4b	<i>Temps de cycle d'un opérateur</i>	164
6.7.5	Dépendances.....	164
6.7.5a	<i>Influence des évènements</i>	164
6.7.5b	<i>Les types de dépendances</i>	165
6.7.5c	<i>Dépendances lâches et fixes</i>	165
6.7.6	Contrôle des dépendances.....	166
6.7.6a	<i>Dépendances lâches</i>	166
6.7.6b	<i>Dépendances fixes</i>	167
6.7.6c	<i>Contrôleur générique</i>	168
6.7.7	Autres structures.....	169
6.8	PROTOCOLE PCFB.....	171
6.8.1	Formalisme.....	171
6.8.2	Architecture du système étudié.....	172
6.8.3	Fonctionnement du protocole.....	173
6.8.3a	<i>A l'initialisation</i>	173
6.8.3b	<i>Phase 1 de l'opérateur B</i>	173
6.8.3c	<i>Phase 2 de l'opérateur B</i>	174
6.8.3d	<i>Phase 3 de l'opérateur B</i>	174
6.8.4	Les phases de l'activité du courant.....	175

6.8.4a Modélisation par CDFG	175
6.8.4b Temps de cycle d'un opérateur	176
6.8.5 Dépendances	176
6.8.5a Influence des évènements	176
6.8.5b Type de dépendances	177
6.8.5c Dépendances fixes et lâches	177
6.8.6 Contrôle des dépendances	178
6.8.7 Autres structures	181
6.9 SYNTHÈSE DE L'ÉTUDE DES PROTOCOLES	183
6.10 UTILISATION DE L'ÉTUDE	184
6.10.1 Périodicité des protocoles.....	184
6.10.2 Protocole de communication et répartition du courant	188
6.11 CONCLUSION.....	188
CONCLUSION.....	189
BIBLIOGRAPHIE	193
BIBLIOGRAPHIE DE L'AUTEUR.....	203
GLOSSAIRE.....	205
ANNEXE	209

Table des illustrations

<i>Figure 1-1 : Effets du champ magnétique</i>	2
<i>Figure 1-2 : Effet du champ électrique</i>	2
<i>Figure 1-3 : Champ électromagnétique [OBER01]</i>	3
<i>Figure 1-4 : Champ rayonné</i>	4
<i>Figure 1-5 : Tout système électrique peut se comporter comme une source de bruit</i>	5
<i>Figure 1-6 : Le phénomène des interférences électromagnétiques</i>	6
<i>Figure 1-7 : Tout système peut être perturbé par son environnement</i>	6
<i>Figure 1-8 : L'émission rayonnée</i>	7
<i>Figure 1-9 : Principales fréquences utilisées</i>	7
<i>Figure 1-10 : L'émission conduite</i>	8
<i>Figure 1-11 : Tout système électrique peut être sujet à des champs électromagnétiques</i>	8
<i>Figure 1-12 : Modélisation du bruit de mode différentiel</i>	9
<i>Figure 1-13 : Modélisation du bruit de mode commun</i>	9
<i>Figure 1-14 : Un circuit est un pollueur et une victime potentielle</i>	10
<i>Figure 1-15 : Appel de courant dans une structure CMOS [BEND03]</i>	12
<i>Figure 1-16 : Le boîtier</i>	13
<i>Figure 1-17 : Interconnexions</i>	13
<i>Figure 1-18 : Modèle ICEM</i>	18
<i>Figure 2-1 : Coûts liés à la résolution des problèmes de CEM [SICA03]</i>	21
<i>Figure 2-2 : Les différentes méthodes de répartition du courant</i>	22
<i>Figure 2-3 : Un signal périodique et son spectre fréquentiel</i>	25
<i>Figure 2-4 : Allure du spectre fréquentiel d'un circuit synchrone</i>	25
<i>Figure 2-5 : signal aléatoire et son spectre fréquentiel</i>	26
<i>Figure 2-6 : Harmonique d'horloge avec et sans modulation [HARD94]</i>	28
<i>Figure 2-7 : Différentes fonctions a, c, e utilisées pour la modulation et résultats obtenus (b, d, f) [HARD94]</i>	28
<i>Figure 2-8 : Arbre d'horloge</i>	30
<i>Figure 2-9 : Groupement des éléments séquentiels</i>	32
<i>Figure 2-10 : Graphe de contraintes</i>	34
<i>Figure 2-11 : Horloge à plusieurs phases</i>	35
<i>Figure 3-1 : Circuit synchrone</i>	40
<i>Figure 3-2 : Circuit asynchrone</i>	41
<i>Figure 3-3 : Communication de type poignée de main (handshake)</i>	41

<i>Figure 3-4: Opérateurs asynchrones</i>	42
<i>Figure 3-5 : Catégories de circuits asynchrones</i>	44
<i>Figure 3-6 : Fourche isochrone</i>	45
<i>Figure 3-7 : Circuit micropipeline</i>	46
<i>Figure 3-8 : Protocole 2 phases</i>	46
<i>Figure 3-9 : Protocole 4 phases</i>	47
<i>Figure 3-10 : Opérateur Asynchrone</i>	47
<i>Figure 3-11 : Protocole WCHB</i>	48
<i>Figure 3-12 : Protocole PCHB</i>	48
<i>Figure 3-13 : Protocole PCFB</i>	49
<i>Figure 3-14 : Codage données groupées</i>	50
<i>Figure 3-15 : Codage 4 états</i>	51
<i>Figure 3-16 : Codage 4 états</i>	51
<i>Figure 3-17 : Codage 3 états</i>	51
<i>Figure 3-18 : Schéma de la porte de Muller</i>	52
<i>Figure 3-19 : Table de vérité d'une porte de Muller à 2 entrées</i>	52
<i>Figure 3-20 : Possible implémentation d'une porte de Muller</i>	53
<i>Figure 3-21 : Implémentation d'une Muller en portes standard</i>	53
<i>Figure 3-22 : Symbole et schéma de structure du dupicateur</i>	54
<i>Figure 3-23 : Symbole et schéma de structure du rendez-vous</i>	54
<i>Figure 3-24 : Architecture d'un latche</i>	55
<i>Figure 3-25 : Architecture d'une flip flop</i>	55
<i>Figure 3-26 : Symbol du délai</i>	55
<i>Figure 3-27 : Le délai classique et sa réponse</i>	56
<i>Figure 3-28 : Délai asymétrique et fonctionnement</i>	56
<i>Figure 3-29 : Spectre de l'activité du courant dans un CAN synchrone et asynchrone</i>	58
<i>Figure 3-30 : Cryptoprocasseur QDI [BOUE02]</i>	58
<i>Figure 3-31 : Flot TAST [RENA03]</i>	61
<i>Figure 3-32 : Le projet TAST</i>	62
<i>Figure 4-1 : Etape de l'asynchronisation dans le flot de conception</i>	69
<i>Figure 4-2 : Processus combinatoire</i>	70
<i>Figure 4-3 : Processus à mémoire</i>	70
<i>Figure 4-4 : Les différents types de processus à mémoire</i>	71
<i>Figure 4-5 : Reset asynchrone et synchrone</i>	72
<i>Figure 4-6 : Structure synchrone</i>	73
<i>Figure 4-7 : Dépendance simple a entre 2 processus</i>	73
<i>Figure 4-8: A et B sont interdépendants.</i>	74
<i>Figure 4-9 : Asynchronisation : principe général</i>	76

<i>Figure 4-10 : Un canal de communication par dépendance</i>	<i>77</i>
<i>Figure 4-11 : Modélisation de la présence d'une requête</i>	<i>77</i>
<i>Figure 4-12: Modélisation par jeton [RENA04].....</i>	<i>78</i>
<i>Figure 4-13 : Bulle et Jeton</i>	<i>78</i>
<i>Figure 4-14 : Jetons valides et invalides dans le protocole non découplé</i>	<i>78</i>
<i>Figure 4-15 : Implémentation du reset synchrone</i>	<i>80</i>
<i>Figure 4-16 : Implémentation du reset asynchrone</i>	<i>80</i>
<i>Figure 4-17 : Structure ARCHI_A0 équivalente à ARCHI0.....</i>	<i>81</i>
<i>Figure 4-18 : Circulation des requêtes dans ARCHI_A0</i>	<i>82</i>
<i>Figure 4-19 : Structure ARCHI_A1 équivalente à ARCHI1</i>	<i>82</i>
<i>Figure 4-20 : Pas de circulation des requêtes dans ARCHI_A1</i>	<i>83</i>
<i>Figure 4-21 : ARCHI_A1' et ARCHI_A1"</i>	<i>84</i>
<i>Figure 4-22 : Un anneau auto séquencé.....</i>	<i>85</i>
<i>Figure 4-23 : L'interface d'entrée assure la cycle compatibilité</i>	<i>86</i>
<i>Figure 4-24 : Rendez-vous et duplicateur.....</i>	<i>87</i>
<i>Figure 4-25 : Un tampon</i>	<i>87</i>
<i>Figure 4-26 : Structure du contrôleur utilisé pour un tampon sans jeton d'initialisation..</i>	<i>88</i>
<i>Figure 4-27 : Structure du contrôleur utilisé pour un tampon avec jeton d'initialisation..</i>	<i>89</i>
<i>Figure 4-28 : Interfaces d'entrée et de sortie.....</i>	<i>90</i>
<i>Figure 4-29 : Structure synchrone</i>	<i>90</i>
<i>Figure 4-30 : Structure asynchrone équivalente.....</i>	<i>91</i>
<i>Figure 4-31 : Boucle utilisant un protocole découplé</i>	<i>91</i>
<i>Figure 4-32 : Structure utilisant le protocole non découplé.....</i>	<i>92</i>
<i>Figure 4-33 : FIR synchrone.....</i>	<i>94</i>
<i>Figure 4-34 : Processus combinatoires et à mémoire du filtre.....</i>	<i>95</i>
<i>Figure 4-35 : Graphe de dépendances des registres du filtre.....</i>	<i>95</i>
<i>Figure 4-36 : Filtre asynchronisé</i>	<i>96</i>
<i>Figure 4-37 : Structure du filtre asynchronisé optimisé.....</i>	<i>97</i>
<i>Figure 4-38 : Réponse indicielle du filtre synchrone.....</i>	<i>97</i>
<i>Figure 4-39 : Réponse indicielle du filtre asynchrone.....</i>	<i>98</i>
<i>Figure 4-40 : Activités électriques des 2 filtres</i>	<i>98</i>
<i>Figure 5-1 : Allure du courant consommé dans un circuit asynchrone.....</i>	<i>101</i>
<i>Figure 5-2 : Niveaux de conception du circuit</i>	<i>103</i>
<i>Figure 5-3 : Valeur de l'activité du courant dans un opérateur</i>	<i>104</i>
<i>Figure 5-4 : N opérateurs effectuant leur traitement en parallèle.</i>	<i>104</i>
<i>Figure 5-5 : Cas où un opérateur est composé de plusieurs opérateurs en série.....</i>	<i>104</i>
<i>Figure 5-6 : Un circuit (a.) et son CDFG (b.)</i>	<i>107</i>
<i>Figure 5-7 : Motifs d'un CDFG</i>	<i>108</i>

<i>Figure 5-8 : Activité du courant dans l'opérateur B</i>	109
<i>Figure 5-9 : Modélisation d'un opérateur par ses phases d'activité en courant (a.)</i> <i>Modélisation de l'activité du courant d'une phase (b.)</i>	110
<i>Figure 5-10 : Définition des pas en temps</i>	112
<i>Figure 5-11 : Pipeline</i>	114
<i>Figure 5-12 : Discrétisation du modèle de courant</i>	115
<i>Figure 5-13 : Modélisation par une force d'une affectation d'une opération à un pas</i>	116
<i>Figure 5-14 : CDFG de l'exemple</i>	116
<i>Figure 5-15 : Pas de contrôle</i>	117
<i>Figure 5-16 : ASAP (à gauche) et ALAP (à droite) de l'exemple de la figure 5.14</i>	117
<i>Figure 5-17 : Time Frame de l'exemple 1</i>	117
<i>Figure 5-18 : Diagrammes de distribution</i>	118
<i>Figure 5-19 : Modélisation d'un DG</i>	119
<i>Figure 5-20 : Modélisation de la probabilité d'une affectation d'une opération à une étape de contrôle</i>	119
<i>Figure 5-21 : Variation d'une force</i>	119
<i>Figure 5-22 : Influence de l'affectation d'une opération sur une autre</i>	120
<i>Figure 5-23 : Modification du time frame 2</i>	121
<i>Figure 5-24 : Méthode pour la répartition du courant</i>	122
<i>Figure 5-25 : Des concurrences sont identifiées</i>	122
<i>Figure 5-26 : Utilisation des modèles de courant</i>	123
<i>Figure 5-27 : Calcul des pas</i>	123
<i>Figure 5-28 : Discrétisation des modèles</i>	123
<i>Figure 5-29 : Allure du courant dans chaque opérateur avant (a.) et après (b.) application de la méthode</i>	124
<i>Figure 5-30 : Allure du courant global avant (a.) et après (b.) application de la méthode</i>	124
<i>Figure 5-31 : Filtre FIR</i>	125
<i>Figure 5-32 : CDFG du filtre</i>	125
<i>Figure 5-33 : Spectres en fréquence du courant dans le filtre RIF avant et après la méthode "mise en forme du courant"</i>	126
<i>Figure 6-1 : Décomposition en 3 phases de l'activité d'un opérateur</i>	131
<i>Figure 6-2 : Détail de la décomposition</i>	132
<i>Figure 6-3 : Implémentation d'un multiplieur micropipeline</i>	132
<i>Figure 6-4 : Activité en courant dans un multiplieur asynchrone micropipeline</i>	134
<i>Figure 6-5 : Phases de l'activité en courant dans un multiplieur asynchrone micropipeline</i>	135
<i>Figure 6-6 : Liaison entre phases</i>	135

<i>Figure 6-7 : Le fonctionnement du circuit peut être représenté par un réseau de nœuds et de branches évoluant sur une dimension.</i>	136
<i>Figure 6-8 : Augmentation du nombre de degrés de liberté par l'utilisation de CDFG raffinés.</i>	137
<i>Figure 6-9 : Exemple de HSE.</i>	137
<i>Figure 6-10 : Exemple de HSE Temporel</i>	138
<i>Figure 6-11 : Exemple de HSE temporel pour 3 opérateurs</i>	139
<i>Figure 6-12 : Exemple de STG</i>	139
<i>Figure 6-13 : Exemple de CDFG</i>	140
<i>Figure 6-14 : Exemple de CDFG déroulé</i>	140
<i>Figure 6-15 : Modélisation structurelle</i>	141
<i>Figure 6-16 : Un contrôleur</i>	143
<i>Figure 6-17 : Délai intrinsèque en entrée</i>	143
<i>Figure 6-18 : Délai pour la faible émission électromagnétique</i>	144
<i>Figure 6-19 : Contrôleur WCHB</i>	145
<i>Figure 6-20 : 3 opérateurs WCHB en série</i>	146
<i>Figure 6-21 : STG des 3 opérateurs utilisant le protocole WCHB - Etat initial</i>	146
<i>Figure 6-22 : STG des 3 opérateurs utilisant le protocole WCHB - Phase 1</i>	147
<i>Figure 6-23 : STG des 3 opérateurs utilisant le protocole WCHB - Phase 2</i>	147
<i>Figure 6-24 : STG des 3 opérateurs utilisant le protocole WCHB - Phase 3</i>	148
<i>Figure 6-25 : CDFG des 3 opérateurs micropipeline utilisant le protocole WCHB</i>	148
<i>Figure 6-26 : Chemin critique pour l'activité en courant de l'opérateur B (protocole WCHB)</i>	149
<i>Figure 6-27 : Evènements dans le protocole WCHB</i>	150
<i>Figure 6-28 : Les dépendances dans le protocole WCHB</i>	150
<i>Figure 6-29 : Dépendances associées à l'étage B</i>	151
<i>Figure 6-30 : Contrôle de ε et β</i>	153
<i>Figure 6-31 : Contrôle de δ et γ</i>	153
<i>Figure 6-32 : Contrôle de α</i>	153
<i>Figure 6-33 : Exploitation de l'architecture du délai asymétrique</i>	154
<i>Figure 6-34 : Implémentations équivalentes</i>	154
<i>Figure 6-35 : Les délais des dépendances peuvent être insérés dans le contrôleur</i>	155
<i>Figure 6-36 : Factorisation des délais</i>	155
<i>Figure 6-37 : CDFG d'un RV en protocole WCHB</i>	156
<i>Figure 6-38 : Duplicateur utilisant le protocole WCHB</i>	157
<i>Figure 6-39 : multiplexeur</i>	158
<i>Figure 6-40 : démultiplexeur</i>	159
<i>Figure 6-41 : Contrôleur PCHB</i>	160

<i>Figure 6-42 : 3 contrôleurs PCHB en série</i>	<i>160</i>
<i>Figure 6-43 : STG des 3 opérateurs utilisant le protocole PCHB – Etat initial.....</i>	<i>161</i>
<i>Figure 6-44 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 1</i>	<i>161</i>
<i>Figure 6-45 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 2.....</i>	<i>162</i>
<i>Figure 6-46 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 3'.....</i>	<i>162</i>
<i>Figure 6-47 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 3".....</i>	<i>163</i>
<i>Figure 6-48 : CDFG des 3 opérateurs micropipeline utilisant le protocole PCHB.....</i>	<i>163</i>
<i>Figure 6-49 : Chemin critique pour l'activité en courant de l'opérateur B (protocole PCHB)</i>	<i>164</i>
<i>Figure 6-50 : CDFG déroulé des 3 contrôleurs A, B, C.....</i>	<i>165</i>
<i>Figure 6-51 : Dépendances dans le protocole PCHB.....</i>	<i>165</i>
<i>Figure 6-52 : Contrôle de β et ε.....</i>	<i>166</i>
<i>Figure 6-53 : Contrôle de γ et δ.....</i>	<i>167</i>
<i>Figure 6-54 : Contrôle de α.....</i>	<i>167</i>
<i>Figure 6-55 : Contrôle de ζ.....</i>	<i>168</i>
<i>Figure 6-56 : Implémentation équivalentes.....</i>	<i>168</i>
<i>Figure 6-57 : Contrôleur générique PCHB</i>	<i>169</i>
<i>Figure 6-58 : Duplicateur– PCHB.....</i>	<i>169</i>
<i>Figure 6-59 : Rendez-vous – PCHB.....</i>	<i>171</i>
<i>Figure 6-60 : Contrôleur PCFB.....</i>	<i>172</i>
<i>Figure 6-61 : 3 Contrôleurs PCFB en série.....</i>	<i>173</i>
<i>Figure 6-62 : STG des 3 opérateurs utilisant le protocole PCFB – Etat initial</i>	<i>173</i>
<i>Figure 6-63 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 1</i>	<i>173</i>
<i>Figure 6-64 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 2</i>	<i>174</i>
<i>Figure 6-65 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 3'.....</i>	<i>174</i>
<i>Figure 6-66 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 3".....</i>	<i>175</i>
<i>Figure 6-67 : CDFG des 3 opérateurs micropipeline utilisant le protocole PCHB.....</i>	<i>175</i>
<i>Figure 6-68 : Temps de cycle de l'opérateur B dans le protocole PCFB</i>	<i>176</i>
<i>Figure 6-69 : CDFG déroulé</i>	<i>177</i>
<i>Figure 6-70 : Les dépendances dans le protocole PCFB.....</i>	<i>178</i>
<i>Figure 6-71 : Contrôle de β et ε.....</i>	<i>179</i>
<i>Figure 6-72 : Contrôle de α.....</i>	<i>179</i>
<i>Figure 6-73 : Contrôle de γ et δ.....</i>	<i>180</i>
<i>Figure 6-74 : Contrôle de la liaison θ dans le contrôleur PCFB</i>	<i>180</i>
<i>Figure 6-75 : Contrôleur générique pour le protocole PCFB</i>	<i>181</i>
<i>Figure 6-76 : Duplicateur PCFB</i>	<i>182</i>
<i>Figure 6-77 : Rendez-vous - PCFB.....</i>	<i>183</i>
<i>Figure 6-78 : protocole WCHB.....</i>	<i>185</i>

<i>Figure 6-79 : WCHB cas où $dA=dB=dC$.....</i>	<i>185</i>
<i>Figure 6-80 : Simulation électrique de A,B,C - protocole WCHB.....</i>	<i>186</i>
<i>Figure 6-81 : WCHB cas où $dA=dC \neq dB$.....</i>	<i>186</i>
<i>Figure 6-82 : WCHB cas où $dA \neq dC \neq dB$.....</i>	<i>187</i>

Introduction

Le phénomène des interférences électromagnétiques s'est révélé dès les années 1930 lorsque certains systèmes électriques, par leur activité, perturbaient les débuts des communications radios. La notion de compatibilité électromagnétique est alors apparue en 1933 avec la création du Comité International Spécial des Perturbations Radioélectriques (CISPR). Depuis, chaque système électrique doit répondre à des normes afin qu'il puisse fonctionner sans perturber son environnement électromagnétique ni être lui-même affecté. L'utilisation des appareils électriques, les fréquences d'utilisation pour les communications s'intensifient et le phénomène des interférences électromagnétiques ne cesse de croître. Ainsi, les normes sont constamment actualisées car elles doivent suivre l'évolution de la technologie.

Par ailleurs, depuis 50 ans, l'effort d'intégration est tel que des milliers de transistors, sont sans cesse intégrés sur un même circuit où se côtoient plusieurs systèmes. Hier, simple composant d'un système électrique, le circuit intégré est aujourd'hui considéré comme un système lui-même et doit aussi répondre aux contraintes de compatibilité électromagnétique. Selon son domaine d'utilisation, les normes peuvent être très strictes. Par exemple, dans l'automobile, la sécurité dépend directement de la conformité des composants utilisés. D'un côté, avec les progrès des applications, les normes de compatibilité électromagnétique sont de plus en plus difficiles à satisfaire. De l'autre, le circuit intégré fonctionne à des fréquences croissantes et l'intégrité des signaux est conservée par des fronts de plus en plus raides ce qui amplifie le phénomène d'interférence électromagnétique.

L'utilisation des circuits sans horloge, connus pour leur faible rayonnement électromagnétique, apparaît alors comme une alternative. Ce faible rayonnement est dû essentiellement à la répartition de leur activité électrique car ils sont séquencés non pas par un signal global tel que l'horloge mais par des communications locales. Ces circuits ont été longtemps mis de côté au profit des circuits synchrones car la quantité d'efforts pour les concevoir était supérieure à celle nécessaire pour élaborer des circuits synchrones. Cependant des recherches sur la conception des circuits asynchrones ont été menées parallèlement au développement des circuits synchrones. Les méthodes de conception et les outils ont été élaborés. Des circuits ont été développés et même commercialisés.

Le groupe Concurrent Integrated Systems (CIS) du laboratoire TIMA fait partie des acteurs qui développent une suite d'outils dédiée à la conception des circuits asynchrones.

Cette thèse s'intègre dans ce travail en proposant des méthodes de conception des circuits asynchrones pour la faible émission électromagnétique. Le projet a été mené selon 2 axes. Le premier suggère de partir de la description haut niveau de circuits synchrones et fournir des circuits asynchrones équivalents compatibles cycle à cycle et de rayonnement électromagnétique moins élevé. Le deuxième axe propose de diminuer les émissions électromagnétiques des circuits asynchrones eux-mêmes en répartissant leur activité électrique dans le temps.

Le chapitre 1 de ce mémoire présente les interférences électromagnétiques (IEM) en général : leurs origines ainsi que leur classement. Il aborde ensuite la notion de compatibilité électromagnétique (CEM). Enfin il se focalise sur le cas des circuits intégrés en décrivant les facteurs qui entrent dans le rayonnement du circuit, les perturbations engendrées, les normes actuelles en terme de modélisation et simulation. La lecture de ce chapitre apporte au lecteur des éléments de compréhension en ce qui concerne le choix de la répartition de l'activité électrique du circuit pour la diminution de l'émission électromagnétique.

Ce choix est d'autant plus explicite lorsque le lecteur aborde le chapitre 2. Celui-ci détaille les méthodes de répartition de l'activité électrique qui existent déjà pour les circuits synchrones. L'étude de ces méthodes et des différents travaux qui leur sont dédiés permet aussi d'apporter des éléments de compréhension sur l'activité du signal d'horloge, son influence sur l'émission électromagnétique du circuit. Ce chapitre permet de comprendre aussi pourquoi il est préférable d'utiliser des circuits asynchrones plutôt que des circuits synchrones.

C'est dans le chapitre 3 que le lecteur trouvera plus de détails sur les circuits asynchrones : les concepts de base, les catégories existantes (classées selon l'implémentation du circuit) et les hypothèses faites. Il présente par ailleurs les performances de ces circuits et l'état actuel des méthodes et outils de conception.

Le fonctionnement des circuits asynchrones est repris dans le chapitre 4 pour une comparaison avec le mode de fonctionnement synchrone. Les circuits synchrones sont étudiés alors dès leur description au niveau registre (*Register Transfer Level*) dans laquelle des structures sont identifiées afin d'être remplacées par des structures asynchrones équivalentes de même niveau de description. La méthode est telle que le nouveau circuit obtenu soit compatible cycle à cycle avec l'environnement du circuit d'origine. Ainsi la validation fonctionnelle du circuit obtenu est rapide et les comparaisons sont pertinentes.

La description du circuit dans un langage standard tel que VHDL permet ensuite d'utiliser les outils d'aide à la conception existants.

Le chapitre 5 se focalise sur le circuit asynchrone lui-même et propose une méthode de conception pour réduire son émission électromagnétique. Cette méthode consiste à répartir l'activité du circuit par ordonnancement des actions dans le circuit. L'architecture du circuit et son activité en courant sont modélisées. Une discrétisation en temps et en amplitude de l'activité du courant permet d'adapter l'ordonnancement au circuit asynchrone. La répartition est ensuite effectuée en adaptant un algorithme déjà existant.

Le niveau de granularité est augmenté dans le chapitre 6 qui traite des protocoles de communications WCHB (*Weak Condition Half Buffer*), PCHB (*Precharge logic Half Buffer*), PCFB (*Precharge logic Full Buffer*) utilisés dans les circuits asynchrones. L'étude des concurrences des actions dans ces protocoles est menée en vue d'apporter des paramètres supplémentaires pour la méthode présentée dans le chapitre 5. L'objectif de cette étude est de quantifier l'influence du protocole sur les profils de courant pour pouvoir choisir un protocole avec une contrainte d'émission électromagnétique.

Enfin, les différents points abordés sont résumés dans la conclusion qui présente aussi les perspectives de ce travail de thèse.

CHAPITRE 1 Interférences électromagnétiques.

1.1 Introduction

Que ce soit dans le domaine domestique, industriel, militaire, des communications ou des transports, chaque système électrique peut provoquer sur son environnement des perturbations de nature électromagnétique. Ainsi, la compatibilité électromagnétique consiste à garantir le fonctionnement correct des équipements électriques, électroniques et radioélectriques qui coexistent [CHAU92]. Avec les progrès des technologies, la multiplication des moyens de transports et de communications, l'accès à de nombreux produits de nature électrique ou électronique au grand public, la compatibilité électromagnétique est essentielle pour la sécurité des biens et des personnes [PERR96]. Chaque constructeur ou équipementier doit vérifier la conformité de ses appareils.

Avec une complexité comparable à la plupart des équipements, les circuits intégrés sont aussi concernés par la compatibilité électromagnétique [CATA01]. Briques de base de la majeure partie des systèmes, ils ne sont plus considérés comme de simples composants, mais comme des systèmes à part entière qui peuvent aussi être affectés par leur propre fonctionnement.

Dans un premier temps, ce chapitre présente les généralités à connaître sur les interférences électromagnétiques (IEM) et introduit la notion de compatibilité électromagnétique (CEM) (Section 1.2). Il se concentre ensuite sur le cas des circuits intégrés (Section 1.3).

1.2 Généralités

Ce paragraphe présente les interférences électromagnétique : leurs origines ainsi que leur conséquences. Les généralités présentées ici peuvent être retrouvées par le lecteur dans [CHAR92] et [MARD03].

1.2.1 Champ électromagnétique

1.2.1a Champ magnétique

Lorsqu'un courant i parcourt un conducteur, il engendre un champ magnétique \vec{H} (Figure 1-1). Ce champ s'exprime en ampère par mètre (A/m). L'effet du champ magnétique est d'induire une tension sur les boucles qui sont perpendiculaires à ses lignes. Cette tension induite est proportionnelle à la surface de la boucle.

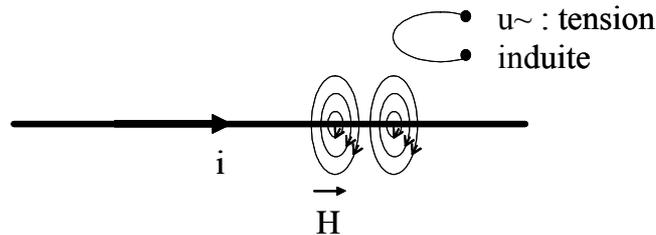


Figure 1-1 : Effets du champ magnétique

1.2.1b Champ électrique

Si une tension u est présente sur un conducteur, elle engendre un champ électrique \vec{E} dont l'intensité est exprimée en Volts par mètre (V/m). L'effet du champ électrique est d'induire un courant sur les fils qui sont parallèles à ses lignes (Figure 1-2).

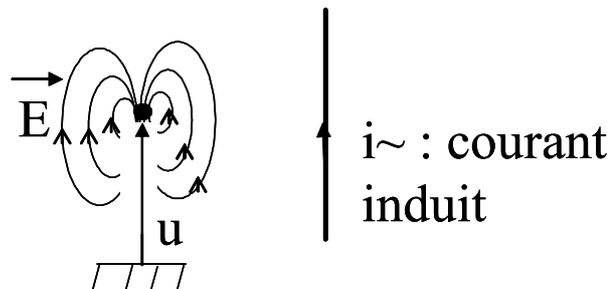


Figure 1-2 : Effet du champ électrique

1.2.1c Champ électromagnétique

Lorsqu'il y a une évolution de tension et de courant dans le temps, les champs électrique et magnétique sont présents et forment le champ électromagnétique (Figure 1-3).

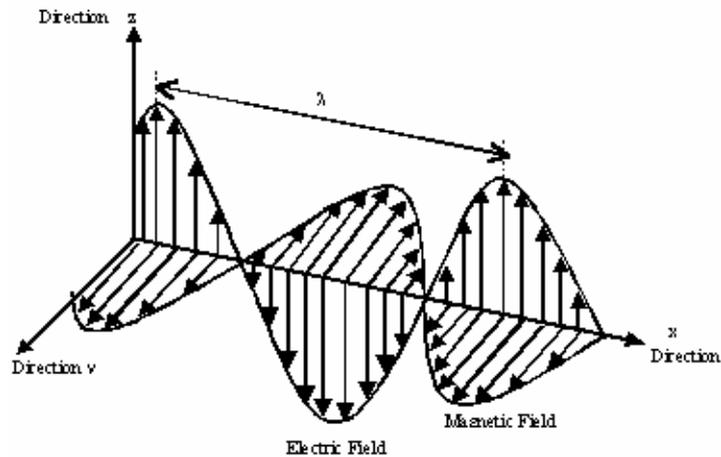


Figure 1-3 : Champ électromagnétique [OBER01]

Les effets respectifs des deux champs sont combinés. L'ampleur des effets de chaque champ dépend des valeurs de leurs composantes. Le conducteur se comporte alors comme une antenne en émettant des ondes électromagnétiques. Le champ électromagnétique s'exprime en Watt par mètre (W/m) ou Tesla.

La longueur de l'onde électromagnétique λ émise par le champ électromagnétique est exprimée en mètres. Elle est définie comme le trajet d'une onde après une période d'oscillation T :

$$\lambda = c \times T = \frac{c}{f \sqrt{\epsilon_r}}$$

Équation 1-1

où c est la célérité de la lumière ($c = 300\text{m/s}$) et f la fréquence d'oscillation de l'onde. ϵ_r est la permittivité diélectrique relative du milieu, elle est égale à $\epsilon_r = \frac{\epsilon}{\epsilon_0}$. ϵ_0 étant la permittivité du vide : $\epsilon_0 = \frac{1}{36\pi 10^9} = 8,85\text{pF/m}$ et ϵ étant la permittivité absolue/

milieu	ϵ_r
air	1
SiO ₂	4
silicium	11.8
Epoxy	4.8

Tableau 1-1 : Permittivités diélectriques relatives de différents milieux

L'effet du champ électromagnétique sur un récepteur dépend de la longueur d'onde émise par le champ électromagnétique et de la distance à la source. Cet effet peut être

caractérisé par l'impédance du champ, définie par le rapport de l'amplitude de son champ électrique à celle de son champ magnétique [CHAR92],[GARE00]:

$$Z_c = \frac{|E|}{|H|}$$

Équation 1-2

Elle est exprimée en Ohms (Ω) (Figure 1-4) et sa valeur limite minimum correspond à

$$Z_{\min} = \mu_0 \omega D$$

Équation 1-3

où μ_0 est la perméabilité magnétique de l'air = $4 \Pi \cdot 10^{-7}$ H/m, et ω , la pulsation = $2\Pi f$ rad/s, et D étant la distance de l'antenne en mètres.

Sa limite maximale est

$$Z_{\max} = \frac{1}{\epsilon_0 \omega D}$$

Équation 1-4

et $Z_{\min} = Z_{\max}$ quand

$$D = \frac{48}{\text{FMhz}} = \frac{\lambda}{2\Pi}$$

Équation 1-5

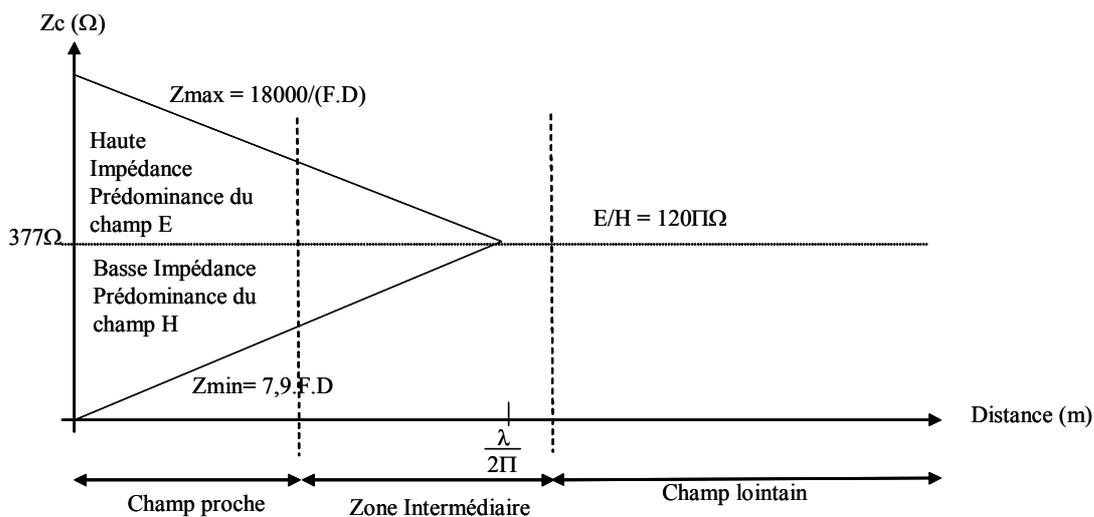


Figure 1-4 : Champ rayonné

A une distance D très inférieure à $\frac{\lambda}{2\pi}$ de l'antenne (à un sixième de la longueur d'onde), le champ est dit proche.

Si $Z_c < 377\Omega$ (basse impédance) : le champ magnétique est supérieur au champ électrique.

Si $Z_c > 377\Omega$, (haute impédance) : le champ électrique est supérieur au champ magnétique.

A une distance $D > \frac{\lambda}{2\pi}$ de l'antenne, le champ est dit lointain (espace libre).

L'impédance du champ électromagnétique est égale à $Z_c = 377\Omega$, elle est donc égale à l'impédance intrinsèque de l'air = $\sqrt{\frac{\mu_0}{\epsilon_0}} = 120\pi\Omega$. Les champs sont couplés.

1.2.2 Interférences électromagnétiques

Il a été vu précédemment que le champ électromagnétique peut induire des tensions et des courants parasites sur des boucles et des conducteurs électriques. Il peut par ailleurs interférer avec d'autres champs électromagnétiques émis intentionnellement pour communiquer (radios, réseau sans fil...). Ce paragraphe décrit les origines et le classement des interférences électromagnétiques. Il est basé sur [TERR97].

1.2.2a Définition

Un signal électrique est l'évolution d'un courant et d'une tension au cours du temps: il est donc associé à une onde électromagnétique. Les équipements électriques et électroniques ne sont pas des systèmes thermodynamiquement clos. (Le mot système est ici pris dans son sens général : un assemblage de systèmes, formé d'un ou plusieurs équipements qui peuvent être une carte PCB (*Printed Circuit Board*) ou un circuit intégré). Comme le montre la Figure 1-5, de l'énergie électromagnétique peut s'échapper des limites de ce système.

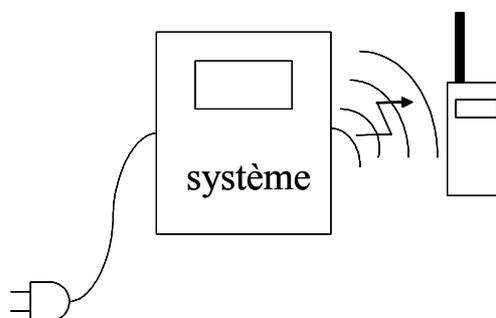


Figure 1-5 : Tout système électrique peut se comporter comme une source de bruit

Par son activité électrique, un circuit peut donc se comporter comme une source de bruit : les ondes électromagnétiques qu'il génère peuvent interférer, se coupler, avec des ondes radios de même fréquence et/ou perturber des circuits récepteurs. Il se comporte alors comme un émetteur, une source de bruit non intentionnelle et participe à la pollution électromagnétique (EM) de son environnement. Ce phénomène est appelé interférence électromagnétique.

Le système à l'origine de la perturbation est appelé "source" alors que le système récepteur qui subit la perturbation est appelé "victime" (Figure 1-6).

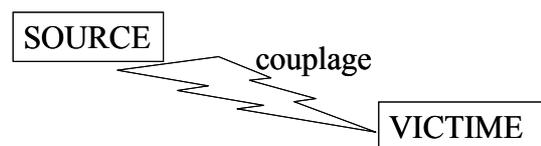


Figure 1-6 : Le phénomène des interférences électromagnétiques

Les conséquences de ce phénomène peuvent aller du dysfonctionnement de la victime jusqu'à sa destruction.

Par ailleurs, tout système opère dans un environnement électromagnétique qui peut produire de l'énergie électromagnétique parasite (Figure 1-7). Le système peut ainsi être une victime.

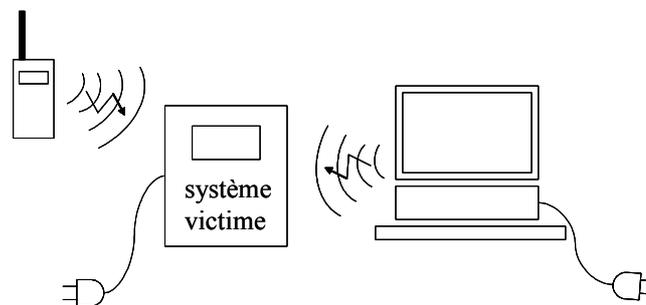


Figure 1-7 : Tout système peut être perturbé par son environnement

Les interférences électromagnétiques dépendent de 2 facteurs : l'émission (1.2.2b) - c'est à dire l'aptitude des systèmes électriques à générer des ondes électromagnétiques - et la susceptibilité (1.2.2c) - l'aptitude des systèmes électriques à réagir aux champs électromagnétiques extérieurs. La notion d'immunité d'un système est aussi employée et n'est autre que l'aptitude du système à résister aux perturbations électromagnétiques extérieures. Les courants et tension parasites qui apparaissent dans les boucles et fils du système sont considérés comme un bruit qui peut être classé selon deux modes : le bruit de mode commun et bruit de mode différentiel (1.2.2d).

1.2.2b Emission

Le bruit électromagnétique généré par un système peut être classé selon son mode de propagation : conduit par un conducteur électrique - émission conduite - ou à travers l'espace - émission rayonnée.

Les émissions rayonnées

Les ondes électromagnétiques générées par le circuit se propagent dans l'air pour interférer avec une victime (Figure 1-8).

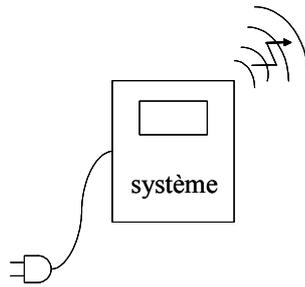


Figure 1-8 : L'émission rayonnée

<i>Fréquence</i>	<i>Utilisation</i>
100 Mhz	Bande FM
400 Mhz	Radio (Talkie Walkie)
900 Mhz	GSM (Global System Mobile)
1,4 GHz	Fréquence d'écoute du ciel par les radioastronomes
1800 Mhz	DCS (Digital Communication System)
1900 Mhz	PCS (Personnal Communication System)
2,45 ; 5,5GHz	Réseau sans fil

Figure 1-9 : Principales fréquences utilisées

Ces ondes peuvent être captées par des circuits récepteurs sur de larges bandes de fréquences allant des fréquences de quelques kHz aux fréquences de plusieurs GHz.

Pour ne pas perturber l'environnement du circuit, les concepteurs de circuits intégrés doivent prêter une attention particulière à la bande de fréquence allant de 30Mhz à 10Ghz (Figure 1-9).

Les émissions conduites

Les ondes électromagnétiques utilisent les câbles et les interconnexions pour se propager. Les émissions conduites mènent aussi au dysfonctionnement de l'environnement ou du système lui-même.

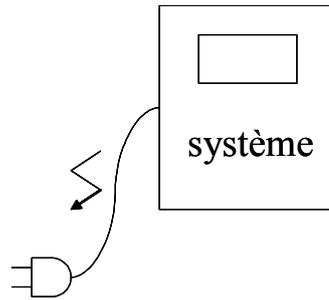


Figure 1-10 : L'émission conduite

1.2.2c Susceptibilité

Interférence RF (RFI)

Beaucoup de circuits doivent opérer dans un environnement où ils sont sujets à des champs électromagnétiques produits par d'autres éléments (Figure 1-11). Ces champs peuvent, lorsqu'ils créent des courants ou tension parasites, causer des erreurs mais aussi détruire le composant. Dans ce cas, le concepteur se doit de protéger le circuit contre les champs extérieurs. En général cette protection s'effectue par blindage ou filtrage des fréquences concernées.

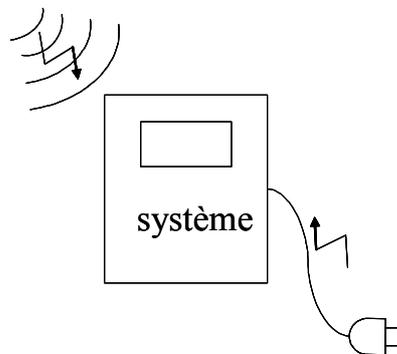


Figure 1-11 : Tout système électrique peut être sujet à des champs électromagnétiques

Décharge électrostatique (ESD)

Dans l'environnement du circuit, des potentiels électrostatiques peuvent s'accumuler. Les décharges électrostatiques qui en résultent peuvent conduire à un fonctionnement erroné du circuit ou, si la décharge a une énergie suffisante, le détruire. Les décharges électrostatiques peuvent endommager le circuit :

- par une variation brusque de la tension (plusieurs kiloVolts)
- par une variation brusque du courant (plusieurs Ampères)

-
- par des courants ou des tensions induits

La foudre est une décharge électrostatique à grande échelle. Une autre source d'interférences EM sont les impulsions électromagnétiques (EMP) provoquées par les explosions nucléaires.

1.2.2d Bruit de mode commun et de mode différentiel

Les interférences électromagnétiques peuvent être aussi classées selon la façon dont elles s'introduisent dans le chemin du signal. Le bruit généré est alors modélisé sous la forme d'un générateur de courant en série ou en parallèle selon qu'il est de mode différentiel ou commun.

Bruit de mode différentiel

Il peut être représenté par un générateur en série avec le générateur de tension désiré.

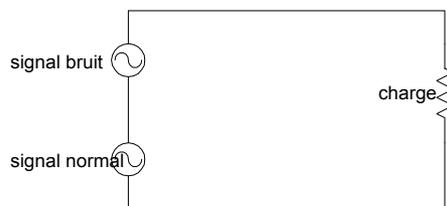


Figure 1-12 : Modélisation du bruit de mode différentiel

Bruit de mode commun

Il apparaît comme un générateur situé entre le potentiel de référence de la source et le récepteur.

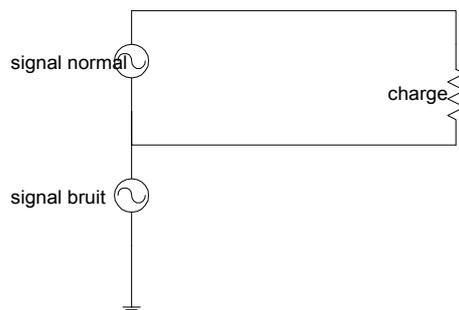


Figure 1-13 : Modélisation du bruit de mode commun

1.2.3 Compatibilité Electromagnétique.

1.2.3a Motivations

Dès les années 30, le problème des interférences électromagnétiques est apparu avec l'utilisation des communications radios. Durant la seconde guerre mondiale, quand l'utilisation d'appareils électroniques s'est accélérée, ce problème s'est manifesté entre les radios et les systèmes de navigation. Depuis, il n'a cessé de s'amplifier avec : l'apparition des composants électroniques dans les années 50, le circuit intégré dans les années 60 et les circuits à haute densité en 1970. Par ailleurs, le spectre fréquentiel utilisé est devenu de plus en plus large allant de 33Mhz (liaison PCI- *Peripheral Component Interconnect*) à quelque Ghz (réseau sans fil) pour subvenir aux besoins croissants de la transmission d'information.

1.2.3b Définition de la compatibilité électromagnétique

Du fait qu'il émet des ondes électromagnétiques, un circuit est un pollueur potentiel. Dépendant de sa susceptibilité aux perturbations électromagnétiques, il est aussi une victime éventuelle (Figure 1-14) : il peut voir ses performances diminuées et/ou avoir un dysfonctionnement.

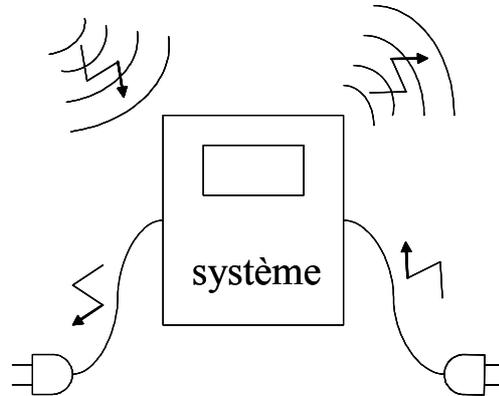


Figure 1-14 : Un circuit est un pollueur et une victime potentielle

Un circuit est compatible avec son environnement si :

- Il ne constitue pas une source de bruit électromagnétique pour les autres circuits. C'est à dire l'énergie électromagnétique qu'il émet n'a pas de conséquences non désirées sur son environnement.
- Il est capable de fonctionner dans un environnement électromagnétique spécifique.

Des normes de compatibilité électromagnétique (CEM) ont été définies pour les constructeurs de circuits, d'appareils de mesure, de transports etc. afin de réguler le bon fonctionnement des appareils [WILL99].

1.2.3c Normes CEM

Le Comité International Spécial des Perturbations Electromagnétiques (CISPR : *International Special Committee on Radio interference*) a été créé en 1933 par la commission électrotechnique internationale (IEC). Celle-ci développe des normes pour éviter les interférences électromagnétiques. En 1979, l'*American Federal Communication Commission* (FCC) a publié des normes limitant les émissions électromagnétiques de tous les appareils électroniques.

Depuis le 1er janvier 1996, tout produit commercialisé sur le marché européen doit satisfaire aux exigences des normes CEM d'émission et d'immunité. Actuellement, le CISPR recommande des valeurs limites d'émission et des techniques de mesures que plusieurs pays ont adoptées.

1.3 CEM des circuits intégrés

Composé de millions de transistors et de plusieurs niveaux de métaux, le circuit intégré est aujourd'hui considéré comme un système à part entière et est il doit aussi répondre à des normes CEM [CATA01]. Ce paragraphe présente les origines (expliquées plus en détail dans [ARAG02]) et les conséquences des interférences électromagnétiques dans le circuit intégré. La fin de ce paragraphe aborde la standardisation des modèles d'émissions des circuits intégrés.

1.3.1 Interférences électromagnétiques

1.3.1a Principaux acteurs des interférences électromagnétiques

Le transistor

La Figure 1-15a et la Figure 1-15b tirées de [BEND03] montrent l'activité du courant dans un transistor CMOS. Lorsque l'entrée passe de 0 à 1 (Figure 1-15a), le courant traverse l'étage pull up et charge la capacité. Lorsque l'entrée passe de 1 à 0 (Figure 1-15b), le courant traverse l'étage pull down et décharge la capacité. Ainsi, lorsqu'il commute, le transistor élémentaire induit une variation de courant di/dt ce qui génère un champ magnétique.

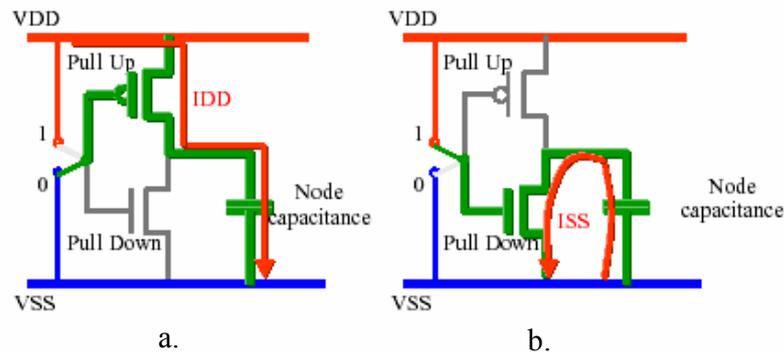


Figure 1-15 : Appel de courant dans une structure CMOS [BEND03]

L'horloge

Dans un circuit synchrone, les éléments séquentiels commutent à la cadence du signal d'horloge. Ainsi les appels de courant sur l'alimentation à chaque front montant/descendant de l'horloge sont élevés. Plus la fréquence d'horloge est grande, plus le nombre de transistors qui commutent simultanément est élevé, plus les variations du courant sont brusques et plus le couplage avec les victimes est efficace. L'horloge est la principale source d'émission rayonnée dans un circuit intégré.

Le plot d'alimentation

Le bruit des commutations simultanées (appelé SSN de l'anglais *Simultaneous Switching Noise*, aussi connu sous le nom de "ground bounce") est une tension induite sur les connexions d'alimentations (VSS, VDD) et qui est due aux courants de commutation qui passent par les inductances parasites du boîtier ou des pistes. Le bruit global généré est source d'énergie RF. Cette énergie potentielle peut être conduite à l'extérieur du circuit par le biais des plots d'alimentation. Ainsi les plots sont des vecteurs d'émission conduite dans les circuits intégrés.

Les rails d'alimentation

Ils sont constamment sollicités par les appels de courant des transistors dans le circuit. Ils sont donc vecteurs d'énergie électromagnétique aussi bien conduite que rayonnée.

Le boîtier

Le rôle du boîtier soit de relier le circuit intégré au circuit imprimé, tout en le protégeant des agressions extérieures, cependant il peut aussi influencer sur l'émission électromagnétique du circuit [RICH02]. La Figure 1-16 montre le boîtier d'un circuit

intégré, des phénomènes de couplage peuvent survenir au niveau des broches et des "bonding". L'émission du circuit intégré peut être transmise facilement à l'extérieur du boîtier selon les matériaux et la géométrie utilisée.

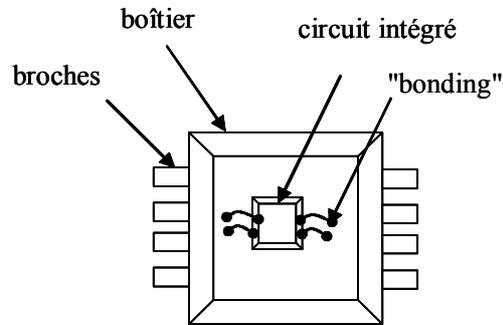


Figure 1-16 : Le boîtier

L'interconnexion

La Figure 1-17, représente deux lignes dans un circuit intégré. Lorsque ces interconnexions sont proches, la variation du courant sur une ligne induit un champ électromagnétique qui peut influencer sur la ligne voisine. Ainsi un changement de valeur d'un signal sur une ligne peut survenir et être transmis dans le circuit. Ce phénomène est appelé diaphonie [SICA99][CHIL00].

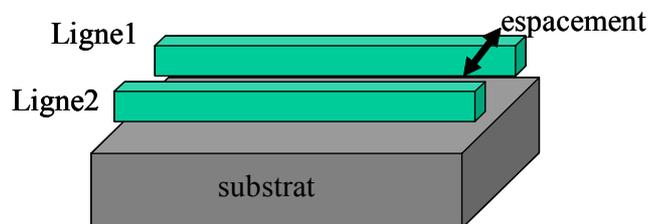


Figure 1-17 : Interconnexions

Le substrat [CHAR99]

Le substrat peut aussi être une source d'émission conduite car le circuit est relié au substrat par des capacités de jonctions des transistors. Les signaux provenant de la commutation d'un nœud peuvent provoquer des variations de la tension du substrat. Celles-ci induisent des pics de courants qui se propagent du nœud en question jusqu'aux contacts qui l'entourent. Ces impulsions de courant peuvent interférer avec des transistors et générer de larges pics de courant. Le phénomène de diaphonie présenté précédemment peut être aussi amplifié.

Le plan de masse

Le plan de masse est une source d'émission rayonnée par mode commun. Dans ce mode, plusieurs circuits ont le même plan de masse et ils peuvent y créer des chutes de tensions. Ces chutes de tensions peuvent induire des courants parasites dans les lignes et des couplages avec les autres circuits reliés au plan de masse (boucles de masse). Le plan de masse se comporte alors comme une antenne pour les interconnexions extérieures. Le bruit dû à la commutation simultanée provoque le rayonnement du plan de masse "*ground bouncing*".

Autres antennes accidentelles

Les boucles fermées à l'intérieur du circuit se comportent comme des antennes qui peuvent émettre mais aussi recevoir et retransmettre. Ainsi, elles peuvent se coupler avec d'autres circuits en induisant des courants et tensions parasites dans ceux-ci. Elles peuvent aussi réagir en induisant des tensions et courant parasites sous l'effet d'ondes électromagnétiques produites par d'autres circuits.

1.3.1b Conséquences du phénomène d'interférence électromagnétique

Dans un circuit intégré, le courant circulant dans les interconnexions et le basculement de tension des portes logiques sont des sources de bruit électromagnétique. Ce phénomène est d'autant plus élevé que le courant circule dans tout le circuit et les parties opératoires traitent plusieurs données en même temps. Le courant dynamique qui résulte de la commutation des transistors est fonction de la fréquence d'horloge, du nombre de transistors, de leurs tailles et de leur tension d'alimentation.

Lorsqu'un circuit est sujet aux interférences électromagnétiques, les conséquences sont les suivantes :

- Un signal peut être non interprété
- Le passage à 0 ou à 1 d'un signal non volontaire peut arriver
- Des délais sont induits dans les portes
- Des chutes de tension sont provoquées dans les transistors
- Un dysfonctionnement des blocs logiques peut apparaître
- Le niveau du signal peut baisser
- Le composant peut être même détruit

Certains facteurs sont aggravants comme :

- Les boucles fermées et le niveau de plus en plus élevé d'interconnexions favorisent les antennes accidentelles.
- Le nombre élevé de transistors et l'augmentation de la fréquence d'horloge favorisent le nombre de commutations simultanées dans le composant.

1.3.1c Circuits numériques et analogiques

En général, la susceptibilité des circuits analogiques est plus grande que celle des circuits numériques. Ceci est dû au faible niveau de leurs signaux qui est de l'ordre du milli voir du micro-volt. Tandis que le niveau des signaux des circuits numériques est bien plus élevé (de l'ordre du volt). Donc quand un circuit contient des parties numériques et analogiques, les variations de courant produites par les parties numériques peuvent provoquer un mauvais fonctionnement des parties analogiques. Ainsi, le phénomène des interférences électromagnétiques est un véritable obstacle au bon fonctionnement des systèmes contenant à la fois des parties numériques et des parties analogiques comme les systèmes sur puce (*SoC-System on Chip*) et les systèmes embarqués.

1.3.1d Réduction des interférences électromagnétiques

D'un point de vue général, il y a 3 façons de réduire les émissions électromagnétiques :

- supprimer l'émission à la source
- rendre le couplage le plus inefficace possible
- rendre le récepteur le moins susceptible aux émissions

Le choix de la ou les solutions dépend de la faisabilité et du coût.

Actuellement, les solutions utilisées se concentrent sur la carte d'application (PCB) qui doit contenir les circuits intégrés :

- Une réduction à la source peut consister à mettre des filtres sur la carte d'application.

- Les variations rapides du signal (montée et descente) di/dt sont les principales causes du contenu haute fréquence du spectre. Plus le spectre du signal s'étend vers les hautes fréquences, et plus le couplage devient efficace. En général, l'addition de capacité de réservoir de forte valeur placée près du composant permet de réduire le changement brusque du courant.

- Aussi, isoler le circuit sur un plan de masse permet de ramener les interférences à la masse. Des capacités peuvent être également additionnées en séries avec une inductance entre la source VCC et la pin d'entrée pour réduire d'avantage les interférences. Cette technique s'applique aussi lors du placement routage du circuit lui-même, ce qui permet de réduire l'addition de composants sur le PCB.

- En ce qui concerne l'horloge, l'addition d'un circuit pour l'étalement de spectre de l'horloge (*Spread Spectrum Clock Generator*)[SHA00][SHA01][SADA02] permet de réduire les émissions (voir chapitre 2).

Certaines solutions agissent au niveau du boîtier du circuit, des pattes et des plots d'alimentation. Une solution consiste à utiliser les circuits asynchrones, qui avec leur

qualité intrinsèque de faible émission électromagnétique, sont une alternative intéressante (Chapitre 3).

Plus le critère de compatibilité électromagnétique est considéré tôt dans la conception du circuit et plus les coûts sont minimisés. Inspirées du *Spread Spectrum Clock* [HARD94], la plupart des méthodes existantes, qui s'appliquent dès la conception, se concentrent sur l'horloge et sa distribution. Le chapitre 2 décrit certaines méthodes qui ont inspiré la seconde partie du travail de thèse.

1.3.2 Normes IEC

1.3.2a Auto-compatibilité électromagnétique

L'avancée des procédés technologiques a permis la réduction de la taille des transistors, le développement des circuits de plus en plus complexes et l'intégration sur une même puce. Les niveaux des interconnexions ont été alors augmentés et les tailles des circuits ont diminué. Le phénomène des émissions électromagnétiques est alors favorisé et croît avec l'augmentation des fréquences de fonctionnement.

Les tensions d'alimentations sont de plus en plus faibles et les signaux commutent de plus en plus rapidement dans les interconnexions. En 0,18 microns, les temps de commutation sont de l'ordre de 100ps pour une tension V_{dd} de 2V. Les amplitudes de courant restent constantes (0,5mA en 100ps) mais les variations sur les rails d'alimentation sont de plus en plus brusques. Un plus grand di/dt est donc observé ainsi que l'augmentation de l'émission électromagnétique du circuit. La tendance pour les circuits va dans l'augmentation de l'énergie contenue dans les pics dues à haute vitesse de commutation de l'horloge, l'augmentation des broches du boîtier, d'une plus grande surface de silicium occupée par le circuit et de sa densité et le boucles d'interconnexion.

La compatibilité du circuit avec son environnement est améliorée sans cesse au niveau des interfaces (carte PCB). Cependant, les mécanismes de couplages internes, inductifs ou capacitifs, la sensibilité à la géométrie des pistes, les caractéristiques des matériaux n'ont pas été encore considérés. A présent, c'est de sa propre activité électrique qu'il faut protéger le circuit intégré.

Des normes CEM ont été élaborées et sont toujours en cours d'élaboration pour modéliser [IEC62] et mesurer l'émission d'un circuit [IEC61], afin qu'il réponde aux contraintes actuelles imposées par les constructeurs automobile, les opérateurs des télécommunications etc.

1.3.2b Standards de modélisation

Bien qu'il existe des normes pour les mesures des émissions et de la susceptibilité des circuits intégrés, ce paragraphe se concentre ici sur les standards de modélisation [CANA00][ROSS02][IBIS][IEC62]. Ces standards permettent aux concepteurs et aux fournisseurs d'outils de simulation d'utiliser les mêmes modèles. Pour une application donnée, la simulation d'une carte dédiée permet de circonscrire le temps de cycle de conception en évitant la fabrication de prototypes non viables.

Les origines du bruit électromagnétique dans un composant intégré, sont l'activité de son cœur et les commutations au niveau de ses interfaces (plots) d'entrées/sorties. Le standard IBIS (*I/O Buffer Information Specification*) [IBIS][MAIO99][HUQ00] a été introduit dans le but d'harmoniser les modèles des émissions électromagnétiques d'un circuit intégré. Ce standard utilise les caractéristiques courant/tension du circuit intégré et contient les informations nécessaires relatives au comportement analogique des interfaces I/O. Prédire le comportement des interfaces n'est pas suffisant car la principale source de bruit reste l'activité du courant de son cœur. Ainsi, le standard ICEM (*Integrated Circuit Electromagnetic compatibility Model*) [IEC62] en cours d'élaboration propose une modélisation du cœur du circuit intégré. IBIS et ICEM sont des modélisations comportementales : elles respectent donc la propriété intellectuelle des constructeurs et peuvent être utilisées par les outils EDA (*Electronic Design Automation*) existants. Ces derniers permettent, au niveau PCB, de prédire le comportement électromagnétique des composants intégrés.

1.3.2c IBIS (*I/O Buffer Information Specification*)[MAIO99]

Cette modélisation, qui est aujourd'hui largement utilisée, concerne les broches des entrées/sortie du circuit intégré.

Pour un *buffer* de sortie, le modèle IBIS contient les informations suivantes :

- Caractéristiques Courant/Tension des *buffers* de sortie lorsque celle ci est à l'état bas.
- Caractéristiques Courant/Tension des *buffers* de sortie lorsque celle ci est à l'état haut.
- Caractéristiques Courant/Tension des *buffers* de sortie lorsque celle ci est forcée à la masse par le rail d'alimentation.
- Le temps de commutation du *buffer*
- La capacité.

Pour un *buffer* d'entrée (**Erreur ! Source du renvoi introuvable.**) le modèle contient ses caractéristiques courant/tension.

1.3.2d ICEM (*Integrated Circuit Electromagnetic compatibility Model*)

Les paramètres ICEM sont une extension du modèle IBIS qui modélisent l'émission conduite et rayonnée de l'activité du cœur du composant intégré. Le modèle comportemental réunit les 3 mécanismes de l'émission suivants :

- L'émission conduite à travers les lignes d'alimentation qui est due aux commutations dans le cœur et les *buffers*. Ceci induit aussi des appels de courants sur les pattes du circuit.
- Le couplage par les plots d'entrée/sortie : les pics de courants provoqués par l'activité du cœur peuvent être mesurés au niveau des I/O bien qu'ils ne soient pas activés. Les lignes PCB connectées peuvent alors se comporter comme des antennes accidentelles.
- Le couplage rayonné : le flux du courant et la fluctuation de la tension de masse qui en résulte sont les causes de l'émission rayonnée.

La Figure 1-18 montre le modèle ICEM d'un circuit intégré dont le cœur et les entrées sorties ont des alimentations différentes.

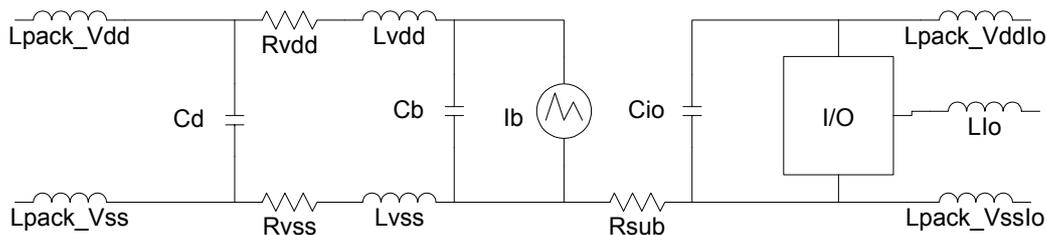


Figure 1-18 : Modèle ICEM

Le modèle caractérise simplement l'émission électromagnétique du circuit.

Il réunit :

- Les caractéristiques du boîtier : $L_{packVdd}$, $L_{packVss}$, $R_{packVdd}$, $R_{packVss}$.
- La capacité de découplage C_d entre les plots V_{dd} et V_{ss} du boîtier C_d .
- La capacité de block C_b qui est grossièrement la somme des capacités de toutes les portes.
- R_{sub} qui représente la liaison entre le réseau des I/O et le substrat et C_{io} la capacité de découplage entre ces deux entités.
- Les mesures et les valeurs par défauts sont utilisées comme paramètres des modèles ICEM.
- I_b représente l'activité électrique du cœur du circuit.

1.3.3 Le générateur I_b dans le modèle ICEM : l'activité du coeur du circuit

Le générateur de courant I_b représente l'activité électrique du coeur du circuit. I_b est au centre du modèle ICEM car c'est la principale source d'émission électromagnétique. Les pics de courants, venant de l'intérieur du cœur, traversent les lignes d'alimentation et se retrouvent au niveau des plots. La forme du courant peut être une description en temps du courant ou bien la forme triangulaire qui lui est équivalente.

L'allure du courant peut être aussi statistiquement évaluée en multipliant l'activité de commutation par un pic de courant typique d'une porte [BEND03] de la manière suivante : La valeur par défaut du courant I peut se calculer en connaissant la densité de cellules par mm^2 , on estime à 10% le nombre de portes qui commutent en même temps. L'appel de courant global à chaque front d'horloge est égal à la somme des pics de courants produits par les commutations simultanées des portes dans le circuit $I_b(t) = \text{nombre de portes totales} * 10\% * \text{le pic de courant maximum d'une porte}$.

Le Tableau 1-2 donne des valeurs typiques pour différentes technologies

Technologie	Année	Alimentation (V)	Densité de cellules (/mm ²)	Fréquence de l'horloge (Mhz)	Courant Crête (mA/porte)
CMOS 0,35 μm	1995	5-3,3	13KGates	16-300	0,6
CMOS 0,25 μm	1997	5-2,5	40KGates	40-450	0,4
CMOS 0,18 μm	1999	3,3-2	130KGates	100-900	0,3
CMOS 0,12 μm	2001	2,5-1,2	200KGates	150-1200	0,2

Tableau 1-2 : Valeurs typiques

La réduction de l'émission électromagnétique du circuit intégré passe principalement par la réduction des pics de l'activité électrique de son coeur.

1.4 Conclusion

Les circuits intégrés produisent des interférences électromagnétiques par leur activité en courant. Les circuits numériques ou digitaux demandent des signaux de niveaux élevés et sont plus susceptibles de polluer par leur énergie électromagnétique plutôt que les circuits analogiques.

Dans les circuits intégrés actuels, la principale source d'émission rayonnée est l'horloge. C'est pour cette raison que la première partie du travail de thèse se concentre sur les circuits asynchrones, affranchis du signal d'horloge et qui semblent être une alternative pour réduire les interférences électromagnétiques dans les circuits intégrés.

Pour des raisons de coût et de temps, il est nécessaire de pouvoir réduire l'émission électromagnétique d'un circuit dès sa conception. Le standard IBIS (*I/O Buffer Information Specification*) a été intégré afin que les outils EDA (*Electronic Design Automation*) de simulation puissent prédire l'émission des circuits intégrés. Bien que très utile, ce standard s'est montré insuffisant car il ne tenait compte que du rayonnement des entrées/sorties et non du rayonnement du coeur. Ainsi, le standard ICEM (*Integrated Circuit Electromagnetic compatibility Model*) est en cours d'intégration afin de combler cette lacune. Tous les paramètres qu'il utilise se concentrent autour de la principale source d'émission électromagnétique du circuit : l'activité électrique du coeur. Cette activité peut être modélisée par une somme de pics de courant dus à la commutation des éléments dans le circuit.

C'est en s'adressant au cœur du circuit lui-même que cette thèse tente d'apporter des solutions pour réduire les émissions électromagnétiques des circuits intégrés digitaux.

CHAPITRE 2 Méthodes de répartition de l'activité du courant

2.1 Introduction

Lorsque le phénomène des interférences électromagnétiques est considéré tôt dans l'élaboration d'un circuit, l'allongement du temps de conception dû au traitement des problèmes liés à ce phénomène est réduit. Par ailleurs, comme le montre la Figure 2-1 tirée de [SICA03], plus les problèmes CEM sont résolus tardivement dans le flot de conception d'un circuit, plus les coûts augmentent. Ainsi, il est nécessaire de traiter le problème des émissions électromagnétiques dans le circuit dès sa conception.

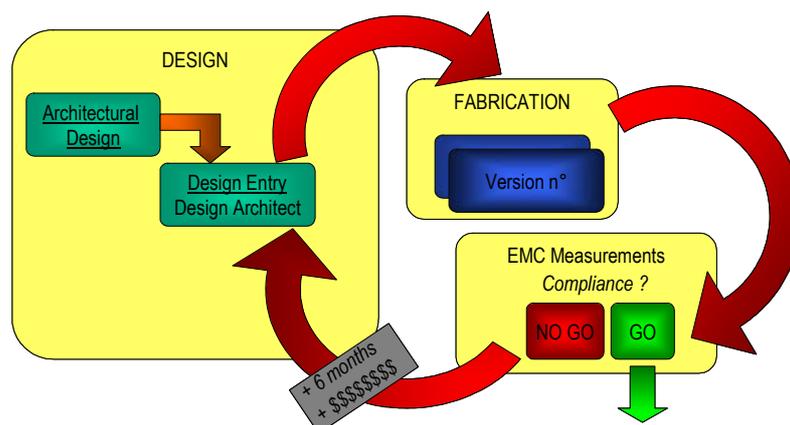


Figure 2-1 : Coûts liés à la résolution des problèmes de CEM [SICA03]

Ce chapitre présente des méthodes déjà existantes pour les circuits synchrones. Celles-ci répartissent l'activité du courant lors de la conception des circuits afin de réduire leurs émissions électromagnétiques. Dans un premier temps, les stades de la conception qui sont concernés sont décrits (paragraphe 2.2). Ensuite, le but général des méthodes de répartition de l'activité du courant est détaillé (paragraphe 2.3). Enfin, les différentes méthodes sont décrites dans les paragraphes 2.4, 2.5 et 2.6.

2.2 Etapes de la conception concernées

La Figure 2-2 présente les différentes méthodes dédiées aux circuits synchrones qui sont étudiées ainsi que les niveaux de conception concernés.

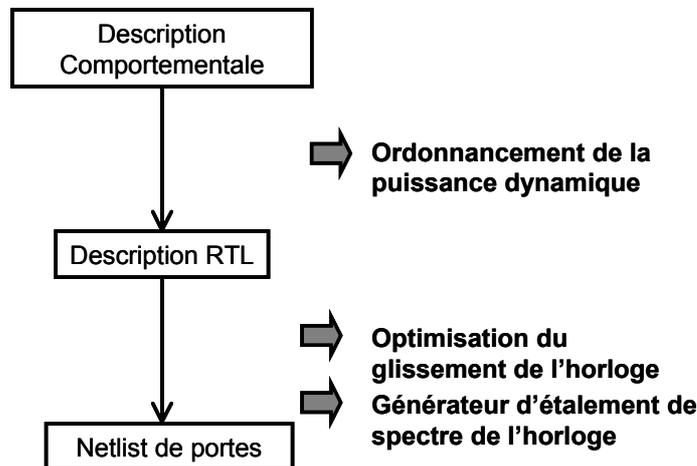


Figure 2-2 : Les différentes méthodes de répartition du courant

2.2.1 Au niveau portes

Au niveau de la description logique (portes), l'implémentation logique du circuit est connue et fixée. A ce stade, le contrôle de la répartition du courant utilise la dynamique des signaux qui circulent dans le circuit. Les méthodes consistent alors à contrôler le signal d'horloge afin que les composantes séquentielles dans le circuit ne soient pas activées au même instant, ce qui réduit ainsi l'appel de courant simultané sur l'alimentation. Ce contrôle peut être effectué par modulation du signal d'horloge - *Spread Spectrum Clock* [HARD94]- ou par le contrôle des retards dans l'arbre d'horloge (Contrôle du déphasage de l'horloge [BENI97][DOUG02][BLUN02]- *Clock skew Optimisation*) (paragraphe 2.5).

2.2.2 Au niveau registres

Lorsque le circuit est décrit au niveau registre (RTL de l'anglais *Register Transfer Level*), les ressources (qui sont aussi appelées opérateurs dans ce mémoire) sont connues ainsi que les traitements qu'elles effectuent. Cependant leur structure logique n'est pas encore connue. C'est l'étape de la synthèse structurelle qui permet d'obtenir les combinaisons de portes logiques qui composent ces ressources et l'ensemble du circuit. Le contrôle du déphasage du signal d'horloge peut aussi s'effectuer dès la synthèse structurelle du circuit.

2.2.3 A la description comportementale

A la description comportementale du circuit, l'implémentation logique n'est pas connue. Le fonctionnement du circuit est décrit dans un langage de haut niveau (VHDL [AIRI98] ou Verilog). C'est la synthèse comportementale qui fait le lien entre la description comportementale et la description RTL [KNAP96]. Cette synthèse permet d'identifier les différents opérateurs qui composent le circuit. Lors de cette étape, une opération appelée ordonnancement [WALK95] permet de répartir les tâches dans le circuit et de les attribuer aux opérateurs en fonction de contraintes définies (temps de cycle, nombre de ressources etc.). Plus de détails à ce sujet sont donnés dans le chapitre 5. La méthode d'ordonnancement de la puissance dynamique s'applique au circuit synchrone à cette étape en vue de contrôler la quantité de courant consommé. Elle ne concerne pas principalement les pics de courants mais utilise le principe de répartition par ordonnancement qui intéresse cette thèse. Cette technique est détaillée dans le paragraphe 2.6.

2.3 But des méthodes de répartition

2.3.1 Spectre de l'activité du courant

L'énergie normalisée $W(t)$ d'un signal $x(t)$ s'exprime de la façon suivante

$$W_x(t_1, t_2) = \int_{t_1}^{t_2} x^2(t) dt$$

Équation 2-1

tandis que la puissance moyenne normalisée de ce signal est :

$$P_x(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} x^2(t) dt$$

Équation 2-2

La distribution spectrale du signal $x(t)$ peut être obtenue grâce à l'application de la transformée de Fourier qui introduit le principe de dualité entre l'espace temps et l'espace fréquence :

$$X(f) = \int_{-\infty}^{+\infty} x(t) \exp(-j2\pi ft) dt$$

Équation 2-3

Cette représentation résulte de la décomposition du signal variant en fonction du temps en ses composantes fréquentielles [DECO99].

La représentation en fréquence de l'activité du courant dans un circuit est pratique pour estimer l'amplitude du courant consommé aux différentes fréquences. L'énergie dissipée par le circuit peut ainsi être déduite rapidement pour les différentes fréquences. Les fréquences auxquelles le circuit rayonne et peut induire des perturbations ainsi que l'ampleur du rayonnement sont identifiées.

2.3.1a Spectre en fréquence d'un signal périodique

La Figure 2-3 montre le spectre fréquentiel [KERR97] d'un signal périodique, tel que le signal d'horloge, et, dont la transformée de Fourier est :

$$f(t) = c_0 + 2 \sum_{n=1}^{\infty} c_n \cos(2\pi n f_c t + \theta_n)$$

Équation 2-4

avec

$$c_n = \frac{1}{T} \int_0^T f(t) e^{-j2\pi n f_c t} dt \text{ et } n = 0, 1, 2, 3, \dots$$

Équation 2-5

$\theta_n = \arctan \frac{\text{Im} c_n}{\text{Re} c_n}$ et $f_c = 1/T$ dans le cas de l'horloge :

$$c_n = -j \frac{V_0}{\pi n} \text{ pour } n = 1, 3, 5, \dots$$

$$c_n = 0 \text{ pour } n = 2, 4, 6, \dots$$

$$c_n = \frac{V_0}{2} \text{ pour } n = 0$$

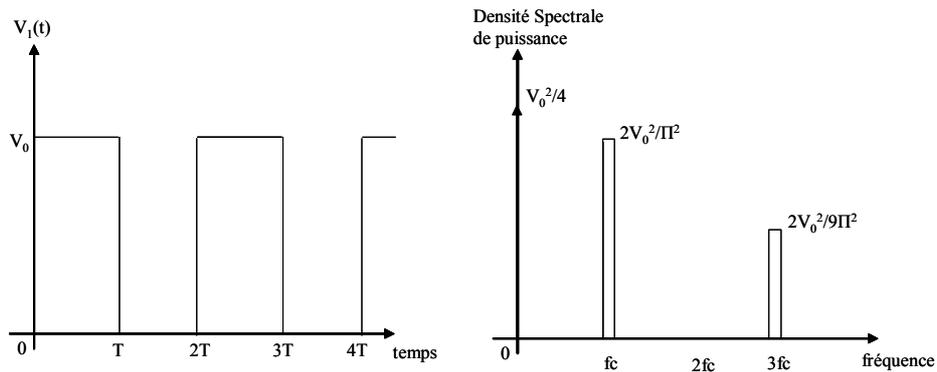


Figure 2-3 : Un signal périodique et son spectre fréquentiel

L'énergie générée par le signal est concentrée dans chaque harmonique de la fréquence d'horloge. L'absence d'énergie sur les harmoniques paires est une conséquence de la demi période nulle de l'allure du signal. Cependant, dans les circuits numériques, les émissions rayonnées apparaissent autant sur les harmoniques correspondant aux fronts montants et descendants de l'horloge (Figure 2-4). Ceci est dû à l'activité électrique générée par les fronts montants et descendants de l'horloge.

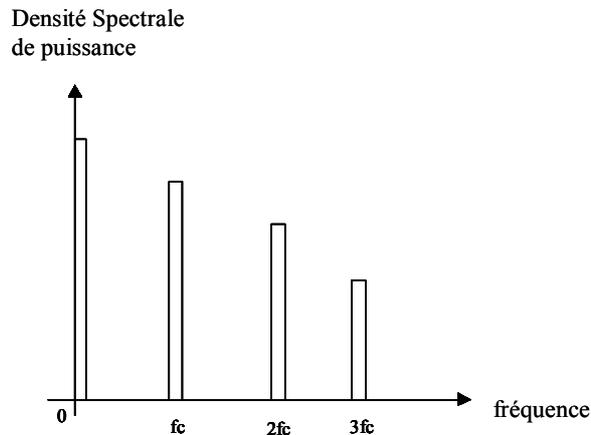


Figure 2-4 : Allure du spectre fréquentiel d'un circuit synchrone

2.3.1b Spectre en fréquence d'un signal aléatoire

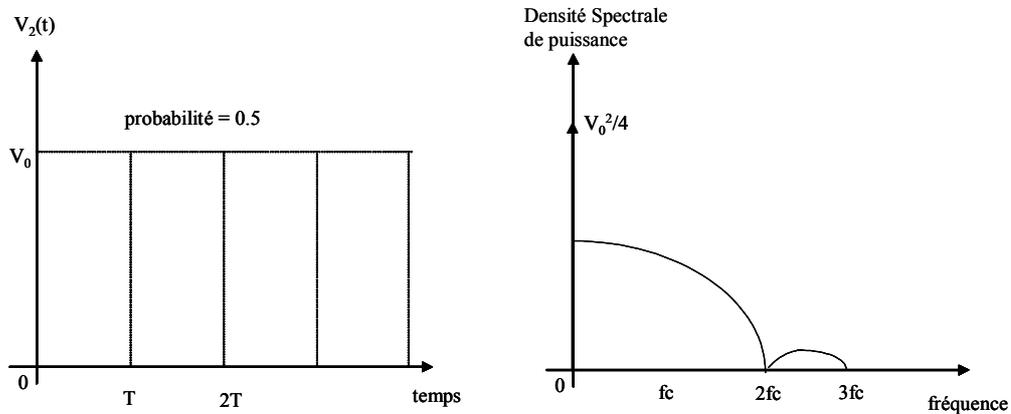


Figure 2-5 : signal aléatoire et son spectre fréquentiel

La Figure 2-5 montre le spectre fréquentiel d'un signal aléatoire. L'énergie du signal est répartie de part et d'autre de chaque harmonique. La valeur maximale du spectre d'un signal aléatoire est inférieure à celle d'un signal périodique. Ainsi, ce qui diffère entre le spectre du signal périodique et du signal aléatoire, c'est la répartition de l'énergie en fréquence : elle est concentrée sur les harmoniques dans le cas d'un signal tel que l'horloge tandis qu'elle est répartie sur la bande de fréquence pour le signal aléatoire. Dans les circuits intégrés, les signaux d'horloge sont donc plus dangereux que les signaux de données. Les variations brusques des signaux (montée et descente) sont les principales causes du contenu haute fréquence du spectre. Plus le spectre s'étend vers les hautes fréquences, plus le couplage devient efficace. En diminuant le contenu haute fréquence du spectre, l'efficacité du couplage donc le niveau du signal perturbateur dans le circuit peut être réduite.

2.3.2 Principe des méthodes de répartition de l'activité du courant

Les méthodes décrites ci après utilisent toutes le même principe : dans un circuit synchrone, l'horloge est la principale source de bruit, il suffit donc de modifier sa fréquence pour répartir son énergie.

2.4 Étalement du spectre de l'horloge

2.4.1 Principe

Le concept d'étalement de spectre n'est pas nouveau, il est déjà employé par les systèmes de communication tels que la radio FM. L'application de cette méthode pour l'usage des circuits intégrés remonte à K.B. Hardin qui établit en 1994 [HARDI94] le principe du *Spread Spectrum Clock* (SSC).

La méthode agit sur la distribution de l'horloge sans pour autant connaître l'architecture ou le fonctionnement du circuit. Seule, la contrainte suivante est respectée : les signaux doivent toujours être valides.

Le spectre de l'énergie consommée est étalé par une modulation en fréquence du signal d'horloge ce qui diminue l'amplitude des harmoniques de l'horloge. En général, la modulation est appliquée grâce à un circuit externe au circuit intégré mais elle peut être aussi intégrée à l'intérieur du circuit lui-même.

Soit $s(t)$ le signal modulé; il peut se mettre sous la forme : $s(t) = S(t) \cdot \cos \theta(t)$ où $S(t)$ est l'amplitude instantanée du signal tandis que $\theta(t)$ et la phase instantanée. Comme la modulation est en fréquence :

$$\frac{d\theta(t)}{dt} = 2\pi f_i(t) = 2\pi f_c + KV(t)$$

Équation 2-6

où f_i est la fréquence instantanée du signal modulé, K l'indice de modulation, et $V(t)$ signal modulant de fréquence f_m .

La déviation du pic en fréquence est

$$\Delta f_c = \frac{1}{2\pi} KV(t)$$

Équation 2-7

La Figure 2-6 tirée de [HARD94] montre un nième harmonique du signal d'horloge.

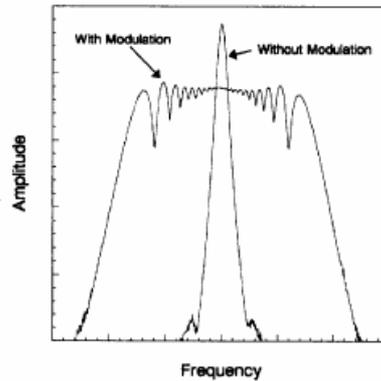


Figure 2-6 : Harmonique d'horloge avec et sans modulation [HARD94]

Dans le domaine fréquentiel, cet harmonique est une fonction centrée sur une fréquence qui est un multiple n de la fréquence du fondamental. Lorsque le signal d'horloge est modulé en fréquence, l'énergie de chaque harmonique est étalée sur une bande de fréquence et l'amplitude de l'harmonique est réduite.

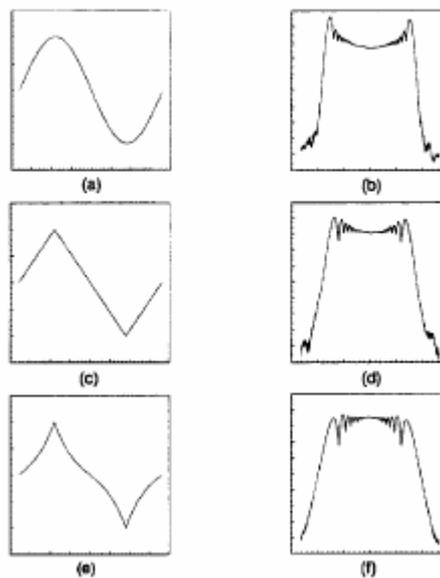


Figure 2-7 : Différentes fonctions a, c, e utilisées pour la modulation et résultats obtenus (b, d, f) [HARD94]

La Figure 2-7 montre le résultat obtenu lors de l'utilisation de différentes fonctions. Pour chaque fréquence, plus la dérivée de la fonction modulante est faible et plus l'énergie sera concentrée sur la fréquence. Ainsi l'utilisation de la fonction sinusoïdale (a) est moins intéressante que celle de la fonction triangulaire (c) ou de la fonction représentée en (e).

Le spectre obtenu peut être calculé par la méthode suivante : Le signal d'horloge modulé est périodique de période $1/f_m$. Sa phase est :

$$\theta(t) = \int_{-\infty}^t 2\pi f_i(t) dt$$

Équation 2-8

Les coefficients de Fourier de la fonction modulé sont [HARD94]:

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} S(t) \cos(n2\pi f_m t) dt$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} S(t) \sin(n2\pi f_m t) dt$$

$$S(t) = \begin{cases} Sm, \sin \theta(t) \geq 0 \\ 0, \sin \theta(t) < 0 \end{cases}$$

Système Équation 2-9

L'amplitude de l'harmonique n est :

$$|C_n| = \sqrt{a_n^2 + b_n^2}$$

Équation 2-10

2.4.2 Applications et résultats

La méthode du SSC est appliquée sur une chaîne expérimentale où un générateur (appelé *Spread Spectrum Clock Generator* – SSCG) fourni la fonction (e) vue précédemment. La chaîne produit également un signal carré représentant une horloge de 20 MHz et qui est modulé par la fonction (e). Les résultats obtenus sont les suivantes : l'amplitude du troisième harmonique (60 MHz) est réduit de 2 dB tandis que celle du 20^{ième} (400 MHz) est réduite de 10 dB.

Dans [STON96], une méthode similaire appelée *Frequency Hopping Spread Spectrum* est utilisée. Elle consiste à moduler le signal d'horloge par une fréquence déterminée pour un nombre de cycles donnés. Cette méthode est utilisée sur un microcontrôleur 87C196KC16 à une fréquence de 16 MHz. Elle permet d'obtenir un gain maximum de 8 dB obtenu sur la troisième harmonique (48 MHz).

2.5 Contrôle du déphasage de l'horloge (*Clock Skew Optimization*)

2.5.1 Déphasage d'horloge (*clock skew*)

Dans un circuit synchrone, les fils et les interconnexions induisent des délais lors de la propagation du signal d'horloge. Le signal d'horloge arrive donc aux différentes parties du circuit à des instants différents. Ce phénomène de déphasage de l'horloge est appelé gigue d'horloge (*clock skew*). La gestion du déphasage de l'horloge est un point important dans la conception des circuits synchrones car la validité des données et la consommation dans le circuit en dépendent.

2.5.2 Arbre d'horloge

Le problème du déphasage d'horloge induit des violations de temps et le dysfonctionnement du système. De bons contrôles du *clock skew* existent, certaines optimisations [CHO93] [CARR97][GUFE97][HSIE97] consistent à accélérer la vitesse du circuit en regroupant les parties séquentielles (bascules appelées aussi *Flip-Flop-FF*) ou réduire la puissance moyenne en contrôlant l'alimentation des différentes parties. La propagation du signal d'horloge dans le circuit est assurée par un arbre d'horloge (Figure 2-8) où des amplificateurs (*buffers*) dans les branches permettent de contenir le déphasage et de redresser le signal. Dans ce cas, les différentes parties séquentielles ne reçoivent pas le signal d'horloge simultanément mais l'arbre d'horloge permet d'égaliser le temps de commutation dans les différentes parties du circuit et de les contrôler. Cependant, les différents étages d'amplificateurs (*buffers*) induisent des délais et augmentent la consommation du circuit.

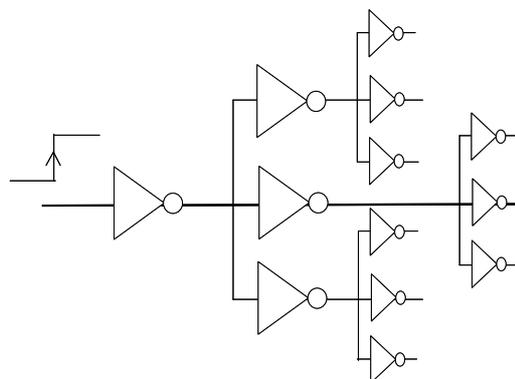


Figure 2-8 : Arbre d'horloge

2.5.3 Contraintes de *zero clocking* et *double clocking*

Lorsque le déphasage de l'horloge dans le circuit est contrôlé, les contraintes de *zero et double clocking* doivent être respectées.

Le *zero clocking* apparaît quand la logique combinatoire dans la conception du circuit est trop lente pour produire un ensemble de sorties valides dans une période d'horloge. Ce phénomène détermine la vitesse maximale de fonctionnement du circuit.

Au contraire, le *double clocking* apparaît quand la logique combinatoire est assez rapide pour produire plus d'un ensemble de sorties valides pour une seule période d'horloge. Ce dernier phénomène est utilisé dans certains circuits où les performances en vitesse sont recherchées. Les éléments dans le circuit sont implémentés de façon à ce qu'ils soient sensibles au front montant et au front descendant de l'horloge.

La vitesse de la logique combinatoire, le déphasage d'horloge et les paramètres en temps des éléments d'état affectent directement ces deux phénomènes.

2.5.4 Principe de la méthode

Le retard du signal d'horloge est exploité et les délais optimisés afin que l'activité des éléments séquentiels dans le circuit soit aux mieux répartie. Les arrivées de l'horloge sur les différents blocs d'éléments séquentiels sont ainsi décalées les unes par rapport aux autres. Ceci a pour effet d'étaler l'activité de l'horloge et ainsi de répartir l'énergie dissipée associée sur une bande de fréquence plus large. Le retard à insérer dans les fils de distribution d'horloge est calculé aléatoirement ou d'une façon déterministe comme dans [BLUN02]. Le paragraphe qui suit décrit quelques travaux étudiés et qui utilisent cette méthode.

2.5.5 Description de différents travaux

2.5.5a Travaux de L. Benini et al. [BENI97]

Pour réduire l'amplitude des pics de courant, les auteurs proposent d'utiliser l'optimisation du déphasage d'horloge en déterminant l'arrivée de l'horloge aux bascules à l'aide d'un algorithme dédié. Les technologies utilisant des fréquences inférieures à 150MHz sont concernées (les auteurs ont appliqué cette méthode sur un circuit de technologie SCMOS 1,2 microns). Le courant maximal sur les lignes d'alimentation est ainsi minimisé tout en satisfaisant les contraintes en temps. Par ailleurs, les différentes parties séquentielles sont regroupées pour être alimentées par le même signal d'horloge (*clock driver*) (Figure 2-9).

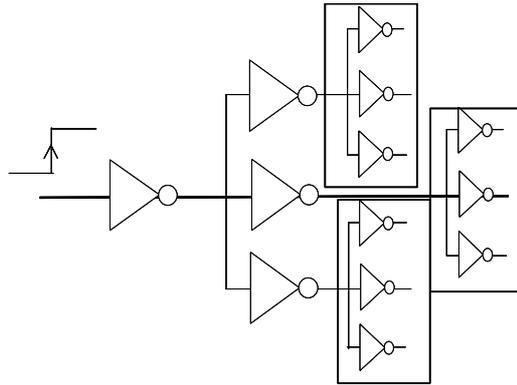


Figure 2-9 : Groupement des éléments séquentiels

La méthode adresse le circuit à sa description RTL. Un algorithme génétique calcule le temps d'arrivée de l'horloge aux éléments séquentiels à chaque cycle.

Le courant dans les éléments séquentiels (*Flip-Flops*) est modélisé par 2 triangles : $\Delta_i^r(t)$ est la fonction triangle représentant l'activité du courant au front montant de l'horloge tandis que $\Delta_i^f(t)$ est celle qui représente l'activité du courant au front descendant de l'horloge.

Les caractéristiques des triangles (temps de montée *setup*, temps de la valeur max, valeur max et temps de descente) sont obtenues par simulation.

L'activité du courant dans les éléments combinatoires peut être considérée comme la somme de plusieurs activités. Elle peut donc être modélisée par une somme de triangles : $\Delta_c(t) = \sum \Delta$. L'activité du courant global est considérée comme la somme des contributions en courant de chaque partie séquentielle et de la partie combinatoire:

$$I_{tot}(t) = \Delta_c(t) + \sum_0^N \Delta_i^r(t) + \sum_0^N \Delta_i^f(t)$$

Équation 2-11

L'activité du courant de la partie combinatoire dépend de la valeur des données et est donc très complexe. Ainsi elle n'est donc pas prise en compte dans ces travaux.

La prise en compte du déphasage d'horloge dans le circuit est effectuée de la façon suivante : t_i représente le délai entre le signal d'horloge et son arrivée sur le groupe i : $\Delta_i^r(t, T_i)$ et $\Delta_i^f(t, T_i)$. Ce délai doit répondre aux contraintes du *double clocking* et du *zero clocking* $0 \leq T_i \leq T_{ck}$, T_{ck} étant la période de l'horloge.

Si T est l'ensemble des retards dans le circuit, le courant global devient alors :

$$I_{tot}(t, T) = \Delta_c(t) + \sum_0^N \Delta_i^r(t, Ti) + \sum_0^N \Delta_i^f(t, Ti)$$

Équation 2-12

.La fonction coût est déterminée par le pic de courant maximum de l'activité du courant :

$$F(t) = \max_{t \in [0, Clkperiod]} \{I_{tot}(t, T)\}$$

Équation 2-13

Ce pic est réduit en trouvant l'ensemble optimal T des retards des arrivées de l'horloge dans le circuit.

Le problème de recherche de l'ensemble T tel que le pic de courant maximum dans le circuit soit minimal est un problème NP-Complet. Les simplifications faites sur les modèles du courant permettent de circonscrire le temps de calcul de l'algorithme choisi. L'algorithme utilisé est ici un algorithme génétique.

Un autre algorithme développé par les auteurs permet de regrouper les éléments séquentiels. Celui ci permet de réduire le nombre de délais à insérer dans le circuit.

Après application de leur méthode sur des réseaux séquentiels de 100 et 50 bascules, les auteurs ont obtenu 30% (1,5dB) de réduction sur les pics de courant maximums sans augmentation significative de la consommation. Il est envisageable d'appliquer ce type de méthode sur des circuits plus complexes, cependant, le temps de calcul reste important car l'algorithme utilisé est ici un algorithme génétique.

2.5.5b Travaux de I. Blunno et al. [BLUN02]

Les travaux décrits ici sont très proches de ceux de [BENI97] dont les auteurs s'inspirent. La modélisation de l'architecture a été ajoutée ainsi qu'un facteur de qualité.

Le courant dans les parties séquentielles du circuit est modélisé comme pour les travaux précédents [BENI97]. (2.5.5a).

Par ailleurs, les dépendances dans le circuit sont modélisées par un graphe de contraintes. La Figure 2-10 montre un graphe de contraintes. Dans ce type de graphe, les nœuds représentent les bascules en entrées (nœuds 1 et 2) et en sortie (nœud 3 et 4) tandis que les arcs (α_i) représentent les dépendances en temps entre ces nœuds.

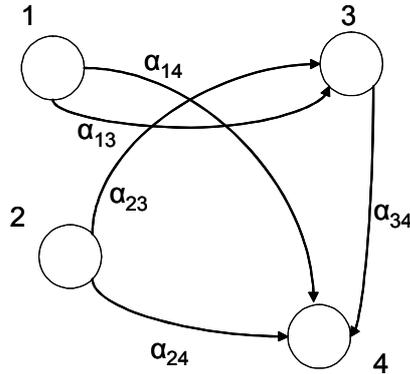


Figure 2-10 : Graphe de contraintes

Ce graphe permet de respecter les contraintes de *zero et double clocking* : [BENI97]

$$T_i + \delta_{ij}^{\max} \langle T_j + T_{ck} - T_s$$

$$T_i + \delta_{ij}^{\min} \rangle T_j + T_h$$

Équation 2-14

Où t_i est la valeur du retard de l'horloge sur l'ensemble des bascules i , T_{ck} la période de l'horloge, T_s et T_h respectivement les temps de montée et de maintien, et δ_{ij} le délai logique entre l'élément séquentiel i et l'élément séquentiel j relié par la liaison a_{ij} .

Le facteur de qualité η , quand à lui, permet de calculer la fonction de coût rapidement en quantifiant la triangularité de la forme du courant. Il est donné par la formule suivante :

$$\eta(T) = \min_{I^T} \sqrt{\int_{-\infty}^{+\infty} (I(t) - I^T(t))^2 dt}$$

Équation 2-15

où I^T est l'ensemble des formes triangulaires $I^T(t)$ qui exprime la différence entre la forme du courant estimée et l'approximation de la meilleure forme.

La partie combinatoire et la partie séquentielle sont ici traitées séparément.

Pour la partie combinatoire, des délais sont insérés après synthèse du circuit. Ils permettent de ralentir le fonctionnement des parties combinatoires. Plus les délais insérés sont grands, et plus la forme du courant est étalée, plus son amplitude est diminuée. Les délais sont calculés grâce à l'estimation de la forme du courant de la partie combinatoire.

Cette dernière s'effectue par un facteur de forme de l'enveloppe spectrale : $\frac{P_W}{T_{ck}}$ qui est le ratio entre la largeur de l'impulsion et de la période de l'horloge.

En ce qui concerne la partie séquentielle : les temps sont analysés après synthèse. Utilisant le graphe des contraintes, un algorithme génétique explore toutes les solutions possibles. La solution optimale est ensuite déterminée par le facteur de triangularisation.

Après application de la méthode sur un filtre à 4 coefficients et un filtre à 8 coefficients, les auteurs ont obtenu 20dB de réduction sur le spectre sur la bande de fréquence après $2F_{ck}$ ($F_{ck} = 50\text{MHz}$). Cependant, la surface a été augmentée de 85%.

2.5.5c Travaux de E. Mussol et J.Cortadella [MUSS96]

Comme pour les travaux précédents, la forme du triangle est utilisée. Chaque élément séquentiel est modélisé par un triangle et non deux triangles (un pour chaque front d'horloge). Le pic du triangle est près de t_s (temps de montée) car les données entrent à cet instant dans les registres.

Le principe de la méthode est de modifier, pour chaque cycle, la phase de l'horloge (Figure 2-11). Un générateur aléatoire est alors utilisé. Il sélectionne les différentes phases pour le signal d'horloge. Le nombre de délais à insérer dans l'arbre d'horloge dépend du nombre de phases désirées.

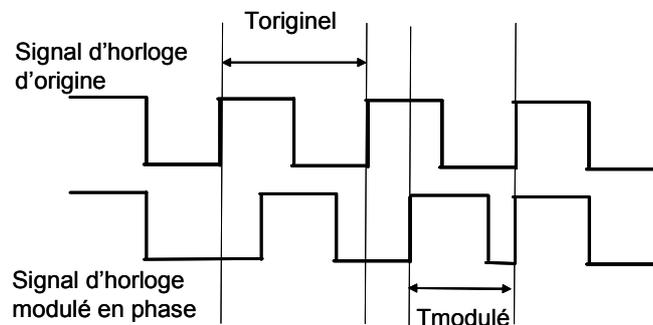


Figure 2-11 : Horloge à plusieurs phases

L'implémentation de la méthode a été faite sur différents circuits pour 2, 4 et 8 phases. Pour 8 phases, les harmoniques hautes fréquences ont été éliminées.

2.6 Ordonnancement de la puissance dynamique

2.6.1 Ordonnancement

L'ordonnancement s'effectue en général lors de la synthèse comportementale du circuit. Il consiste à répartir les traitements dans le temps de cycle du circuit en les affectant aux différentes ressources. En général, une description du circuit est utilisée par l'algorithme. Cette description représente les dépendances entre les différentes opérations effectuées par le circuit dans le temps. Elle utilise pour échelle temporelle un ensemble de pas de temps appelés pas de contrôle. Celui-ci représente le temps de cycle du circuit. Chaque pas est égal à une période de l'horloge. L'ordonnancement est vu plus en détail dans le paragraphe 5.3.4b.

2.6.2 Principe

Plus il y a de basculements de signaux simultanés, plus les variations du courant global sont brusques et plus il y a d'interférences électromagnétiques. Certaines recherches pour la faible consommation des circuits intégrés [AZEVO1][MART96][GUPT01] rejoignent aussi le but de cette thèse par la manière dont elles procèdent pour la répartition de la consommation dynamique. Pour lisser la consommation, les méthodes décrites ci après attribuent les tâches aux ressources et les répartissent dans le temps.

Cet ordonnancement s'applique dès la synthèse comportementale du circuit. Les ressources ne sont ni fixées, ni allouées. L'ordonnancement consiste à répartir les activités des ressources du système dans le temps afin de lisser au mieux la puissance dynamique consommée. Comme le nombre de ressources qui consomment dans un cycle donné est diminué, le nombre de pics de courant simultanés est aussi diminué.

2.6.3 Description de différents travaux

2.6.3a Travaux de A. Azevedo et al. [AZEVO1]

Le but de ces travaux est de répartir au mieux la consommation du circuit. Le circuit est modélisé sous la forme d'un graphe flot de données (*Data Flow Graph*). La contrainte en temps est respectée pour répartir les traitements des différentes ressources. L'algorithme *Force Directed Scheduling* qui sera détaillé dans le chapitre 5 est utilisé.

2.6.3b Travaux de R.S. Martin et J.P. Knight [MART96]

Les auteurs proposent un outil appelé Power Profiler dont le but est de minimiser en particulier:

- la consommation moyenne avec des contraintes en surface, temps et pic de consommation
- le pic de consommation avec des contraintes en surface, temps et consommation moyenne

A partir d'une description flot de données du circuit, un algorithme génétique permet de déterminer la répartition des traitements et leurs affectations aux ressources. Une réduction du pic de consommation de 66% (3,6 dB) a été obtenue en appliquant cet outil sur un circuit DCT (*Direct Cosinus Transform*).

2.7 Conclusion

Dans ce chapitre, différentes méthodes de réduction des émissions électromagnétiques dans les circuits synchrones ont été étudiées. Ces méthodes adressent le circuit dès sa conception et sont appliquées à différents niveaux : niveau portes, niveau registres, niveau comportemental. Elles utilisent le principe qui consiste à contrôler la forme du courant en répartissant les traitements dans le circuit.

La séquence des traitements dans le circuit dépend directement de l'activité de l'horloge. Ainsi, certaines méthodes s'attaquent au signal d'horloge lui-même : en le modulant (*Spread Spectrum Clock*), ou en optimisant les différents retards lors de la propagation de ce signal (*Clock skew Optimisation*). Ces dernières répartissent l'activité du courant dans le but de réduire les pics de courant maximum dans le circuit.

L'ordonnancement de la puissance dynamique permet aussi de contrôler la forme du courant en répartissant les tâches des ressources du circuit. Le temps de calcul du circuit est discrétisé par l'activité de l'horloge et les traitement des ressources sont répartis dans ces pas de temps. Ainsi l'activité du courant est répartie dans le temps et dans le circuit.

Ainsi, dans les circuits synchrones, la gestion de l'activité du courant est fonction de celle de l'activité de l'horloge. Ceci n'est pas le cas dans les circuits asynchrones dépourvus de ce signal de contrôle. Dans le chapitre 5, cette thèse propose une méthode pour la réduction des émissions électromagnétiques dans les circuits asynchrones. Ces circuits sont d'abord étudiés (Chapitre 3) avant d'être traités. Une méthode (*Asynchronisation*) permettant le passage d'une description RTL d'un circuit synchrone à un circuit asynchrone– ou plutôt dé-synchronisé - est ensuite proposée (Chapitre 4).

CHAPITRE 3 Circuits asynchrones

3.1 Introduction

L'histoire des circuits asynchrones remonte aux premiers ordinateurs, en particulier le fameux ENIAC élaboré vers 1945 par Eckert et Mauchly. Le fonctionnement de ce calculateur reposait sur des synchronisations locales entre les différentes unités et n'était donc pas géré par une horloge globale.

Depuis, les architectures des ordinateurs ont évolué et des fonctions de calcul sans cesse croissantes en complexité sont intégrées sur une puce dont la surface s'amenuise. Dans la course à l'intégration, les circuits asynchrones n'ont pas su s'imposer car les méthodes de conception qui leur étaient dédiées ont rapidement montré leur lourdeur face à la gestion des événements aléatoires. En effet, dans un circuit asynchrone, tout événement de ce type peut être interprété comme une donnée à traiter et déclencher un fonctionnement erroné. C'est ainsi que la méthode suivante, plus simple, a été largement employée : cadencer les traitements dans le circuit par un signal global de période régulière – l'horloge – afin d'assurer la validité des données.

Depuis, malgré l'omniprésence des circuits synchrones, des recherches sur les circuits asynchrones ont tout de même été menées. Deux faits marquants peuvent être cités : le développement durant les années 1950 d'une méthode de conception pour les circuits asynchrones utilisant le mode fondamental (*fundamental-mode*) par Huffman et celui d'une théorie de base sur les circuits indépendants de la vitesse par Muller [HAUC95]. Ensuite, Unger publia le premier livre sur la conception de circuits asynchrones en 1969 et beaucoup d'autres publications suivirent. Par ailleurs, de nombreux circuits asynchrones ont été développés.

Aujourd'hui, avec l'évolution technologique qui suit les prévisions de Moore, les circuits synchrones montrent leurs faiblesses et les circuits asynchrones apparaissent comme une alternative intéressante.

Ce chapitre est dédié aux circuits asynchrones. Il présente les concepts de base dans le paragraphe 3.2, tandis que l'implémentation de ces circuits est présentée paragraphe 3.3. Le paragraphe 3.4 traite de l'émergence de ces circuits.

3.2 Concepts de base

Ce paragraphe présente le principe de fonctionnement des circuits asynchrones et les caractéristiques d'un opérateur asynchrone. Le lecteur peut trouver plus de détails dans [RENA00][BERK99] [HAUC95] [SPAR01] dont cette partie est largement inspirée.

3.2.1 Principe de fonctionnement

Les circuits asynchrones diffèrent des circuits synchrones par leur gestion des séquences des données.

3.2.1a Fonctionnement d'un circuit synchrone

Dans un circuit numérique synchrone, le traitement des actions est régi par un signal périodique appelé horloge (Figure 3-1).

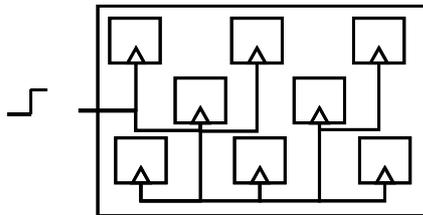


Figure 3-1: Circuit synchrone

Ce signal contrôle les composantes qui lui sont sensibles. Ces composantes sont dites séquentielles car elles sont fonction de leurs entrées et du temps. A l'opposé, les composantes combinatoires dépendent uniquement de leurs entrées.

Dans un circuit synchrone, le temps est défini par la cadence du signal d'horloge. Selon la convention utilisée, les composantes séquentielles opèrent soit au front montant du signal d'horloge (c'est à dire lorsqu'il passe de la valeur 0 à la valeur 1) soit à son front descendant (lorsqu'il passe de la valeur 1 à la valeur 0). Une fois activées, ces composantes libèrent alors les données calculées par les parties combinatoires en amont. Les données résultantes sont ensuite traitées par les parties combinatoires en aval. Ainsi, l'horloge cadence le fonctionnement du circuit en libérant périodiquement les données et simultanément dans toutes les parties du circuit par le biais des composantes séquentielles.

3.2.1b Fonctionnement d'un circuit asynchrone, notion de canal

Dans un circuit asynchrone (Figure 3-2), le traitement des actions n'est pas régi par un signal global mais par des communications locales entre les différents éléments du circuit.

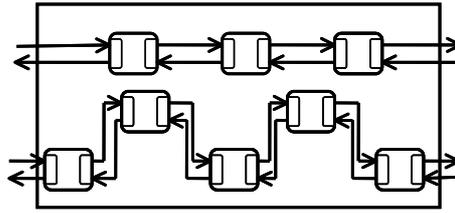


Figure 3-2 : Circuit asynchrone

Ces communications assurent la séquence des traitements dans le circuit par des protocoles de type poignée de main (*handshake*). Elles sont synchronisées par le biais de canaux qui permettent à 2 opérateurs d'échanger des données par une connexion point à point gérée par le protocole de communication. Chaque canal est composé de signaux de contrôle (requête et acquittement) qui permettent d'assurer le protocole et du chemin des données.

L'exemple suivant (Figure 3-3) décrit une communication dans un circuit asynchrone.

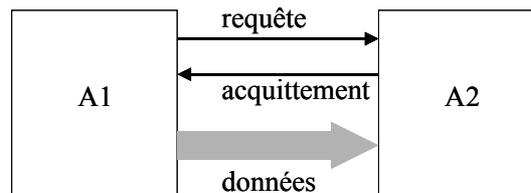


Figure 3-3 : Communication de type poignée de main (*handshake*)

A1 et A2 sont 2 opérateurs asynchrones. A2 doit effectuer son calcul en utilisant une donnée résultante de A1.

Lorsque l'opérateur A1 fini son traitement, il envoie un signal de requête à l'opérateur A2 pour lui indiquer qu'il fournit des données valides. Si l'opérateur A2 est prêt, alors il effectue son traitement et indique à A1 qu'il peut recevoir de nouvelles données en utilisant un signal d'acquittement.

Les composantes séquentielles sont donc ici commandées par des signaux de contrôle locaux (3.3.4). La validité des données doit donc être assurée localement (3.3.5).

3.2.2 Caractéristiques d'un opérateur asynchrone

La Figure 3-4 représente 2 opérateurs asynchrones qui communiquent non seulement entre eux, mais aussi avec le reste du circuit. Elle montre le cas où les données sont groupées avec la requête (*bundled data*). Ce paragraphe décrit les différentes caractéristiques d'un opérateur asynchrone.

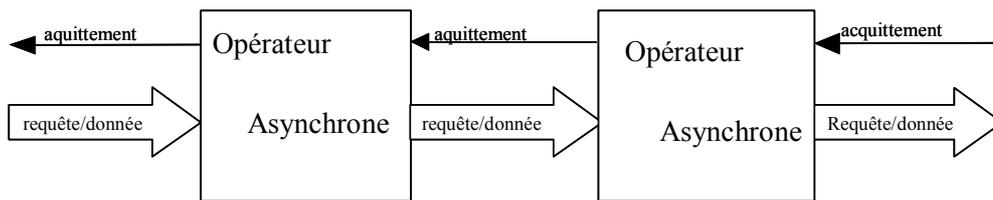


Figure 3-4: Opérateurs asynchrones

3.2.2a Latence

La latence d'un opérateur asynchrone correspond au temps nécessaire à une sortie pour être l'image de l'entrée. Il est égal au temps de calcul de la chaîne combinatoire.

3.2.2b Cycle

Le temps de cycle d'un opérateur asynchrone correspond au temps nécessaire entre la prise en compte de deux entrées successives.

3.2.2c Profondeur du pipeline

La profondeur du pipeline correspond au nombre d'informations que l'opérateur peut mémoriser.

3.3 Implémentation des circuits asynchrones

Lors de l'implémentation des circuits asynchrones [DAVI97][RENA00][LINE95], les aléas ou phénomènes de *glitches* dont la nature est décrite paragraphe 3.3.1 doivent être circonscrits afin d'assurer le fonctionnement correct du circuit. Ce paragraphe explique quelles sont les hypothèses faites pour l'implémentation d'un circuit asynchrone ainsi que les notions de base à connaître sur les protocoles de communication et le codage des données.

3.3.1 Aléas

Dans un état stable du circuit, la validité des données est garantie, mais lorsque les données évoluent, des phénomènes transitoires peuvent se produire. Ces phénomènes, appelés aussi *aléas* peuvent survenir suite à des changements de signaux en entrée des parties combinatoires ou des retards induits par les fils et interconnexions. Ils sont de différents types selon leur origine et leurs comportements :

Les *aléas combinatoires logiques* dépendent de la distribution des délais dans la logique, donc du placement des portes ou de l'implémentation du circuit. Ainsi une attention particulière est apportée lors de la conception d'un circuit asynchrone pour qu'il soit exempt d'aléas de ce type. Les aléas combinatoires logiques peuvent être soit statique soit dynamique.

- L'*aléa statique ou spike* est un changement de valeur transitoire d'un signal qui doit rester constant. Par exemple, il peut apparaître à la sortie d'une porte OU/ET lorsque les deux entrées changent simultanément.
- Les *aléas dynamiques* apparaissent lorsque le signal doit changer une seule fois de niveau. Un phénomène de rebond apparaît : il y a plusieurs changements de niveau.
- Les *aléas combinatoires fonctionnels* apparaissent suite à une mauvaise spécification de la fonction. Ils peuvent être supprimés en faisant des hypothèses sur les délais dès la spécification.
- Les *aléas séquentiels* apparaissent dans les signaux bouclés. Ils peuvent être détectables de la façon suivante : si l'entrée change 1 à 3 fois et que l'état final n'est pas le même dans les deux cas, alors il y a un aléa séquentiel.
- Les aléas de type *Single Input Change (SIC)* apparaissent lorsqu'une seule entrée change.
- Les aléas de type *Multiple Input Change (MIC)* apparaissent lorsque plusieurs entrées changent.

Les aléas peuvent s'éliminer d'eux-mêmes et ne pas influencer sur le fonctionnement du circuit, mais ils peuvent aussi avoir des conséquences sur les sorties des parties combinatoires du circuit. Les aléas peuvent aussi être identifiés et supprimés à plusieurs stades de la conception d'un circuit. S'ils sont dus au fonctionnement du circuit, alors ils peuvent être supprimés en changeant la spécification de la fonction.

3.3.2 Gestion des aléas

Le changement du signal d'horloge est régulier dans le temps ce qui permet, lors de la conception des circuits synchrones, de tenir compte de la discrétisation de leur fonctionnement dans le temps. Le pas de discrétisation correspond à la période du signal d'horloge. Sa valeur est déterminée par la latence maximale des parties combinatoires contenues entre 2 composantes séquentielles à laquelle elle doit être supérieure. Ce temps doit respecter par ailleurs la mise en place de la valeur des signaux (*setup*) et le maintien de cette valeur (*hold*). Ces conditions sont nécessaires pour assurer le fonctionnement correct du circuit et garantir la validité des données. Par leur conception, les circuits synchrones sont exempts d'aléas.

Quand aux circuits asynchrones, ils sont plus vulnérables face aux aléas car le temps n'est pas discrétisé. Une attention particulière est donc portée à la conception des

circuits asynchrones afin qu'ils soient exempts d'aléas. La validité des données est assurée localement par des délais dans les communications qui respectent la latence des parties combinatoires. Les délais sont rarement "purs" car ils ont une certaine inertie par rapport aux changements. Ils sont alors appelés délais inertiels et permettent d'éliminer les aléas équivalents à de courtes impulsions. Un délai fixe possède une valeur constante. Un délai borné possède une valeur limite inférieure et supérieure. Le délai non borné ne possède pas de limite.

3.3.3 Catégories de circuits asynchrones

Concevoir des circuits asynchrones robustes face aux aléas consiste donc à contrôler les hypothèses temporelles faites pour l'élaboration du circuit. Les différentes hypothèses faites sur les délais et le mode de fonctionnement de l'environnement permettent de classer les circuits asynchrones selon les catégories expliquées dans les paragraphes suivants et récapitulées Figure 3-5. Plus les hypothèses temporelles sont fortes et plus la robustesse du circuit est faible.

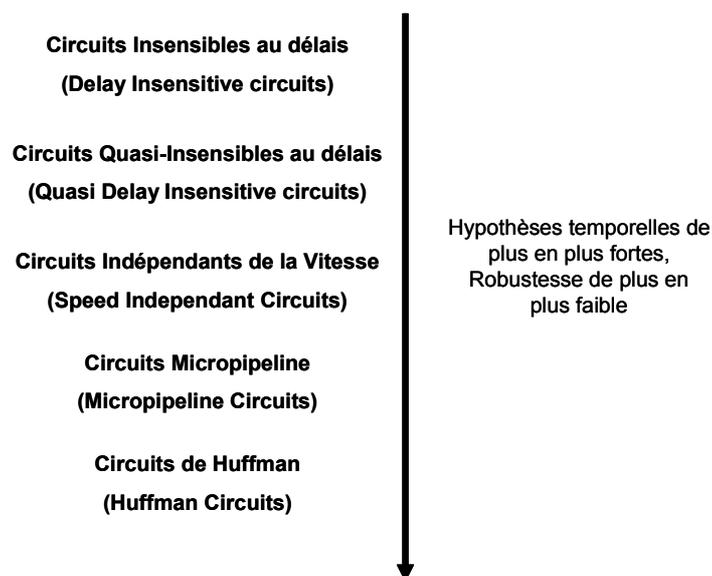


Figure 3-5 : Catégories de circuits asynchrones

3.3.3a Circuits Insensibles aux Délais (DI)

Ces circuits sont fonctionnellement corrects quels que soient les délais introduits par les fils ou les éléments logiques. Le modèle de délai utilisé est donc non borné : aucune hypothèse temporelle n'est faite. Cela impose que les circuits récepteurs doivent informer les circuits émetteurs que l'information a bien été reçue. Dans le cas où une seule entrée

change, elle doit pouvoir être acquittée indépendamment des autres. Ainsi, la réalisation de ce modèle de circuit demande une implémentation lourde par des portes complexes.

3.3.3b Circuits Quasi Insensibles au Délais (QDI)

Ce type de circuit utilise le même modèle de délai non borné pour les connexions, mais ajoute la notion de *fourche isochrone* (Figure 3-6). Une *fourche isochrone* est un fil qui connecte un expéditeur à 2 récepteurs. Elle est dite isochrone lorsque les délais allant de l'émetteur à chaque récepteur sont identiques. C'est donc un modèle de porte simple sans hypothèse temporelle pour les composants du circuit. Ceci permet de réaliser les circuits avec des portes standard.

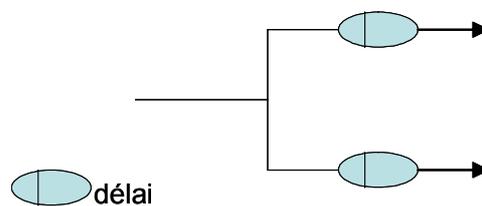


Figure 3-6 : Fourche isochrone

3.3.3c Circuits indépendants de la vitesse (SI)

Les délais dans les fils sont supposés négligeables et les délais dans les portes non bornés : toutes les fourches sont isochrones. Ce modèle est donc équivalent au modèle QDI mais n'est pas implantable à cause de l'hypothèse sur les fourches car dans les technologies actuelles, l'hypothèse des délais nuls dans les fils n'est pas réaliste.

3.3.3d Circuits micropipelines

Les circuits micropipeline ont été introduits par Sutherland en 1989 [SUTH89]. Dans ce type de circuit, les blocs combinatoires traitent les données, et la partie contrôle gère leur flux (Figure 3-7). Des délais sont insérés dans les requêtes afin de respecter les latences des parties combinatoires. Les délais sont donc bornés. La motivation première de ce type de circuit était de permettre un pipeline élastique. En effet, le nombre de données présentes dans le pipeline peut être variable.

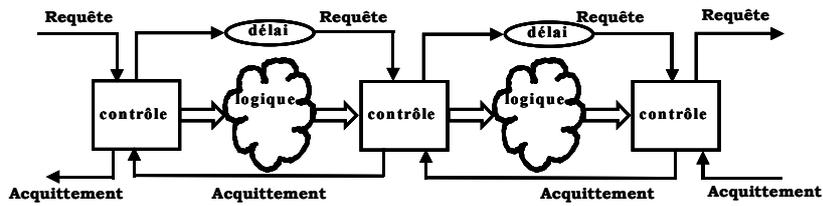


Figure 3-7 : Circuit micropipeline

3.3.3e Circuits de Huffman

Ces circuits ont un modèle de délais identique aux circuits synchrones : ils supposent que les délais dans tous les éléments du circuit sont bornés ou de même valeurs connues. Le contrôle du chemin de données s'apparente donc à un réseau d'horloges locales dont la période est estimée selon la latence maximale trouvée dans les parties combinatoires des circuits.

3.3.4 Protocoles de communication

Les communications dans les circuits asynchrones sont basées sur un protocole de type poignée de main (requête–acquittement, *handshake*). Le protocole de communication utilisé peut être 2 phases ou 4 phases. Le protocole 2 phases est sensible aux événements des signaux tandis que le protocole 4 phases est sensible à leurs niveaux.

3.3.4a Protocole 2 phases

Il est aussi appelé protocole *NRZ* pour *Non Retour à Zéro*, ou encore *half handshake*. Ce protocole est sensible aux événements des signaux (Figure 3-8).

- Phase 1 : Le récepteur est actif : il traite l'information et produit le signal d'acquittement.
- Phase 2 : L'émetteur est actif : il détecte le signal d'acquittement et envoie les nouvelles données.

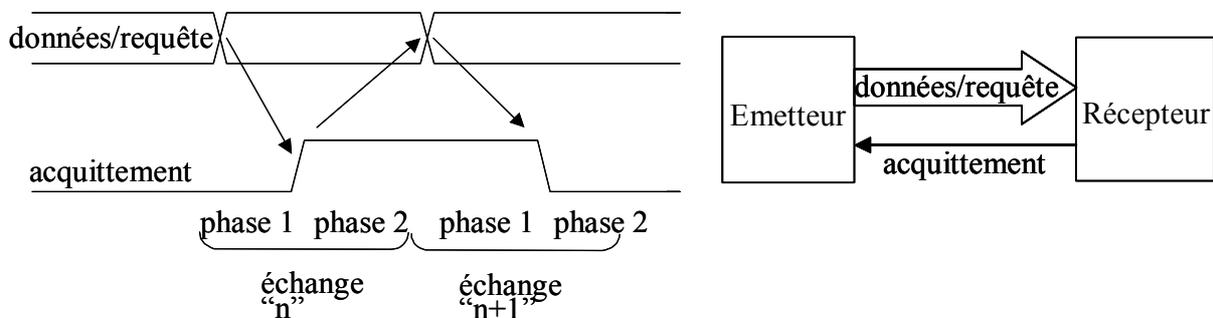


Figure 3-8 : Protocole 2 phases

3.3.4b Protocole 4 phases

Il est aussi appelé protocole *RZ* pour *Retour à Zéro*, ou *Full-Handshake* (Figure 3-9).

- Phase 1 : L'émetteur est passif. Le récepteur est actif : il détecte la présence de données valides, effectue son traitement et envoie un signal d'acquittement à l'émetteur.
- Phase 2 : L'émetteur est actif : il détecte l'acquittement du récepteur et invalide les données. Le récepteur est, quand à lui, passif.
- Phase 3 : L'émetteur est passif. Le récepteur détecte l'invalidité des données et mets son signal d'acquittement à l'état initial.
- Phase 4 : L'émetteur a détecté le changement du signal d'acquittement. L'émetteur et le récepteur sont passif : fin du protocole.

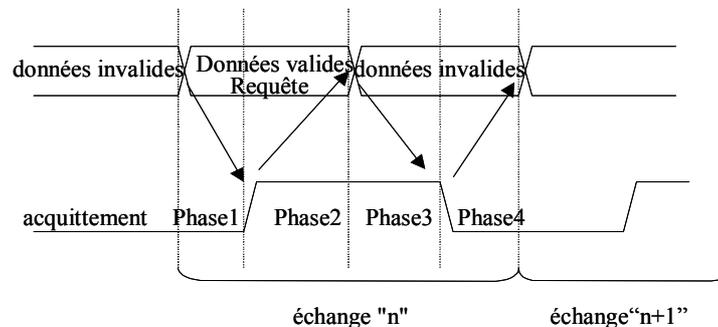


Figure 3-9 : Protocole 4 phases

Les protocoles WCHB, PCHB, PCFB

Parmi les nombreuses variantes du protocole 4 phases, cette thèse utilise les protocoles WCHB, PCHB, PCFB décrits ci-après. Pour plus de détails, le lecteur se reportera au Chapitre 6.

La Figure 3-10 représente un opérateur asynchrone et les canaux qui relient son entrée et sa sortie. Le fonctionnement de cet opérateur utilisant les différents protocoles est décrit ci-après.

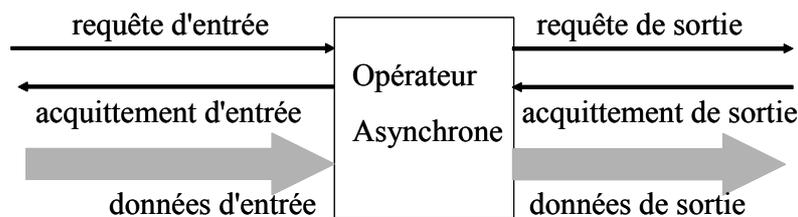


Figure 3-10 : Opérateur Asynchrone

Le protocole WCHB (*Weak Condition Half Buffer*) est appelé aussi protocole "early"[FURB96] ou protocole standard. La validité des données commence sur le front montant du signal de requête et se termine sur le front descendant du signal d'acquittement (Figure 3-11).

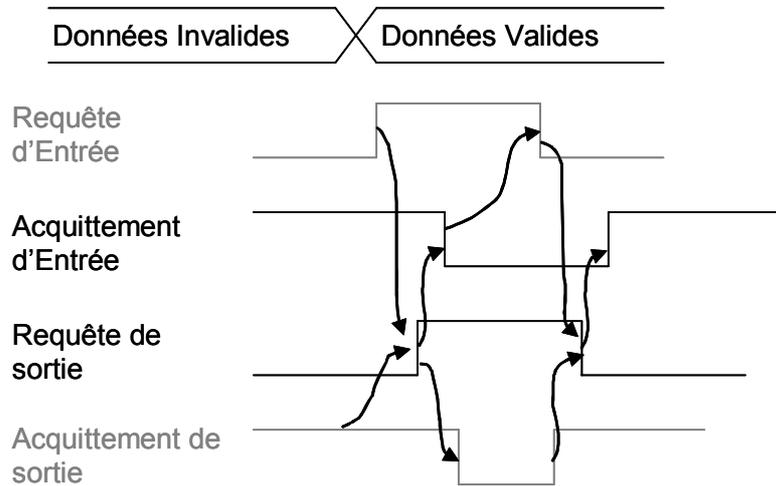


Figure 3-11 : Protocole WCHB

Le protocole PCHB (*Precharge logic Half Buffer*) utilise des phases montantes identiques au protocole WCHB. C'est au niveau des phases descendantes qu'il diffère : les remises à 0 des requêtes et la remise à 1 des acquittements (dans ce mémoire, les acquittements sont inactifs à 1) sont désynchronisées. La remise à zéro du canal d'entrée s'effectue avant celle du canal de sortie ce qui permet d'autoriser la communication au niveau de la sortie rapidement et de gagner de la vitesse dans le pipeline (Figure 3-12).

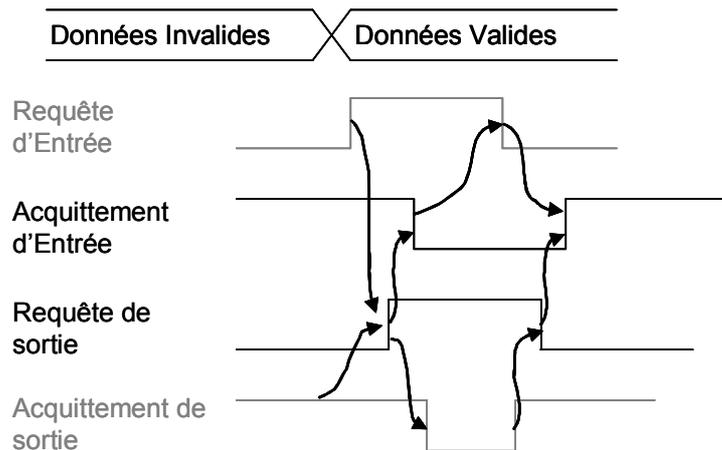


Figure 3-12 : Protocole PCHB

Le protocole PCFB (*Precharge Logic Full Buffer*) utilise des phases montantes identiques aux protocoles précédents. Comme le protocole PCHB, il est découplé lors de la remise à 0 des canaux d'entrée et de sortie. Cependant celle-ci est effectuée d'une manière totalement concurrente qui permet de découpler l'entrée de la sortie.

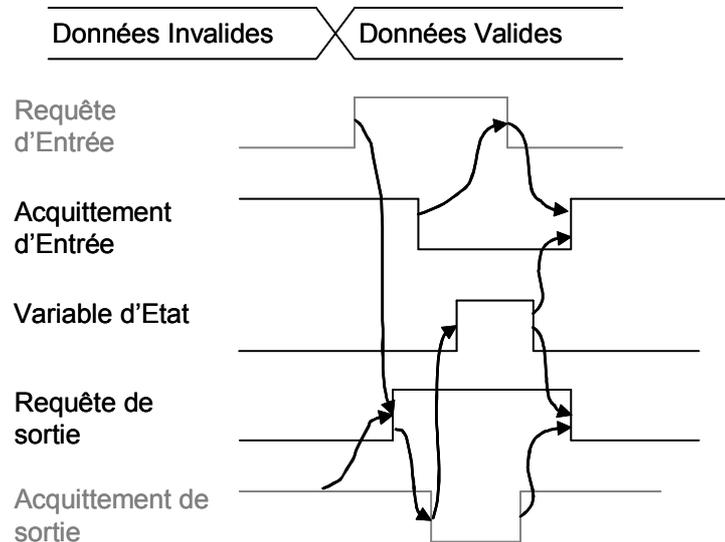


Figure 3-13 : Protocole PCFB

Choix du protocole

Le tableau ci-dessous synthétise les avantages et les inconvénients des différents protocoles 4 phases énumérés. L'implémentation utilisée est celle décrite dans [RIGA02]. Le nombre de portes indiqué représente celui utilisé dans un contrôleur de communication d'un opérateur. Il contient des porte standard et des portes de muller qui sont décrites dans le paragraphe 3.3.6.

protocole	portes utilisées	avantages	inconvénient
WCHB	portes de Muller : 1 inverseurs : 1	implémentation peu coûteuse	protocole lent dû au couplage des E/S
PCHB	portes de Muller : 2 inverseurs : 1	protocole rapide du au semi-découplage des E/S	implémentation complexe
PCFB	portes de Muller : 3 inverseurs : 2	protocole rapide du au découplage des E/S	implémentation complexe (gestion d'une variable interne)

L'implémentation du protocole 2 phases demande une logique beaucoup plus complexe que le protocole PCHB, due à la logique à évènements. Le choix d'un protocole pour l'implémentation d'un circuit dépend du choix entre des performances en vitesse plus élevées ou une implémentation moins complexe.

3.3.5 Codage des données

Dans un circuit asynchrone, la validité des données est assurée localement de la manière suivante : la donnée est groupée avec l'information de sa validité. Les différents types de codages utilisés sont décrits ci après.

3.3.5a Codage données groupées

Il se retrouve dans les circuits de type micropipeline. Un fil est ajouté au bus de données afin de spécifier l'information de validité (Figure 3-14). Ce type de codage peut être indifféremment utilisé avec le protocole 2 phases ou le protocole 4 phases.

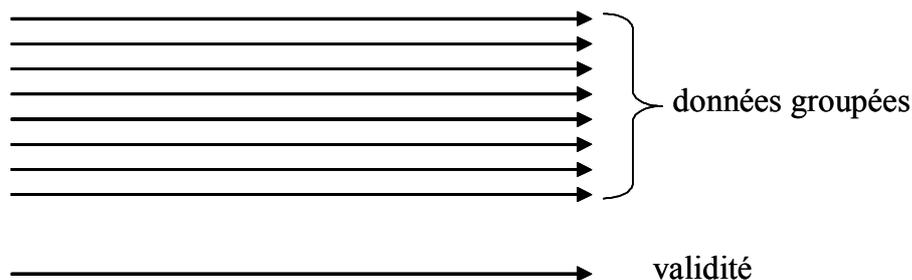


Figure 3-14 : Codage données groupées

3.3.5b Codage quatre états

Chaque bit de donnée est représenté par deux fils. Un de ces fils est réservé à la valeur '0' tandis que l'autre moitié est réservée à la valeur '1'. L'émission d'une nouvelle donnée se traduit par un changement sur un fil.

Deux conventions peuvent être utilisées :

- La convention liée au évènement des fils : un fil est dédié à une valeur et tout évènement sur un des fils signifie qu'une nouvelle valeur est valide. La Figure 3-15 représente deux fils D0 codant la valeur 0 et D1 codant la valeur 1: un évènement sur D0 signifie qu'un 0 est émis ou reçu tandis qu'un évènement sur D1 signifie qu'un 1 est émis ou reçu.

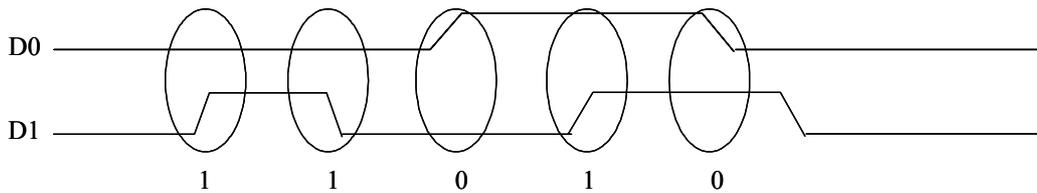


Figure 3-15 : Codage 4 états

- La convention liée à la parité des bits : un bit est dédié à la valeur et l'autre la parité. Si le bit de valeur change, alors la valeur change. Si le bit de parité change, alors la valeur ne change pas. La validité des données est assurée par le changement de parité du couple des fils. Dans Figure 3-16, la valeur est assurée par *D1* tandis que la parité est déterminée par *D0*.

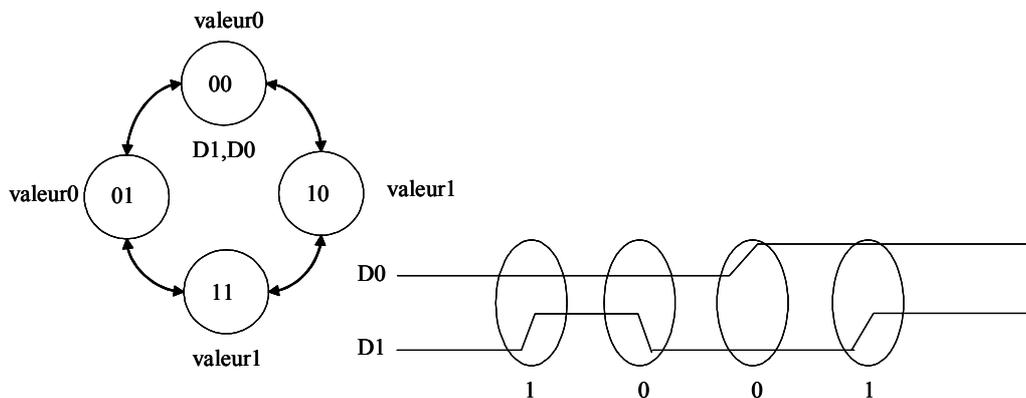


Figure 3-16 : Codage 4 états

Le codage quatre états se prête bien au protocole deux phases car chaque évènement sur un des deux fils qui codent le bit transporte une information.

3.3.5c Codage 3 états

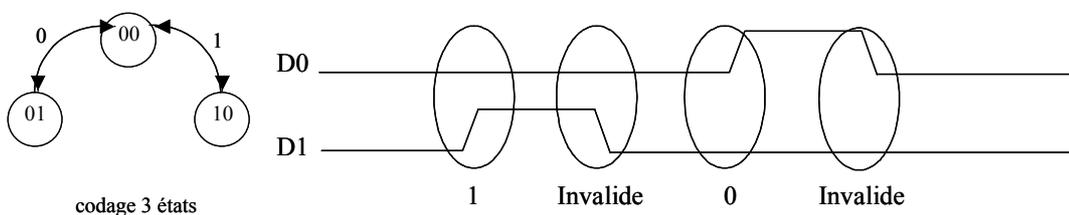


Figure 3-17 : Codage 3 états

Chaque bit est représenté par 2 fils mais contrairement au codage 4 états, les valeurs représentées ne sont pas dupliquées. L'état 11 est interdit. Lorsque les 2 fils sont à

0, il n'y a pas de valeur valide. Chaque changement de valeur valide implique de repasser par l'état 00. Il est adapté au protocole 4 phases.

3.3.6 Porte de Muller

Introduite par Muller en 1965, elle est un élément essentiel pour l'implémentation de la logique asynchrone. Appelée aussi C élément (Figure 3-18), elle permet d'obtenir l'image de la valeur à sa sortie lorsque ses entrées sont égales, quelque soit le nombre de ses entrées. Elle est donc indispensable pour l'implémentation du contrôle des circuits asynchrones (gestions des requêtes et acquittements de sortie en fonction des requêtes et acquittements d'entrée).

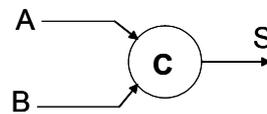


Figure 3-18 : Schéma de la porte de Muller

Cette fonction séquentielle est représentée par la table ci-dessous, pour 2 entrées A et B , où S est la valeur en sortie. S^l indique que la valeur de la sortie est égale à sa valeur précédente.

A	B	S
0	0	0
0	1	S^l
1	1	1
1	0	S^l

Figure 3-19 : Table de vérité d'une porte de Muller à 2 entrées

La porte de Muller peut être aussi dissymétrique dans le cas où toutes les entrées n'interviennent pas dans le changement de la sortie

La Figure 3-20 montre l'implémentation possible d'un porte de Muller [SPAR01]:

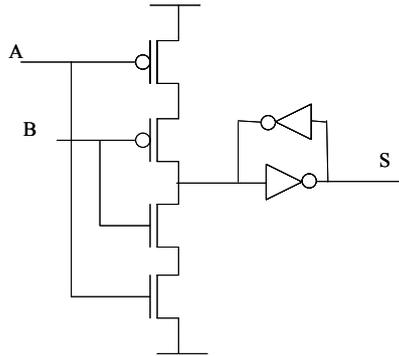


Figure 3-20 : Possible implémentation d'une porte de Muller

La porte de Muller peut être implémentée par des portes standard comme le montre la Figure 3-21.

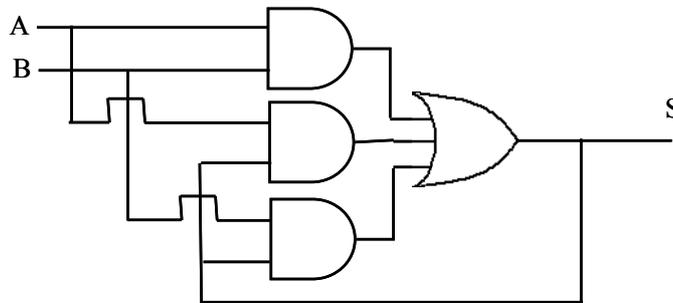


Figure 3-21 : Implémentation d'une Muller en portes standard

3.3.7 Duplicateur et Rendez-vous

Les structures suivantes permettent de gérer plusieurs signaux de requête et d'acquiescement en entrée et en sortie.

3.3.7a Duplicateur

Appelé aussi *Fork* dans la littérature, il permet d'envoyer plusieurs requêtes simultanément à partir d'une seule requête. Il acquiesce celle ci lorsqu'il reçoit les acquiescements de toutes les requêtes envoyées.

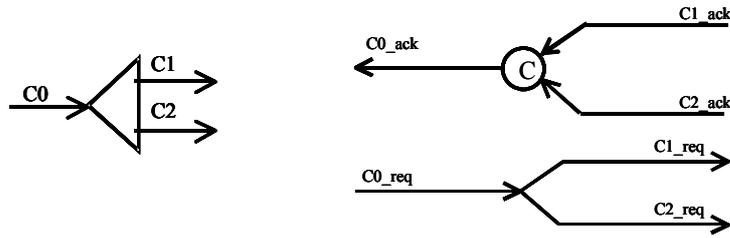


Figure 3-22 : Symbole et schéma de structure du duplicateur

Comme le montre la Figure 4-24, le duplicateur peut être implémenté grâce à une porte de Muller qui synchronise les acquittements des sorties.

3.3.7b Rendez-vous

Appelé aussi *Join*, il permet de générer une requête à partir de plusieurs requêtes. Dès qu'il reçoit l'acquiescement de sortie, il acquiesce toutes les requêtes en entrées.

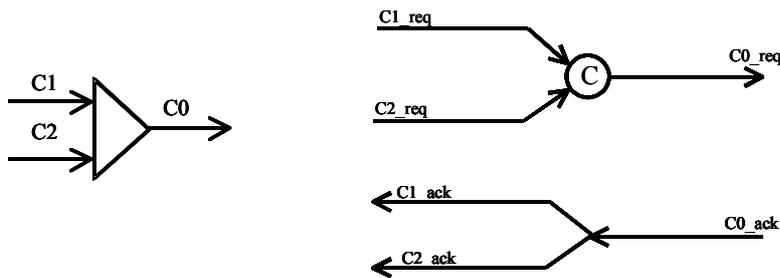


Figure 3-23 : Symbole et schéma de structure du rendez-vous

3.3.8 Latches vs Flip-flop

La Figure 3-24 présente l'architecture d'un *latch* et la Figure 3-25 celle d'une *Flip-flop*.

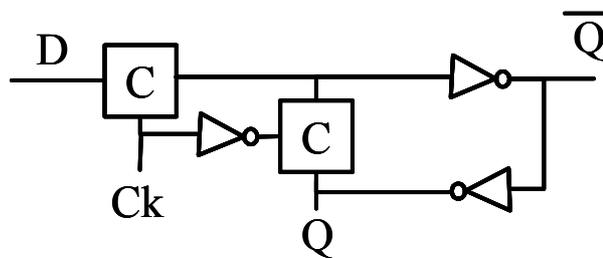


Figure 3-24 : Architecture d'un *latch*

Ces structures sont utilisées pour implémenter respectivement le processus à mémoire sensible au niveau du signal et celui sensible au front. La structure d'une *Flip-Flop* utilise 2 structures *latch*, donc prend environ 2 fois plus de surface.

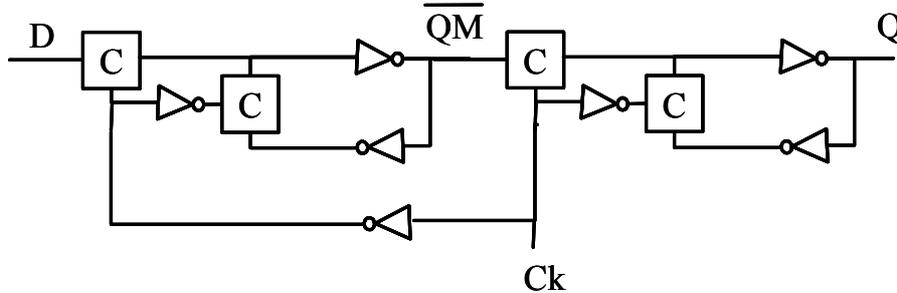


Figure 3-25 : Architecture d'une *flip flop*

Moins coûteuses en surface et plus adaptés par leur déclenchement au niveau d'un signal, les latches sont utilisées dans la conception des circuits asynchrones de type micropipeline. Le signal de déclenchement énable est géré alors par une partie de contrôle.

3.3.9 Délais

L'introduction de délais dans les fils, permet d'assurer la robustesse des circuits asynchrones face aux aléas. En particulier, pour la catégorie micropipeline; des délais sont insérés dans les communications. Leur valeur est supérieure à celle des parties combinatoires. Ceci assure la validité des données transmises. La Figure 3-26 représente le symbol du délai (asymétrique) utilisé dans cette thèse.



Figure 3-26 : Symbol du délai

3.3.9a Le délai classique

Le délai classique retarde le changement de la valeur du signal quelle que soit cette valeur. Sa structure est composée d'une mise en série de cellules délai de la technologie cible (Figure 3-27). Ce délai a pour avantage d'avoir une structure simple. Cependant son utilisation peut ralentir le fonctionnement du circuit inutilement. C'est le cas quand il est utilisé comme délai intrinsèque. La mise à 1 de la requête en entrée signale la validité des données qui entrent dans le contrôleur. Retarder la requête permet ainsi d'assurer la validité des données résultant de la partie combinatoire. Cependant, retarder la remise à zéro de la

requête s'avère inutile car la partie combinatoire n'effectue pas de calculs. Dans ce cas, il est préférable d'utiliser un délai asymétrique décrit dans le paragraphe suivant.

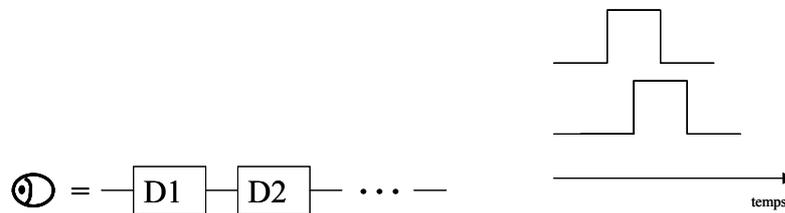


Figure 3-27 : Le délai classique et sa réponse

3.3.9b Le délai asymétrique

Ce délai permet de retarder différemment la mise à 1 et à 0 d'un signal. La Figure 3-28 représente une structure proposée dans [VIVE01].

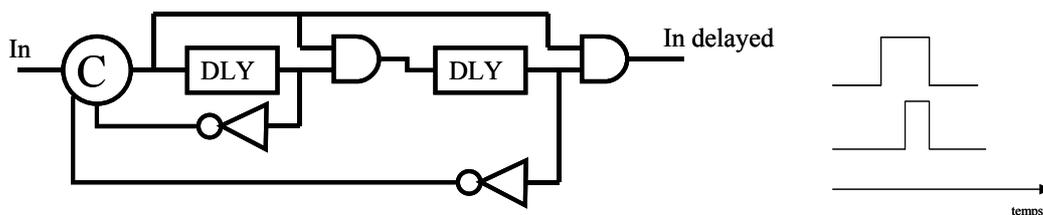


Figure 3-28 : Délai asymétrique et fonctionnement

Cette structure convient parfaitement au problème énoncé précédemment. Elle permet d'avoir une remise à 0 immédiate du signal de requête en entrée du contrôleur. L'utilisation de ce type de délai permet de ne pas ralentir inutilement le système mais il consomme plus et requiert plus de surface de silicium que le délai précédent.

3.4 Emergence des circuits asynchrones

Pourquoi les circuits asynchrones sont-ils attrayants ? Comment les introduire dans les applications actuelles ? Ce paragraphe tente de répondre à ces questions en listant d'une part les performances des circuits asynchrones et en abordant d'autre part les méthodes qui tentent d'insérer les circuits asynchrones dans les applications actuelles.

3.4.1 Performances des circuits asynchrones

3.4.1a Emission électromagnétique

Dans un circuit asynchrone, l'activité du courant dans le circuit est répartie grâce aux communications locales entre les opérateurs. Les forts appels de courant dus aux commutations simultanées provoquées par le signal d'horloge n'existent pas dans les circuits asynchrones, contrairement aux circuits synchrones.

Beaucoup d'expérimentations ont démontré la propriété de faible émission électromagnétique des circuits asynchrones par rapport aux circuits synchrones et sont décrites ci-après.

Dans [BERK99] K. V. Berkel et al. comparent le spectre en fréquence du*es microcontrôleurs 80c51 version synchrone et asynchrone [GAGE98]. Pour plusieurs harmoniques du spectre en fréquence de l'activité en courant du circuit, la différence entre les deux versions atteint les 30dB. Philips Semiconductor a exploité la qualité de faible émission électromagnétique du 80c51 asynchrone pour concevoir des pagers.

Un circuit DSP (*Digital Signal Processing*) asynchrone [PAVE98] a démontré sa faible émission électromagnétique par rapport à la version synchrone. Le spectre du courant du premier DSP fournit un pic de courant maximum 1.8 fois (2,5dB) moins élevé que le celui du synchrone.

AMULET2e [FURB99] est un système embarqué qui contient un processeur asynchrone compatible ARM 32-bit. Les mesures des émissions rayonnées sont conformes aux normes de compatibilité électromagnétique contrairement aux versions synchrones.

Par ailleurs, un convertisseur analogique numérique (CAN) asynchrone élaboré au sein du groupe Concurrent Integrated System a montré sa faible émission électromagnétique [ALLI03]. La Figure 3-29 montre les résultats de simulations électriques du courant consommé sur l'alimentation des deux convertisseurs (technologie 0.18micron STMicroelectronics). Le premier spectre correspond à l'activité du courant dans le CAN synchrone qui a été simulé après placement routage, donc tenant compte de l'activité de l'arbre d'horloge. Le second spectre correspond à l'activité du courant de l'implémentation du CAN qsynchrone. Du premier au second spectre, la diminution des amplitudes des harmoniques est significative et s'élève à environ 63% (4,3dB).

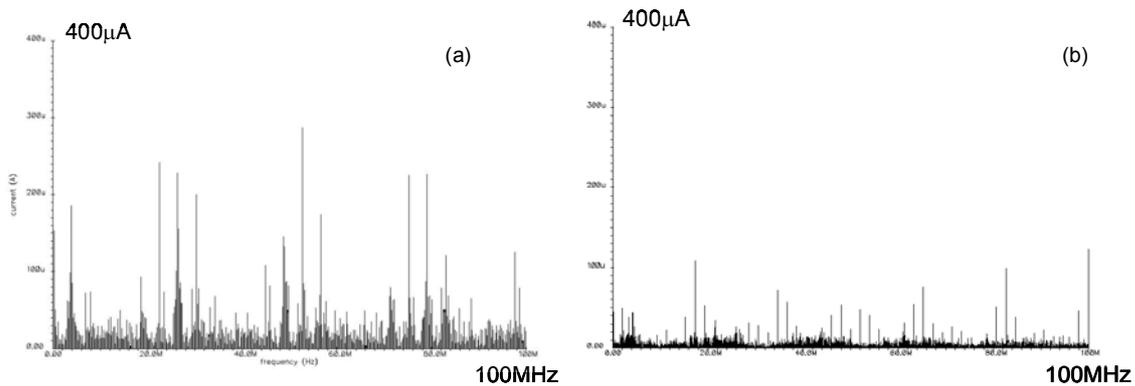


Figure 3-29 : Spectre de l'activité du courant dans un CAN synchrone et asynchrone

3.4.1b Signature électrique

En ce qui concerne le domaine de la sécurité, les circuits asynchrones sont attractifs par leur faible signature électrique [MOOR02]. Comme ils fonctionnent d'une manière flot de données et n'utilisent pas d'horloge globale, toute attaque par analyse de temps est quasiment impossible. L'analyse de la signature électromagnétique s'avère difficile car les circuits asynchrones ont un faible rayonnement électromagnétique dû à l'utilisation des communications de type poignée de main.

Par ailleurs, les circuits QDI sont très intéressants par leur caractéristique d'insensibilité aux délais où le fonctionnement correct ne dépend pas des délais des éléments. L'analyse électrique du microcontrôleur Mica [RENA02] a prouvé que les circuits QDI sont résistants aux attaques DPA (*Differential Power Analysis*) (Figure 3-30).

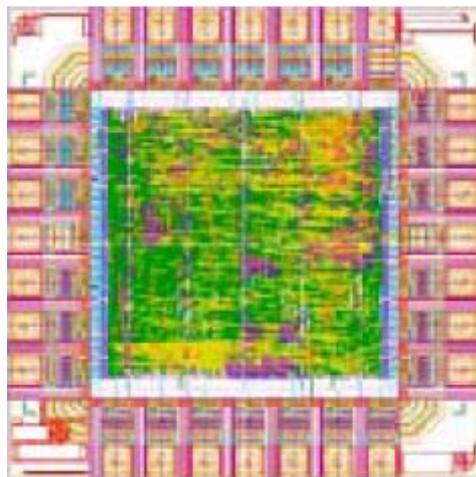


Figure 3-30 : Cryptoprocresseur QDI [BOUE02]

Les travaux du groupe CIS vont plus loin en exploitant cette propriété des circuits QDI pour explorer les architectures et élaborer des méthodes dédiées à la cryptographie [BOUE04].

3.4.1c Consommation

Lorsque l'implémentation asynchrone est choisie plutôt que la solution synchrone, la consommation du circuit liée au signal d'horloge (arbre d'horloge, bascules) est supprimée.

Chaque élément consomme uniquement lorsqu'il est sollicité par une requête. Ainsi, un opérateur ne consommera pas inutilement comme c'est le cas dans les circuits synchrones. Cependant, les circuits asynchrones n'ont pas prouvé leur qualité de faible consommation face aux circuits synchrones. En effet, selon les implémentations des contrôles locaux, la consommation peut être supérieure. Ainsi des méthodes sont développées au sein du groupe *Concurrent Integrated Système* pour réduire la consommation des circuits asynchrones : au niveau architecture [SLIM03] et au niveau de la gestion du système d'exploitation [ESSA02].

3.4.1d Surface

Dans un circuit asynchrone, toute la surface occupée par le signal d'horloge n'existe pas, mais le contrôle des communications dans les canaux occupe une place considérable car les contrôleurs doivent gérer d'une part, les protocoles de requête - acquittement, et, d'autre part, la validité des données. L'utilisation de cellules standard pour l'implémentation des circuits asynchrones comme c'est le cas dans le chapitre 5 a démontré que les circuits asynchrones occupent beaucoup plus de surface que les circuits synchrones. Ainsi, il est nécessaire, voir indispensable de disposer de cellules complexes réalisées et optimisées en surface pour les circuits asynchrones. Une partie du travail mené par le groupe CIS se focalise sur l'optimisation de ces cellules [MAUR03] [FRAG03].

3.4.1e Dissipation thermique

Avec l'évolution des technologies submicroniques, la puissance dissipée par les circuits intégrés ne cesse d'augmenter [TOKI02]. Ce problème peut être remarqué en particulier sur les processeurs où des moyens de refroidissement sont mis en oeuvre : dissipateurs, ventilateurs. Les circuits intégrés sont destinés à fonctionner dans une certaine gamme de température, et, au-delà de celle ci, les risques de dysfonctionnement existent. Ainsi, la dissipation thermique d'un circuit est aujourd'hui un critère qu'il faut prendre en compte et certains circuits dépassent les 100Watts. Du fait de la répartition de leurs communications, les circuits asynchrones pourraient être une alternative à ce problème. Cependant, peu d'études ont été effectuées à ce sujet.

3.4.1f Vitesse

Dans un circuit asynchrone, lorsqu'un opérateur fini son traitement, il en informe immédiatement l'opérateur en aval. Ainsi, la vitesse du circuit correspond à la somme des opérations à effectuer. Le temps de réponse du circuit n'aura pas comme contrainte d'être un multiple de la période d'horloge. Cette performance en vitesse peut être augmentée par le type d'architectures ou le protocole de communication utilisé, par exemple un pipeline élastique peut utiliser un protocole découplé.

3.4.2 Concevoir et produire des circuits asynchrones

Pour pouvoir produire des circuits asynchrones, la conception de ce type de circuits doit être assistée et automatisée grâce à des outils tels qu'il en existe pour les circuits synchrones. Ce paragraphe décrit les méthodes élaborées et employées pour les circuits asynchrones ainsi que les outils dédiés en cours d'élaboration et en constante amélioration.

Dans l'attente de la maturité de tels outils, d'autres solutions proposent de mélanger le style de conception synchrone et asynchrone (3.4.2b) ou de relâcher les synchronisations dans les circuits synchrones afin d'en faire des circuits asynchrones (3.4.2c). Dans le chapitre suivant, une alternative similaire à cette dernière solution est proposée.

3.4.2a Méthodes et outils de conception dédiés aux circuits asynchrones

Parmi les nombreux outils et les méthodes de conception dédiés aux circuits asynchrones, les principaux sont les suivants [SPAR01] : Tangram, Balsa [BARD00], Petrify et Minimalist [NOWI02].

L'outil TAST (*TIMA Asynchronous Synthesis Tool*) est une suite d'outils développée par le groupe CIS (*Concurrent Integrated System*) au sein du laboratoire TIMA [TIMA]. L'organigramme du flot de conception est donné Figure 3-31.

A partir d'une description haut niveau du circuit, le compilateur permet de produire différentes catégories de circuits. Ces circuits sont modélisés en utilisant un langage CHP (*Communicating Hardware Process*) dont une description est donnée dans [VIVE01]. C'est un langage proche du CSP (*Communicating Sequential Process*) qui est très utilisé pour décrire le fonctionnement des circuits asynchrones. Ce dernier gère les compositions parallèles et la notion de canal. Les CHP est quand à lui enrichi avec une syntaxe utile pour la simulation et la synthèse. Il est à noter que le compilateur se veut général [DINH02] et permettra à l'avenir d'accepter plusieurs langages en entrée (VHDL, CHP).

Les descriptions du circuit en entrée sont d'abord analysées puis contrôlées sur les caractéristiques suivantes : affectations concurrentes des variables, les déclarations de canaux, la cohérence des actions de communication, la concordance des types, etc.. Si cette

phase de vérification ne montre pas d'erreur, alors la description est transformée en une forme intermédiaire basée sur les Réseaux de Pétri et les graphes de flots de données.

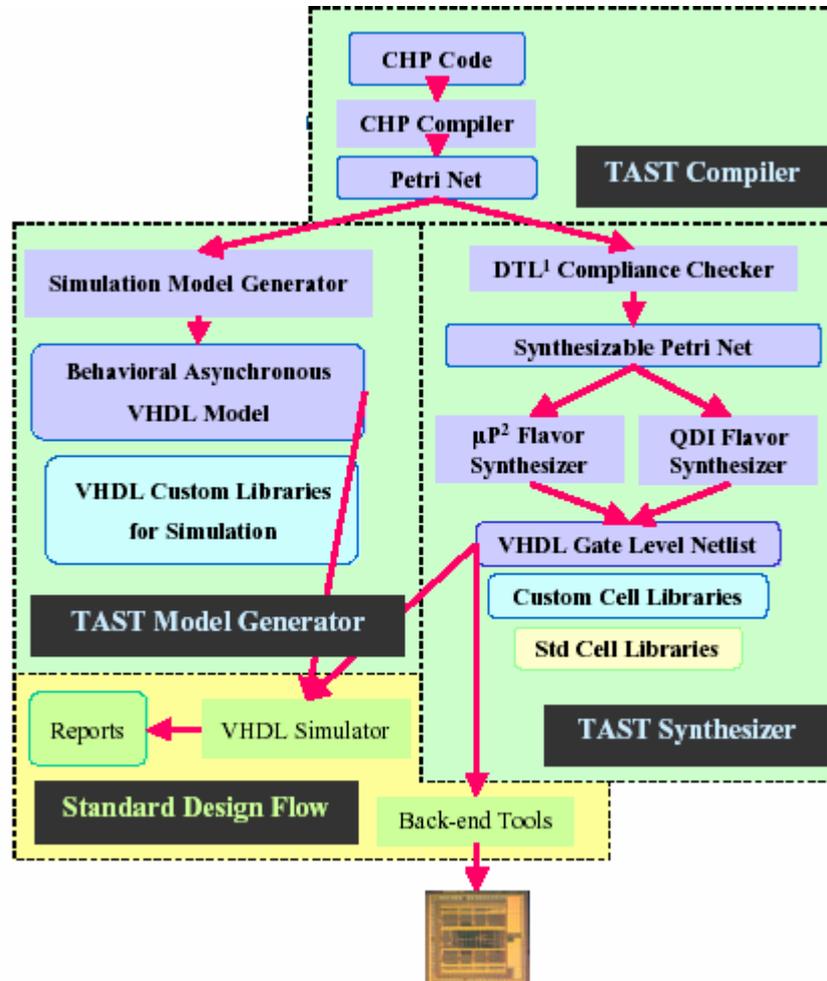


Figure 3-31 : Flot TAST [RENA03]

À ce stade, le concepteur peut choisir de produire une description fonctionnelle en VHDL du modèle qui peut être simulée afin de vérifier la fonctionnalité du circuit. Une option permet au concepteur de produire une description VHDL au niveau registre du circuit, qui peut être employée pour viser des technologies ASICs ou FPGAs à l'aide des outils standard. Le concepteur peut également décider de viser les circuits asynchrones. Le compilateur analyse alors la forme interne pour vérifier si les modèles peuvent être implémentés sur les circuits numériques asynchrones ou pas. Pour être acceptés pour ce type de synthèse, les descriptions CHP doivent respecter quelques règles de programmation qui assurent le ciblage (*mapping*) correct de la forme interne sur les circuits asynchrones. Si c'est le cas, alors la catégorie de circuit micropipeline ou QDI est

choisie. L'outil génère ainsi un fichier appelé *netlist* de portes et qui peut être simulé en utilisant les outils standard EDA (*Electronic Design Automation*).

Mais TAST n'est pas seulement un outil d'aide à la synthèse des circuits. Il permettra à l'avenir d'améliorer les performances des circuits obtenus : la surface [FRAG02] la consommation [SLIM03][ESSA02], l'émission électromagnétique [PANY02], la sécurité [BOUE02], leur durcissement face aux radiations [MONN04]. La Figure 3-32 montre l'envergure du projet TAST et les possibilités d'aide à la conception qui sont offertes par TAST.

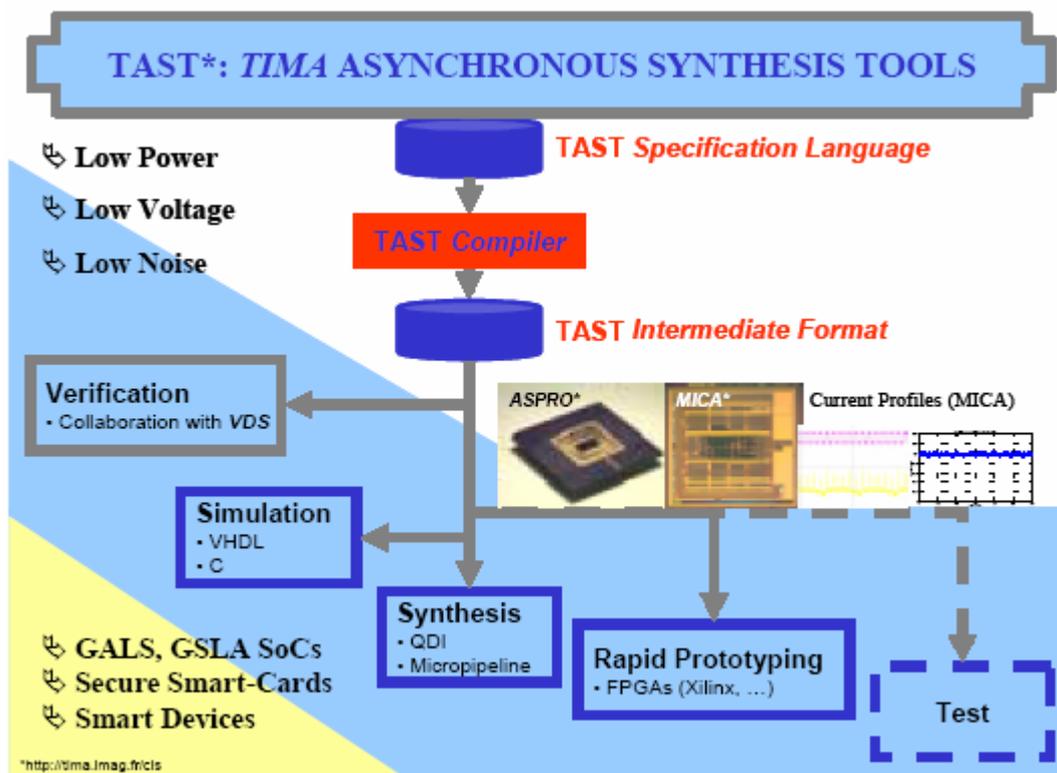


Figure 3-32 : Le projet TAST

Cette thèse se concentre sur la partie amélioration des performances en émission électromagnétique et propose des méthodes. Ces méthodes pourront être automatisées et intégrées dans le flot TAST.

3.4.2b GALS, GSLA

Pour profiter des avantages des circuits asynchrones sans toutefois ralentir le temps de conception du circuit, une solution consiste à utiliser des circuits Globalement Asynchrones et Localement Synchrones (GALS), ou Globalement Synchrones et

Localements Asynchrones (GSLA) [MEIN98][HEAT02]. La principale difficulté réside dans la synchronisation des parties entre elles.

3.4.2c Circuits Désynchronisés

La dé-synchronisation consiste à supprimer la synchronisation globale d'un circuit synchrone par des communications locales du type « poignée de mains ». Ceci permet dans un premier temps de disposer de circuits asynchrones (non optimisés) mais de performances intéressantes comme la faible émission électromagnétique. Beaucoup d'études sont menées à ce sujet.

Dans [CORT03] et [BLUN04], les auteurs proposent d'éliminer le signal d'horloge. Ils appliquent pour cela 2 procédures différentes selon le type de structure étudiée :

- La dé-synchronisation des registres consiste à remplacer l'horloge par des communications locales. Ces communications sont gérées par des contrôleurs commandant des *latches*. Les contrôleurs respectent le temps de calcul des parties combinatoires.

- La dé-synchronisation de la partie combinatoire consiste à implémenter celle ci en ajoutant un signal d'achèvement du calcul. Ce signal est en réalité un acquittement informant que la partie donnée est prête. Ce signal est généré par un réseau de contrôle composé de cellules muller et de portes logiques. L'implémentation de ce réseau est insensible aux délais. La surface du circuit est donc augmentée et des procédures d'optimisation doivent alors être envisagées.

La méthode proposée permet de dé-synchroniser des parties de circuits en vue d'obtenir des GALS ou GLSA.

Dans la même optique, [SOTI03] propose d'implémenter des circuits asynchrones en utilisant des outils EDA et des spécifications synchrones au niveau portes :

- Les pipelines synchrones sont remplacés par des pipelines asynchrones. La logique combinatoire et les délais dans les bascules sont respectés par des délais rajoutés dans les communications. Il est cependant précisé que la version synchrone est plus rapide que la version asynchrone car cette dernière nécessite l'ajout des délais dans les communications. Une optimisation en vitesse consiste alors à remplacer les bascules *Flip-Flops* par des *latches*.

Les circuits de contrôle sont implémentés en machine à état asynchrone par l'outil AFSMGEN Pipefitter [BLUN03].

C. Sotiriou, propose d'augmenter le niveau de granularité du pipeline en plusieurs modules en vue d'augmenter la vitesse en découplant les étages. Son approche a été validée par plusieurs implémentations sur des circuits DES (*Data Encryption Standard*).

[BREJ03] propose une technique de translation des circuits synchrones en circuits asynchrones où chaque élément synchrone est remplacé par un circuit équivalent

asynchrone grâce à l'utilisation d'une table de conversion. Cette méthode ne prend pas en considération les dépendances bouclées dans le circuit ainsi que son initialisation..

Enfin, un travail intéressant sur la dé-synchronisation des pipelines est à remarquer. Décrit dans [KOL97], ce travail compare des structures de pipelines asynchrones équivalentes à une structure synchrone et aborde la question des Fourches et *Join* dans la dé-synchronisation. En particulier, les auteurs proposent de remplacer les bascules par des couples de *latches* maître-esclaves.

Ainsi les méthodes de dé-synchronisation proposées s'adressent à la description au niveau portes des circuits synchrones pour obtenir des circuits asynchrones. Elles ne sont pas générales car elles ne traitent ni des boucles dans le circuit, ni de son initialisation.

3.5 Conclusion

Contrairement aux circuits synchrones, les circuits asynchrones utilisent une synchronisation locale pour assurer la séquence des traitements. Cette synchronisation est gérée par des protocoles de communication de type poignée de main dans des canaux de communication.

La principale difficulté lors de l'implémentation de tels circuits est la gestion des aléas. En effet, tout changement de valeur non désiré qui se produit peut être pris en considération et générer un traitement erroné. Ainsi les circuits asynchrones peuvent être classés en différentes catégories plus ou moins robustes face aux aléas selon les hypothèses temporelles qui sont faites.

TAST, l'environnement de synthèse des circuits asynchrones développé par le groupe CIS au sein du laboratoire TIMA, permet de produire deux catégories de circuits asynchrones (QDI et micropipeline) à partir d'une description haut niveau en CHP du circuit. Mais son but est aussi de proposer une aide à l'amélioration des performances des circuits asynchrones. En particulier, cette thèse exploite la qualité de faible émission électromagnétique des circuits asynchrones et propose une méthode d'amélioration de cette propriété (Chapitre 5).

La répartition de l'activité du courant est naturelle dans un circuit asynchrone et de plus le signal d'horloge n'existe pas. Les variations brusques de courant simultanées sont réduites par rapport aux circuits synchrones, ce qui en fait des circuits à faible émission électromagnétique naturellement.

CHAPITRE 4 De la description RTL au circuit asynchrone : *Asynchronisation* ou De-synchronisation

4.1 Introduction

Le style de conception synchrone s'est imposé depuis l'apparition des circuits intégrés car pour assurer la séquence des traitements et la validité des données, le choix le plus simple était l'utilisation d'un signal de contrôle global : l'horloge. Les outils pour la conception synchrone (outils EDA, de l'anglais *Electronic Design Automation*) se sont alors développés depuis plus de dix ans conjointement avec l'évolution des méthodes de conceptions dédiées à ce type de circuits.

Mais, aujourd'hui par les ondes électromagnétiques qu'il génère, le signal d'horloge est devenu un obstacle à la compatibilité du circuit avec son environnement. Comme le décrit le chapitre 3, les circuits sans horloges sont une alternative pour répondre aux contraintes de la compatibilité électromagnétique. Cependant leur utilisation est freinée par le manque d'outils dédiés à ce type de circuit et l'inexpérience des concepteurs dans ce domaine. Pour profiter des avantages des circuits asynchrones sans toutefois consacrer des temps excessifs pour leur conception, des solutions décrites dans le chapitre précédent ont été proposées :

- Les circuits Globalement Synchrones et Localement Asynchrones (GSLA)
- Les circuits Globalement Asynchrones et Localement Synchrones (GALS)
- Les circuits dé-synchronisés

La méthode présentée dans le chapitre suivant exploite cette qualité de faible émission électromagnétique des circuits asynchrones en proposant une méthode similaire à la dé-synchronisation (paragraphe 3.4.2c) et dont le but est de remplacer un circuit synchrone par son équivalent asynchrone. L'approche proposée par cette thèse est complète et plus performante que celles précédemment proposées dans le paragraphe 3.4.2c. En particulier, l'analyse et les solutions sont exhaustives : elles traitent aussi bien des structures linéaires que des boucles, de la compatibilité de cycle et de l'initialisation du circuit. Elle est par ailleurs automatisable.

Le paragraphe 4.2 décrit le contexte d'élaboration de la méthode. Les concepts de fonctionnement synchrones sont étudiés paragraphe 4.3. La méthode *Asynchronisation* qui résulte de cette étude est présentée à travers les paragraphes 4.4, 4.5, 4.6, 4.7 et 4.8. L'implémentation d'un filtre à Réponse Impulsionnelle Finie (RIF) présenté au paragraphe 4.9 permet de valider la méthode *Asynchronisation*.

4.2 Contexte

4.2.1 Principe

Le principe de la méthode proposée est le suivant : adresser une description du circuit synchrone existante, supprimer le signal d'horloge et le remplacer par des communications locales de type poignées de main. La séquence des traitements des ressources étant gérée par les communications locales, l'activité du courant est répartie dans le circuit : les variations de courant simultanées sont réduites par rapport aux circuits synchrones. Les émissions électromagnétiques dues à l'horloge sont ainsi diminuées.

4.2.2 Contraintes

La méthode a été élaborée pour disposer de circuits asynchrones tout en répondant aux contraintes suivantes :

- Circonscrire au maximum le temps de conception des circuits asynchrones :
 - en appliquant une procédure quasi-générale et répétitive
 - en utilisant les outils standard lorsque cela est possible.
- Assurer la cycle compatibilité du circuit afin qu'il convienne aux applications
 - en exploitant les avantages des communications locales pour la faible émission électromagnétique
 - en ayant une possibilité de relâcher le fonctionnement du circuit afin qu'il ne soit pas cadencé par l'horloge. Le circuit pourra être utilisé dans un environnement où la cycle compatibilité n'est pas demandée (par exemple un environnement asynchrone)

4.2.3 Choix de la méthode

Le tableau suivant présente l'ensemble des choix qui ont été faits pour la méthode et dont les raisons sont détaillées ci-après.

Cette méthode va plus loin que [CORT03][SOTI03][BREJ03][KOL97], car elle ajoute l'initialisation, la gestion des boucles et la contrainte de cycle compatibilité du circuit. De plus, elle adresse le circuit synchrone à sa description RTL afin d'être automatisable.

<i>Choix</i>	<i>Raisons</i>
Description au niveau RTL (<i>Register Transfer Level</i>) du circuit synchrone	Cycle compatibilité Simplicité et répétition de la méthode Réduction du temps de conception
Approche par graphe de dépendance	Modularité Garantie du fonctionnement Simplicité et répétition de la méthode
Description du circuit obtenu dans un langage standard	Utilisation des outils standard Validation dans l'environnement synchrone Optimisations

Tableau 4-1: Choix effectués pour l'asynchronisation

4.2.3a Description du circuit synchrone

La description du circuit synchrone peut être adressée à différents niveaux d'abstraction : comportemental, RTL, porte... Elle est ici adressée au niveau RTL pour les raisons décrites ci-après.

Garantie de compatibilité cycle à cycle du circuit

La description RTL répond au cahier des charges du fonctionnement global du circuit pour une/des application(s) donnée(s). Si les séquences des traitements dans le circuit sont maintenues cycle à cycle, alors le circuit *asynchronisé* obtenu possède un fonctionnement identique qui répond au même cahier des charges. Il sera vu dans le paragraphe 4.5.3a comment le fonctionnement cycle à cycle du circuit peut être conservé.

Simplicité et répétition de la méthode

Au niveau RTL, les ressources gérées par le signal d'horloge sont déterminées ainsi que la séquence de leur traitement. Ceci facilite l'application de la méthode dont le but est de remplacer le signal d'horloge par des communications locales. La procédure est simple : remplacer des ressources ou des ensembles de ressources synchrones par des ressources asynchrones ayant le même fonctionnement.

Réduction du temps de conception

Les outils de synthèse dédiés aux circuits synchrones ne possèdent pas la notion de canaux. La synthèse comportementale d'un circuit asynchrone doit donc être effectuée à la main ce qui rallonge le temps de conception. Partir d'une description RTL d'un circuit synchrone permet d'obtenir une description niveau "RTL" du circuit *asynchronisé* dans laquelle les canaux sont implémentés rapidement ce qui circonscrit le temps de conception.

4.2.3b Approche par graphe des dépendances

Un graphe des dépendances entre les registres peut être utilisé pour les raisons suivantes :

Modularité

L'utilisation des blocs de propriété intellectuelle IP (*Intellectual Property*) ou la limitation du temps de conception imposent l'*asynchronisation* partielle du système/circuit. Celui-ci n'est donc pas *asynchronisé* totalement. Qu'elle s'opère à plat ou d'une manière hiérarchique, la méthode permet d'*asynchroniser* certains blocs sans pour autant modifier le reste du circuit.

Garantie du bon fonctionnement

Le bon fonctionnement du système est garanti par l'analyse des dépendances dans le circuit (voir paragraphe 4.3.4). Le signal d'horloge peut être supprimé sur certains blocs d'une façon incrémentale, et la validation fonctionnelle peut se faire au fur à mesure.

Compatibilité cycle à cycle

L'approche par graphe de dépendances des processus à mémoire représente le comportement du circuit : les interactions entre les blocs et la séquence des traitements. L'*Asynchronisation* s'effectue en conservant ces caractéristiques et permet d'obtenir un circuit cycle compatible.

Automatisation

Le graphe de dépendance peut être déduit de la description au niveau registres du circuit synchrone. Donc tôt dans la conception. Il peut être l'entrée d'un outil automatique.

4.2.3c Utilisation d'un langage standard (VHDL ou Verilog)

La description du circuit *asynchronisé* est écrite dans un langage standard. Ceci permet d'utiliser les outils standard, de valider le circuit dans l'environnement synchrone ainsi que d'optimiser éventuellement l'architecture.

Utilisation des outils standards

Le circuit peut fonctionner dans un mode synchrone car il peut être géré par le signal d'horloge. Ce fonctionnement pourra être donc simulé par les outils EDA standard. Une fois celui-ci validé, la synthèse structurelle à l'aide des outils standard pourra être effectuée.

Validation dans l'environnement synchrone

Décrit dans le même langage que le circuit synchrone, le circuit asynchrone pourra également être simulé dans l'environnement de test fonctionnel du circuit synchrone.

Optimisation éventuelle de l'architecture

Du point de vue du concepteur, les briques asynchrones sont considérées comme des boîtes noires dont il ne connaît pas l'architecture mais la fonction. Comme la procédure d'*asynchronisation* appliquée est simple, le circuit obtenu n'est pas optimisé en surface ou en vitesse, mais il est décrit dans un langage standard, il peut donc être facilement repris et optimisé. Cependant, avec quelques connaissances en conception asynchrone, le concepteur peut explorer et modifier ces structures afin d'améliorer s'il le souhaite et si possible les performances en consommation, en émission électromagnétique ou en surface de son circuit.

4.2.4 Introduction de la méthode dans le flot de conception

La Figure 4-1 résume le flot de conception d'un circuit synchrone et le rôle que tient l'*Asynchronisation* dans celui-ci en vue d'obtenir un circuit *asynchronisé*.

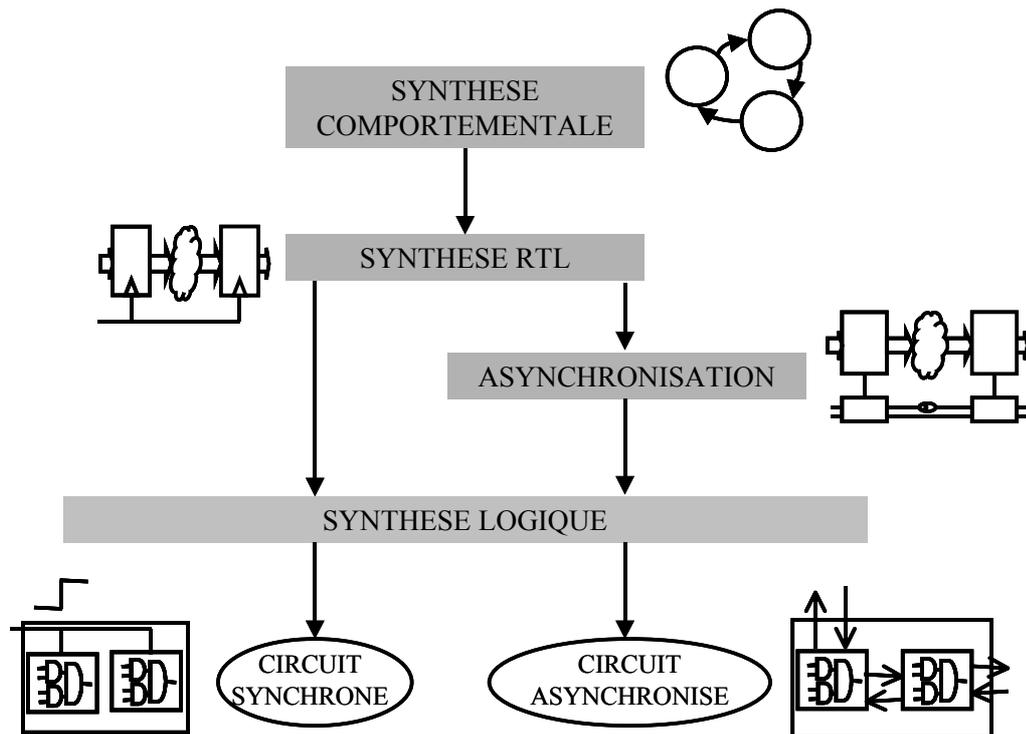


Figure 4-1 : Etape de l'*asynchronisation* dans le flot de conception

L'*asynchronisation* est une étape supplémentaire dans la conception des circuits. Elle se situe entre la synthèse RTL et la synthèse logique. Les délais de conception ajoutés par l'application de cette méthode incluent les délais d'une synthèse RTL auxquels il faut ajouter le temps d'analyse du code RTL d'origine et celui de la vérification par simulation fonctionnelle de la description obtenue.

4.2.5 Elaboration de la méthode

La procédure élaborée consiste à remplacer toutes les briques synchrones dans la description RTL du circuit pour les remplacer par des briques asynchrones dont le comportement est identique.

Le plan de l'étude est le suivant :

- Comprendre les concepts du fonctionnement synchrone pour identifier les analogies synchrones et asynchrones
- Proposer une procédure d'*asynchronisation*

4.3 Concepts du fonctionnement synchrone

Ce paragraphe tente de réunir les concepts du fonctionnement synchrone afin de trouver les structures asynchrones équivalentes. Les signaux d'entrée et de sortie du circuit sont synchronisés.

4.3.1 Briques de bases : les processus

L'objet fondamental manipulé par le simulateur est ici considéré comme brique de base des structures rencontrées. Selon la norme VHDL [AIRI98]:

- un processus ne contient que des instructions séquentielles
- toute instruction concurrente peut toujours être traduite par un processus équivalent, elle sera donc appelée par abus de langage "processus".
- un processus vit toujours. Sa durée de vie est celle de la simulation.

4.3.1a Processus combinatoire

Les sorties d'un processus combinatoire (Figure 4-2) dépendent uniquement des données en entrée. Les données de sorties sont calculées dès qu'il y a un changement sur son/ses entrée(s).

L'assignation d'une donnée *data_in* à une donnée *data_out* s'écrit :

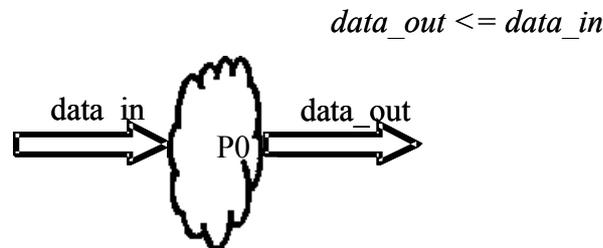


Figure 4-2 : Processus combinatoire

La condition de bon fonctionnement d'un processus combinatoire est la suivante : les entrées doivent se présenter suffisamment espacées dans le temps pour lui laisser le temps d'effectuer son calcul. Si une erreur arrive à l'entrée d'un processus combinatoire, alors elle provoque une sortie erronée et une consommation inutile de courant pour son traitement.

4.3.1b Processus à mémoire

Ses sorties du processus à mémoire (Figure 4-3) dépendent de ses données en entrée et du temps. Deux types peuvent être rencontrés : la logique sensible aux événements de l'horloge (*Flip Flop*), la logique sensible au niveau de l'horloge (*Latch*) décrites dans la Figure 4-4.

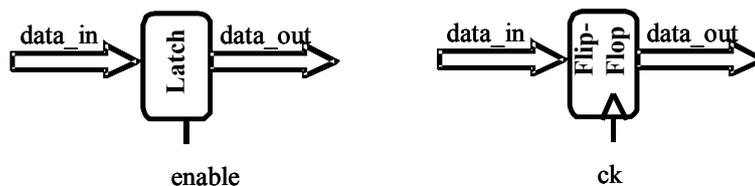


Figure 4-3 : Processus à mémoire

<pre> <i>process (<Clk>)</i> <déclaration> <i>begin</i> <i>if (<Clk='1' and Clk'event>) then</i> $Q \leq D$ <i>else</i> $Q \leq Q$ <i>end if;</i> <i>end process;</i> </pre>	<pre> <i>process (<E, enable>)</i> <déclaration> <i>begin</i> <i>if (<enable=passant>) then</i> $Q \leq E$ <i>else</i> $Q \leq Q$ <i>end if;</i> <i>end process;</i> </pre>
<p>processus à mémoire sensible aux évènements (FlipFlop)</p>	<p>processus à mémoire sensible aux niveaux (Latche)</p>

Figure 4-4 : Les différents types de processus à mémoire

Dans la Figure 4-4, la description VHDL du processus à mémoire sensible aux évènements indique que la logique est sensible au front montant de l'horloge (*if Clk='1'and Clk'event*). La logique peut être aussi sensible au front descendant de l'horloge (dans le cas, ceci s'écrit *if Clk='0'and Clk'event*)

La condition de bon fonctionnement d'un processus à mémoire sensible au front d'un signal est la suivante : les entrées doivent changer suffisamment à l'avance afin de respecter le temps d'établissement du régime (*setup*) et doivent rester stables assez longtemps pour respecter le temps de maintien (*hold*).

Les conditions de bon fonctionnement d'un processus à mémoire sensible au niveau d'un signal sont les suivantes :

- les données en entrée doivent être stables suffisamment à l'avance avant que les signaux *enable* soit actifs,
- les données en entrées doivent être stables assez longtemps pour respecter la latence de la partie combinatoire.

Dans un circuit synchrone, le flot de données est régulé par un signal régulier appelé "horloge", et, selon la convention utilisée à chaque front montant ou descendant de celui-ci, les données entrantes ou sortantes des processus combinatoires sont garanties valides. Le processus à mémoire sensible au front d'horloge – appelé processus synchrone dans le reste de ce mémoire- est utilisé en amont ou en aval des processus combinatoires. Il mémorise ses sorties lorsqu'il reçoit le signal d'horloge et permet ainsi la gestion du flux des données. C'est donc à ce type de processus que l'étude s'intéresse et cherche à remplacer.

4.3.2 Signaux d'horloge

L'horloge est un signal qui permet de cadencer la séquence des actions en commandant les processus séquentiels. Les processus sensibles au front montant/descendant de la même horloge ont un fonctionnement dit concurrent.

4.3.3 Signaux d'initialisation

En général, dans un circuit synchrone, l'initialisation des processus est synchrone (Figure 4-5). Ceci est dû principalement aux règles d'écriture pour la synthèse [KEAT98]. Cependant elle peut être aussi asynchrone (comme c'est le cas d'une interruption). Dans le cas où l'initialisation est synchrone, le *reset* est alors considéré comme un signal de donnée. Dans le cas où l'initialisation est asynchrone, il est alors considéré comme un signal spécial tel que les signaux d'horloge. Cependant, que l'initialisation soit synchrone ou asynchrone, elle adresse les processus concurremment.

<pre>process (<signal_horloge>, <reset>) <déclaration> begin if (<reset>) then <séquence de traitements> elsif (<front_d'horloge>) then <séquence de traitements> end if; end process;</pre>	<pre>process (<front_d'horloge>, <reset>) <déclaration> begin if (<front_d'horloge >) then if (<reset >) then <séquence de traitements> else <séquence de traitements> end if; end if; end process;</pre>
reset asynchrone	reset synchrone

Figure 4-5 : *Reset* asynchrone et synchrone

4.3.4 Compositions

Un circuit synchrone est composé de processus à mémoire et de processus combinatoires (Figure 4-6). La logique combinatoire appelée aussi *glue logic* est contenue dans la logique séquentielle sensible aux niveaux, aux événements des signaux et à l'horloge (*Flips-Flops*).

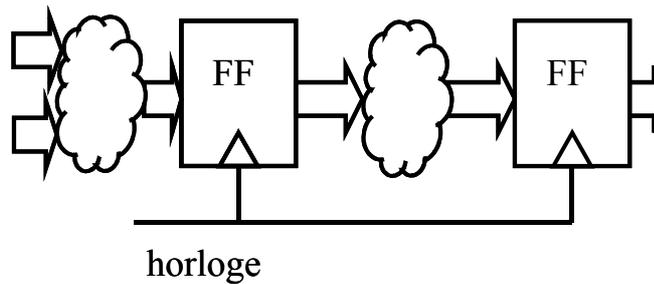


Figure 4-6 : Structure synchrone

L'étude s'intéresse aux compositions des processus à mémoire car ils gèrent la séquence des opérations dans le circuit par l'utilisation du signal d'horloge qui doit être supprimé.

Les structures de processus synchrones de bases rencontrés dans le circuit sont énumérées et modélisées sous forme de système d'équations décrits ci après. Ces structures suivent les directives écrites pour la synthèse des circuits [IEEE10]. Il est aussi supposé que les signaux entrant dans les structures sont valides à chaque front montant/descendant des horloges. Il en est de même pour les signaux qui sortent des structures.

4.3.4a Dépendance

Dans cette composition, les données vont d'un processus synchrone à un autre dans un seul sens. La dépendance est dite simple lorsque les données entrantes d'un processus sont les sorties d'un seul processus. La dépendance est dite multiple lorsqu'elles proviennent de plusieurs processus. Les données en sortie des processus synchrones avals sont les images des données des processus synchrones en amont dont ils dépendent.

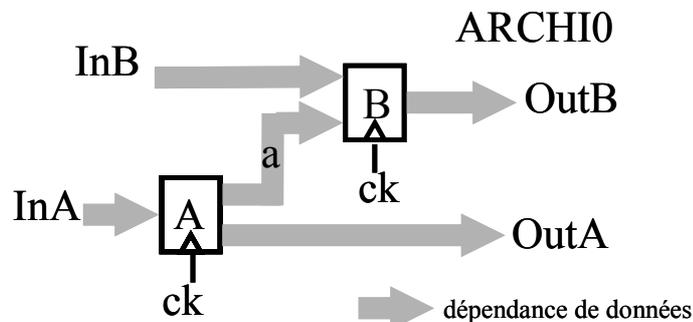


Figure 4-7 : Dépendance simple a entre 2 processus

Dans l'architecture *ARCHIO* (Figure 4-7), deux processus synchrones sont représentés. Les dépendances s'effectuent à sens unique. On rencontre ici 2 types de dépendances, les dépendances simples comme celle de "A" qui ne dépend que de InA et les dépendances multiples (cas de B dépendant de InB venant de l'extérieur et de a venant de A).

Soit f_A la fonction exécutée par A :

$$\begin{cases} OutA(0) = a_0 \\ OutA(t) = f_A(InA(t-1)) \end{cases}$$

Système Équation 4-1

ou $InA(t-1)$ est l'entrée fournie au front d'horloge précédent et f_B la fonction exécutée par B :

$$\begin{cases} OutB(0) = b_0 \\ OutB(t) = f_B(InB(t-1)) \end{cases}$$

Système Équation 4-2

La structure *ARCHI0* peut donc être modélisée par le système d'équations suivant :

$$\begin{cases} OutA(t) = a(t) = f_A(InA(t-1)) \\ OutB(t) = f_B(InB(t-1), a(t-1)) \\ OutA(0) = a_0 \\ OutB(0) = b_0 \end{cases}$$

Système Équation 4-3

ou encore :

$$\begin{cases} OutA(t) = f_A(InA(t-1)) \\ OutB(t) = f_B(InB(t-1), f_A(InA(t-2))) \\ OutA(0) = a_0 \\ OutB(0) = b_0 \end{cases}$$

Système Équation 4-4

4.3.4b Interdépendance

Dans cette composition (exemple de la Figure 4-8), d'un processus synchrone à un autre, les données vont dans les deux sens.

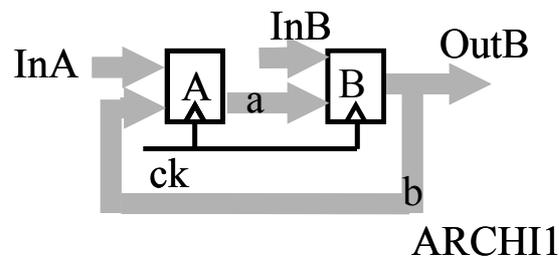


Figure 4-8: A et B sont interdépendants.

Les composantes à mémoire sont activées au front montant ou descendant de l'horloge (ceci dépend de la convention). Les données sont donc prises en compte au même moment. Ainsi, à l'instant t , les données en sorties de processus synchrones interdépendants sont les images des données fournies par les mêmes processus à $t-1$. Le détail le fonctionnement de l'interdépendance de la Figure 4-8 est détaillé ci dessous.

Soit f_A la fonction exécutée par A :

$$\begin{cases} OutA(0) = a_0 \\ OutA(t) = a(t) = f_A(InA(t-1), b(t-1)) \end{cases}$$

Système Équation 4-5

et f_B la fonction exécutée par B :

$$\begin{cases} OutB(0) = b_0 \\ OutB(t) = b(t) = f_B(InB(t-1), a(t-1)) \end{cases}$$

Système Équation 4-6

La structure *ARCHII* peut donc être modélisée par le système d'équations suivant :

$$\begin{cases} OutA(t) = a(t) = f_A(InA(t-1), b(t-1)) \\ OutB(t) = b(t) = f_B(InB(t-1), a(t-1)) \\ OutA(0) = a_0 \\ OutB(0) = b_0 \end{cases}$$

Système Équation 4-7

$$\text{soit } \begin{cases} OutA(t) = a(t) = f_A(InA(t-1), b(t-1)) \\ OutB(t) = b(t) = f_B(InB(t-1), InA(t-2), b(t-2)) \\ OutA(0) = a_0 \\ OutB(0) = b_0 \end{cases}$$

Système Équation 4-8

L'étude des structures synchrones permet à présent de définir l'*asynchronisation* détaillée ci après.

4.4 Architecture générale pour l'*Asynchronisation*

Le principe de l'*asynchronisation* est de remplacer un circuit ou une partie de circuit par son équivalent asynchrone (4.2.1). Du point de vue architecture, cette méthode s'adresse donc directement aux processus synchrones (4.3.1b) et considère les processus combinatoires comme transparents. Comme le montre la Figure 4-9, chaque opérateur (processus) synchrone est

remplacé par un opérateur (processus) asynchrone équivalent. L'horloge qui gère les parties à mémoire est remplacée par un contrôle local composé d'un contrôleur, de latches et des canaux de communication. Les parties à mémoire sensibles au front de l'horloge (*Flip-Flop*) sont remplacées par des *latches*, parties à mémoire sensibles au niveau du signal de contrôle local (*Enable*) géré par les communications.

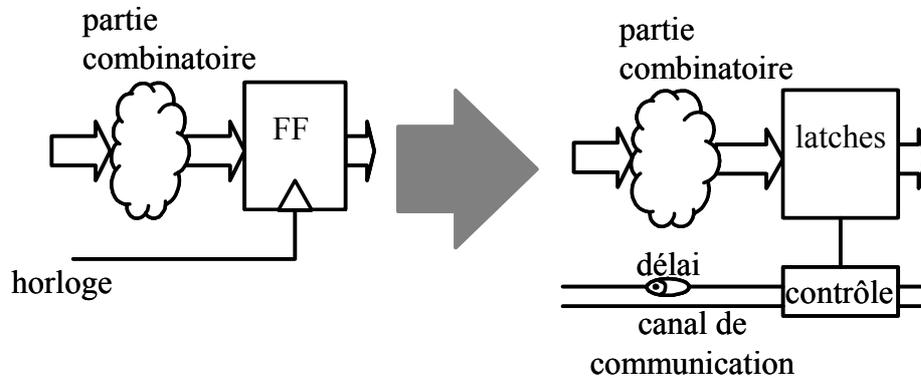


Figure 4-9 : *Asynchronisation* : principe général

Chaque processus sensible à l'horloge est équivalent, après *asynchronisation*, à une partie contrôle et une partie donnée (Figure 4-9). La partie contrôle du circuit a pour rôle de gérer les communications entre les processus *asynchronisés* et le processus considéré. La partie donnée représentera le traitement effectué à l'intérieur du processus activé et les *latches* commandées par la partie contrôle.

4.5 *Asynchronisation*

4.5.1 Séquence des évènements

4.5.1a Dépendances et canaux

Dans un circuit synchrone, la séquence des traitements dans les parties combinatoires est régulée par les éléments à mémoire contrôlés par le signal d'horloge. Lorsqu'il y a un évènement sur le signal d'horloge, les opérateurs qui lui sont sensibles effectuent leur traitement concurremment. Au contraire, dans un circuit asynchrone, les traitements des opérateurs s'effectuent séquentiellement. Une attention particulière est donc portée pendant l'*asynchronisation* lorsque le signal d'horloge est remplacé par des communications de type poignées de mains. En effet, ces communications doivent respecter la séquence des traitements donc respecter le fonctionnement des structures étudiées précédemment (dépendances, interdépendances): Quand un processus *B* dépend d'un processus *A*, il utilise les entrées générées par *A* à *t* et génère ses propres sorties à *t+1*. Ainsi, le fonctionnement du circuit est garanti correct si tout processus vérifie si toutes ses entrées sont disponibles et une fois toutes ses sorties utilisées. Il peut alors effectuer son traitement.

Comme l'horloge est supprimée et la séquence des traitements doit être respectée, chaque dépendance de donnée requiert un canal de communication (Figure 4-10). Une requête sur ce dernier signifiera que les données sont disponibles. Ainsi, si un processus possède une dépendance multiple, il aura plusieurs canaux à son entrée (c'est le cas du processus *A3* utilisant un élément appelé rendez-vous (paragraphe 3.3.7b) pour générer une requête *r1* à son entrée). Inversement, si plusieurs processus dépendent d'un même processus, alors il y aura plusieurs canaux en sortie de ce processus (c'est le cas du processus *A3* qui utilise un élément appelé duplicateur). Les éléments duplicateurs et rendez-vous décrits dans le paragraphe 3.3.7 permettent d'implémenter les dépendances multiples.

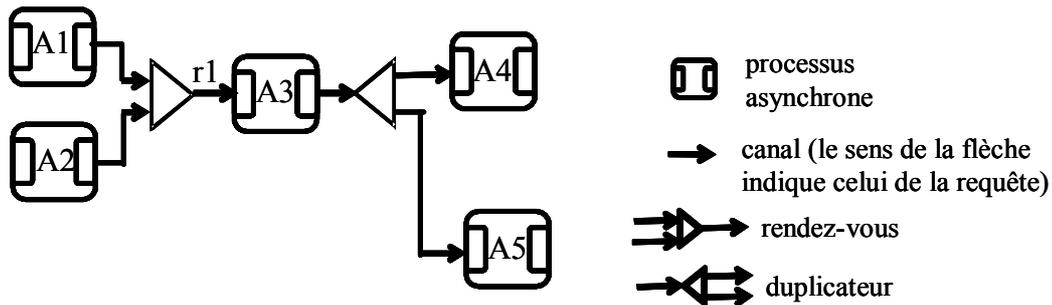


Figure 4-10 : Un canal de communication par dépendance

4.5.1b Jetons et bulles

La Figure 4-10 montre le type de représentation utilisé pour définir l'*asynchronisation*. Ce schéma représente une structure asynchrone avec des dépendances multiples. Le processus asynchrone, représenté par le motif  modélise un opérateur asynchrone qui contient de la logique combinatoire et des éléments mémoires ainsi que ses ports d'entrée/sortie. Ces éléments mémoires sont gérés par les protocoles de communication s'effectuant à travers les canaux. Ceux-ci sont représentés par une flèche \rightarrow , dont la tête indique le sens de la requête dans le canal, la présence d'une requête est indiquée par \bigcirc (Figure 4-11). En pratique, celle-ci est implémentée par la mise à 1 du signal de requête. Les rendez-vous et duplicateur sont respectivement représentés par \triangleright et \triangleleft .

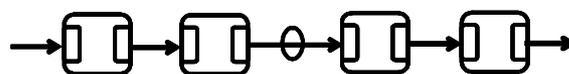


Figure 4-11 : Modélisation de la présence d'une requête

La circulation des requêtes dans le circuit peut être aussi modélisée par une circulation de jetons dans le circuit [RENA04] comme le montre la Figure 4-12. Cette modélisation est équivalente à celle de la Figure 4-11.  représente le processus et sa partie mémorisante (partie grisée).  représente les données transmises d'un processus à l'autre. Le jeton est porteur d'information qu'il stocke dans les éléments mémoires.

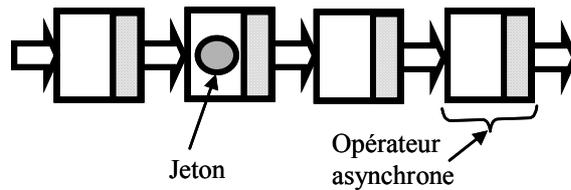


Figure 4-12: Modélisation par jeton [RENA04]

La circulation des jetons représente les communications qui s'effectuent dans le circuit. Lorsqu'un jeton circule, il laisse derrière lui une place vacante pour un autre jeton et remplit une place qui était disponible. Ces places vides sont appelées bulles (Figure 4-13). Le flot des données est régi par 2 règles [RENA04]:

Règle sur les jetons : la mémoire d'un opérateur *A* peut recevoir un jeton et stocker les données qui lui sont attribuées si et seulement s'il possède une bulle.

Règle sur les bulles : la mémoire d'un opérateur *A* devient vide (apparition d'une bulle) si et seulement si l'opérateur suivant *B* reçoit et stocke le jeton contenu par *A*.

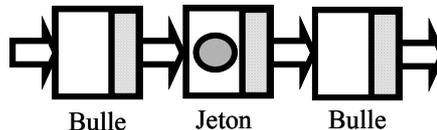


Figure 4-13 : Bulle et Jeton

4.5.1c Protocole découplé, protocole non découplé

Le cas d'une chaîne d'opérateurs représentée Figure 4-11 illustre le cas d'utilisation de différents protocoles. Si, lorsqu'une requête en entrée est envoyée continuellement, tous les opérateurs peuvent être actifs à la fois, le protocole est alors dit découplé. Si, lorsqu'une requête en entrée est envoyée continuellement, 1 opérateur sur 2 est actif à la fois, alors le protocole est dit non découplé. Dans ce cas, l'entrée et la sortie de chaque opérateur asynchrone sont couplées.

Dans un circuit utilisant le protocole découplé, tous les jetons seront dits valides. Dans un protocole non découplé, deux types de jetons peuvent circuler (Figure 4-14): les jetons valides et les jetons invalides (permettant la remise à zéro du protocole de communication).

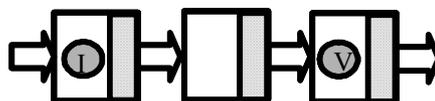


Figure 4-14 : Jetons valides et invalides dans le protocole non découplé

Le choix du protocole a un impact direct sur la vitesse et la surface du circuit. Par exemple, un pipeline d'un débit de sortie donné nécessite plus d'opérateurs dans le cas du protocole non découplé par rapport au protocole découplé. Ce point sera repris plus en détail dans la section 4.7.

4.5.1d Validité des données

Pour faire son exécution, chaque processus doit disposer de toutes ses entrées valides. En synchrone, l'horloge assure la validité des données. Dans le circuit *asynchronisé*, des retards équivalents à la latence maximum des parties combinatoires sont insérés pour assurer la validité des données. Comme le montre la Figure 4-9, ces délais sont insérés dans les canaux de communication. Les valeurs de ces retards peuvent être typiques ou extraites après synthèse du circuit (d'après les fichiers *Standard Delay File* - sdf). Il est à noter que ces dernières sont elles-mêmes des fonctions de valeurs typiques mais leur valeur est plus précise.

4.5.2 Initialisation

L'initialisation du fonctionnement de la structure comprend celle des données et celle du fonctionnement de la structure. *Asynchroniser* l'ensemble est une étape délicate car une mauvaise initialisation entraîne forcément un mauvais fonctionnement du circuit.

4.5.2a Initialisation des signaux de contrôle

Une attention particulière est donnée aux valeurs initiales des canaux de communications. Des requêtes doivent être présentes dans les canaux de communication en entrée des opérateurs d'une part pour assurer un fonctionnement correct du circuit et d'autre part pour prévenir du problème d'inter-blocage comme l'expliquent les paragraphes suivants. Par ailleurs, le choix du protocole de communication reste un acteur principal dans le choix des structures équivalentes et de l'initialisation du contrôle. Pour assurer l'initialisation lors du protocole WCHB, les contrôleurs utilisés sont les suivants :

- les contrôleurs d'initialisation qui possèdent une requête en sortie à l'initialisation et laissent passer les données à travers les *latches* (*enable='1'*)
- les contrôleurs non initialisant qui ne possèdent pas de requête en sortie au *reset* et retiennent les données (*enable='0'*)

4.5.2b Initialisation des signaux de donnée

L'initialisation des signaux de donnée doit suivre celle qui est effectuée dans la structure synchrone.

En général, le *reset* est synchrone pour des raisons de synthèse [KEAT98]: il se comporte comme un signal en entrée (synchronisé identiquement aux autres signaux). S'il est actif, ce signal sera traité au bon moment si le flot des données dans le circuit asynchronisé est respecté, et si par ailleurs l'initialisation du contrôle est correcte.

Le *reset* asynchrone est un signal spécial au même titre que le signal d'horloge. Il reste asynchrone lors de l'application de la méthode et ne pose pas de difficulté.

4.5.2c Convention pour l'initialisation

Les paragraphes précédents permettent de définir les conventions d'initialisation des protocoles. La structure représentée Figure 4-15 remplace la structure avec *reset* synchrone. Dans cette structure, l'initialisation du contrôle est assurée par l'envoi d'une requête en sortie. Ce point

sera repris plus en détail dans le paragraphe suivant (4.5.3). Les données initialisées sont transmises. La stabilité de celles ci est assurée par un contrôleur non initialisant en amont.

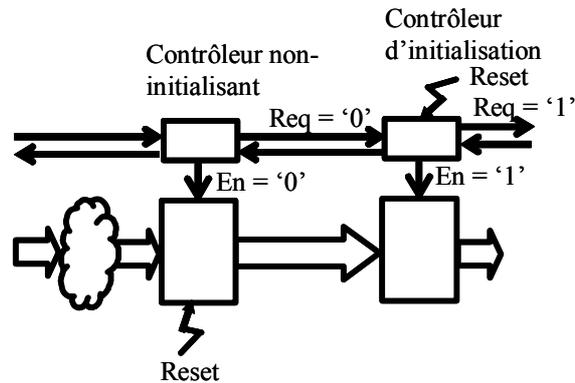


Figure 4-15 : Implémentation du *reset* synchrone

La structure représentée Figure 4-16 sera utilisée pour remplacer la structure synchrone avec *reset* asynchrone. Le *reset* sur les *latches* commandées par le contrôleur non initialisant assure l'initialisation asynchrone des données. Celles ci sont garanties stables car les *latches* sont verrouillées ($En=0$). L'initialisation du contrôle est assurée par un contrôleur initialisant en aval de la structure.

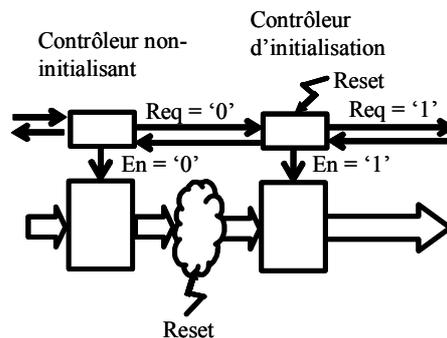


Figure 4-16 : Implémentation du *reset* asynchrone

4.5.3 Conditions de vivacité et bon fonctionnement

Ce paragraphe explique les problèmes d'inter-blocage et de fonctionnement erroné.

4.5.3a L'inter-blocage

Le problème de l'inter-blocage se révèle lors du fonctionnement du circuit, de deux manières différentes:

- A la mise en route, lorsque le circuit ne démarre pas.
- A l'état transitoire, lorsque le circuit démarre et s'arrête pendant son premier cycle.

Dans les deux cas, le circuit se trouve dans un état bloqué. Dans cet état, les jetons ne peuvent pas circuler dans le circuit : soit parce qu'il manque des jetons, soit parce qu'il y a trop de jetons.

Le manque de jetons se manifeste en amont des rendez-vous ou plusieurs jetons sont nécessaires pour produire un jeton en sortie. L'inter-blocage se manifeste aussi à l'intérieur des boucles du circuit : trop de jetons sont présents et ne peuvent pas circuler. Le choix de l'architecture détermine le nombre de jetons présents à l'initialisation (contrôleur d'initialisation), le nombre de jetons nécessaires (rendez-vous, boucle). Le protocole de communication utilisé a également un impact sur le nombre de bulles présentes dans les boucles.

Illustration du problème d'inter-blocage à la mise en route

La Figure 4-17 représente la structure *ARCHI0* et sa version asynchrone.

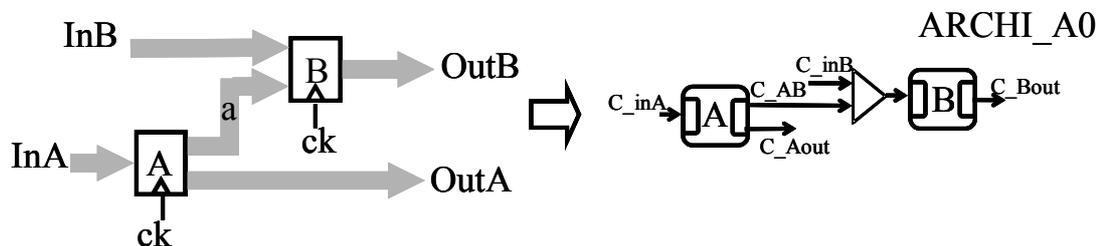


Figure 4-17 : Structure *ARCHI_A0* équivalente à *ARCHI0*

Dans ce type de structure, lorsqu'un jeton arrive sur le canal C_{inA} , il est pris en compte par A . A considère alors que ses données en entrée sont valides. Il effectue son traitement et envoie une requête à B (mise à 1 du signal de requête) et un acquittement à son entrée (mise à 0 du signal d'acquittement). B effectue ensuite son traitement lorsqu'il reçoit une requête sur le canal venant de A (C_{AB}) et venant de l'extérieur (C_{InB}). Le problème de mise en route ne se pose donc pas car le système démarre (Figure 4-18).

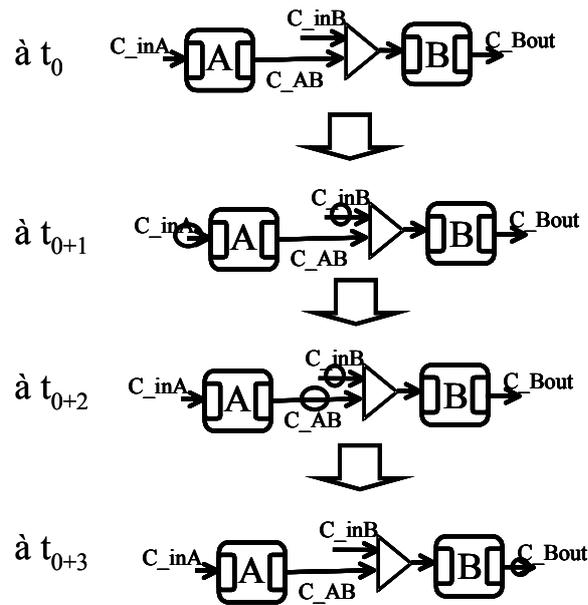


Figure 4-18 : Circulation des requêtes dans ARCHI_A0

Une nouvelle structure asynchrone *ARCHI_A1*, présentée Figure 4-19 illustre le problème de l'inter-blocage à la mise en route. Celle-ci est équivalente à la structure *ARCHI1* présentée précédemment.

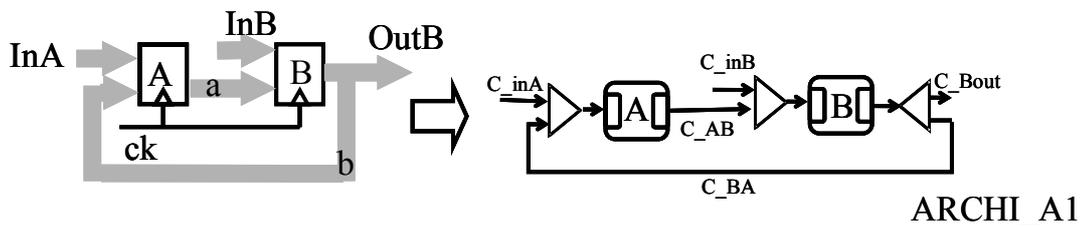


Figure 4-19 : Structure ARCHI_A1 équivalente à ARCHI1

Le processus *A* ne peut fonctionner que s'il reçoit une requête venant de l'extérieur (sur le canal C_{inA}), mais aussi une requête de *B*. De même, *B* ne peut fonctionner que s'il reçoit une requête venant de l'extérieur (sur C_{inB}) et une requête de *A*. S'il n'y a pas de requête présente ni sur le canal C_{BA} , ni sur le canal C_{AB} , la structure ne démarre pas.

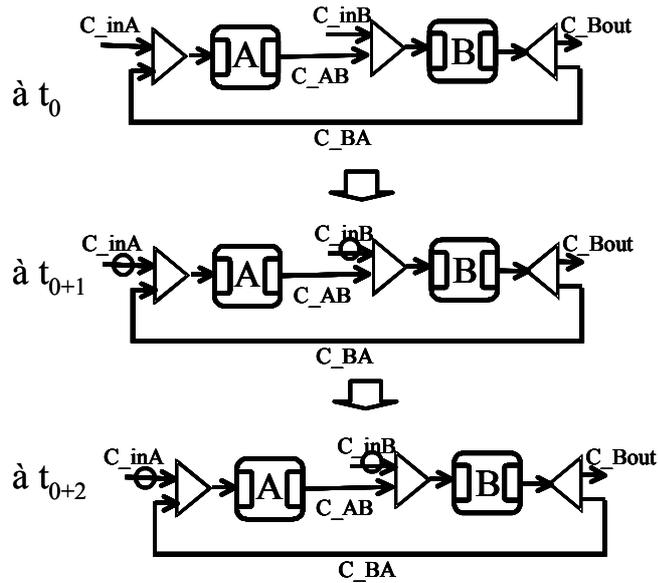


Figure 4-20 : Pas de circulation des requêtes dans ARCHI_A1

Illustration du problème d'inter-blocage à l'état transitoire

Dans Figure 4-19, si le protocole de communication utilisé entre les contrôleurs est un protocole non découplé, la structure ne peut pas fonctionner : il y a un inter-blocage (Figure 4-20).

4.5.3b La correction fonctionnelle

La correction fonctionnelle consiste à assurer le fonctionnement correct du circuit.

Les données doivent être prises au bon moment. Par exemple, dans la structure *ARCHI_A1* de la Figure 4-7, il faut assurer que ce sont $a(t-1)$ et $InB(t-1)$ qui sont utilisées pour calculer $b(t)$ et non $a(t-2)$ et $InB(t-1)$.

Illustration du problème de fonctionnement erroné

Deux structures très semblables sont proposées Figure 4-21. La différence réside en l'ajout d'un tampon supplémentaire dans la structure 2. L'initialisation dans la boucle se fait par l'envoi d'une requête 0 par B dans la *ARCHI_A1'*. L'initialisation dans la boucle de la structure *ARCHI_A1''* se fait par l'envoi d'une requête par un tampon supplémentaire C ajouté dans la boucle.

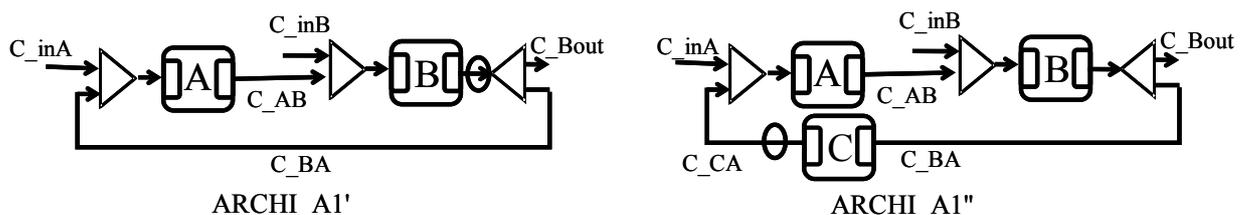


Figure 4-21 : ARCHI_A1' et ARCHI_A1''

Si le protocole utilisé dans les 2 structures est un protocole découplé, les jetons peuvent circuler, il n'y a pas d'inter-blocage. Cependant, ces structures ont un fonctionnement erroné :

InA , InB , $OutA$ (ou a), $OutB$ (b) et c sont les données circulant dans les structures et signalées respectivement par les requêtes sur les canaux C_InA , C_InB , C_OutB , C_AB , C_BA et C_CA .

Les équations qui modélisent $ARCHI_A1'$ et régissent par le flot des données sont :

$$OutA(t)=a(t)=f_A(InA(t-1),b(t-1))$$

$$OutB(t)=b(t)=f_B(InB(t-1),a(t-1))$$

Système Équation 4-9

Au démarrage, A effectue son traitement en premier, B effectue le sien ensuite.

$$OutA(1)=a(1)=f_A(InA(0),b_0)$$

$$OutB(1)=b(1)=f_B(InB(0),a(1))$$

Système Équation 4-10

La valeur d'initialisation a_0 n'est donc pas prise en compte au démarrage de la structure de $ARCHI_A1'$. La structure de la $ARCHI_A1'$ n'a donc pas un fonctionnement identique au fonctionnement de la structure $ARCHI1$.

Les équations qui modélisent la $ARCHI_A1''$ sont les suivantes:

$$OutA(t)=a(t)=f_A(InA(t-1),c(t-1))$$

$$OutB(t)=b(t)=f_B(InB(t-1),a(t-1))$$

Système Équation 4-11

Le processus C a pour fonction f_c . C'est une fonction identité qui induit un retard d'un jeton : elle insère une bulle dans la boucle de contrôle.

$$c(t)=f_c(b(t-1))=b(t-2).$$

donc

$$OutA(t)=a(t)=f_A(InA(t-1),b(t-2))$$

$$OutB(t)=b(t)=f_B(InB(t-1),a(t-1))$$

Système Équation 4-12

Ces équations sont différentes de celles modélisant la structure *ARCHII*. Le fonctionnement de *ARCHI_AI''* n'est donc pas équivalent au fonctionnement de la structure *ARCHI*.

4.5.3c Conditions de vivacité et de bon fonctionnement

Voici l'analyse de la structure *ARCHII* qui permet de résoudre les problèmes de fonctionnement erroné et d'inter-blocage décrits précédemment. Les équations qui modélisent *ARCHII* sont les suivantes :

$$\begin{cases} OutA(t) = a(t) = f_A(InA(t-1), b(t-1)) \\ OutB(t) = b(t) = f_B(InB(t-1), InA(t-2), b(t-2)) \\ OutA(0) = a_0 \\ OutB(0) = b_0 \end{cases}$$

Système Équation 4-13

Par conséquent, $b(t)$ dépend de la valeur produite par B à $t-2$: $b(t-2)$. La boucle contient deux étages de *Flip-Flop*. Deux jetons doivent donc pouvoir circuler en même temps dans la boucle pour actionner les 2 étages. Ceci rejoint les règles des anneaux auto séquencés expliqués dans l'encadré ci-dessous tiré de [RENA04].

Un anneau auto séquencé (Figure 4-22) est composé d'étages pipelinés dans lesquels des données différentes circulent. Il est supposé fermé sans interaction avec l'extérieur, ceci pour simplifier le raisonnement.

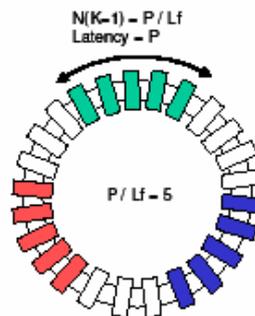


Figure 4-22 : Un anneau auto séquencé

La latence totale de l'anneau est la somme des temps de propagation d'un jeton pour faire le tour de l'anneau : $\lambda = N \times Lf$. Le débit global de l'anneau est égal au nombre de jetons qu'il contient par la latence totale : $T = \frac{K}{\lambda} = \frac{K}{N \times Lf}$. Le débit maximal ne peut pas dépasser le débit

local de chaque étage qui est $T_{\max} = \frac{1}{P}$. Le nombre optimal d'étage pour traiter K données (tout en respectant la vitesse) est : $N_{\text{optimal}} = K.P / Lf$. Si le nombre d'étages est suffisamment grand, ($>K.P/Lf$), les données se propagent alors comme dans une structure purement combinatoire, en s'étendant sur environ P/Lf étages chacune. Si le nombre d'étages est inférieur à $2K+1$, alors l'anneau ne peut pas fonctionner, il y a un inter-blocage. Dans le cas d'un protocole 4 phases, chaque donnée s'étend au minimum sur 2 étages. De plus, la présence d'un étage vide est nécessaire pour la circulation des données. Dans le cas d'un nombre d'étages intermédiaire compris entre $2K$ et $K.P/Lf$, les jetons ne peuvent pas se propager à la vitesse maximale et leur extension est limitée par la place disponible. La vitesse de circulation est alors déterminée pas le nombre de bulles $N-2K$ (étages disponibles pour la circulation des acquittements). Dans le cas du protocole 4 phases, il faut deux circulations complètes autour de l'anneau à une bulle pour faire avancer un jeton complet. Le débit global de l'anneau vaut ainsi : $T = (N-2K)/(2.N.Lr)$.

Ainsi la structure équivalente à la boucle équivaut à 3 tampons lorsque le protocole PCFB est utilisé et 5 tampons lorsque le protocole WCHB est utilisé.

Dans le cas particulier du passage d'un circuit synchrone à un circuit asynchrone, les conditions de vivacité et de bon fonctionnement ont pu être définies et sont les suivantes :

Quel que soit le bloc considéré :

- Pour toute consommation de jetons sur toutes les entrées, il y a production de jetons sur les sorties.
- Tout registre émet une requête à l'initialisation
- Toute boucle doit respecter les règles de vivacité des anneaux auto séquencés [RENA04] qui est fonction de la structure du nombre de jetons circulant dans la boucle et du protocole.

4.5.4 Compatibilité de cycle

La cycle compatibilité du circuit est assurée par l'interface d'entrée du circuit (Figure 4-23). Cette interface permet d'envoyer des jetons à la cadence d'une horloge (fonctionnement synchrone) pour que le circuit réponde cycle à cycle comme un circuit synchrone ou bien d'envoyer continuellement des jetons (fonctionnement libre appelé *free-running*).

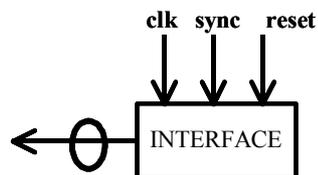


Figure 4-23 : L'interface d'entrée assure la cycle compatibilité

4.6 Composants pour l'asynchronisation

4.6.1 Eléments pour les dépendances multiples

Afin de supporter les dépendances multiples entre les processus, les éléments duplicateur et rendez-vous (Figure 4-24) décrits dans le Chapitre 3 paragraphe 3.3.7 sont utilisés.

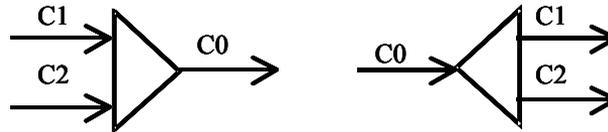


Figure 4-24 : Rendez-vous et duplicateur

4.6.2 Tampons (*buffer*, *half buffer*).

Le tampon est composé d'un contrôleur qui assure les communications, des *latches* commandés par celui-ci qui gèrent le passage des données traitées par la partie combinatoire.

Il est appelé *Half buffer* ou *buffer* selon le protocole utilisé. En effet dans le cas d'un protocole découplé, il assurera un rôle complet de *buffer* tandis que dans un protocole non découplé, il assurera un rôle d'un demi-*buffer*.

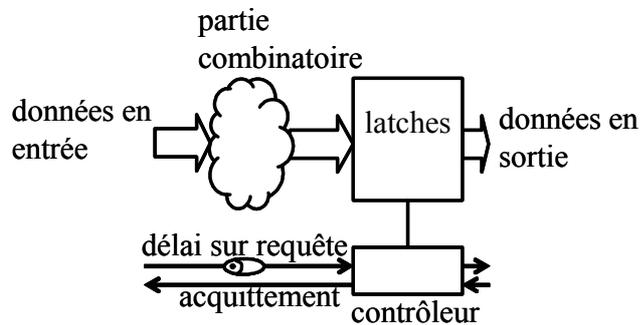


Figure 4-25 : Un tampon

Les contrôleurs utilisés pour l'asynchronisation ont été élaborés au sein du groupe CIS [RIGA02]. Leur structure dépend du choix du protocole de communication. Pour assurer la mise en route et le bon fonctionnement, pour chaque protocole les contrôleurs peuvent être de 2 types : avec ou sans jeton d'initialisation. Les conventions choisies sont les suivantes : *enable* = '1' les *latches* sont déverrouillées, *resetb* = '0' ou *set* = '1' le contrôleur est initialisé (*resetb* actif à 0).

Dans un circuit synchrone, les éléments à mémoire sensibles à l'horloge utilisés sont actuellement les bascules (*flip-flop*). Ce sont des éléments sensibles aux événements donc qui conviennent plus au mode de fonctionnement synchrone que les *latches*. Les *latches* sont plus adaptés aux communications asynchrones car ils sont sensibles aux niveaux des signaux et non à leurs événements. En effet, les niveaux des signaux de communications sont stables et assurés par

les protocoles dans les circuits asynchrones. De plus les *latches* prennent moins de surface (4.3.1b).

4.6.2a Tampon sans jeton d'initialisation

Quand *resetb* est actif, les *latches* sont initialisés et le tampon ne laisse pas passer les données (*enable='0'*).

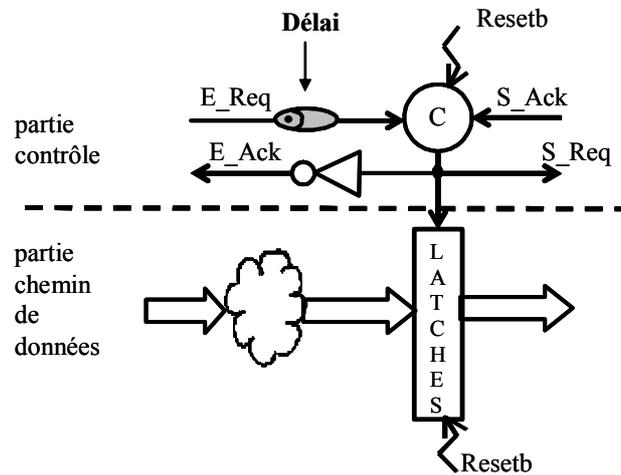


Figure 4-26 : Structure du contrôleur utilisé pour un tampon sans jeton d'initialisation

A titre d'exemple, la Figure 4-26 présente la structure du tampon utilisant le protocole 4 phases WCHB. Un inverseur permet de respecter la convention : acquittement actif niveau bas tandis que le délai respecte la latence de la partie combinatoire.

4.6.2b Tampon avec jeton d'initialisation

Quand *set* est actif, les *latches* sont passants et une requête est envoyée en sortie du contrôleur.

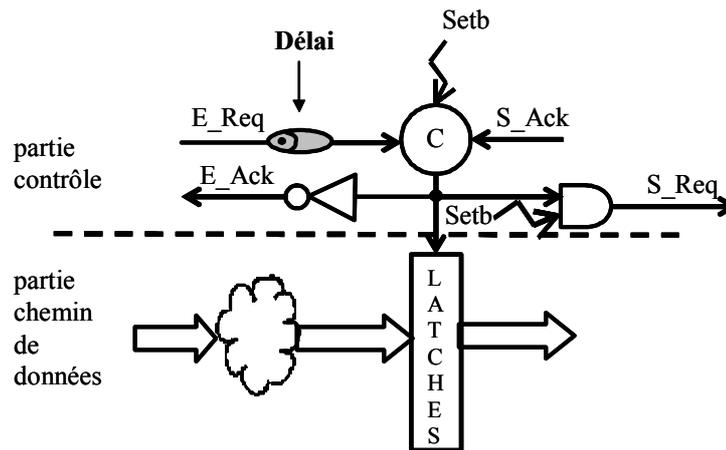


Figure 4-27 : Structure du contrôleur utilisé pour un tampon avec jeton d'initialisation

A titre d'exemple, la Figure 4-27 présente la structure du tampon avec jeton d'initialisation du protocole 4 phases WCHB. Comme pour le contrôleur précédent un inverseur permet de respecter la convention et le délai respecte la latence de la partie combinatoire. Une porte ET en sortie permet d'assurer la requête en sortie à 1 lors de l'initialisation.

4.6.2c Interfaces d'entrée et de sortie

Pour gérer le signal de *reset* et la synchronisation du système sur l'horloge, une interface est utilisée en entrée. Cette interface a été décrite dans le paragraphe 4.5.3a sur la compatibilité cycle à cycle (Figure 4-23). Toute la gestion est faite en entrée car ceci facilite le contrôle d'initialisation du système. L'interface d'entrée permet de gérer le signal de *reset*, et d'envoyer les jetons dans le circuit selon le mode de fonctionnement choisi. Les conditions de fonctionnement de l'interface d'entrée sont les suivantes :

- Les signaux d'entrée satisfont les conditions synchrones.
- Des requêtes sont générées afin de signaler à la structure la présence des signaux valides en entrée. Ceci est fait continuellement et selon la validité des signaux en entrée si le mode de fonctionnement de la structure est asynchrone, sinon à chaque front d'horloge si le mode de fonctionnement choisi est synchrone.
- Les acquittements des requêtes en entrées des requêtes envoyées doivent arriver avant le front d'horloge suivant afin que les nouvelles données soient bien prises en compte.

L'interface de sortie permet de gérer la compatibilité cycle à cycle en sortie en délivrant les données traitées à chaque front montant ou descendant de l'horloge.

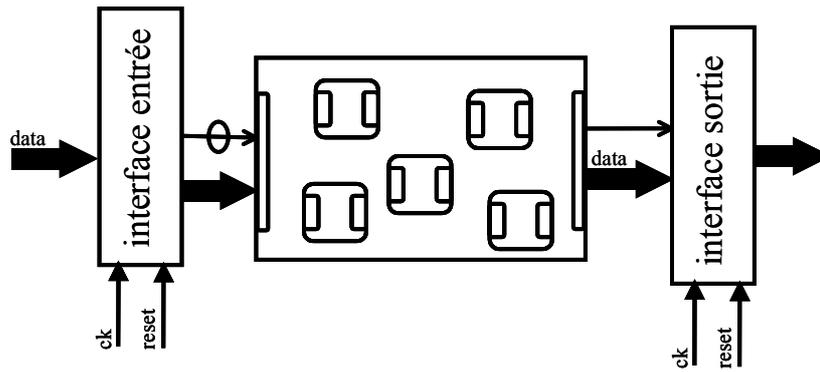


Figure 4-28 : Interfaces d'entrée et de sortie

4.7 Choix d'un protocole

4.7.1 Latence du circuit

Dans le circuit synchrone, le flux du passage des données est géré par l'horloge. Lorsque les éléments à mémoire sont remplacés par des contrôleurs et des *latches*, le flux du passage des données est déterminé par le protocole de communication. La latence d'un circuit utilisant un protocole découplé est plus faible que celle dans le cas d'un protocole non découplé.

Les Figure 4-29 et Figure 4-30 comparent deux structures équivalentes synchrone et asynchrone.

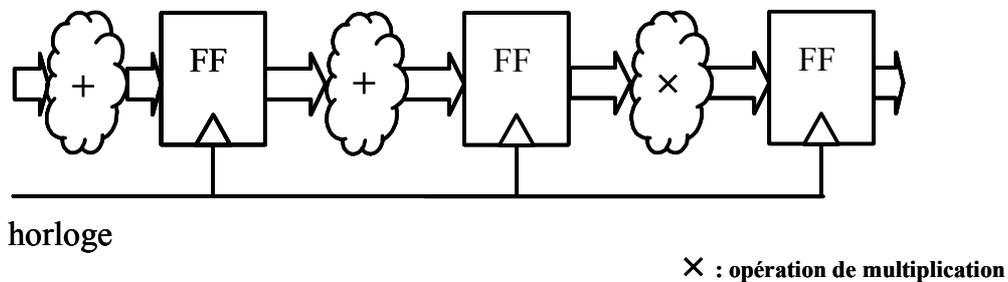


Figure 4-29 : Structure synchrone

Dans le circuit synchrone, l'image de l'entrée est obtenue en sortie au bout de 3 périodes d'horloge ($3T$). La valeur de T est supérieure ou égale au temps maximum de la partie combinatoire contenue entre 2 éléments à mémoire.

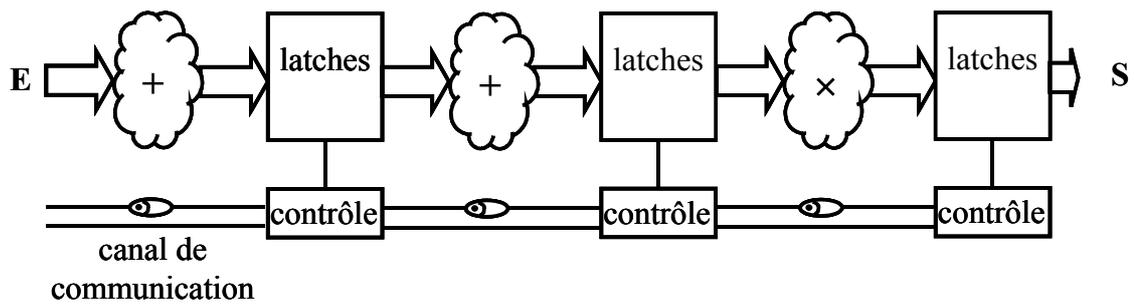


Figure 4-30 : Structure asynchrone équivalente

La Figure 4-30 montre la structure asynchrone équivalente. Le temps n'est pas contraint par la période du signal d'horloge mais par la durée des protocoles entre les contrôleurs. Par exemple, pour un contrôleur utilisant le protocole WCHB, l'acquittement d'entrée n'est pas remis à 1 avant la remise à 0 de la requête de sortie. Il est à noter que le temps d'exécution dépend du protocole : il est différent pour un protocole WCHB, PCHB ou PCFB. Ceci dépend de la synchronisation entre les signaux d'entrée et de sortie. Un protocole sera dit découplé lorsque la remise à zéro des signaux d'entrée des contrôleurs est indépendante de la remise à zéro de leurs signaux de sortie. Ce point est repris plus en détail dans le chapitre 6.

4.7.2 Surface

Dans le cas des structures linéaires telles que les pipelines vus précédemment, le choix du protocole influe sur la vitesse des données uniquement. En général, les circuits contiennent d'autres structures comme les boucles (interdépendances). Lorsque le signal d'horloge est supprimé dans une boucle formée par plusieurs éléments à mémoire, il faut non seulement tenir compte de l'ordre de prise en compte des données mais aussi assurer leur circulation dans la boucle. Dans le cas de la structure de la Figure 4-31 où un protocole découplé est utilisé par les contrôleurs de la boucle, le jeton peut circuler. Ce protocole peut être par exemple, le protocole 4 phases PCFB.

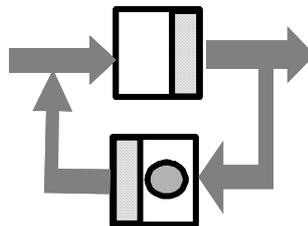


Figure 4-31 : Boucle utilisant un protocole découplé

Mais lorsque la même structure utilise un protocole non découplé (WCHB), aucun des contrôleurs des processus ne peut consommer le jeton qui se trouve en amont car il reste un jeton présent en aval.

Pour avoir un système non bloquant dans le cas du protocole WCHB, une solution consiste à introduire des bulles dans la boucle afin que les jetons puissent circuler (Figure 4-32).

L'introduction de bulles dans la boucle permet de satisfaire les conditions de vivacité et bon fonctionnement vues en 4.5.3c.

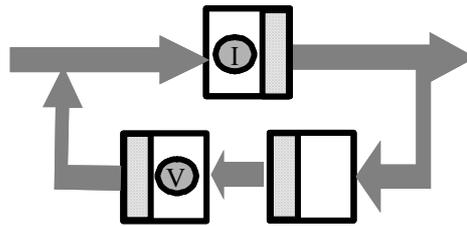


Figure 4-32 : Structure utilisant le protocole non découplé

Ceci est assuré par l'ajout de tampons supplémentaires (tampon bulle) dont le nombre dépend du protocole utilisé.

Un protocole non découplé nécessite plus de tampons qu'un protocole découplé. L'ajout de tampon permet d'optimiser la vitesse du protocole mais augmente la surface du circuit. L'implémentation des contrôleurs découplés est plus complexe et nécessite plus de surface que celle des contrôleurs non découplés.

Le choix du protocole de communication influe donc sur la vitesse de la réponse du circuit, la surface et sur la consommation de celui-ci.

4.8 Méthode

L'approche qui consiste à analyser le circuit synchrone, trouver une équivalence asynchrone, choisir un protocole, permet de définir la procédure. Celle-ci part d'une description RTL synthétisable du circuit synchrone afin d'obtenir un circuit asynchrone.

4.8.1 Analyse du code RTL

- L'analyse du code RTL consiste à identifier en premier lieu les signaux de contrôle qui gèrent le circuit : les horloges, les signaux d'initialisation. L'analyse doit déterminer la nature de ces derniers.
- Ensuite, l'analyse du code permet d'identifier la composition du circuit : la nature des processus (à mémoire ou combinatoires) en fonction de leurs entrées (signaux d'horloge, données, *reset*).
- Les dépendances entre processus sont ensuite déduites de la connexion de leurs ports. Les dépendances entre les processus à mémoire sont identifiées : simple, multiple, interdépendances.

Dans cette thèse, l'analyse du code a été effectuée à la main. Mais celle-ci peut s'effectuer automatiquement à l'aide d'outils disponibles dans le commerce pour la compilation, la synthèse ou la vérification formelle. Dans ce cas, il faut que le code analysé suive rigoureusement les directives du [IEEE10] pour être analysables par les outils standard.

4.8.2 Description et définition de l'interface

Les signaux de contrôle ayant été identifiés, les signaux d'entrées et de sortie du circuit étant connus, l'interface synchrone-asynchrone peut donc être définie.

En mode synchrone sur un signal d'horloge, l'interface d'entrée envoie une requête et les données dans le circuit. L'interface de sortie renvoie un acquittement et transmet les données en sortie à la cadence de l'horloge.

En mode asynchrone, l'interface envoie les requêtes et les données valides en entrée du circuit dès que ces dernières sont prêtes et que le signal d'acquittement de la requête précédente a bien été reçu, quelque soit la valeur des signaux d'horloge. L'interface de sortie renvoie un acquittement aussitôt qu'elle reçoit une requête et transmet les données en sortie.

Le signal *sync* est un signal de contrôle ajouté qui permet de sélectionner le mode de fonctionnement du circuit. *sync*=1 : mode synchrone, *sync*=0 mode asynchrone. Ce signal peut être déclaré comme un signal d'entrée supplémentaire (port du circuit supplémentaire).

4.8.3 Passage du synchrone à l'asynchrone

Un graphe des dépendances entre processus à mémoire est déduit de l'analyse du code RTL. Les processus à mémoire sont remplacés par des contrôleurs commandant des *latches* selon la convention choisie en 4.5.2, et par l'utilisation des composants décrits en 4.6 selon le choix du protocole (4.7). Les structures bouclées asynchrones respectent les conditions de vivacité et de bon fonctionnement définies en 4.5.3.

4.8.4 La synthèse : utilisation des outils standards.

La synthèse des interfaces et de la partie contrôle doit être effectuée à la main car les outils d'aide à la conception ne gèrent pas la notion de canal. Elle pourra à l'avenir être traitée par TAST. Celle des parties combinatoires s'effectue à l'aide d'outils standard (Synopsys Design Compiler, Mentor Graphics Leonardo) ce qui permet de limiter le temps de conception du circuit. L'ensemble est ensuite réuni pour pouvoir effectuer la simulation fonctionnelle.

Comme le circuit asynchrone obtenu possède un mode de fonctionnement synchrone, il peut être validé dans l'environnement de test fonctionnel synchrone. Le nouveau circuit contenant les structures synchrones changées en structures asynchrones et l'interface à une déclaration identique au précédent (ports entrées et sorties), il peut donc remplacer l'ancien circuit dans son environnement. Si le signal *sync* est rendu accessible (port supplémentaire), l'environnement doit donc être légèrement modifié pour accepter ce nouveau port.

4.9 Implémentation

4.9.1 Filtre à Réponse impulsionnelle finie 4 coefficients

La méthode est implémentée sur l'exemple pédagogique d'un filtre RIF synchrone dont la structure est donnée Figure 4-33.

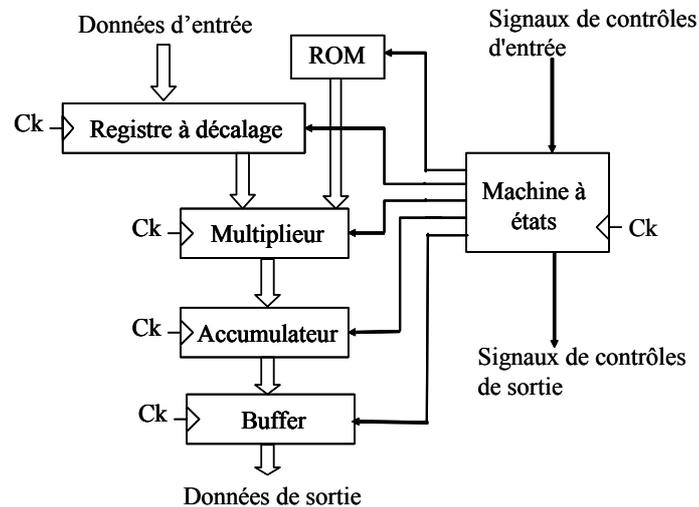


Figure 4-33 : FIR synchrone

Le fonctionnement du filtre répond à l'équation suivante : $Out(n) = \sum_{i=0}^{N-1} a(i).x(n-i)$.

Les coefficients sont stockés dans la ROM. Le registre à décalage (REGDEC) exécute le décalage des données en entrées et les stocke pour les fournir au multiplieur. Celui-ci calcule le produit des données x_i et des coefficients a_i . L'accumulateur (ACCU) permet de faire la somme des résultats des multiplications $x_1a_1 \dots x_n a_n$. Le *buffer* (BUFFER) fournit le résultat de la dernière addition en sortie. La machine à état (*Finite State Machine* – FSM) permet de contrôler la séquence des évènements dans le circuit. Les signaux de contrôle (ADC_busy, ADC_convstb, ADC_Rd_csb) échangés avec l'extérieur permettent de synchroniser le filtre avec le convertisseur.

4.9.2 Application de la méthode

4.9.2a Analyse du code RTL

La description RTL du filtre permet d'identifier les processus combinatoires et les processus à mémoires (sensibles au front d'horloge). La Figure 4-34 montre la structure du filtre et les processus identifiés. Le reset est synchrone pour tous les processus à mémoire.

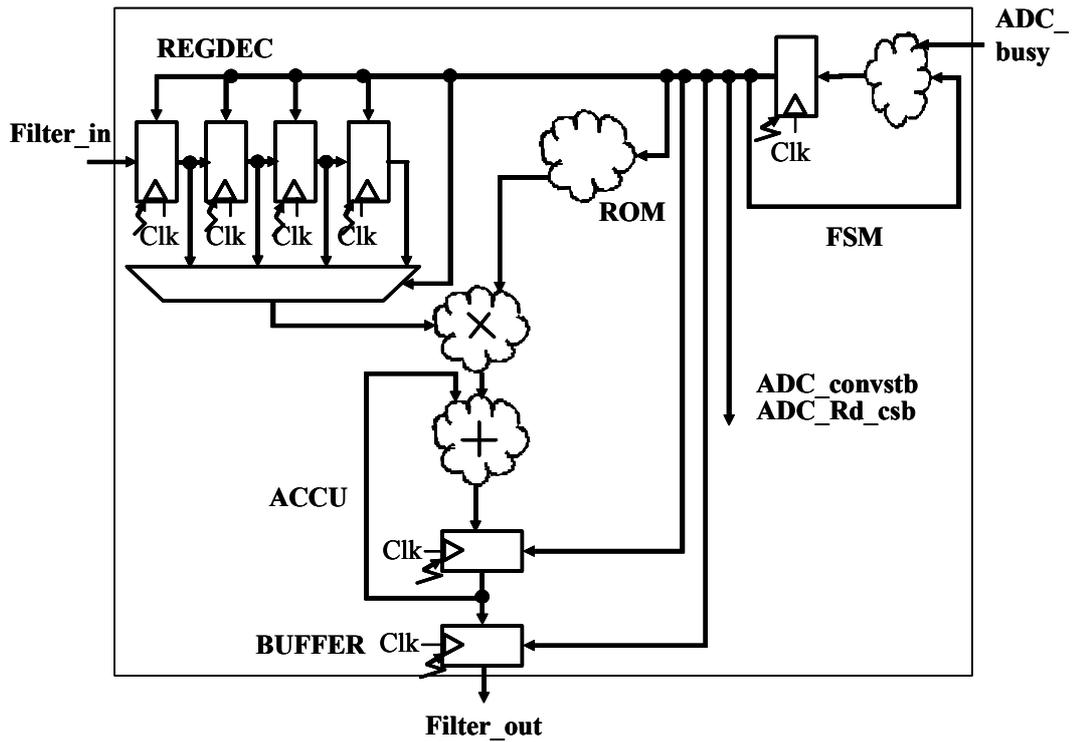


Figure 4-34 : Processus combinatoires et à mémoire du filtre

Les dépendances entre registres sont identifiées à la main. La Figure 4-35 montre le graphe des dépendances correspondant. La machine à état (FSM) et l'accumulateur (ACCU) contiennent chacun une boucle.

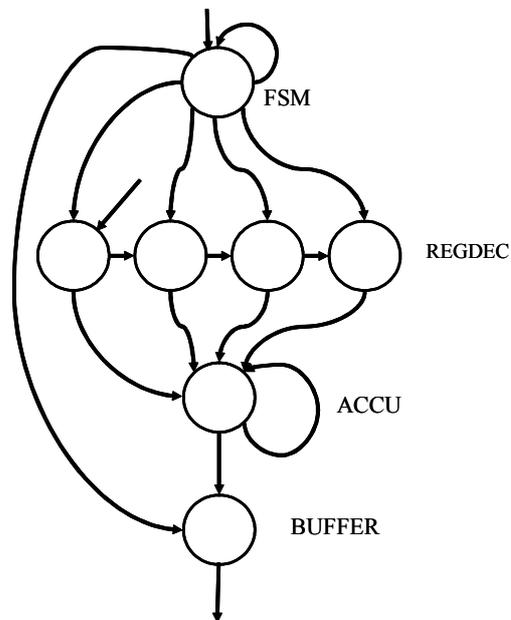


Figure 4-35 : Graphe de dépendances des registres du filtre

4.9.2b Description et définition de l'interface entrées/sorties

L'interface d'entrée et de sortie assurent la synchronisation entre l'extérieur du filtre et FSM, REGDEC et BUFFER.

L'interface d'entrée envoie une requête à la machine à état et une requête au registre à décalage. L'interface de sortie reçoit un acquittement de la FSM et du BUFFER. En fonction du signal *sync*, ces interfaces se synchronisent ou non sur le signal d'horloge.

4.9.2c Passage du synchrone à l'asynchrone

La Figure 4-36 montre la structure du filtre après application de la méthode et utilisation du protocole 4 phases WCHB. Pour que le schéma soit lisible, seuls les tampons et les canaux qui les relient ont été dessinés.

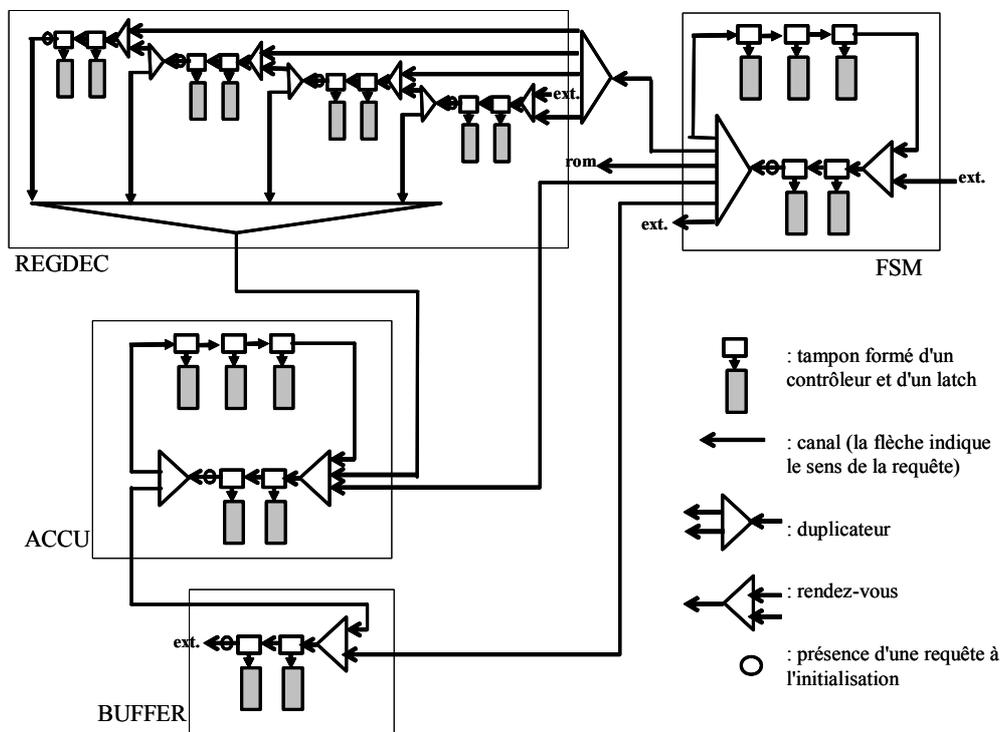


Figure 4-36 : Filtre asynchronisé

Dans cette solution, 5 tampons sont présents dans les structures bouclées, ce qui augmente considérablement la surface du filtre. Celle-ci peut être réduite en optimisant l'architecture. La Figure 4-37 montre la structure après une optimisation en surface de l'architecture. Des contrôleurs 2 phases (complètement découplés) ont été utilisés dans les boucles. Le registre a été optimisé de la manière suivante : les latches du registre à décalage sont commandées alternativement par *enable* et *not(enable)*, les données sont lues dans l'ordre inverse et le contrôle est optimisé par l'utilisation d'un multiplexage des requêtes et des acquittements.

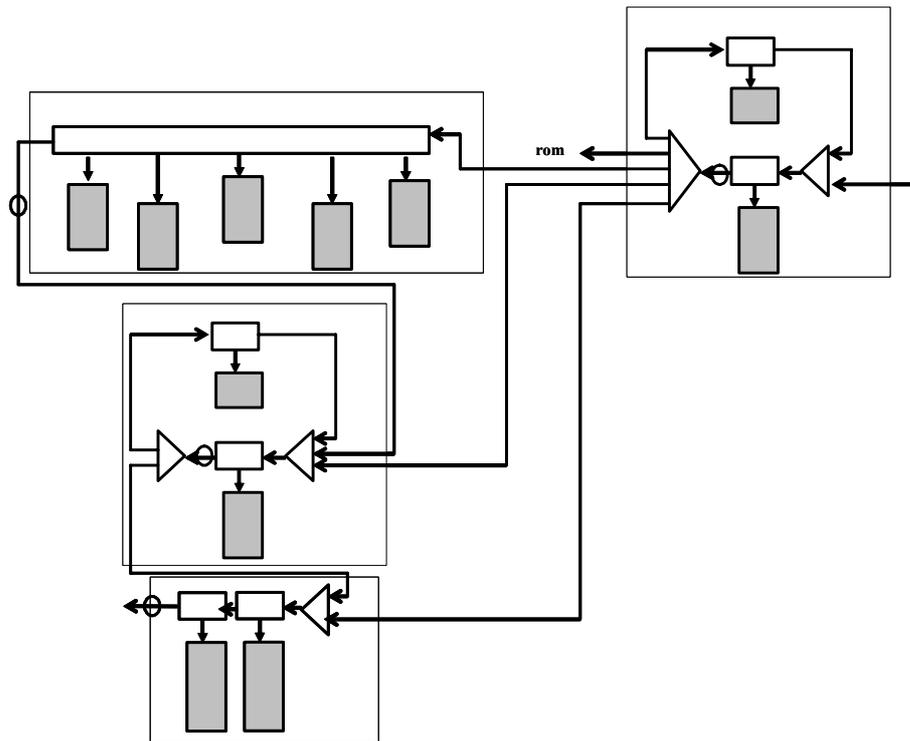


Figure 4-37 : Structure du filtre *asynchronisé optimisé*

4.9.2d Implémentations et Résultats

Le filtre synchrone et le filtre asynchronisé optimisé ont été implémentés dans la technologie 0,18 microns de STMicroelectronics. Les résultats ont été obtenus après synthèse, avant placement routage.

Les Figure 4-38 et Figure 4-39 présentent les réponses indicielles des deux filtres à un fonctionnement de 33MHz. Le filtre *asynchronisé* est en mode synchrone : les réponses sont identiques.

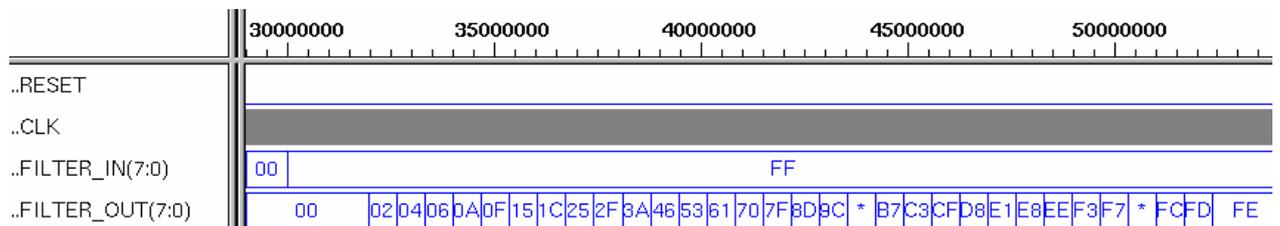


Figure 4-38 : Réponse indicielle du filtre synchrone

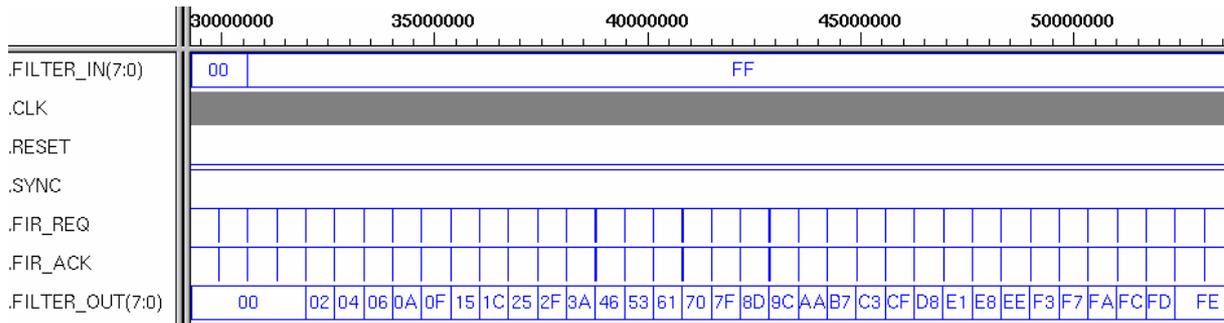


Figure 4-39 : Réponse indicielle du filtre asynchrone

L'activité électrique du circuit asynchronisé est présentée Figure 4-40, l'émission électromagnétique entre le circuit synchrone et le circuit asynchrone a été nettement réduite sur la bande allant de 33Mhz jusqu'à plus de 2GHz. A noter que l'arbre d'horloge n'a pas été implémenté dans le filtre synchrone. Les spectres ont été obtenus après simulation d'une série de 2 convolutions (FF.FF, FF.AA+AA.FF). Analog Artist (Cadence) a été utilisé avec des netlists Spectre en entrée, les simulations ont duré moins de 15heures chacune. Différentes alimentations ont été définies la simulations. Les courbes de courant de chaque opérateurs ont été sommées afin d'obtenir l'allure du courant global (le courant consommé par l'environnement n'est donc pas pris en compte). La DFT a été effectuée sur cette somme.

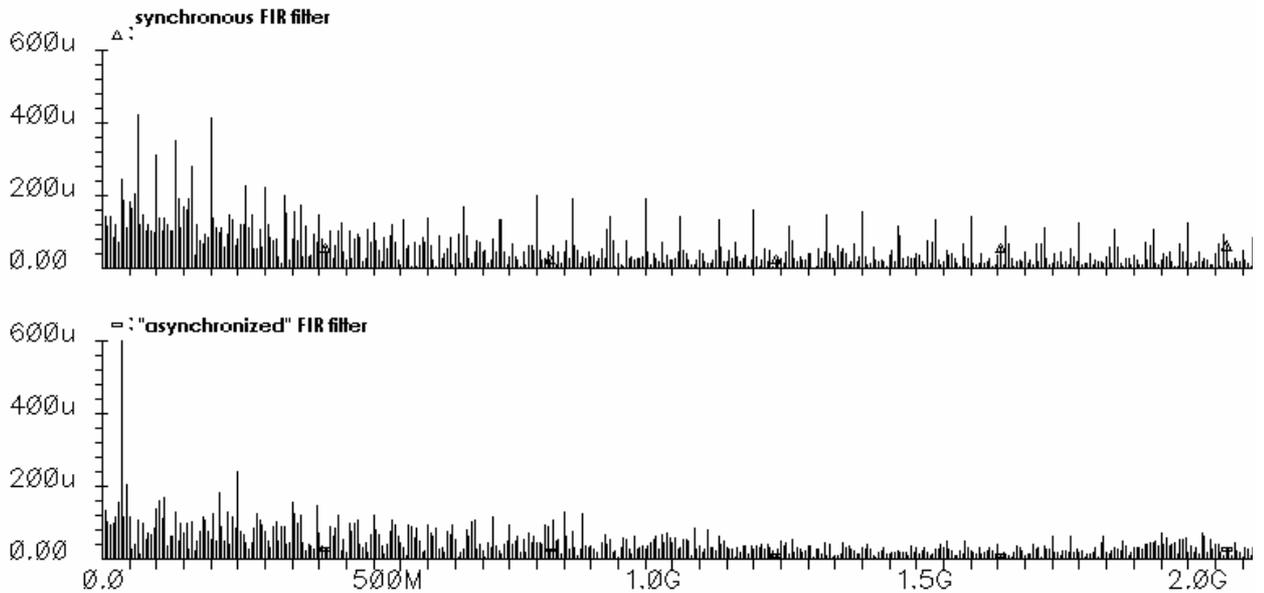


Figure 4-40 : Activités électriques des 2 filtres

4.10 Conclusion

La méthode *Asynchronisation* été élaborée en vue d'exploiter la propriété de faible émission des circuits asynchrones. Elle consiste à relâcher les communications dans le circuit en

remplaçant le signal de contrôle global (l'horloge) par des protocoles locaux (communications de type poignée de main). La consommation du circuit est ainsi mieux répartie.

Cette méthode gère les structures pipelines et séquentielles comme le font les méthodes de dé-synchronisation présentées en 3.4.2c. Elle va plus loin en proposant des structures équivalentes pour toutes les formes de dépendances (interdépendances, dépendances multiples). Elle gère par ailleurs l'initialisation du circuit.

De plus, le circuit obtenu à l'avantage d'être cycle compatible avec l'environnement du circuit synchrone grâce aux structures équivalentes et l'interface d'entrée.

Adressant le circuit à sa description RTL et utilisant un graphe de dépendances, la méthode est tout à fait automatisable.

Le gain en terme d'émission électromagnétique est significatif. Cependant l'augmentation de surface est indéniable et l'optimisation des structures en surface est indispensable tant que les cellules pour le contrôle de communication sont implémentées avec des portes standard. L'utilisation de différents protocoles dans le circuit est aussi envisageable.

CHAPITRE 5 Méthode "mise forme du courant" (*Current Shaping*)

5.1 Introduction

Les circuits asynchrones sont attractifs pour leur qualité de faible émission électromagnétique. Cependant, comme le montre la Figure 5-1, des forts pics de courant peuvent subsister lors de leur fonctionnement. A certaines fréquences, et selon leur amplitude, ces pics sont indésirables. Pour certaines applications, disposer alors d'un circuit asynchrone pour résoudre les problèmes d'émissions électromagnétiques liés à l'utilisation d'un circuit synchrone peut s'avérer une solution inefficace.

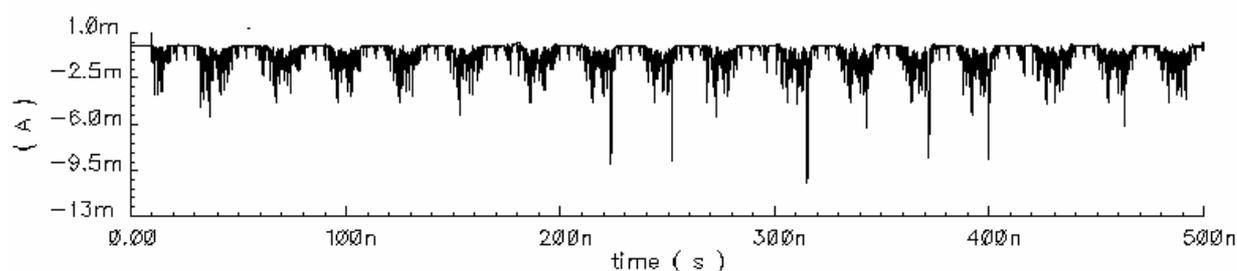


Figure 5-1 : Allure du courant consommé dans un circuit asynchrone

Ainsi, la diffusion industrielle des circuits asynchrones est possible non seulement si les outils pour leur élaboration sont disponibles, mais aussi s'il existe des méthodes d'optimisation de leurs performances. Les méthodes pour réduire le phénomène d'interférence électromagnétique dans les circuits asynchrones n'existent pas encore. Tenant compte de ce fait, une méthode pour améliorer leur performance en compatibilité électromagnétique a été élaborée avec l'intention d'être intégrée dans l'environnement d'outils TAST.

Ce chapitre présente dans un premier temps le contexte d'élaboration de la méthode "Mise en forme du courant" (5.2). Il détaille ensuite l'approche utilisée (5.3) et décrit la méthode définie à l'aide d'un exemple simple (5.4). L'implémentation (5.5) sur un filtre RIF quatre coefficients micropipeline valide la méthode.

5.2 Contexte

Ce paragraphe explique les orientations du projet pour l'élaboration de la méthode "mise en forme du courant" : le principe de la méthode, les contraintes qui ont été définies afin de minimiser la fonction de coût, et les choix qui ont été adoptés.

5.2.1 Contraintes

Certaines contraintes ont été ajoutée dans l'élaboration de la méthode tout en gardant le but de réduire les émissions électromagnétiques.

La première est évidemment d'adresser spécifiquement les circuits asynchrones. En effet, la méthode doit pouvoir être intégrée dans l'environnement TAST.

La seconde est de modifier au minimum l'architecture des circuits car l'amélioration des performances ne doit pas augmenter disproportionnellement les temps de conception des circuits.

La dernière contrainte impose au circuit optimisé de respecter le fonctionnement de l'architecture d'origine afin de convenir aux mêmes applications.

5.2.2 Choix

Ce paragraphe présente les choix qui ont été faits lors de l'élaboration de la méthode "mise en forme du courant".

5.2.2a La méthode s'inspire de l'ordonnement de la puissance dynamique

Comme décrit le chapitre 2, des méthodes pour la réduction des interférences électromagnétiques dans les circuits synchrones existent déjà. Cependant, les méthodes qui se basent sur l'optimisation de la répartition de l'horloge (*Spread Spectrum Clock*, *Clock skew Optimisation*) ne peuvent pas s'appliquer aux circuits asynchrones. Par contre, l'ordonnement de la puissance dynamique est une technique qui est intéressante. Elle a l'avantage d'exploiter la séquence des événements dans le circuit et non la répartition du signal d'horloge. Elle peut donc s'adapter aux circuits asynchrones afin d'ordonner le courant consommé. Il sera vu plus tard comment cette technique peut être appliquée aux circuits asynchrones.

5.2.2b La méthode s'applique au niveau architecture

Il est possible d'agir à plusieurs niveaux du circuit pour réduire les émissions électromagnétiques comme l'illustre la Figure 5-2.

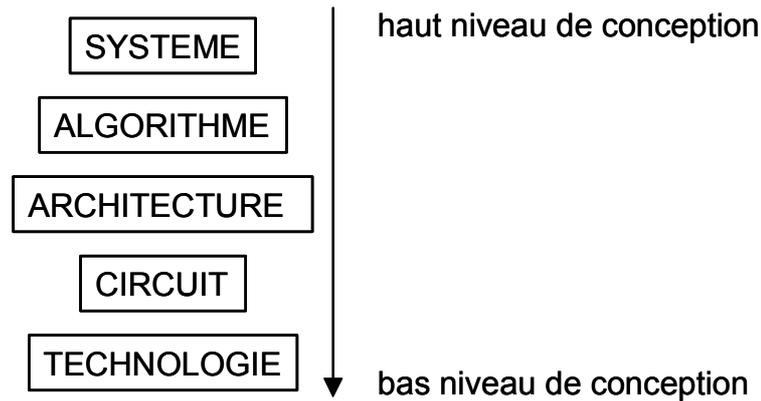


Figure 5-2 : Niveaux de conception du circuit

La méthode élaborée lors de cette étude est une méthode qui, non seulement, s'applique aux circuits asynchrones, mais qui est aussi complémentaire à la méthode *Asynchronisation* proposée dans le chapitre précédent. L'architecture du circuit est donc ici adressée et facilite l'application de la méthode : une représentation de l'architecture est utilisée, les ressources, leurs traitements et leurs latences sont connues.

Adressant les circuits au niveau registre (RTL), la technique circonscrit le temps de conception. Le circuit optimisé en vue d'une faible émission électromagnétique est obtenu dans le même langage (VHDL ou Verilog) que le langage de description du circuit d'origine ce qui permet de le simuler et de le réintégrer dans le flot de conception.

5.2.3 Principe

Les composantes élevées du spectre en courant présenté dans la Figure 5-1 proviennent principalement des appels de courant simultanés dans le circuit. Dès la conception du circuit, la concurrence des appels de courant dans le circuit peut être diminuée afin de minimiser ces pics. Les paramètres structurels des circuits sur lesquels il est possible d'agir sont, au niveau architectural, les délais caractéristiques des temps de calcul des blocs de l'architecture, et, au niveau portes logiques, les délais caractéristiques de chacune des portes.

Contrôler ces délais permet de mettre en forme le courant consommé au profit d'une réduction de l'émission électromagnétique. Et bien qu'ils soient dépendants de la technologie, les délais sont les seuls paramètres sur lesquels la méthode peut intervenir. En effet, les autres solutions comme l'introduction d'un séquenceur modifieraient l'architecture du circuit, ce qui ne permettrait pas de contenir la surface du circuit ni le temps de conception.

5.2.4 Fonction de coût

Ce paragraphe présente la fonction de coût à minimiser.

5.2.4a Activité du courant

Pour chaque élément d'un circuit, un modèle $i(t)$ (t allant de 0 à d_{max} - latence maximum de l'élément) de l'activité du courant peut être obtenu. Ce modèle (Figure 5-3) est soit de type liste de points significatifs, soit de type expression analytique.

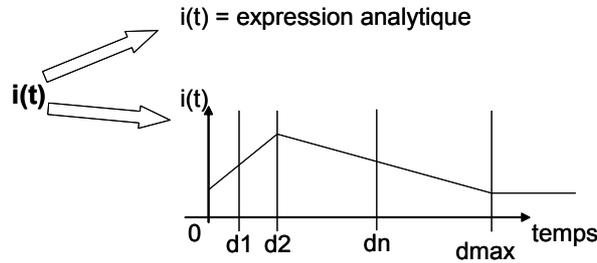


Figure 5-3 : Valeur de l'activité du courant dans un opérateur

5.2.4b Activité globale du courant

Dans un circuit où des traitements concurrents s'effectuent, l'activité globale du courant peut être estimée à l'instant t .

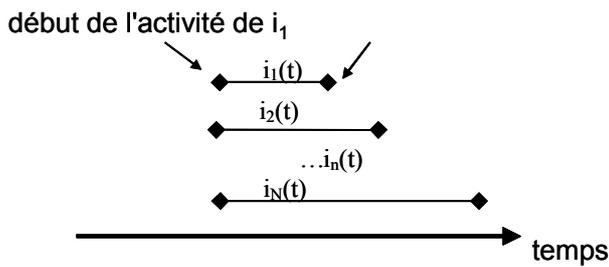


Figure 5-4 : N opérateurs effectuant leur traitement en parallèle.

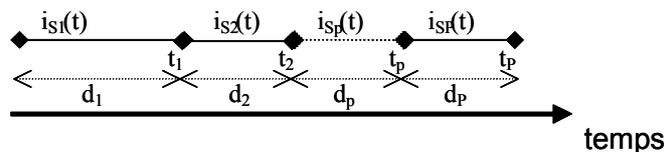


Figure 5-5 : Cas où un opérateur est composé de plusieurs opérateurs en série

La Figure 5-4 représente le cas où N éléments fonctionnent en parallèle. Les latences maximums des traitements des différents opérateurs sont modélisées par des segments. Chaque opérateur j possède sa propre latence maximum d_{jmax} ($d_{1max} < d_{2max} < \dots < d_{nmax} < \dots < d_{Nmax}$) et a pour activité électrique $i_j(t)$. Par ailleurs, il peut être lui-même une composition en série de P entités/opérateurs (Figure 5-5). Son activité devient alors :

$$i_j(t) = \begin{cases} i_{s1}(t) \text{ pour } t \in [0, t_1] \\ i_{s2}(t) \text{ pour } t \in [t_1, t_2] \\ i_{sp}(t) \text{ pour } t \in [t_{p1}, t_p] \end{cases}$$

Système Équation 5-1

et sa latence maximum : $d_j \max = \sum_{i=0}^P d_i$ (avec d_p : temps de latence maximum de l'élément p).

L'activité globale du courant à l'instant t du système représenté Figure 5-4 est égale à la somme des activités du courant de tous les opérateurs à l'instant t :

$$I_{tot}(t) = i_1(t) + i_2(t) + \dots + i_n(t)$$

Équation 5-2

$$I_{tot}(t) = \sum_{j=0}^N i_j(t)$$

Équation 5-3

Dans le système de la Figure 5-4, si le déclenchement le traitement d'un opérateur j est décalé de $D_j t$, le modèle en courant de l'élément devient :

$$I_{D_j}(t) = i_j(t - D_j t)$$

Équation 5-4

$$\text{avec : si } t < D_j t \text{ alors } I_{D_j}(t) = i_j(t - D_j t) = 0$$

Équation 5-5

Si le traitement de l'opérateur 1 est décalé de $D_1 t$, alors l'activité globale du courant dans le circuit est la suivante :

$$I_{tot_{D_1}}(t) = I_{D_1}(t) + i_2(t) + \dots + i_n(t) + \dots + i_N(t) \text{ avec } I_{D_1}(t) = i_1(t - D_1 t) \\ \text{avec } i_{D_1}(t) = 0 \text{ si } t < D_1(t)$$

$$I_{tot_{D_1}}(t) = \sum_{j=0}^N i_j(t) - i_1(t) + i_{D_1}(t)$$

Équation 5-6

D'une façon générale, lorsque le traitement de l'opérateur k est retardé de $D_k t$ ($D_k t < d_{Nmax} - d_{kmax}$) :

$$\begin{aligned}
I_{tot_{Dk}}(t) &= i_1(t) + i_2(t) + \dots + I_{Dk}(t) + \dots + i_n(t) + \dots + i_N(t) \\
\text{avec } I_{Dk}(t) &= i_k(t - D_k t) ; I_{Dk}(t) = 0 \text{ si } t < k(t) \\
I_{tot_{Dk}}(t) &= \sum_{j=0}^N i_j(t) - i_k(t) + I_{Dk}(t)
\end{aligned}$$

Équation 5-7

D'une manière générale, si tous les traitements des opérateurs dans le circuit sont retardés, l'activité du courant global dans le circuit devient :

$$\begin{aligned}
I_{tot}(t) &= I_{D1}(t) + I_{D2}(t) + \dots + I_{Dn}(t) + \dots + I_{DN}(t) \\
\text{avec } I_{Dj}(t) &= i_j(t - D_j t) ; I_{Dj}(t) = 0 \text{ si } t < D_j(t) \text{ et } I_{Dj}(t) = i_j(t) \text{ si } j = n. \\
I_{tot}(t) &= \sum_{j=0}^N I_{Dj}(t)
\end{aligned}$$

Équation 5-8

Ainsi, pour optimiser l'activité du courant en réduisant au maximum les pics de courant, il faut déterminer l'ensemble des retards $[\Delta 1t, \Delta 2t, \dots, \Delta N-1t]$ tels que :

$$\max_{t \in [0, d_n \text{ max}]} (I_{tot}(t)) = \min_{D_j t \in [0, d_n \text{ max} - Dn]} (\max_{t \in [0, d_n \text{ max}]} (I_{tot}(t)))$$

Équation 5-9

avec $j \in [0, N-1]$

5.3 Approche

Le problème d'optimisation nécessite d'aborder quatre points fondamentaux :

- La modélisation structurelle de l'architecture qui est nécessaire pour identifier des processus/opérateurs/ressources concurrents et pour répartir leur traitement.
- L'estimation du temps de calcul des blocs de l'architecture : les retards à l'intérieur du circuit permettent d'estimer le profil du courant.
- La modélisation des profils de courant des blocs de l'architecture : l'activité du courant de chaque opérateur est étudiée pour évaluer le profil du courant global.
- L'optimisation du profil du courant global par répartition des traitements des ressources afin de modéliser le courant global et de distribuer la consommation en courant.

5.3.1 Modélisation structurelle de l'architecture

Les forts pics de courant qui subsistent lors du fonctionnement des circuits asynchrones résultent de la concurrence des traitements des opérateurs dans le circuit. Ce parallélisme peut être identifié en analysant le fonctionnement du circuit. Pour cela, un graphe de représentation de l'architecture du circuit doit être utilisé et représenter :

- Le circuit dont les communications entre les blocs sont les "arcs" du graphe utilisé pour la modélisation.
- Les conditions de branchement du circuit (*if-then-else* et instruction *case*) qui ajoutent des "arcs" au graphe qui modélise l'architecture.

Ainsi un choix entre 2 types de graphes se présentent : le *Control Data Flow Graph* ou le PN/DFG (Petri Net/Data flow Graph) présentés ci-après.

5.3.1a DFG/CDFG

Le CDFG (*Control Data Flow Graph*) est un graphe qui représente le fonctionnement du circuit. Ce type de représentation est déjà utilisé pour l'ordonnancement lors de la synthèse comportementale. Il est cette fois réutilisé pour l'analyse et l'ordonnancement de la méthode élaborée.

La Figure 5-6 illustre un circuit (a.) et son CDFG (b.).

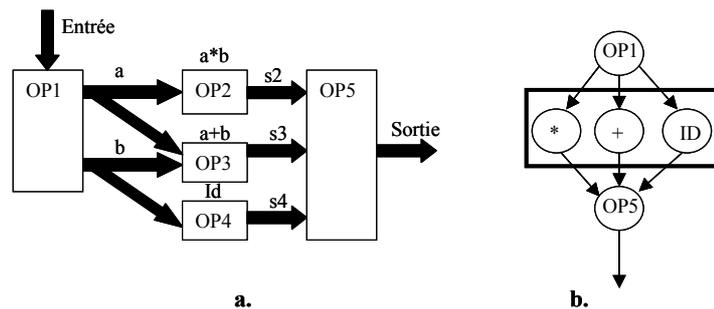


Figure 5-6 : Un circuit (a.) et son CDFG (b.)

Dans ce graphe, chaque place, également appelé noeud, est marquée par une opération et chaque arc représente une liaison entre les ressources. Le *Data Flow Graph* (DFG) est une version simplifiée du CDFG qui ne tient compte que de la séquence des traitements sans les choix de contrôle et qui ne prend pas en compte la cyclicité.

5.3.1b PN/DFG

Le PN/DFG est une représentation qui rassemble deux types de graphes : les Réseaux de Pétri [DIAZ01] et les *Data Flow Graph*. Cette représentation est fournie par l'outil TAST. Le compilateur fournit cette forme intermédiaire pour la synthèse, la vérification...

5.3.1c Choix du graphe de représentation de l'architecture

Les CDFG et PN/DFG remplissent les conditions énumérées précédemment, ils peuvent être l'un et l'autre utilisés pour la méthode de répartition en courant. Mais le choix entre les 2 types de représentations dépend aussi de l'algorithme utilisé pour la répartition. C'est ainsi que le choix va se tourner vers la représentation couramment utilisée pour l'ordonnancement : le CDFG. Ce choix est effectué dans un premier temps et n'exclue pas l'utilisation des PN/DFG à l'avenir.

5.3.1d Compositions

Les motifs suivants peuvent être rencontrés dans le CDFG qui représente l'architecture du circuit :

La séquence (Figure 5-7 a.)

Le parallélisme (Figure 5-7 b.)

Le choix (Figure 5-7 c.)

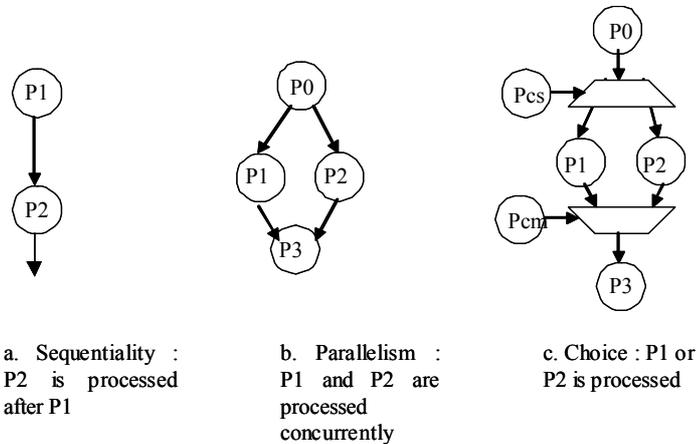


Figure 5-7 : Motifs d'un CDFG

Les concurrences des traitements sont les principales cibles de ce travail et sont facilement identifiables (motif de parallélisme) dans la représentation du circuit. Cependant, il faut noter que d'autres concurrences moins explicites existent et sont rencontrées dans les autres motifs :

- La concurrence des activités dans les canaux de communication. En effet, lorsqu'un protocole de communication s'effectue, les portes dans le contrôle peuvent être activées en parallèle.

- La concurrence des traitements des ressources dues à l'architecture pipeline du circuit ou à son fonctionnement cyclique. Il faut dans ce cas explorer la globalité de l'architecture.

5.3.2 Estimation du temps de calcul des blocs de l'architecture

Les délais dans les parties combinatoires sont estimés au niveau fonctionnel soit à partir de délais typiques, soit après synthèse structurelle de l'architecture à l'aide d'outils estimant les délais au niveau porte logique. Dans ce dernier cas, les délais sont plus précis, mais un travail de rétro annotation doit être effectué.

5.3.3 Modélisation des profils de courants des opérateurs

5.3.3a Analyse du protocole : Phases de l'activité en courant

L'activité en courant d'un opérateur dépend du protocole de transmission utilisé. Elle peut être décomposée en différentes phases (Figure 5-9). Ce point sera pris plus en détail dans le chapitre 6.

Un protocole 4-phases WCHB [RIGA02], par exemple, peut être décomposé en 3 phases. Chaque phase correspond à une action principale d'un étage : arrivée de données, ouverture du *latch*, et fermeture du *latch*. L'analyse du comportement de l'opérateur *B* de la Figure 5-8 justifie cette décomposition :

Phase 1: Les données valides arrivent de l'étage précédent *A*, qui envoie le signal de requête (*ReqA*). La logique combinatoire de l'opérateur *A* traite les données reçues. La fin du calcul marque l'accomplissement de la phase.

Phase 2: Le signal *enable* commute et laisse les données passer à travers les *latches A*. L'étage *B* envoie une requête (*ReqB*) à l'étage suivant *C* et un acquittement (*AckB*) à l'étage *A*.

Phase 3: L'étage *B* attend le signal d'acquiescement (*AckC*) de l'étage *C* et la remise à zéro du signal de requête (*ReqA*) de l'étage *A*. Ensuite, le signal *enable* passe à zéro : les *latches* sont bloqués. Puis l'étage *B* cesse d'envoyer le signal de requête (*ReqB*) à l'étage *C* et le signal d'acquiescement (*AckB*) à l'étage *A*.

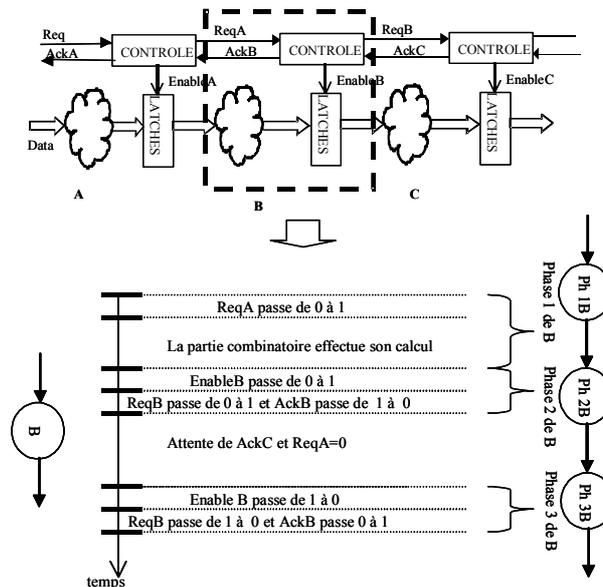


Figure 5-8 : Activité du courant dans l'opérateur B

5.3.3b Raffinement du CDFG en phases de courant

Par conséquent, chaque noeud du CDFG peut être raffiné en utilisant trois noeuds modélisant les phases du protocole. Cependant, ce modèle peut être simplifié selon les

synchronisations qui existent. Par exemple, dans le protocole WCHB, la phase 3 peut être fusionnée avec la phase 2 car elle est synchronisée avec elle.

5.3.3c Modèle de courant dans une phase

Le courant consommé à chaque phase peut être représenté par un triangle (Figure 5-9).

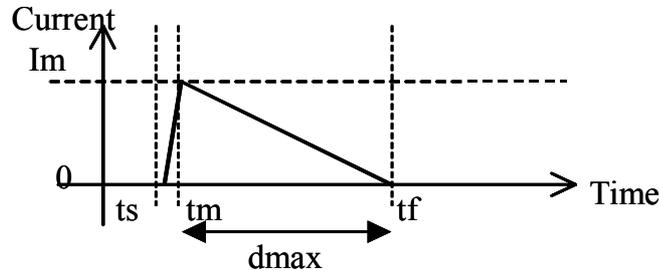


Figure 5-9 : Modélisation d'un opérateur par ses phases d'activité en courant (a.) Modélisation de l'activité du courant d'une phase (b.)

Comme dans [BENI97] : les caractéristiques du triangle sont les suivantes :

ts est l'instant où le courant croît et atteint 1% de sa valeur maximale

tf est l'instant où le courant décroît et descend en dessous de 1% de sa valeur maximale.

Im est la valeur maximale du courant

tm est l'instant où la valeur maximale du courant est obtenue.

Cette représentation permet de déterminer approximativement la position des pics de courant.

Cette estimation du courant est suffisante à ce stade de conception. Le triangle $\Delta_i^{phase1}(t)$, représentant la première phase de l'opérateur i , capture l'activité de la logique combinatoire. Les triangles, représentant la deuxième phase $\Delta_i^{phase2}(t)$ et la troisième phase $\Delta_i^{phase3}(t)$, capturent essentiellement les profils courants dus au protocole de communication.

5.3.3d Allure du courant global

En annotant le CDFG représentant le circuit avec les modèles de courant des opérations et des phases de communication et en utilisant l'information de temps, le profil du courant global du circuit peut être estimé. Le courant total consommé par le circuit est :

$$I_{tot}(t) = \sum_0^N \Delta_i^{phase1}(t) + \sum_0^N \Delta_i^{phase2}(t) + \sum_0^N \Delta_i^{phase3}(t)$$

Équation 5-10

5.3.3e Influence de la modélisation des profils de courant sur la fonction de coût

Le principe de la méthode est de décaler localement le temps d'exécution des blocs qui n'appartiennent pas au chemin critique. Si un opérateur i est retardé d'un temps Di , l'activité du courant de chacune de ses phases est $\Delta_i^{phase}(t, Di)$. En considérant que tous les opérateurs du

circuit sont retardés, le courant total s'exprime par : $I_{tot}(t,D)=\sum_0^N \Delta_i^{phase1}(t,Di)+\sum_0^N \Delta_i^{phase2}(t,Di)+\sum_0^N \Delta_i^{phase3}(t,Di)$ où t est le temps et D l'ensemble de retards, avec $D_j=0$ quand l'opérateur j est sur le chemin critique. Le courant crête maximum est déterminé par la fonction de coût trouvée en [BENI97]. Le problème de la répartition du courant est maintenant formulé comme suit : étant donné $F(t)=\max_{t \in [0, CircuitLatency]} \{I_{tot}(t,D)\}$, trouver l'ensemble de retard (d) qui réduit au minimum $F(t)$ sans augmenter la latence de circuit. Les retards seront insérés dans les canaux de communications afin de retarder l'exécution des opérateurs.

5.3.4 Optimisation du profil du courant global par ordonnancement

A partir d'une description RTL du circuit, la séquence des opérations est connue, le parallélisme entre celles ci aussi. Dans le cas d'un parallélisme, le déclenchement simultané des opérations provoque un fort appel de courant. Pour réduire l'amplitude de cet appel de courant dans ce type de structure, les traitements des opérations peuvent être déclenchés à des instants différents. L'activité du courant est ainsi répartie, le problème considéré est donc un problème d'ordonnancement.

5.3.4a Définition des pas en temps

L'ordonnancement des activités dans le circuit est ici un problème d'optimisation contraint par la latence du circuit et l'activité en courant des opérateurs.

Lors de la synthèse comportementale des circuits synchrones, l'algorithme d'ordonnancement répartit les traitements des ressources dans des pas de temps qui décomposent le cycle de fonctionnement du circuit. Ces pas, appelés aussi pas de contrôle, ont une valeur définie par la période de l'horloge.

Pour pouvoir appliquer un algorithme d'ordonnancement à la description d'un circuit asynchrone, le temps de calcul du circuit doit être divisé en pas de temps. Une fois ces pas définis, l'algorithme peut leur affecter les traitements des ressources afin de répartir l'activité du courant comme cela se passe lors de l'ordonnancement dans les circuits synchrones. Les pas de temps T_{step} sont définis tels que :

$$N \cdot T_{step} = \min_{i \in I} (mobility(OP_i))$$

Équation 5-11

où I est l'ensemble des opérateurs qui ne sont pas sur le chemin critique. $Mobility(OP_i)$ est la mobilité de l'opérateur i . Cette mobilité est définie comme le délai entre les déclenchements d'un opérateur i après ordonnancement ASAP (*As Soon As Possible*) et ALAP (*As Late As Possible*) (voir 5.3.4b). N définit la limite inférieure de la gamme de mobilité par $N-1$.

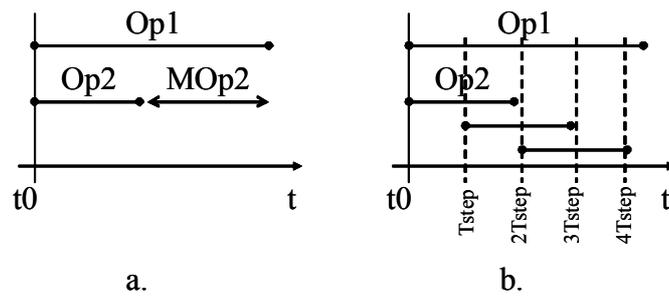


Figure 5-10 : Définition des pas en temps

La Figure 5-10 illustre comment le pas de temps unitaire $Tstep$ est calculé sur un exemple simple. $Op1$ et $Op2$ sont deux opérateurs parallèles. $Op1$ est sur le chemin critique donc sa mobilité est 0. Quant à l'opérateur $Op2$, celui-ci a pour mobilité $MOp2$.

Le calcul est le suivant : $I=\{1\}$ et $mobility(Op2)=MOp2$. N est choisi arbitrairement à 2. Par conséquent la valeur du pas de temps est : $Tstep=MOp2/2$. Le temps de calcul est découpé en tranches avec $Tstep$, et $Op2$ peut être affecté de trois façons différentes comme représenté sur la Figure 5-10b. Plus N est élevé, plus il y a de flexibilité pour répartir les opérations à l'aide du processus d'ordonnancement, et plus l'optimisation est meilleure. Cependant, ceci se fait au prix d'un calcul plus complexe.

5.3.4b Choix de l'algorithme

L'algorithme généralement utilisé pour l'étalement de spectre dans les circuits synchrones est un algorithme génétique qui requiert beaucoup de temps de calcul. Ici, les algorithmes d'ordonnancement utilisés pour la synthèse comportementale conviennent à la méthode élaborée car leur coût en temps de calcul est faible.

Rappel sur l'ordonnancement [WALK95][SUGA00][RAHM97]

A la synthèse comportementale d'un circuit synchrone, l'algorithme d'ordonnancement permet de déterminer l'ordre et la séquence des opérations à effectuer durant un cycle de calcul. En fonctionnement synchrone, celui-ci est divisé en étapes de contrôle de durée égale à la période d'horloge. Ces étapes correspondent aux différents états du séquenceur qui régit l'ensemble du système. Affecter une opération à une étape de contrôle revient à déterminer l'état du séquenceur qui envoie le signal de commande de l'opération.

Les principales approches d'ordonnancement lors de la synthèse comportementale sont décrites ci-après (Tableau 5-1).

Différentes contraintes peuvent être imposées durant l'ordonnancement :

- La contrainte en nombre de ressources (RCS de l'anglais *Resources Constraint Scheduling*)
- La contrainte en temps (TCS de l'anglais *Timed Constraint Scheduling*)
- La contrainte en temps et en nombre de ressources (TRCS)
- Aucune contrainte (UCS de l'anglais *Unconstrained Scheduling*)

Classement effectué pour les approches d'ordonnancement	
Arrangement d'ordonnancement et allocation indépendants	L'ordonnancement est traité indépendamment de l'allocation:
Arrangement d'ordonnancement et allocation interdépendants	L'ordonnancement est traité simultanément avec l'allocation:
Ordonnancement et Allocation par raffinement pas à pas	Allocation, puis estimation de l'ordonnancement etc. cas du système HAL [] qui comprend l'algorithme Force Directed Scheduling.
Ordonnancement Pipeline	L'algorithme global est divisé en plusieurs étapes. Chaque étapes de l'algorithme est appliquée sur le chemin de donnée d'une façon pipelinée.

Tableau 5-1 : Types d'ordonnancement

Algorithmes [GOVI95]

Pour répondre à ces contraintes, différents types algorithmes peuvent être utilisés, les principaux sont :

ASAP/ALAP (*As-Soon/As-Late As Possible*) : ces types d'algorithmes ordonnent les opérations dans un ordre fixe en affectant une opération dans l'étape de contrôle qui va s'exécuter le plus tôt/tard possible. Ce sont des algorithmes généralement utilisés pour UCS car ils sont rapides et simples à implémenter. Cependant, ils ne conviennent pas aux problèmes de type TCS et RCS.

Le *List Scheduling* ordonne les opérations d'une manière séquentielle : il choisi dans une liste d'opérations prêtes à être affectées, sans violation de contrainte de ressources et selon un ordre de priorité qui dépend beaucoup de la structure du graphe du flot de données. Cet algorithme convient pour RCS.

Le *Force Directed Scheduling* (FDS) ordonne en minimisant des forces. Une force est attribuée à chaque opération pour chaque étape de contrôle. Cette force est proportionnelle au nombre d'opérations de même type qui peut prendre place pendant cette étape. Cet algorithme convient pour TCS.

Integer Linear Programming (ILP) [SHIU00] est une méthode de programmation en nombres entiers qui utilise des variables de décisions x_{ij} . Celle-ci valent 1 si l'opération est placée dans l'étape de contrôle et 0 sinon. Les équations sont paramétrées par les contraintes en temps et en ressources. Cet algorithme convient pour TCRS. Il demande un temps de calcul long.

Pour satisfaire le projet, le choix entre ces différents algorithmes va être déterminé par les critères suivants :

- L'algorithme doit pouvoir gérer les pipelines : la répartition du courant doit se faire sur tous les groupes qui s'effectuent concurremment. Le pipeline est donc une forme de concurrence qui ne doit pas être négligée. La Figure 5-11 montre le cas du filtre asynchrone implémenté pour le projet. La structure pipeline induit des concurrences des traitements de plusieurs parties du

circuit. La fsm, le buff2 et le multiplieur vont fonctionner en même temps ainsi que l'ensemble : accumulateur, rom, multiplexeur et buff1.

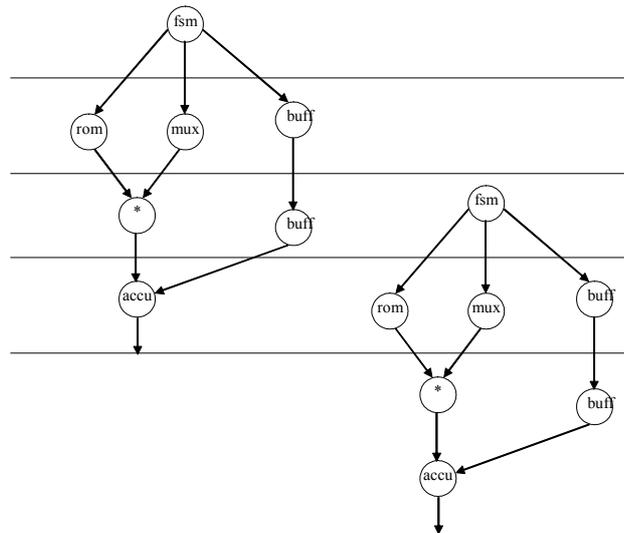


Figure 5-11 : Pipeline

L'algorithme doit aussi tenir compte des parallélismes imbriqués, c'est-à-dire des parallélismes contenus dans des structures elles-mêmes parallèle. Dans le cas de la Figure 5-11, la ROM et le multiplexeur sont concurrents et le groupe formé par ceux-ci et le multiplieur est lui-même concurrent avec les deux *buffers* en série. Dans le cas où l'architecture possède des parallélismes imbriqués, la méthode de répartition du courant doit être appliquée d'une façon globale ou incrémentale selon les algorithmes utilisés.

Les Eléments à répartir

Dans le cas de l'étude, plusieurs éléments peuvent être sujets à l'ordonnancement.

- Les traitements des opérateurs dans le circuit
- Les phases d'activité du courant de ces opérateurs
- Les pics de courant significatifs produits par les ces opérateurs

Le choix dépend des performances souhaitant être obtenues en terme d'émission électromagnétique. Il est à noter que plus le niveau de granularité est élever, plus l'optimisation est efficace, mais plus les temps de calcul sont longs.

Ainsi le choix suivant est fait : les phases d'activité du courant des opérateurs sont adressées et les pics de courant significatifs sont pris en compte grâce à une pondération des modèles des phases en courant. La Figure 5-12 montre la discrétisation d'un modèle de courant. L'amplitude de chaque échantillon est égale à la valeur de sa surface sur un pas. C'est cette amplitude qui détermine le poids d'une opération dans un pas.

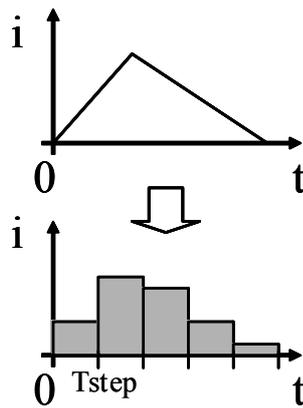


Figure 5-12 : Discretisation du modèle de courant

5.3.4c Force Directed Scheduling

L'algorithme d'ordonnancement pour la synthèse de haut niveau proposé par Pierre Paulin (*Force Directed Scheduling*)[PAUL86][PAUL89] permet de répondre aux besoins de la méthode. En effet, il satisfait les ordonnancements contraints sur le temps de fonctionnement global du circuit comme c'est le cas dans cette étude (ce temps ne doit pas être augmenté). Il calcule une solution optimale tout en utilisant des capacités de calcul raisonnables. Son but est de réduire la concurrence entre les opérations afin d'utiliser le moins de ressources possibles. Dans son concept, il cherche à répartir au mieux la charge de calcul au cours du temps, ce qui est l'objectif de cette méthode. Les paragraphes suivants détaillent l'algorithme *Force Directed Scheduling*.

Les forces calculées dans l'algorithme FDS

L'impact de l'affectation d'une opération à une étape de contrôle sur l'ensemble du circuit peut être modélisé par une somme de forces appelée force totale. Celle ci regroupe:

- La force propre de l'opération qui modélise les contraintes dues à la présence l'opération à l'étape de contrôle.
- Les forces induites : L'affectation d'une opération à une étape de contrôle impose parfois l'affectation des opérations précédentes et successives à d'autres étapes de contrôle. Les forces induites modélisent les contraintes dues aux placements engendrés des opérations précédentes et/ou successives.

Chaque force peut être illustrée grâce au système mécanique masse-ressort (Figure 5-13). La raideur du ressort représente la probabilité qu'un type d'opération soit présent à une étape de contrôle et la masse représente l'opération à affecter.

Plus la raideur du ressort est élevée, plus la force à appliquer pour positionner la masse sur le ressort sera élevée. Si une opération est placée à l'étape de contrôle dans laquelle elle a le plus la probabilité d'être présente, alors la concurrence entre les opérations est élevée. Par conséquent, plus une force est élevée, plus des tensions sont créées dans le système et plus la concurrence des opérations est élevée. Le but de l'algorithme est d'ordonner les opérations afin d'équilibrer le système en minimisant les forces en présence.

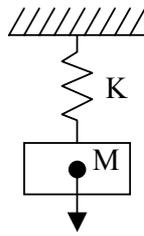


Figure 5-13 : Modélisation par une force d'une affectation d'une opération à un pas

Description de l'algorithme FDS [PAUL89]

L'exemple de l'article illustre bien le comportement de l'algorithme. Voici un circuit représenté par son (CDFG) (Figure 5-14) et dont la description algorithmique est :

```

y'' + 3xy' + 3y = 0 :
tant que (x < a) faire
  xl = x + dx ;
  ul = u - (3*x*u*dx) - (3*y*dx) ;
  yl = y + (u*dx) ;
  x = xl; u = ul; y = yl;
fin tant que.
  
```

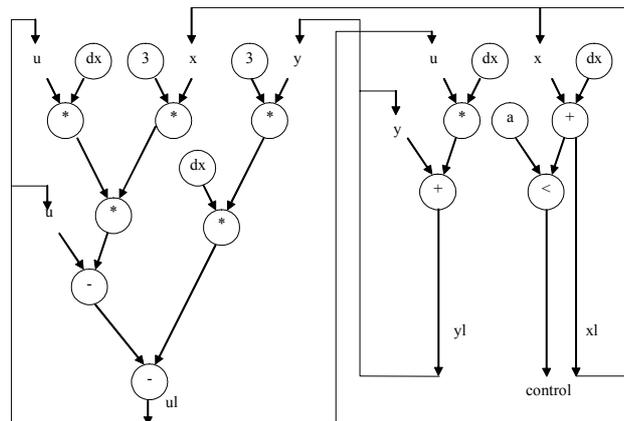


Figure 5-14 : CDFG de l'exemple

Le nombre d'étapes de contrôle nécessaires pour effectuer un cycle de calcul est déterminé par le chemin critique du circuit. Ici le calcul le plus long, correspondant à la partie gauche de la Figure 5-14, s'effectue en 4 étapes de contrôles (appelés aussi c-step). La dernière soustraction ne peut être effectuée qu'à la dernière étape de contrôle.

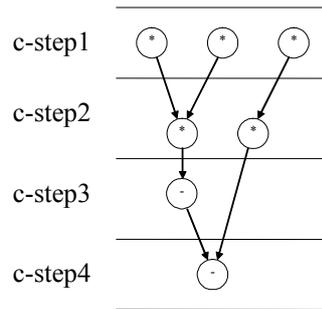


Figure 5-15 : Pas de contrôle

L'ordonnancement s'effectue en 5 étapes qui se rebouclent tant toutes les opérations ne sont pas affectées à des étapes de contrôle.

Etape 1 : "time Frames".

L'ordonnancement ASAP (*As Soon As Possible*) et l'ordonnancement ALAP (*As Late As Possible*) sont donnés Figure 5-16. Pour la version ASAP de gauche, un maximum de calculs est effectué aux étapes initiales (*c-step1*, *c-step2*). Pour la version ALAP de droite, un maximum de calculs est effectué aux étapes finales (*c-step3*, *c-step4*).permettent d'évaluer les étapes de contrôle pendant lesquelles peut s'exécuter une opération (*time frame*).

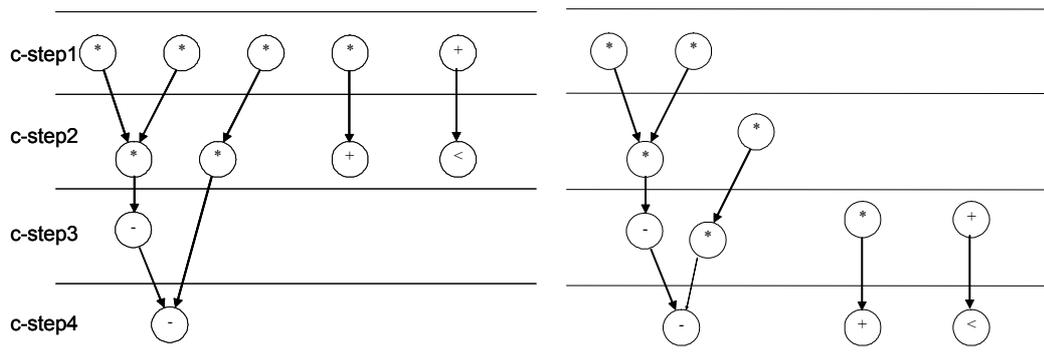


Figure 5-16 : ASAP (à gauche) et ALAP (à droite) de l'exemple de la figure 5.14

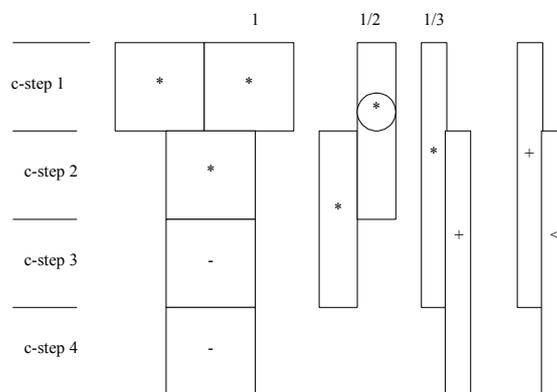


Figure 5-17 : Time Frame de l'exemple 1

La Figure 5-17 représente les *times frames* des opérations. Par exemple, l'opération de multiplication entourée peut être effectuée à l'étape de contrôle *c-step1* ou au *c-step2* (confrontation des ordonnancement ASAP et ALAP). Ainsi une probabilité d'1/2 lui est attribuée pour qu'elle soit assignée à chacune de ces étapes de contrôle.

Étape 2 : Génération des diagrammes de distribution

Pour chaque type d'opération, le diagramme de distribution (DG de l'anglais *Distribution Graph*) représente la somme des probabilités d'affectation de ce type d'opération sur chaque étape de contrôle. Les étapes de contrôles encadrées correspondent aux étapes où l'opération peut être effectuée.

Dans la Figure 5-18, le DG de la multiplication montre qu'à la première étape de contrôle, la probabilité d'avoir une opération de multiplication est de 2.833. Le cadre indique les étapes de contrôle où l'opération peut être effectuée. Ici, aucune opération de multiplication ne peut être effectuée dans le quatrième c-step.

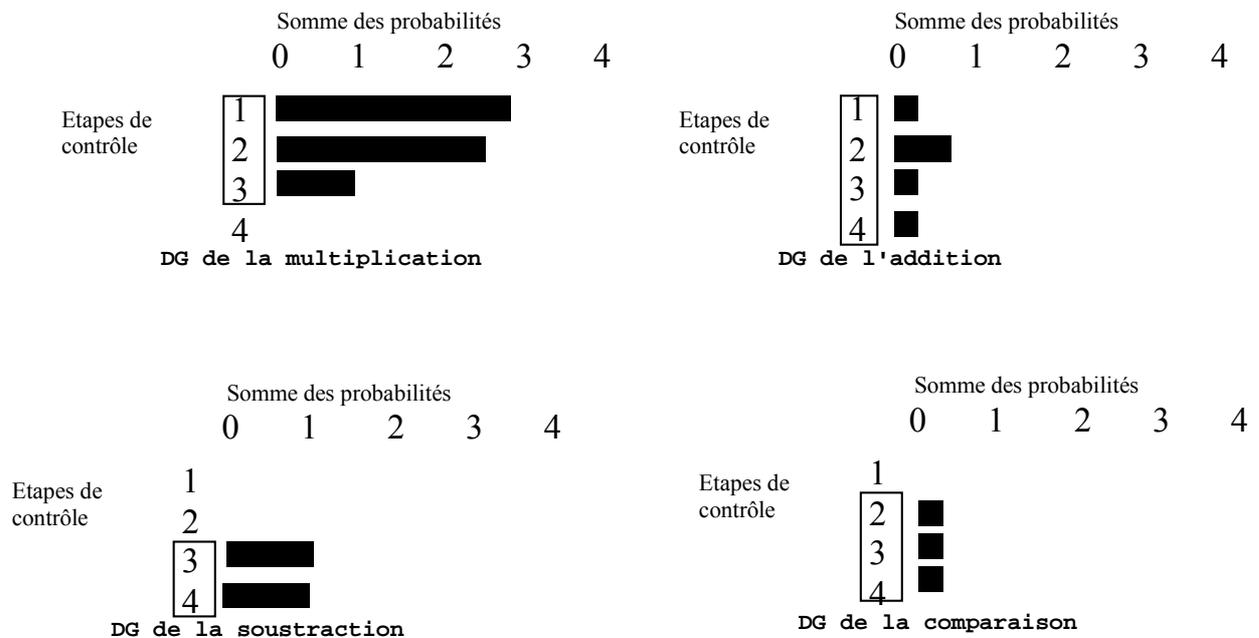


Figure 5-18 : Diagrammes de distribution

Étape 3 : Calcul des forces propres des opérations pour toutes les étapes de contrôle où elles peuvent être affectées.

Le diagramme de distribution d'un type d'opération peut être modélisé par une série de ressorts. Chaque ressort a une raideur K_i (somme des probabilités du type d'opération à l'étape i considérée = $DG(i)$). La Figure 5-19 représente le système de ressorts équivalent au DG de la multiplication.

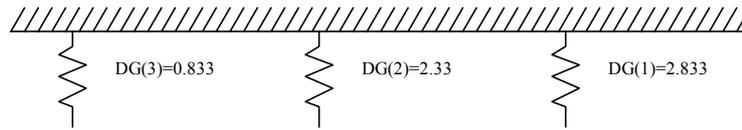


Figure 5-19 : Modélisation d'un DG

Lorsqu'une opération n'est pas encore affectée à une étape : le système ressort-masse (diagramme de distribution/opération) est à l'équilibre. Sa masse ou probabilité d'être affectée à l'étape i est p_i (Figure 5-20).

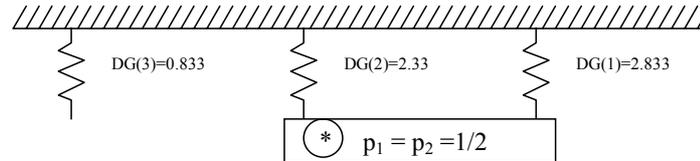


Figure 5-20 : Modélisation de la probabilité d'une affectation d'une opération à une étape de contrôle

Si l'opération est affectée à un c -step i , alors sa probabilité est forcée à 1. La probabilité de la masse à l'étape i varie donc de $x_i = 1 - p_i$. La force F_i appliquée à l'étape i est égale à :

$$K_i \cdot x_i.$$

Équation 5-12

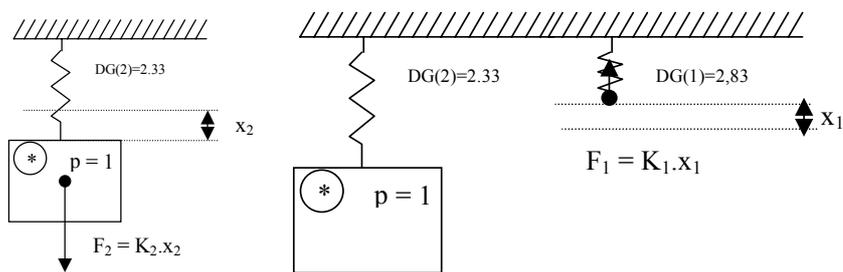


Figure 5-21 : Variation d'une force

Dans l'exemple de la Figure 5-21, l'opération de multiplication entourée est affectée à l'étape 2. Sa probabilité est passée de 0.5 à 1 donc $x_2 = +0.5$. La force associée est :

$$F_2 = DG(2) \cdot x_2 = 2,33 \cdot 0,5 = 1.165.$$

Équation 5-13

Si l'opération avait une probabilité d'être présente aux étapes suivantes $j, k \dots$ alors les forces respectives $F_j, F_k \dots$ appliquées aux ressorts suivants sont égales à

$$F_j = K_j \cdot x_j = K_j \cdot (0 - p_j), F_k = K_k \cdot x_k = K_k \cdot (0 - p_k) \dots$$

Équation 5-14

La force propre d'une opération lorsqu'elle est affectée à une étape i est égale à la somme des forces appliquées :

$\text{Self_force}_{\text{opération}}(\text{étape } i) = \text{somme des forces appliquées à la série de ressorts.}$

Équation 5-15

Pour le cas de la Figure 5-21, lorsque la multiplication entourée est affectée à l'étape 2, la force exercée sur le ressort de l'étape 1 est :

$$F1 = DG(1).x1 = 2.83 * -0.5 = -1.415.$$

Équation 5-16

La force propre de la multiplication entourée affectée à l'étape 2 est:

$$\text{Self force (2)} = F1 + F2 = DG(1)*x(1) + DG(2)* x(2) = -0.25$$

Équation 5-17

Etape 4 : Calcul des forces totales pour chaque affectation d'opération.

Si une opération succède ou précède une autre opération, alors lorsqu'elle est affectée à une étape, ceci influe sur la distribution de probabilité de l'autre opération. Il faut donc tenir compte des forces appliquées sur les opérations précédentes et successives.

Force totale appliquée au système est donc:

$$\text{Total_force} = \text{self_force} + \text{succ_force} + \text{pred_force.}$$

Équation 5-18

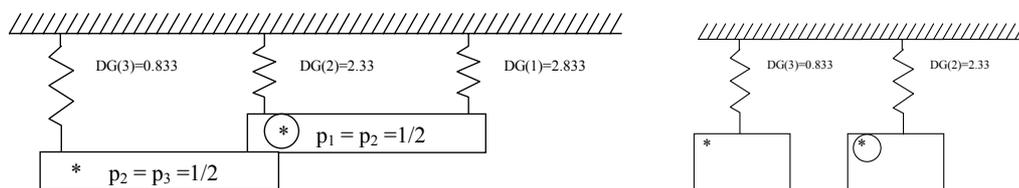


Figure 5-22 : Influence de l'affectation d'une opération sur une autre

Pour l'exemple précédent (Figure 5-22), une multiplication succède à la multiplication entourée et ne peut être effectuée qu'à la troisième étape de contrôle.

$$\text{Succ_force (2)} = \text{self_force}_{\text{opération suivante}}(3) = K2.x2 + K3.x3 = 2.33*-0.5 + 0.33*0.5 = -1$$

Équation 5-19

donc

$$\text{Totalforce (2)} = \text{self_force}(2) + \text{succ_force}(2) = -1.25$$

Équation 5-20

Etape 5 : Ordonnancement.

Une fois toutes les totales forces calculées, l'opération ayant la plus petite force totale est affectée à l'étape correspondante. Son time frame est donc modifié. L'algorithme retourne ensuite à l'étape 1.

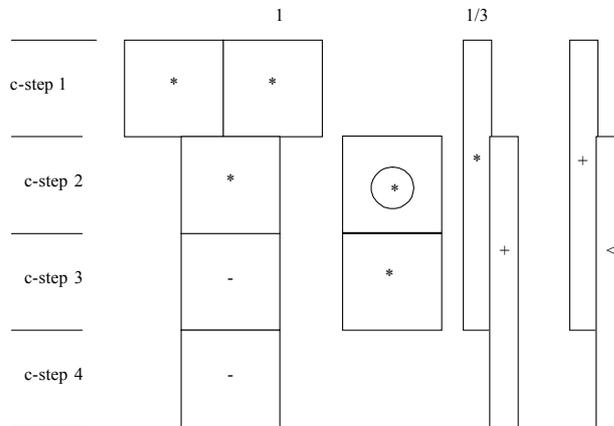


Figure 5-23 : Modification du *time frame 2*

5.4 Méthode

Les informations nécessaires décrites précédemment (modèle de l'architecture, temps de calcul des blocs, modèles de courant, algorithme d'ordonnancement) permettent de définir la méthode "mise en forme du courant".

5.4.1 Définition de la méthode

La méthode se compose des étapes suivantes :

- Le modèle de l'architecture est analysé : les concurrences sont identifiées grâce aux motifs décrits dans 5.3.1d.
- L'utilisation des modèles de courant (5.3.3) qui sont définis auparavant grâce à l'analyse de l'activité électrique dans les opérateurs et les protocoles de communication ainsi que l'extraction des délais après synthèse.
- Les pas de contrôle sur la latence du circuit sont calculés (5.3.4a). Les modèles de courant sont discrétisés en temps et en amplitude (Figure 5-12). La répartition des tâches des opérateurs peut être effectuée.
- L'algorithme d'ordonnancement (ici le *Force Directed Scheduling* ([PAUL89]) est appliqué.

La méthode (Figure 5-24) permet ainsi de calculer les délais à insérer dans les communications afin de lisser l'activité du courant. Les valeurs de ces délais sont des multiples de celle du pas de temps. Certaines communications sont donc retardées par rapport à d'autres dans les traitements concurrents.

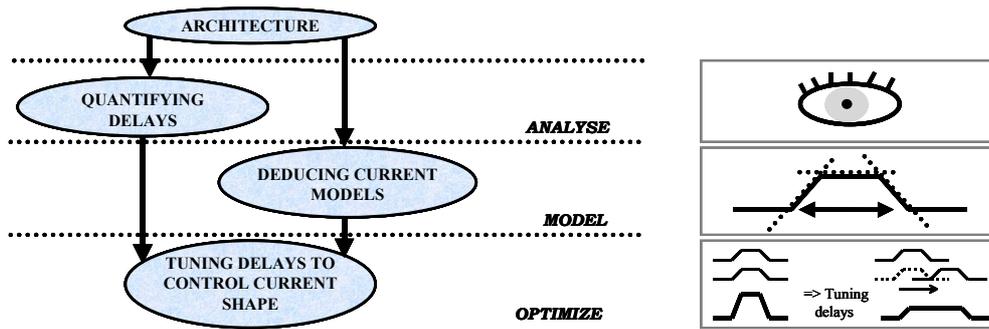


Figure 5-24 : Méthode pour la répartition du courant

5.4.2 Illustration de la méthode

L'exemple de la Figure 5-6 est repris pour illustrer la méthode et ses différentes étapes.

5.4.2a Analyse du modèle de l'architecture

Le modèle est avant tout analysé pour permettre d'identifier le type de liaison entre les opérations et de définir la probabilité des affectations d'une opération à un pas. A cette étape, les concurrences de l'exemple peuvent être directement remarquées (Figure 5-25): la multiplication, l'addition et l'identité ont un fonctionnement parallèle. Il est à noter que le fonctionnement récurrent de la structure n'est pas pris en compte : pour des raisons de simplification, le circuit traite une donnée en entrée lorsqu'il délivre sa sortie.

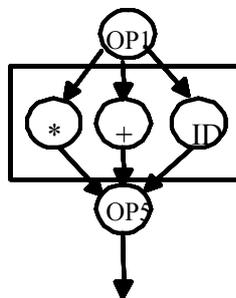


Figure 5-25 : Des concurrences sont identifiées

A ce stade, l'architecture du circuit et les portes qui le composent sont connues.

5.4.2b Utilisation des modèles de courant

Les opérateurs peuvent être modélisés par les phases de leur activité en courant. Ici, le protocole WCHB est utilisé, le modèle simplifié (contenant seulement 2 phases) peut être utilisé. Les latences des différents opérateurs peuvent être grossièrement déterminées après synthèse. Les latences des différentes phases des activités du courant dans les opérateurs ainsi que la forme du courant (qui va permettre de définir les triangles) sont déterminées après simulation électrique.

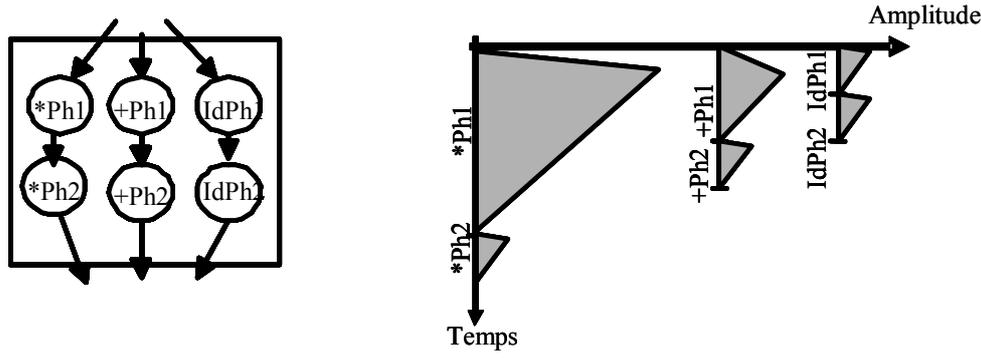


Figure 5-26 : Utilisation des modèles de courant

La Figure 5-26 montre l'utilisation des modèles en phases : le CDFG a été raffiné et les places ne représentent plus les opérateurs, mais leur phase d'activité.

5.4.2c Définition de pas en temps (ou pas de contrôle) et discrétisation des modèles.

Des pas de temps sont définis en fonction de la latence des différents opérateurs et calculés selon 5.3.4a (Figure 5-27).

Sub Block latencies :

- *Ph1 = 8ns
- *Ph2 = 2ns
- +Ph1 = 4ns
- +Ph2 = 2ns
- IdPh1 = 2ns
- IdPh2 = 2ns
- ➔ Tstep = 2ns

Figure 5-27 : Calcul des pas

Une fois les pas de temps connus, les modèles en courant peuvent être discrétisés en temps et amplitude comme le montre la Figure 5-28

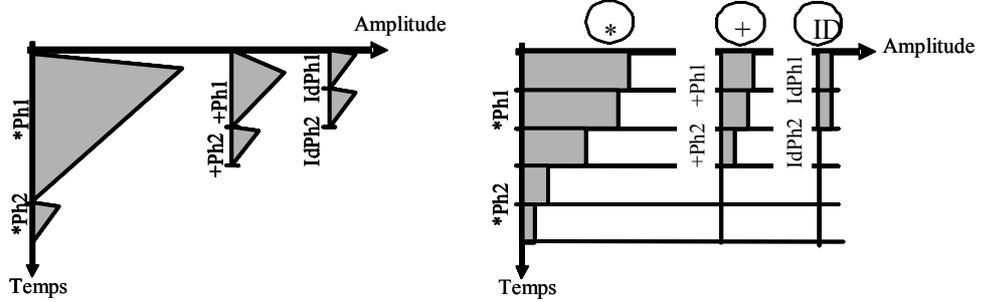


Figure 5-28 : Discrétisation des modèles

5.4.2d Application de l'algorithme

Au final, l'algorithme *Force Directed Scheduling* est appliqué pour répartir l'activité du courant d'une manière uniforme. La Figure 5-29 montre la répartition des activités des différents opérateurs.

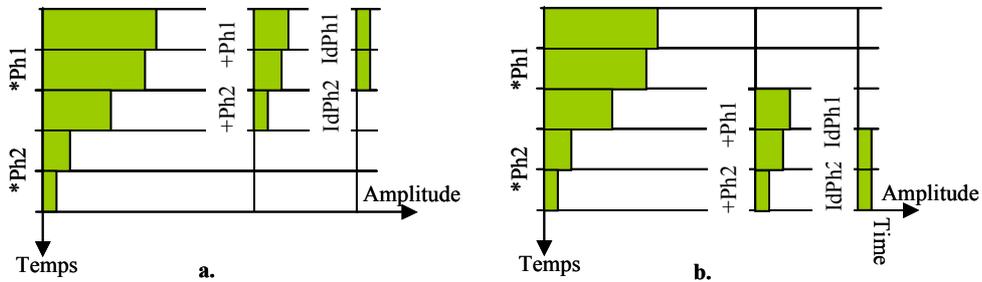


Figure 5-29 : Allure du courant dans chaque opérateur avant (a.) et après (b.) application de la méthode

La Figure 5-30 montre l'allure du courant global après application de l'algorithme.

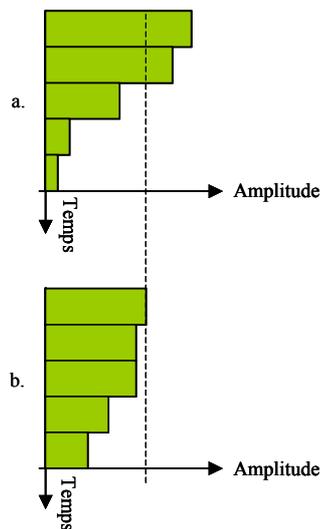


Figure 5-30 : Allure du courant global avant (a.) et après (b.) application de la méthode

5.5 Implémentation

5.5.1 Filtre à Réponse Impulsionnelle 4 coefficients

Un filtre dont le fonctionnement est identique à celui étudié dans le Chapitre 4 est utilisé. Le filtre asynchrone développé est un filtre de catégorie micropipeline dont la représentation est donnée Figure 5-31. Ce filtre utilise un protocole standard 4 phases WCHB, ainsi les rectangles noirs représentent la présence de *half-buffers*.

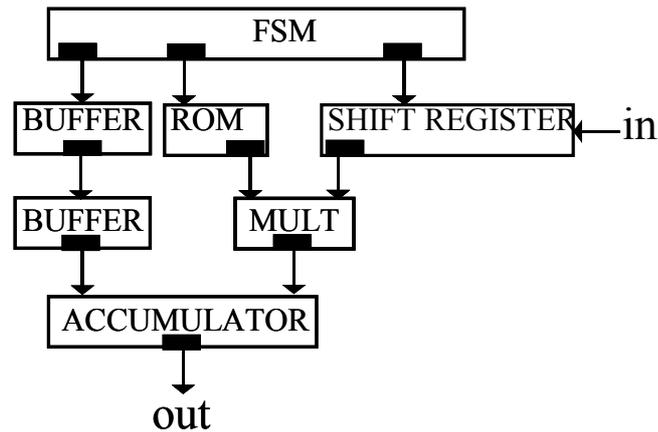


Figure 5-31 : Filtre FIR

La Figure 5-32 représente le modèle de l'architecture sous forme de *Control Data Flow Graph* où 2 concurrences sont identifiées : $\{fsm, regdec, buff1\}$; $\{mult, buff2, mult\}$.

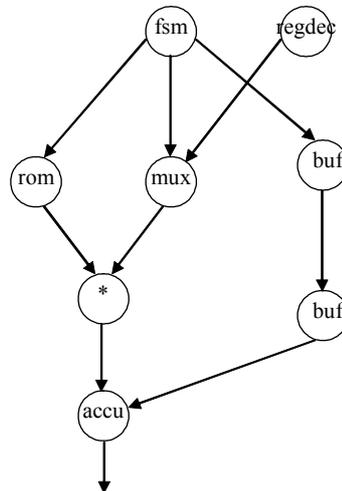


Figure 5-32 : CDFG du filtre

5.5.2 Application de la méthode

Le filtre a été développé dans la technologie 0,18 microns de STMicroelectronics. Les simulations électriques ont pu fournir les temps de latence et forme du courant de chaque opérateur et ses phases.

Le calcul du pas de temps est effectué. L'algorithme est appliqué. Les délais à ajouter sont calculés et ajoutés dans les communications au niveau des requêtes ainsi qu'au niveau de l'ouverture et la fermeture des *latches*.

5.5.3 Résultats

La Figure 5-33 montre le spectre en fréquence de l'activité du courant dans les 2 filtres : avant et application de la méthode "mise en forme du courant". Le gain sur l'amplitude maximum et de 9dB et 1dB pour les fréquences allant au-delà de 500Mhz.

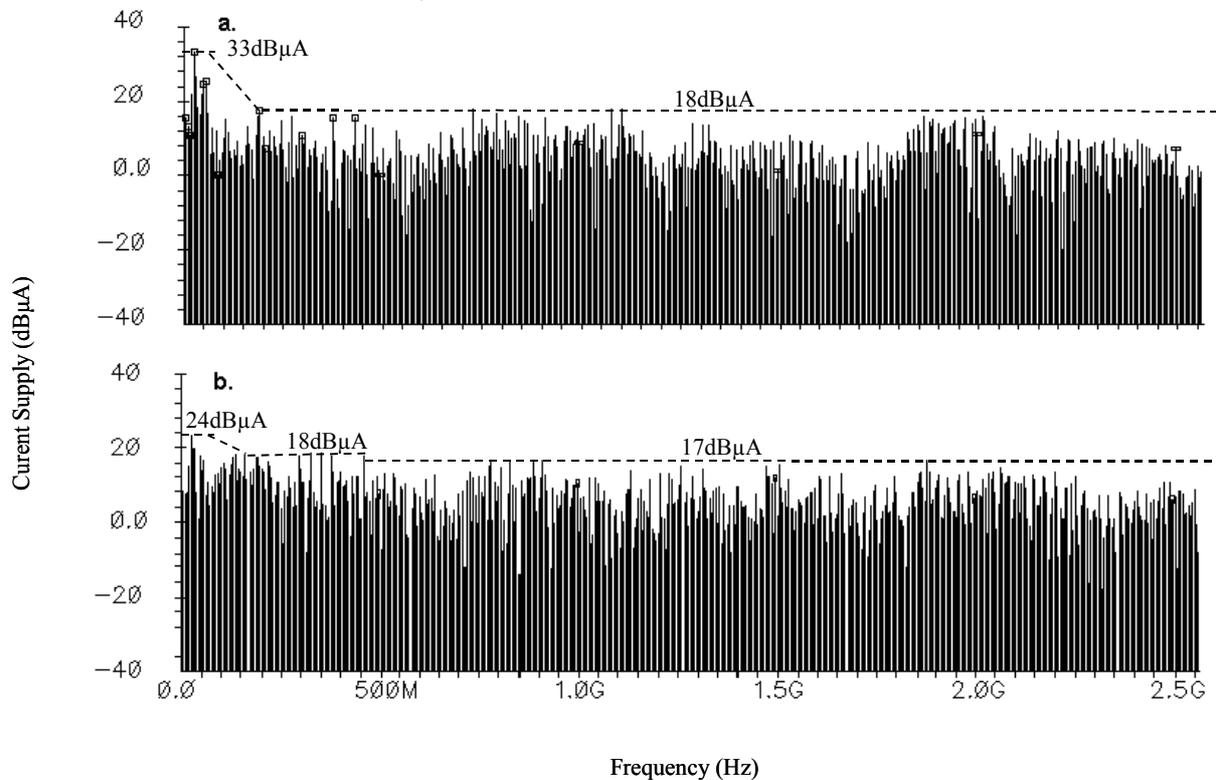


Figure 5-33 : Spectres en fréquence du courant dans le filtre RIF avant et après la méthode "mise en forme du courant"

5.6 Conclusion

Ce chapitre a présenté la méthode "mise en forme du courant". Son but est de minimiser les pics de courant dans les circuits asynchrones afin de réduire les émissions électromagnétiques.

Les variations brusques du courant dans le circuit sont principalement dues aux commutations simultanées de plusieurs portes donc aux concurrences des traitements.

La méthode "mise en forme du courant" (*Current Shaping*) a le même but que les méthodes *Spread Spectrum Clock* et *Clock skew Optimisation* : l'étalement du spectre de l'activité du courant dans le circuit. Elle s'inspire de l'ordonnancement de la puissance dynamique utilisé lors de la synthèse comportementale.

Ici les ressources/opérateurs de l'architecture du circuit sont déjà fixés. La modification de la structure du circuit est minimale : des délais sont insérés dans les communications. Ces délais permettent de retarder des traitements par rapport à d'autres et de diminuer ainsi les concurrences.

La méthode utilise des modèles de l'architecture et de l'activité du courant, et calcule les délais à insérer par l'algorithme *Force Directed Scheduling*.

L'application de la méthode sur un Filtre RIF micropipeline donne des gains intéressants pour seulement 2 concurrences traitées. L'application de la méthode sur des circuits ayant plus de concurrences donnerait des résultats plus significatifs. Pour cela, l'automatisation de la méthode est nécessaire. L'annexe A propose une procédure à ce sujet.

Cette méthode peut être encore améliorée en traitant les autres concurrences qui existent dans le circuit et qui sont source de rayonnement. En particulier, le chapitre suivant traite des protocoles de communications.

CHAPITRE 6 **Activité du courant dans les protocoles 4 phases**

6.1 Introduction

Dans le chapitre 5, la méthode "mise en forme du courant" s'intéresse aux traitements des ressources dans le circuit asynchrone. Elle cible la concurrence de ces traitements et les répartit dans le temps afin de réduire le nombre d'appels de courant simultanés.

Pour réduire davantage les appels de courant simultanés, le nombre de paramètres utilisables pour cette méthode peut être augmenté. Ainsi, les concurrences des événements dans les protocoles de communication peuvent être traitées.

L'étude présentée dans ce chapitre se focalise sur des protocoles de communication 4 phases et la concurrence de leurs événements. Elle apporte une méthode générale de décomposition et de modélisation de l'activité en courant dans les protocoles.

Ce chapitre présente comment à partir d'une représentation STG (*Signal Transition Graph*) du protocole, l'analyse du fonctionnement permet d'obtenir une modélisation par CDFG (*Control Data Flow Graph*) des phases de l'activité en courant. Ensuite, les dépendances entre les différentes phases d'activité sont identifiées et les positionnements des délais à insérer dans ces dépendances sont donnés. Le modèle décomposé obtenu est alors directement utilisable par la méthode "mise en forme du courant" qui calculera la valeur des délais à insérer.

Cette méthode d'analyse des protocoles est générale et peut s'appliquer à n'importe quel type de protocole de communication. L'étude se focalise ici sur les protocoles 4 phases WCHB, PCHB et PCFB déjà définis structurellement dans [RIGA02] et utilisés dans les circuits de l'équipe CIS.

Les paragraphes 6.2, 6.3, 6.4, décrivent le contexte de l'étude : les protocoles 4 phases et leurs phases en courant, les délais insérés pour la méthode ainsi que les modélisations utilisées. Pour chaque protocole WCHB, PCHB et PCFB, l'étude est décrite respectivement dans les paragraphes 6.6, 6.7 et 6.8.

6.2 Protocoles 4 phases

Les protocoles étudiés dans ce chapitre sont les protocoles 4 phases décrits en 3.3.4b : le protocole WCHB (*Weak Condition Half Buffer*), PCHB (*Precharge logic Half Buffer*) et PCFB (*Precharge logic Full Buffer*). Ils sont par ailleurs longuement étudiés dans [RIGA02] d'où les structures de base des contrôleurs sont tirées.

Ces protocoles se différencient les uns des autres par le découplage de leur entrées par rapport à leur sortie. Le Tableau 6-1 rappelle les caractéristiques d'entrée/sortie de ces protocoles.

Protocole WCHB (<i>Weak Condition Half Buffer</i>)	Il est aussi appelé protocole standard ou <i>early</i> . Les remises à zéro des requêtes et acquittements sont synchronisées : ce protocole est non découplé.
Protocole PCHB (<i>PreCharge Logic Half Buffer</i>)	Se différencie du protocole WCHB au niveau de ses phases descendantes où les remises à zéro sont désynchronisées : il est semi découplé.
Protocole PCFB (<i>PreCharge Logic Full Buffer</i>)	Se différencie du protocole WCHB au niveau de ses phases descendantes où les remises à zéro sont désynchronisées. En plus du protocole PCHB, cette remise à zéro est effectuée d'une manière concurrente qui permet de découpler l'entrée de la sortie: il est découplé.

Tableau 6-1 : Rappel sur les protocoles WCHB, PCHB, PCFB

6.2.1 Décomposition des protocoles en phases

Les protocoles dans les communications sont composés d'évènements. Ces évènements sont les commutations des portes dans les contrôleurs. Ces commutations provoquent, dans les contrôleurs et les canaux qui les relie, une activité en courant. Cette étude propose une décomposition (déjà vue en 5.3.3a) de l'activité en courant d'un contrôleur utilisant un protocole 4 phases. Par ailleurs, le déclenchement d'une phase dans un contrôleur dépend non seulement des autres phases du contrôleur, mais aussi des phases des contrôleurs adjacents. Les dépendances sont ici définies et modélisées.

6.2.2 Principales phases de l'activité du courant dans un contrôleur

Trois phases principales de l'activité du courant sont dénombrées dans les protocoles 4 phases. L'exemple de la Figure 6-1 montre la décomposition de la représentation de l'activité de l'opérateur *B* en 3 phases d'activité.

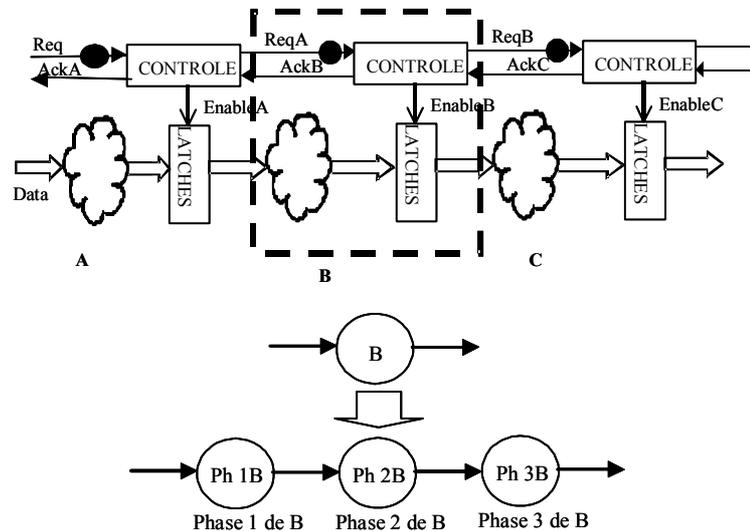


Figure 6-1 : Décomposition en 3 phases de l'activité d'un opérateur

La décomposition est détaillée ci-dessous et représentée par la Figure 6-2 pour le cas de l'opérateur *B* :

- La Phase 1 est caractérisée par la validité des données : Les données valides arrivent. Elles sont signalées par la mise à 1 de la requête d'entrée (début de la phase 1). La fin de traitement des données indique la fin de la phase 1.
- La Phase 2 est caractérisée par le déverrouillage des *latches* : La requête retardée et l'acquittement en sortie à 1 arrivent au contrôleur (début phase 2). Le signal *Enable* laisse passer les données traitées. Le signal de requête en sortie est mis à 1 et le signal d'acquittement en entrée est mis à 0 (fin de la phase 2)
- La Phase 3 est caractérisée par le verrouillage des *latches* : La mise à zéro de l'acquittement de l'étage suivant et le *reset* du signal de requête en entrée sont reçus (début de la phase 3). Le signal *Enable* repasse à 0. Le signal d'acquittement en entrée est remis à 1 et le signal de requête en sortie est remis à 0 (fin de la phase 3).

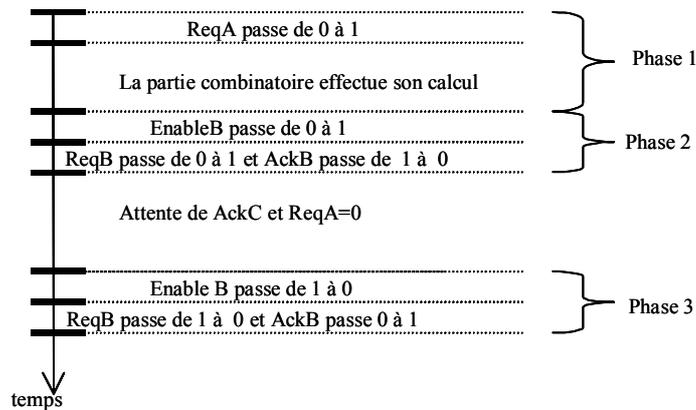


Figure 6-2 : Détail de la décomposition

6.2.3 Activité du courant dans un multiplieur utilisant un contrôleur WCHB

A titre d'exemple pour la décomposition en phases, un multiplieur 8x8 bits a été implémenté en technologie 0,18 microns. Il est de catégorie micropipeline et utilise le protocole 4 phases.

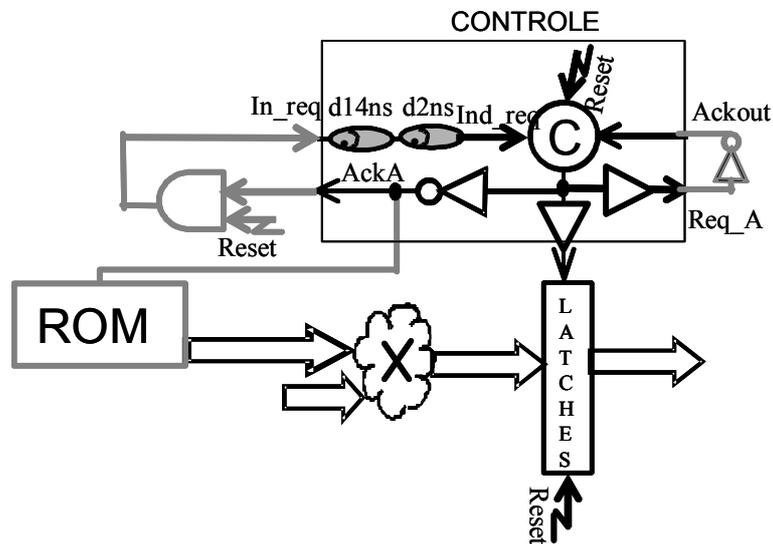


Figure 6-3 : Implémentation d'un multiplieur micropipeline

La Figure 6-3 montre le schéma de l'implémentation du multiplieur ainsi que son environnement de simulation. L'implémentation du contrôle WCHB est montrée dans la partie

encadrée de la figure. Celle-ci utilise dans le cas de l'exemple : une cellule de muller , des buffers  un inverseur  pour l'acquiescement d'entrée (actif à 0), les délais  sur la requête d'entrée dont le rôle est expliqué paragraphe 6.4.

Les conditions de simulations sont représentées en grisé sur la Figure 6-3 et sont les suivantes : Une requête *In_req* est envoyée à l'entrée systématiquement dès que la requête précédente est prise en compte. L'acquiescement *AckA* en sortie est aussi automatiquement délivré dès que *ReqA* est actif. Le Tableau 6-2 donne les valeurs des amplitudes maximales des pics de courants dans les différents éléments de l'opérateur, obtenues par simulation.

<i>buffer</i>	260 μ A
muller2r	230 μ A
délai de 14ns (d14ns)	1,4 mA
délai de 2 ns, d2ns	1,2 mA
latch (8 bits)	1,8 mA
multiplieur (8x8)	15 mA

Tableau 6-2 : Amplitude maximale des pics de courants

La Figure 6-4 montre les activités du courant dans l'opérateur.

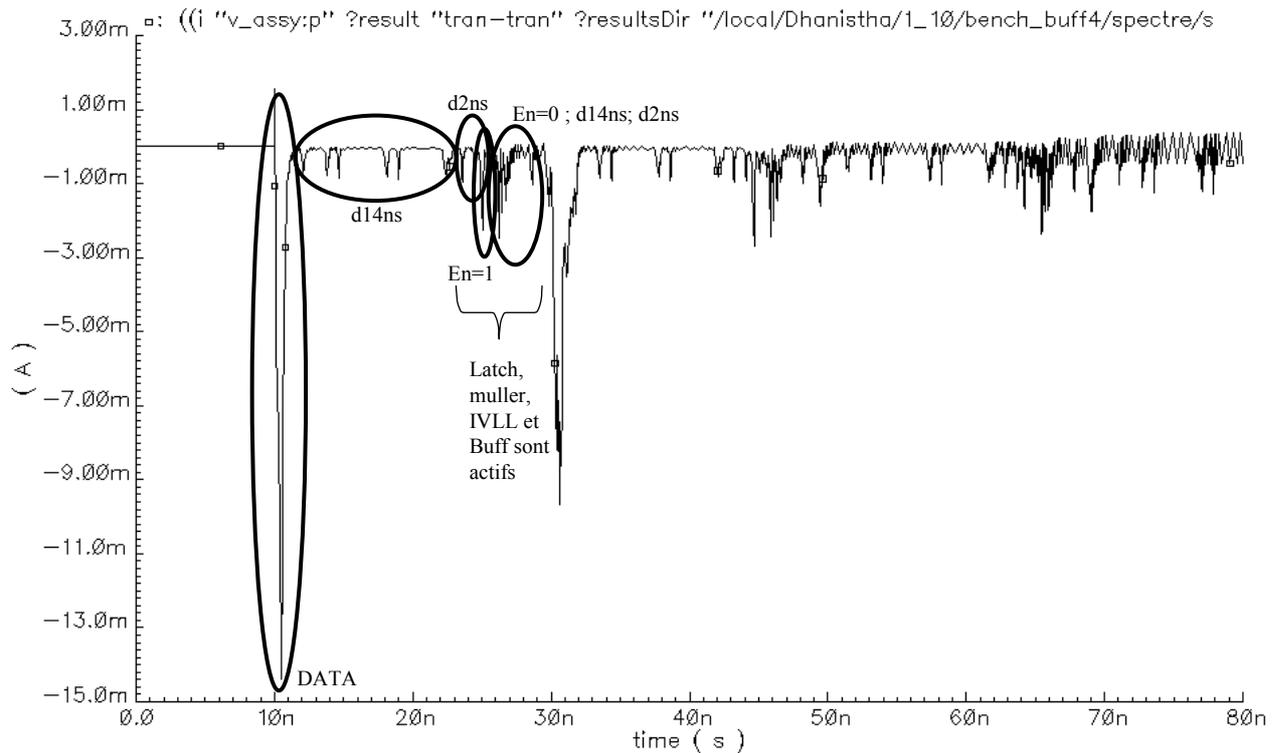


Figure 6-4 : Activité en courant dans un multiplieur asynchrone micropipeline

Les activités des différents éléments de l'opérateur ont été entourées : le premier pic de courant correspond à l'arrivée des données valides. Ensuite, arrive le déclenchement des deux délais dissymétriques pour la mise à 1 de la requête. Le contrôleur est actif et les *latches* laissent passer les données. Finalement, le verrouillage des *latches* et la remise à zéro de la requête d'entrée s'effectuent. Dans la Figure 6-5, les triangles représentant les différentes phases ont été dessinés :

- Phase 1 : La requête d'entrée à 1 (In_req+) est retardée (Ind_req+) : $In_req+ \rightarrow Ind_req+$. Les données arrivent et sont immédiatement traitées par la partie combinatoire. Les délais qui retardent la requête sont actifs.
- Phase 2 : La présence de la requête retardée et l'acquiescement de sortie non actif ($Ackout+$) à la porte de muller déclenche le déverrouillage des *latches* ($En+$), la mise à zéro de l'acquiescement d'entrée ($AckA-$) et l'envoi de la requête de sortie ($ReqA+$) : $Ind_req+ \wedge Ackout+ \rightarrow En+, AckA-, ReqA+$. Les délais et le contrôleur sont actifs pour la remise à zéro et les *latches* sont transparents (*enable* actif).
- Phase 3 : La remise à zéro de la requête d'entrée (Ind_req-) et la remise à 1 de l'acquiescement de sortie ($Ackout-$) en entrée de la porte de Muller déclenche le verrouillage

des *latches* ($En-$), la mise à un de l'acquittement de sortie ($AckA+$) et la remise à zéro de la requête de sortie ($ReqA-$) : $Ind_req- \wedge Ackout- \rightarrow En-, AckA+, ReqA-$. Les *latches* sont verrouillés et le contrôleur assure la mise à zéro.

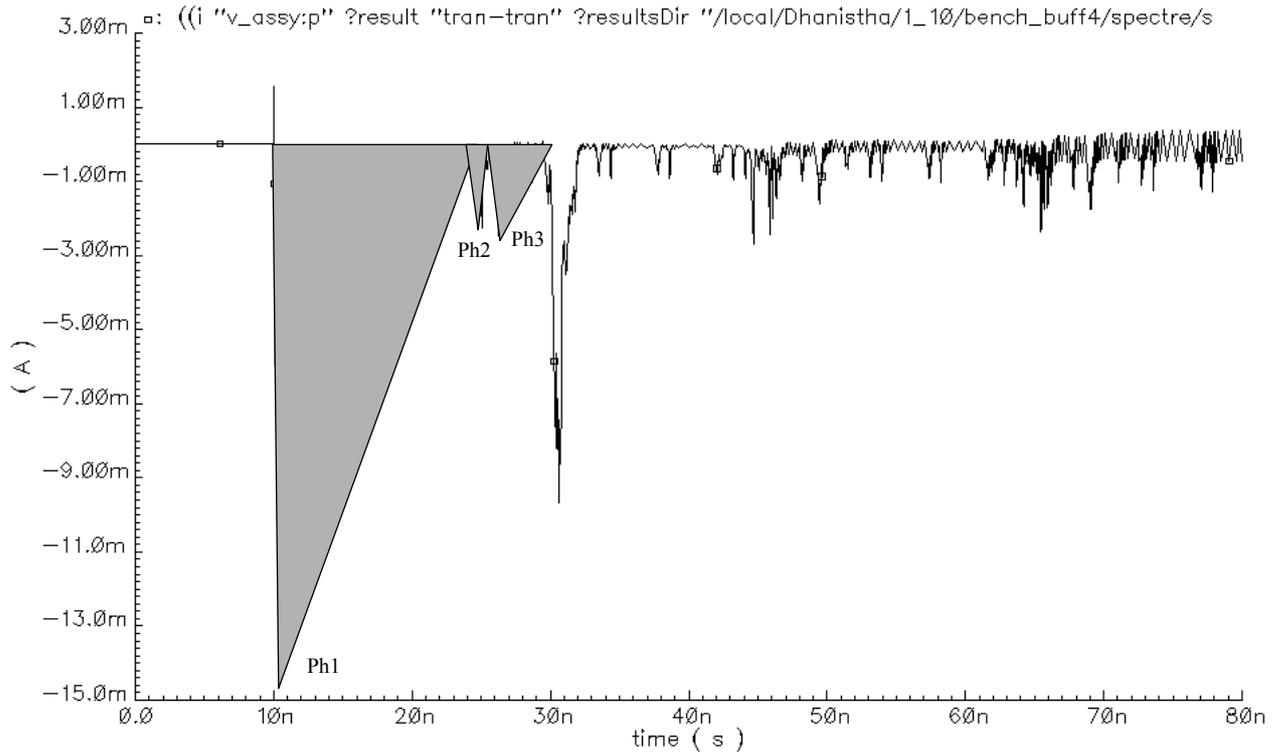


Figure 6-5 : Phases de l'activité en courant dans un multiplieur asynchrone micropipeline

6.2.4 Nature des dépendances entre les phases de l'activité du courant

Les dépendances (appelées aussi liaisons) entre les phases (Figure 6-6) sont ici étudiées et les concurrences sont identifiées.

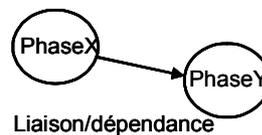


Figure 6-6 : Liaison entre phases

Pour chaque protocole, l'élasticité des dépendances entre les phases en courant des opérateurs est identifiée. Cette élasticité correspond à la flexibilité des liaisons. Plus cette

élasticité est élevée, et plus les phases sont mobiles : elles peuvent donc être réparties plus facilement.

Une liaison sera dite fixe si elle ne peut pas être modifiée. Aucun délai ne peut y être inséré afin de retarder la phase en aval.

Une liaison sera dite lâche si un délai pour répartir le courant peut y être ajouté (6.4.1b). Plus il y a de concurrences dans les phases, plus il y a de commutations simultanées donc moins le protocole est avantageux pour la compatibilité électromagnétique. Cependant, plus les dépendances entre les phases sont lâches, plus le protocole est manipulable par la méthode de "mise en forme du courant". Lors de l'ajout de délai, le passage d'une phase à l'autre est retardé. De cette façon, le fonctionnement du circuit peut être représenté par un réseau de nœuds (phases) et de branches (liaisons) évoluant sur une dimension (Figure 6-7) : le temps. Lors de l'ajout de délais, certaines branches seront allongées et les nœuds se déplaceront les uns par rapport aux autres.

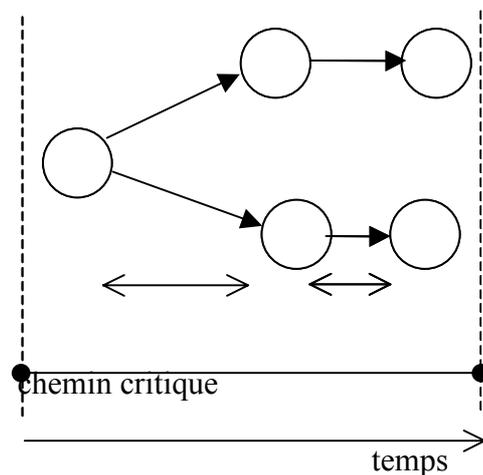


Figure 6-7 : Le fonctionnement du circuit peut être représenté par un réseau de nœuds et de branches évoluant sur une dimension.

La longueur du réseau correspond au chemin critique du circuit. Si elle est conservée lors de l'ajout de délai, alors le fonctionnement du circuit n'est pas ralenti.

Lors de l'utilisation de la méthode "mise en forme du courant" à ce niveau de granularité, la répartition des phases dans les protocoles s'opère comme celle des opérateurs : les ressources sont fixées et le temps est contraint.

Le CDFG utilisé est dit raffiné : ce ne sont plus les opérateurs qui sont adressés, mais leurs phases en courant comme le montre la Figure 6-8. Le nombre de degrés de liberté pour la méthode "mise en forme du courant" est alors augmenté.

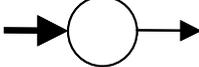
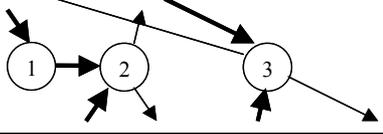
	CDFG	CDFG raffiné
Niveau d'abstraction	Opérateurs 	Phases de courants 
Nombre de degrés de liberté par nœud (cas d'opérateurs en série)	1	supérieur ou égal à 1

Figure 6-8 : Augmentation du nombre de degrés de liberté par l'utilisation de CDFG raffinés

6.3 Modélisations utilisées

L'étude des protocoles de communication utilise plusieurs types de modélisation. Ce paragraphe permet de familiariser le lecteur avec les différentes modélisations utilisées.

6.3.1a HandShaking Expansion (HSE)

Comme le montre l'exemple de la Figure 6-9, le HSE s'apparente à un langage de description haut niveau. Il permet de décrire le comportement des signaux de communications les uns en fonction des autres. Cette représentation est basée sur les événements des signaux.

$$*[[E1 \wedge E2]; S+; [-E1 \wedge \neg E2]; S-]$$

Figure 6-9 : Exemple de HSE.

Les conventions de cette modélisation sont les suivantes :

- * : le processus est répété
- ; : séquentialité
- , : concurrence
- \wedge : et logique
- \neg : non logique
- + : mise à 1
- : mise à 0

Le HSE de la Figure 6-9 représente les actions suivantes :

$[E1 \wedge E2]$: si les requêtes E1 et E2 sont actives, alors la requête de sortie S est mise à 1 (active).

$[\neg E1 \wedge \neg E2]$: si les requêtes E1 et E2 sont inactives, alors la requête de sortie S est mise à 0 (inactive).

La requête de sortie est remise à 0 (inactive) après la remise à 0 des requêtes en entrée.

Cette représentation est syntaxique. Elle ne permet pas au lecteur de comprendre rapidement le comportement du système par manque de lisibilité, en particulier lorsque plusieurs processus composent le système. Ainsi, il a été défini pour l'étude une représentation graphique qui s'inspire des HSE et qui est décrite dans le paragraphe suivant.

6.3.1b HSE Temporel

Cette représentation a été définie afin de déduire rapidement le comportement des signaux dans les communications entre plusieurs opérateurs. Les descriptions HSE des processus des opérateurs sont développées et réparties sur l'axe du temps. De ce fait, le caractère "*" (répétition) n'est plus utilisé car les processus sont développés dans le temps. Le HSE temporel de la Figure 6-10 représente le comportement de l'opérateur décrit en Figure 6-9. L'axe du temps n'a pas d'unité définie, il sert uniquement à montrer la séquence des évènements.

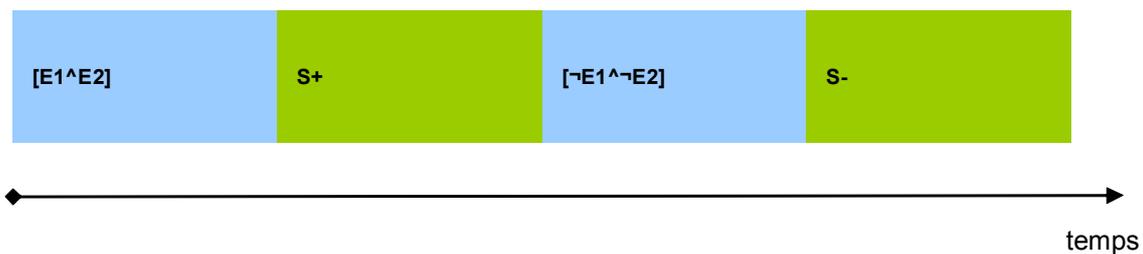


Figure 6-10 : Exemple de HSE Temporel

La Figure 6-11 représente un système de 3 opérateurs ayant les comportements suivants :

$OP1 : *[[A]; E1+; [\neg A]; E1-]$

$OP2 : *[[E1 \wedge E2]; S+; [\neg E1 \wedge \neg E2]; S-]$

$OP3 : *[[C]; E2+; [\neg C]; E2-]$



Figure 6-11 : Exemple de HSE temporel pour 3 opérateurs

La Figure 6-11 montre l'existence de deux régimes lors du fonctionnement du système : le régime transitoire (de $t0$ à $t3$) et le régime établi (au delà de $t3$). Ce type de modélisation est donc pratique pour analyser le comportement de structures pipelinées.

6.3.1c STG (Signal Transition Graph)

Comme le *HandShaking Expansion*, cette modélisation graphique est basée sur les évènements de signaux. Cependant, elle est plus riche car les états des signaux (0 et 1) dans le contrôle sont explicitement représentés. Elle est assez utilisée, en particulier pour la synthèse des circuits asynchrones [PIGU01][SPAR01].

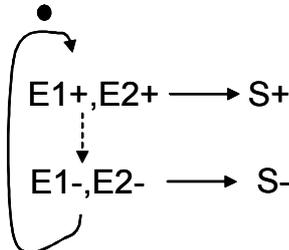


Figure 6-12 : Exemple de STG

Les conventions pour ce type de graphe sont les suivantes :

- : état initial du circuit
- ⋯▶ : transitions précédentes
- ▶ : transitions courantes

6.3.1d CDFG (Control Data Flow Graph) et DFG (Data Flow Graph)

La modélisation par graphe de flot de données (*DFG : Data Flow Graph*) est une description de l'architecture du système qui ne tient pas compte du cycle du circuit. Dans le cas étudié, les noeuds du graphe représentent les traitements (calculs) effectués dans le circuit et les

arcs du graphe représentent les dépendances entre les données. Le CDFG (*Control Data Flow Graph*), quand à lui, est un DFG enrichi qui incorpore des noeuds de décision (contrôle) et des DFG comme sous-graphes. Il permet en particulier de modéliser des circuits complexes.

Le *Control Data Flow Graph* a déjà été abordé dans le paragraphe 5.3. Dans le cas de l'exemple de la Figure 6-13, les noeuds représentent les états des signaux et les arcs les événements sur ces signaux.

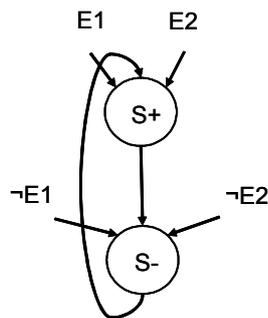


Figure 6-13 : Exemple de CDFG

Les conventions utilisées sont les suivantes : les flèches qui arrivent au noeud représentent les événements qui déclenchent l'activation de ce noeud.

6.3.1e CDFG déroulé dans le temps

Les CDFG et DFG présentés dans le paragraphe précédent permettent de comprendre le fonctionnement du système, mais dans le cas d'un fonctionnement cyclique, il ne permet pas d'identifier les traitements concurrents dus à l'architecture pipeline du système. Le *Control Data Flow Graph* déroulé qui a été défini dans cette étude, permet de représenter l'activité des noeuds dans le temps : les motifs sont répétés autant de fois que l'opérateur est actif (Figure 6-14). En utilisant cette représentation, le lecteur visualise directement le comportement du circuit dans le temps.

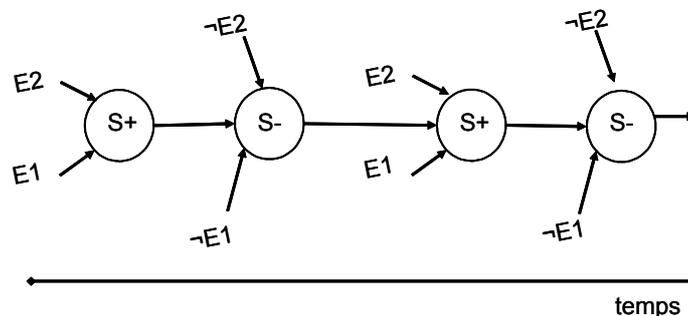


Figure 6-14 : Exemple de CDFG déroulé

6.3.1f Modélisation structurelle

Cette modélisation est simplement la représentation de la structure du circuit (Figure 6-15).

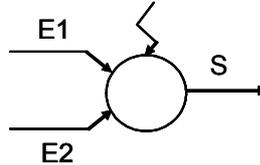


Figure 6-15 : Modélisation structurelle

6.3.1g Choix d'une modélisation

Le choix d'une modélisation dépend du niveau d'abstraction abordé et du but de l'utilisation de celle-ci. Dans cette étude, le HSE et le HSE temporel seront utilisés pour décrire le fonctionnement du protocole, l'enchaînement des signaux et sa périodicité (6.9). Les CDFG/DFG et les CDFG/DFG déroulé seront utilisés ensuite en vue d'exploiter les interactions entre les phases. Les nœuds représenteront alors les phases de l'activité en courant des contrôleurs et les arcs les événements agissant sur ces phases.

Le Tableau 6-3 résume les différentes modélisations qui sont utilisées dans l'étude :

BUT	Modélisation utilisée	
Description du fonctionnement du protocole, enchaînement des signaux	HandShaking Expansion (HSE)	$*[[E1 \wedge E2]; S+; [-E1 \wedge -E2]; S-]$
	HSE temporel	
	Signal Transition Graph (STG)	
Enchaînement des phases	HSE temporel	
	Control Data Flow Graph (CDFG)	
	CDFG déroulé	
Structure	Schéma de structure	

Tableau 6-3: Modélisations utilisées

6.4 Délais

Le principe de la méthode "mise en forme du courant" est d'insérer des délais pour réduire les concurrences. Ce paragraphe décrit les différents délais utilisés ainsi que leurs architectures.

6.4.1a Délais intrinsèques (délais fonctionnels)

La Figure 6-16 représente la structure d'un contrôleur [RIGA02] assurant la gestion des communications qui utilise le protocole WCHB.

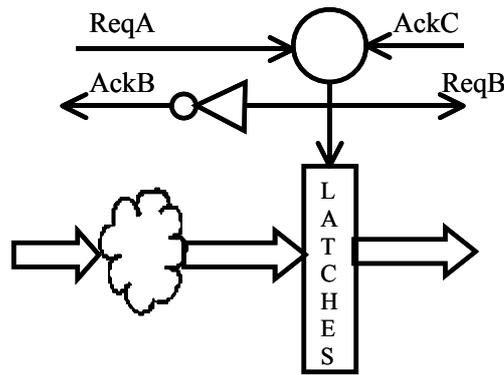


Figure 6-16 : Un contrôleur

Un délai est inséré dans ce contrôleur lorsque celui-ci est utilisé dans un circuit micropipeline. Ce délai a une valeur égale à la latence de la logique combinatoire contenue entre le contrôleur concerné et celui en amont ou en aval selon la convention utilisée (Figure 6-17). Ce délai permet, selon la convention, de retarder la requête en entrée ou en sortie du contrôleur donc de respecter la latence de la partie combinatoire. Ce délai sera appelé délai fonctionnel ou intrinsèque.

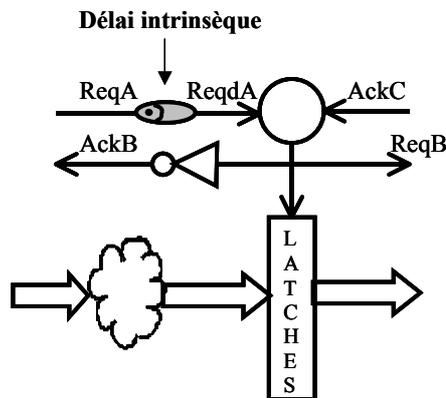


Figure 6-17 : Délai intrinsèque en entrée

6.4.1b Délais pour la faible émission électromagnétique

Lors de l'application de la méthode "mise en forme du courant", des délais sont calculés par l'algorithme *Force Directed Scheduling* [PAUL89] de façon à optimiser la forme du courant. Ils peuvent être insérés dans les signaux de communication (Requêtes et Acquittements) ou dans les signaux de contrôle comme le montre la Figure 6-18.

Délais pour la faible émission électromagnétique

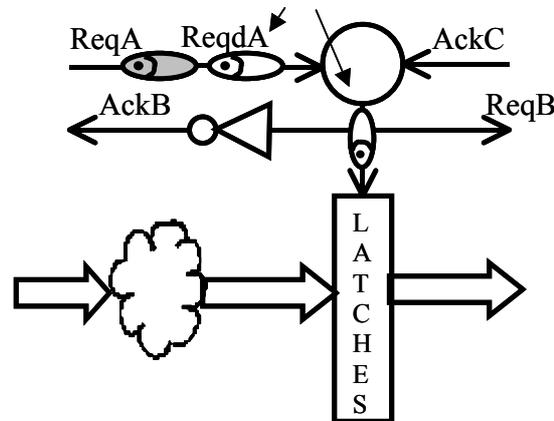


Figure 6-18 : Délai pour la faible émission électromagnétique

Pour pouvoir être efficaces, ces délais doivent consommer très peu. Dans la suite de l'étude ces délais seront appelés délais pour la faible émission électromagnétique.

6.5 Description de l'étude

L'étude des protocoles 4 phases se fait sur un système de 3 contrôleurs en série. Cette structure est simple et permet de prendre en compte le couplage entrée-sortie du protocole WCHB.

Pour chaque protocole :

- Son formalisme est abordé. Il comprend : la description HSE du contrôleur et sa structure ([RIGA02]).
- Une vue d'ensemble de la structure des 3 contrôleurs et le STG correspondant sont donnés. Ce dernier permet de comprendre le fonctionnement des contrôleurs et de déterminer les phases en courant décrites dans le paragraphe 6.2.2. L'analyse se focalise alors sur le fonctionnement du contrôleur central (appelé *B*) qui reflète le fonctionnement d'un contrôleur dépendant de contrôleurs amont et aval (ici *A* et *C*).
- L'activité du courant est ensuite analysée pour caractériser les phases de l'activité du courant dans les contrôleurs et le type de dépendances qui existent entre ces phases. Les CDFG sont alors utilisés pour identifier les dépendances tandis que les CDFG déroulés sont utilisés pour la description du fonctionnement du circuit dans le temps.
- Le temps de cycle de l'opérateur est identifié. Il permet de définir la contrainte temporelle qui doit être utilisée pour l'ordonnancement.

- Selon l'élasticité des liaisons, l'insertion de délais et la modification du contrôleur correspondante sont proposées pour le critère de faible émission électromagnétique. Ce délai pourra être calculé par l'algorithme FDS.

6.6 Le protocole WCHB

Le protocole WCHB de l'anglais *Weak Condition Half Buffer* est le protocole 4 phases dit standard. Le contrôleur utilisé est appelé contrôleur simple.

6.6.1 Formalisme

La Figure 6-19 montre la structure d'un contrôleur utilisant le protocole WCHB.

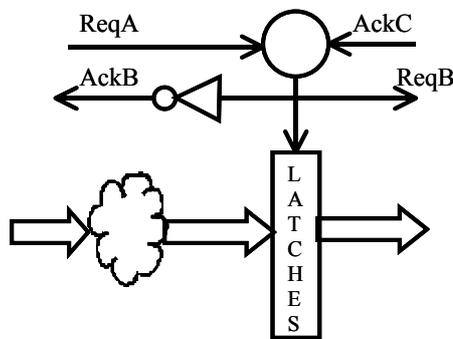


Figure 6-19 : Contrôleur WCHB.

Le HSE qui décrit le fonctionnement du contrôleur est le suivant :

$$*[[AckC \wedge ReqA]; ReqB+; AckB-; [\neg AckC \wedge \neg ReqA]; ReqB-; AckB+]$$

Comme le montre la structure Figure 6-19, la valeur du signal de commande des *latches* (*Enable*) est la même que celle de la requête de sortie *ReqB*. Donc, lorsqu'il y a *ReqB+*, il y a *EnableB+*, de même pour *ReqB-* et *EnableB-*.

6.6.2 Architecture du système étudié

La Figure 6-20 montre 3 *half buffer* *A*, *B*, *C* en série. Pour le *half-buffer B*, *ReqA* est retardé par un délai intrinsèque permettant de respecter la latence de la partie combinatoire. *ReqdA* est la requête venant de *A* et retardée de ce délai.

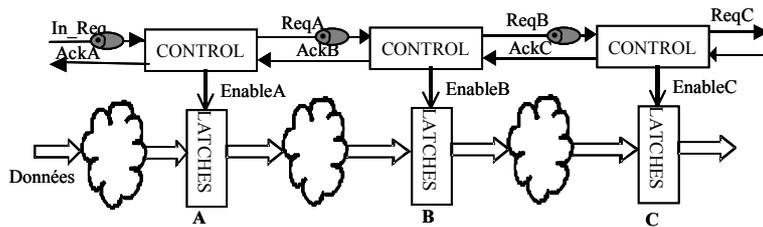


Figure 6-20 : 3 opérateurs WCHB en série

6.6.3 Fonctionnement du protocole

6.6.3a A l'initialisation

La Figure 6-21 montre le système des trois *half buffers* à l'initialisation. Lors de cet état, il n'y a aucune activité dans le système. L'état des 3 *half buffers* est identique.

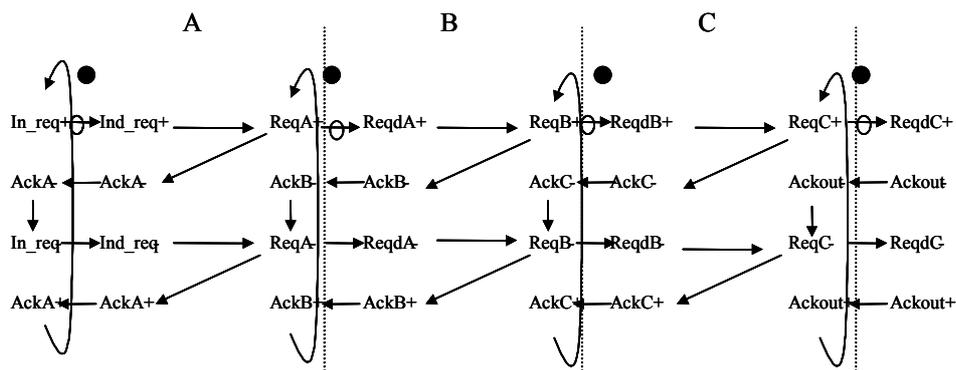


Figure 6-21 : STG des 3 opérateurs utilisant le protocole WCHB - Etat initial

6.6.3b Phase 1 de l'opérateur B

La Figure 6-22 montre le déclenchement de la Phase 1 du *half buffer B*. Le *half-buffer A* achève sa phase 2 en envoyant une requête au contrôleur B ($ReqA+$). Cette requête est retardée par le délai intrinsèque. Les *latches* du *half-buffer A* sont déverrouillées ($EnableA+$) ainsi les parties combinatoires de l'opérateur B en amont des *latches* sont actives.

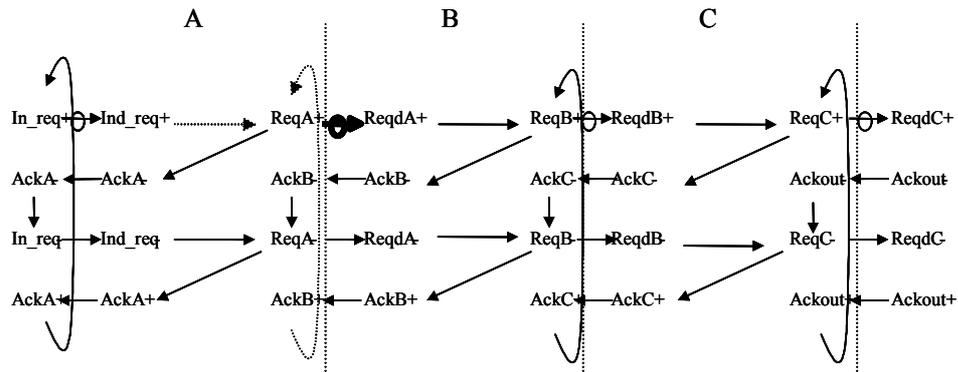


Figure 6-22 : STG des 3 opérateurs utilisant le protocole WCHB - Phase 1

6.6.3c Phase 2 de l'opérateur B

La phase 2 de l'étage B (Figure 6-23) est déclenchée à la fin de sa phase 1 lorsque le signal de requête retardée *ReqdA* est actif. La phase 2 requiert également *AckC+*. Cela correspond à la fin de la phase 3 du contrôleur C qui a terminé le traitement des données fournies par B à $t-1$. *AckC+* et *ReqdA* activent le contrôleur (porte de muller) qui génère *ReqB+* et déverrouille les *latches* (*EnableB+*). Le contrôleur C passe en Phase 1.

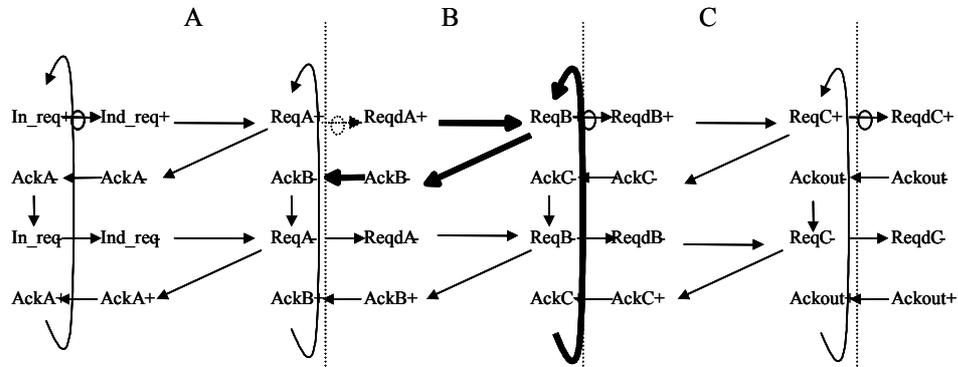


Figure 6-23 : STG des 3 opérateurs utilisant le protocole WCHB - Phase 2

6.6.3d Phase 3 de l'opérateur B

La phase 3 de l'étage B s'effectue lorsque le contrôleur C passe en phase 2 (*AckC-*) et lorsque le contrôleur A fini sa phase 3 (*ReqdA-*). La requête de sortie est alors remise à zéro (*ReqB-*) et l'acquittement d'entrée est remis à 1 (*AckB-*).

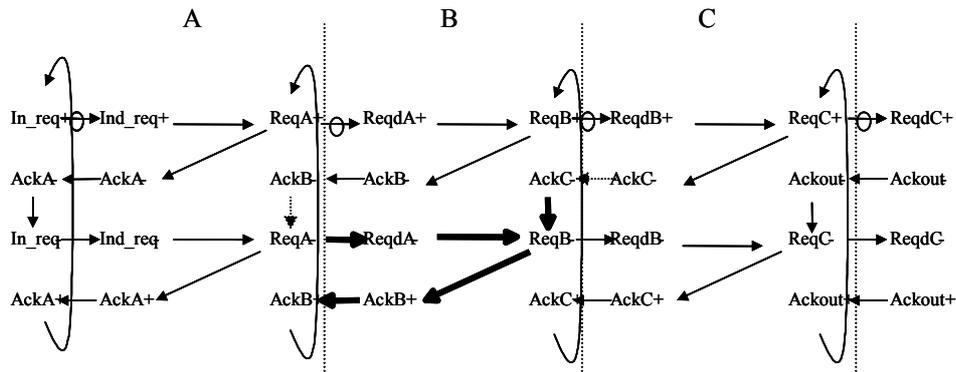


Figure 6-24 : STG des 3 opérateurs utilisant le protocole WCHB - Phase 3

6.6.4 Phases de l'activité du courant

6.6.4a Modélisation par CDFG

A présent, ayant déterminé le déclenchement des différentes phases du *Buffer B*, l'étude du CDFG de la structure permet de comprendre les dépendances entre les phases (Figure 6-25).

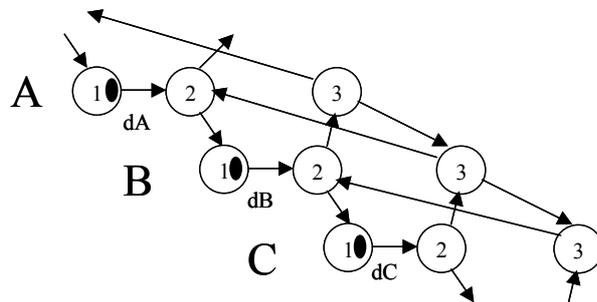


Figure 6-25 : CDFG des 3 opérateurs micropipeline utilisant le protocole WCHB

Dans la Figure 6-25, les 3 phases de l'activité du courant dans chaque contrôleur sont représentées sous forme de noeuds. Les dépendances correspondent aux liaisons induites par les protocoles (Requêtes et Acquittement). dA , dB et dC sont les délais intrinsèques des contrôleurs. Comme ces délais sont asymétriques, ils sont uniquement dans les phases 1 des opérateurs.

Les liaisons allant des phases 3 aux phases 2 représentent la synchronisation entre la gestion des signaux d'entrée et de sortie des contrôleurs. Ces phases sont couplées : le protocole WCHB est un protocole dit non découplé.

6.6.4b Temps de cycle d'un opérateur

Le temps de cycle d'un opérateur peut être déduit du CDFG de ses phases en courant.

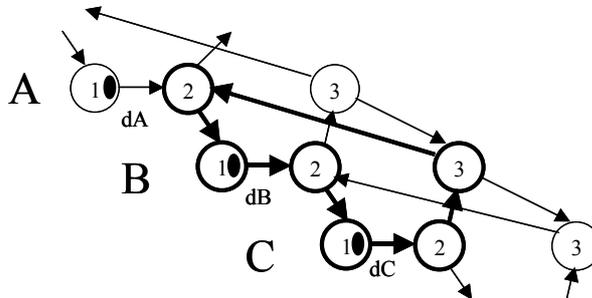


Figure 6-26 : Chemin critique pour l'activité en courant de l'opérateur B (protocole WCHB)

Le chemin critique de l'opérateur B est représenté en gras sur la Figure 6-26. Le temps que prend l'étage B pour effectuer un cycle est le suivant :

$$TB = dB + \text{délai_phase_2B} + dC + \text{délai_phase_2C} + \text{délai_phase3B} + \text{délai_phase2A}$$

Équation 6-1

Ce temps de cycle dépend donc des phases de l'activité en courant du contrôleur considéré, mais aussi des phases des contrôleurs voisins.

Ceci est l'inconvénient majeur de l'utilisation du protocole WCHB non découplé : dans le temps de cycle d'un opérateur, il faut prendre en compte la latence de l'ensemble des phases de l'opérateur suivant et la phase 2 de l'opérateur précédent.

6.6.5 Dépendances

6.6.5a Influence des évènements

La Figure 6-27 reprend le CDFG de la Figure 6-26 dans laquelle les évènements correspondant aux liaisons entre les phases ont été ajoutés.

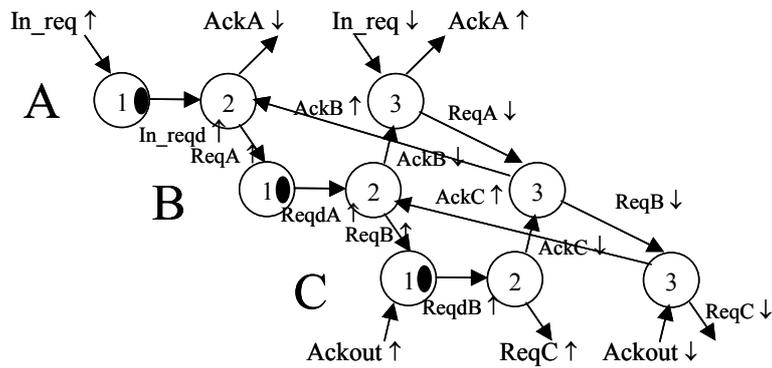


Figure 6-27 : Evènements dans le protocole WCHB

Les dépendances entre les phases des différents contrôleurs peuvent être classées selon le type d'évènement qu'elles représentent.

A noter que vu la faible complexité du protocole WCHB, la représentation CDFG déroulé n'a pas été utilisée.

6.6.5b Les types de dépendances

Dans la Figure 6-27 qui représente le système utilisant un protocole WCHB, 5 types de liaisons peuvent être remarquées. Elles sont appelées α , β , γ , δ , ϵ pour des raisons de simplicité et sont représentées Figure 6-28.

α correspond à l'arrivée d'une requête ($0 \rightarrow 1$) et des données valides sur un contrôleur.

β correspond à l'arrivée d'une requête retardée ($0 \rightarrow 1$)

γ correspond à l'acquittement d'une requête ($1 \rightarrow 0$) par le contrôleur

δ signifie que l'acquittement est remis à 1 ($0 \rightarrow 1$)

et ϵ correspond à la remise à zéro d'un signal de requête ($1 \rightarrow 0$) (le délai du signal retardé est asymétrique)

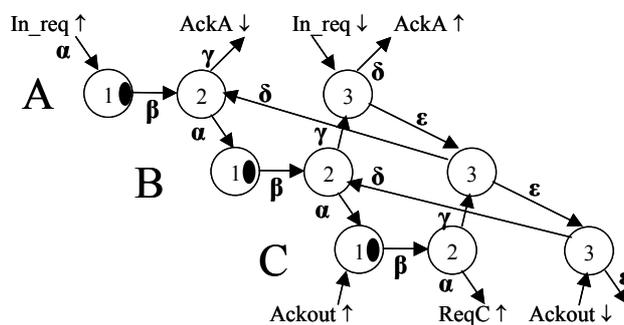


Figure 6-28 : Les dépendances dans le protocole WCHB

Pour le *Half-Buffer B* (Figure 6-29), les liaisons ont pour dénominations :

- α_B la liaison qui correspond à l'envoi de la requête *ReqB* et la libération des données.
- β_B la liaison qui correspond au passage de 0 à 1 du signal *ReqdA* dans le *half-buffer*.
- γ_B la liaison qui correspond à l'acquittement de *ReqA* par l'envoi de *AckB*.
- δ_B la liaison qui correspond à la fin d'acquittement de *ReqA* par le passage de 1 à 0 de *AckB*.
- ϵ_B la liaison qui correspond à la remise à zéro du signal *ReqB*.

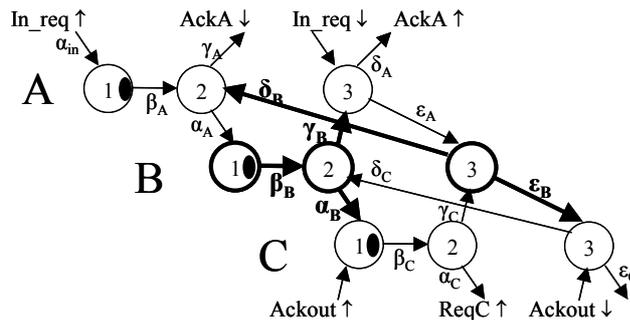


Figure 6-29 : Dépendances associées à l'étage B

Le Tableau 6-4 résume quelles phases sont directement retardées lors de l'ajout de délais dans les liaisons appartenant à *B*.

<i>Liaison</i>	<i>Phase directement retardée</i>
α_B	Phase 1 de C
β_B	Phase 2 de B
γ_B	Phase 3 de A
δ_B	Phase 2 de A
ϵ_B	Phase 3 de C

Tableau 6-4 : Phases retardées par l'insertion de délais dans les liaisons

Cinq dépendances ont donc été identifiées pour le protocole WCHB. Elles correspondent à différents événements dans le protocole. Ces 5 dépendances se retrouvent dans les autres protocoles 4 phases.

6.6.5c Dépendances lâches et fixes

Connaissant les dépendances entre les nœuds, il faut déterminer leur nature (lâches ou fixes) pour savoir où des délais peuvent être insérés. L'insertion d'un délai dans une dépendance lâche ne nécessite pas de modification de l'architecture du contrôleur car il peut être inséré dans

les communications, tandis que l'insertion d'un délai dans une dépendance fixe requiert une modification interne du contrôleur.

Dépendances Lâches

Les dépendances suivantes ont été identifiées comme lâches :

β dépend du délai sur la requête en entrée du contrôleur. Un délai peut donc être inséré en amont ou en aval du délai existant.

ε dépend de la dissymétrie du délai qui retarde la requête dans le contrôleur. La remise à zéro de la requête peut donc être retardée en jouant sur la dissymétrie du délai.

γ et δ représentent respectivement la mise à 0 et la mise à 1 de l'acquittement d'entrée. Un délai peut donc retarder cet acquittement.

Dépendance fixe

Dans le protocole WCHB, seul la liaison α est fixe car elle dépend uniquement du flux des données. Si la requête qui arrive au contrôleur peut être retardée, le calcul des données ne peut être retardé qu'en touchant à l'implémentation du contrôleur en amont (c'est-à-dire en retardant la libération des données en amont).

6.6.6 Contrôle des dépendances

Une fois les dépendances et leurs natures identifiées, une solution pour insérer des délais dans les liaisons est proposée.

6.6.6a Dépendances lâches

La Figure 6-30 représente le cas où ε et β sont modifiées. Des délais sont respectivement insérés dans les fils de requête arrivant et sortant du contrôleur. La modification de β s'effectue au niveau du délai intrinsèque (délai en gris sur la figure). La modification de ε s'effectue par l'ajout d'un délai sur la requête de sortie.

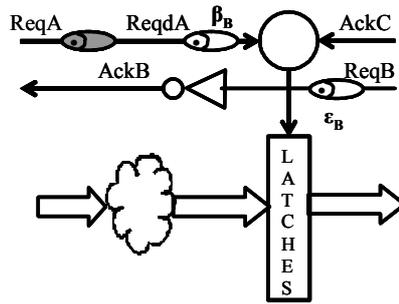


Figure 6-30 : Contrôle de ϵ et β

La Figure 6-31 représente le cas où δ et γ sont modifiées. Un seul délai est inséré. Sa dissymétrie gère la mise à 1 et à 0 de l'acquiescement de la requête d'entrée.

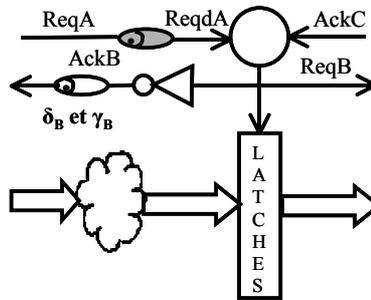


Figure 6-31 : Contrôle de δ et γ

6.6.6b Dépendance fixe

La modification de l'implémentation du contrôleur permet de modifier la dépendance α_B dont dépend la phase 1 de C comme le montre la Figure 6-32.

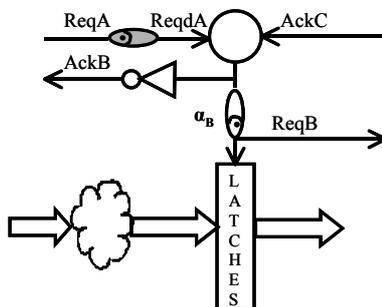


Figure 6-32 : Contrôle de α

Ce délai permet de retarder à la fois la requête de sortie *ReqB* et la libération des données en sortie de l'étage.

6.6.6c Contrôleur générique

Dans les paragraphes précédents, des solutions ont été proposées pour retarder chacune des dépendances présentes à l'intérieur du protocole WCHB.

Dans le cas où toutes les dépendances sont modifiées, les solutions proposées peuvent être cumulées. Une factorisation de tous les délais peut être effectuée. Elle suppose l'utilisation de délais dissymétriques dont l'architecture est décrite en 3.3.9. La Figure 6-33 montre comment, le délai peut être exploité pour gérer 5 retards différents.

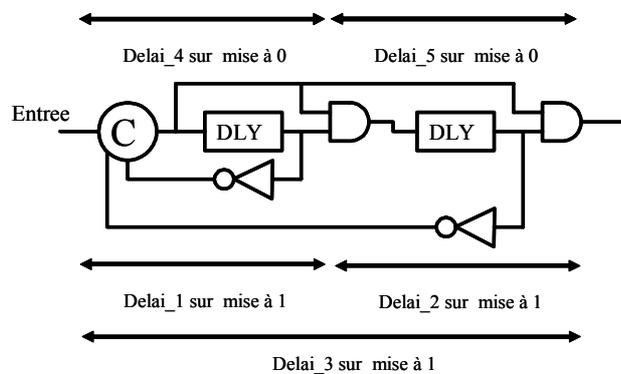


Figure 6-33 : Exploitation de l'architecture du délai asymétrique

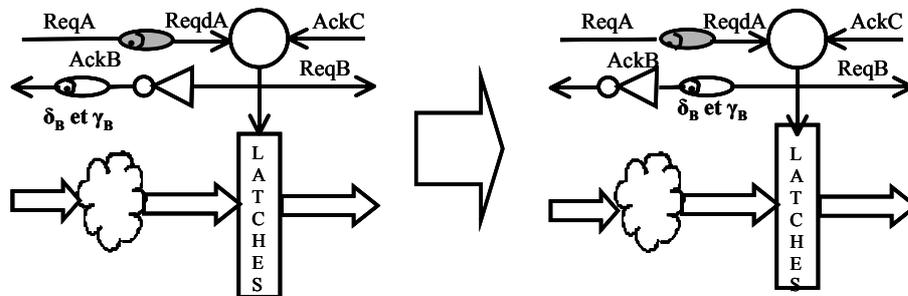


Figure 6-34 : Implémentations équivalentes

La Figure 6-34 montre comment les délais sur les liaisons δ et γ sont introduits à l'intérieur du contrôleur. La Figure 6-35 réunit l'ensemble des solutions après insertion dans le contrôleur.

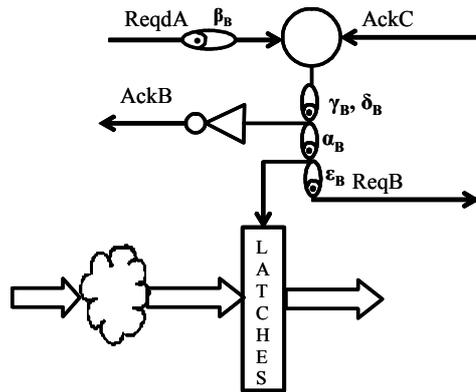


Figure 6-35 : Les délais des dépendances peuvent être insérés dans le contrôleur

En exploitant l'architecture des délais dissymétriques (Figure 6-33), le même délai peut être utilisé pour gérer plusieurs dépendances comme le montre la Figure 6-36.

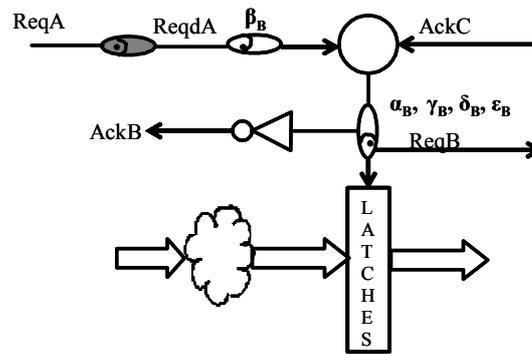


Figure 6-36 : Factorisation des délais

En comparaison à l'utilisation de plusieurs délais, ceci permet de réduire la surface, la consommation et les pics de courant résultant de la somme de plusieurs délais actifs simultanément.

Une implémentation de contrôleur générique possédant les paramètres génériques "délais à ajouter dans les liaisons α , β , γ , δ ϵ " est envisageable.

6.6.7 Autres structures

L'utilisation des protocoles de communication ne s'arrête pas à la mise en série de plusieurs opérateurs. Dans la suite, les structures rendez-vous, duplicateur, multiplexeur et démultiplexeur sont étudiées.

6.6.7a Le rendez-vous (*Join*)

Dans la structure présentée Figure 6-37, les sorties des contrôleurs *A1* et *A2* sont connectées à l'entrée du contrôleur *B*.

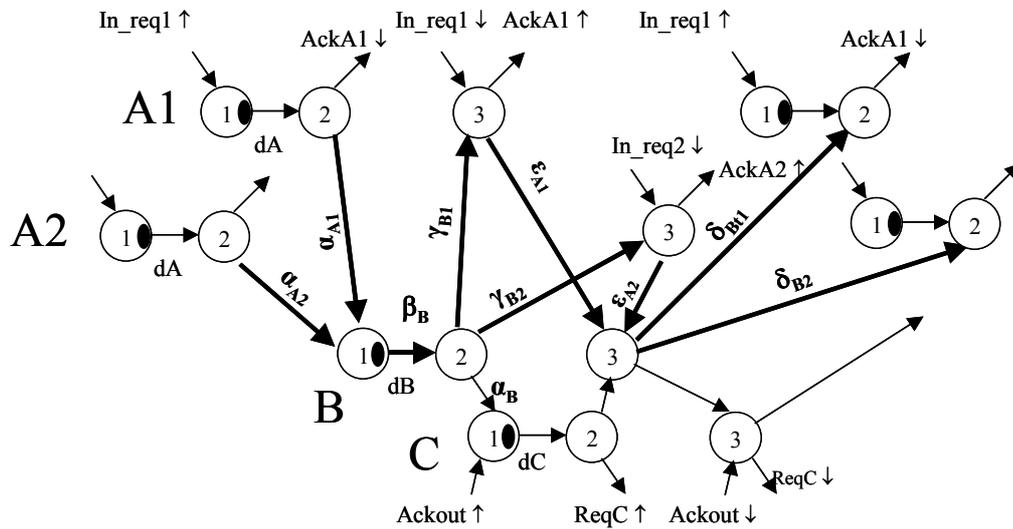


Figure 6-37 : CDFG d'un RV en protocole WCHB

Les liaisons et leur nature restent identiques à celle du contrôleur de la structure en série. Seul le temps de cycle de l'opérateur B change. Temps de cycle :

$$dB + Ph2b + dC + Ph2C + Ph3b + \max(Ph2A1/A2)$$

Équation 6-2

$\alpha A1$ et $\alpha A2$ sont fixes l'une par rapport à l'autre. Elles ne peuvent pas être traitées indépendamment. Cependant le couple $\{\alpha A1, \alpha A2\}$ forme une dépendance lâche.

$\epsilon A1$ et $\epsilon A2$ sont fixes l'une par rapport à l'autre. Elles ne peuvent pas être traitées indépendamment. Cependant le couple $\{\epsilon A1, \epsilon A2\}$ forme une dépendance lâche.

$\gamma B1$ et $\gamma B2$ déclenchent respectivement les phases 3 des contrôleurs *A1* et *A2*. Les déclenchements peuvent être décalés en jouant sur les délais.

$\delta B1$ et $\delta B2$ déclenchent respectivement les phases 3 des contrôleurs *A1* et *A2*. Les déclenchements peuvent être décalés en jouant sur les délais.

Les dépendances $\beta B1$ et $\beta B2$ sont indépendantes et peuvent être modifiées.

6.6.7b Le duplicateur

Dans la structure présentée Figure 6-38, les entrées des contrôleurs $B1$ et $B2$ sont connectées à la sortie du contrôleur A .

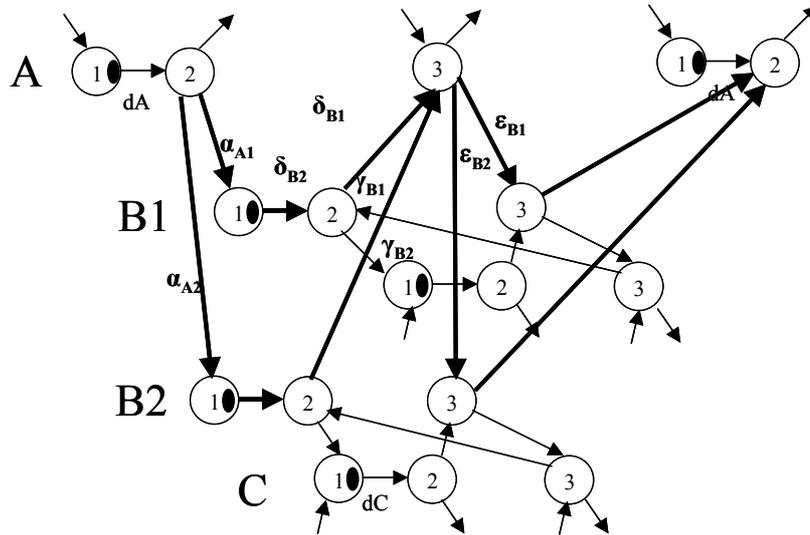


Figure 6-38 : Duplicateur utilisant le protocole WCHB

La structure du duplicateur est symétrique à celle du Rendez-vous. 2 protocoles sont concurrents : $B1$ et $B2$. Le temps de cycle de cette concurrence est égal au cycle de l'opérateur le plus long.

γ_{B1} et γ_{B2} sont fixes l'une par rapport à l'autre. Elles ne peuvent pas être traitées indépendamment. Cependant, le couple $\{\gamma_{B1}, \gamma_{B2}\}$ forme une dépendance lâche.

Idem pour δ_{B1} et δ_{B2}

Les dépendances β_{B1} et β_{B2} sont indépendantes et peuvent être modifiées.

α_{A1} et α_{A2} déclenchent respectivement les phases 1 des contrôleurs $B1$ et $B2$. En allongeant les délais dans ces dépendances, ces déclenchements peuvent être décalés.

Même remarque pour ϵ_{B1} et ϵ_{B2} .

Les dépendances β_{B1} et β_{B2} sont indépendantes et peuvent être modifiées.

6.6.7c Multiplexeur

Dans la structure présentée Figure 6-39, la sortie du contrôleur A est reliée soit à l'entrée du contrôleur $B1$, soit à l'entrée du contrôleur $B2$.

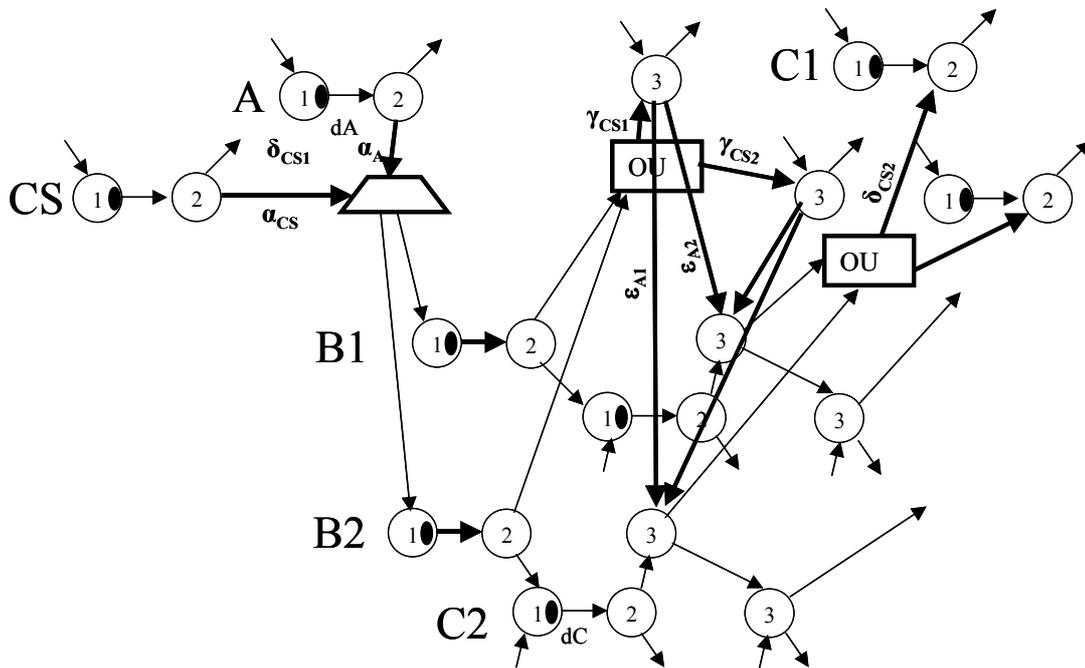


Figure 6-39 : multiplexeur

Les chemins utilisant $B1$ ou $B2$ sont traités comme 2 chemins séparés en prenant garde au fait que α_{CS} et α_A sont communes aux 2 chemins.

6.6.7d Démultiplexeur

Dans la structure présentée Figure 6-40, l'entrée du contrôleur B est reliée soit à la sortie du contrôleur $A1$, soit à la sortie du contrôleur $A2$.

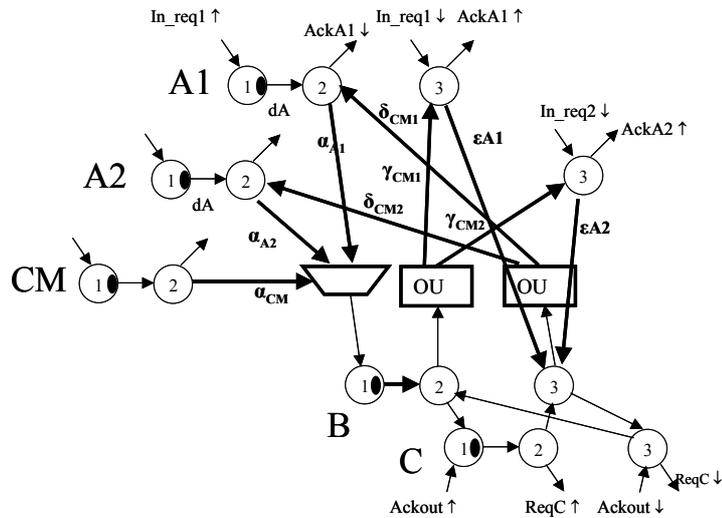


Figure 6-40 : démultiplexeur

Les chemins utilisant A1 ou A2 sont traités comme 2 chemins séparés en prenant garde au fait que γ_B et δ_B sont communes aux 2 chemins.

6.7 Protocole PCHB

Le protocole *PreCharge logic Half Buffer (PCHB)*, diffère du protocole WCHB lors de sa phase de remise à zéro où les entrées et les sorties sont désynchronisées.

6.7.1 Formalisme

Le HSE qui décrit le contrôleur du protocole PCHB est le suivant :

$$*[[Sacq \wedge Ereq]; Sreq+; Eacq-; [\neg Sacq]; Sreq-; [\neg Ereq]; Eack+]$$

La Figure 6-41 montre la structure d'un contrôleur utilisant le protocole PCHB.

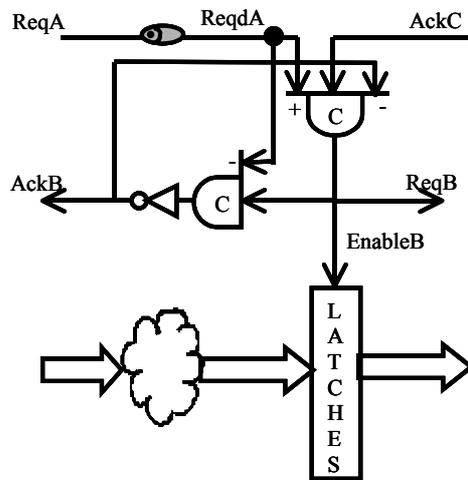


Figure 6-41 : Contrôleur PCHB

Comme pour le protocole WCHB, ReqB et *EnableB* sont liés. Donc, lorsqu'il y a ReqB+, il y a *EnableB*+, de même pour ReqB- et *EnableB*-.

6.7.2 Architecture du système étudié

La Figure 6-42 présente 3 *half buffers* A, B, C en série. Pour le *half-buffer* B, ReqA est retardé par un délai intrinsèque permettant de respecter la latence de la partie combinatoire. ReqdA est la requête venant de A et retardée de ce délai.

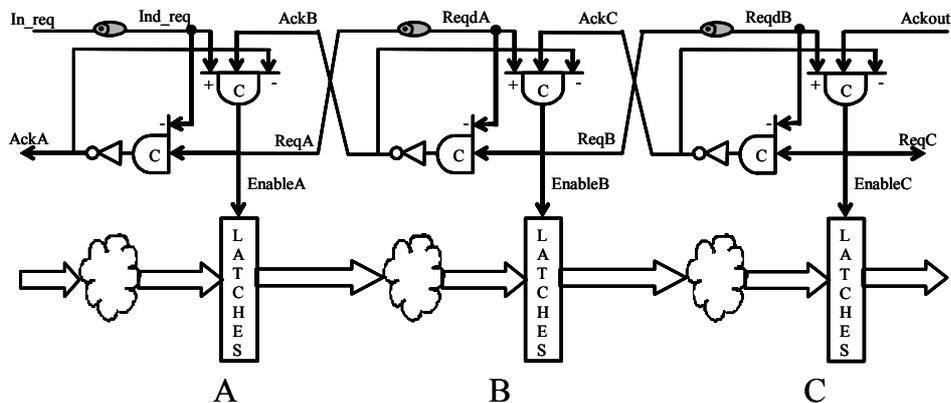


Figure 6-42 : 3 contrôleurs PCHB en série

6.7.3 Fonctionnement du protocole

L'étude du fonctionnement du protocole se fait sur le STG comme pour le protocole WCHB.

6.7.3a A l'initialisation

La Figure 6-43 montre le système à l'initialisation.

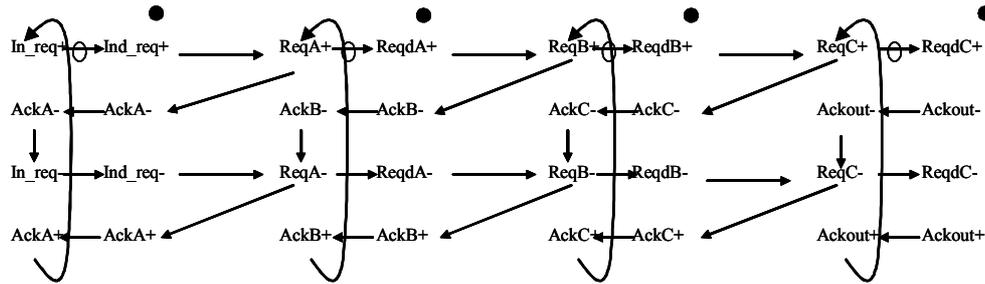


Figure 6-43 : STG des 3 opérateurs utilisant le protocole PCHB – Etat initial

6.7.3b Phase 1 de l'opérateur B

La Figure 6-44 montre le déclenchement de la phase 1 du *half buffer B*. Comme pour le protocole WCHB, celle-ci est déclenchée après la phase 2 de l'étage *A* qui envoie une requête au contrôleur *B* ($ReqA+$). Cette requête est retardée par le délai intrinsèque. Les *latches* du *half-buffer A* sont déverrouillées ($EnableA+$). Ainsi les parties combinatoires de l'opérateur *B* en amont des *latches* sont actives.

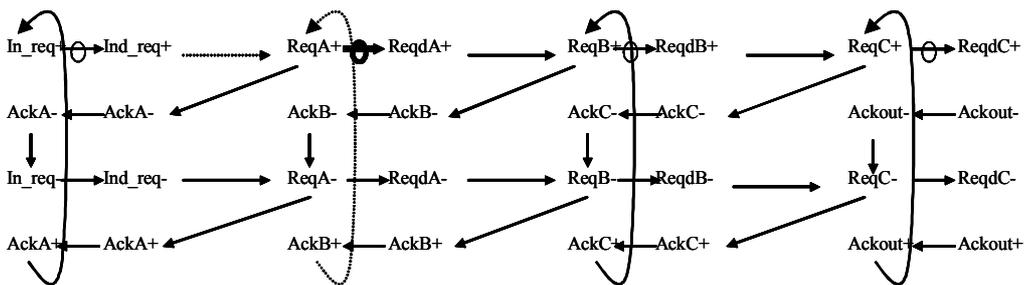


Figure 6-44 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 1

6.7.3c Phase 2 de l'opérateur B

La phase 2 de l'étage B (Figure 6-45) est déclenchée lorsque le signal de requête retardée *ReqdA* est actif. La phase 2 requiert également *AckC+* qui correspond à la fin de la phase 3 du contrôleur C. *AckC+* et *ReqdA* activent le contrôleur qui génère *ReqB+* et déverrouille les latches (*EnableB+*). Le contrôleur C passe alors en phase 1.

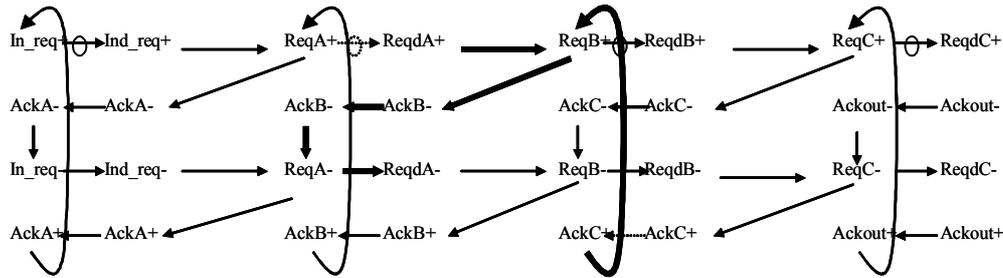


Figure 6-45 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 2

6.7.3d Phase 3 de l'opérateur B

La troisième phase est décomposable en 2 phases (Figure 6-46 et Figure 6-47).

La phase 3' (Figure 6-46) est déclenchée à la fin de la phase 2 de l'étage C (*AckC-*). La remise à 0 de la requête en sortie (*ReqB-*) est alors effectuée.

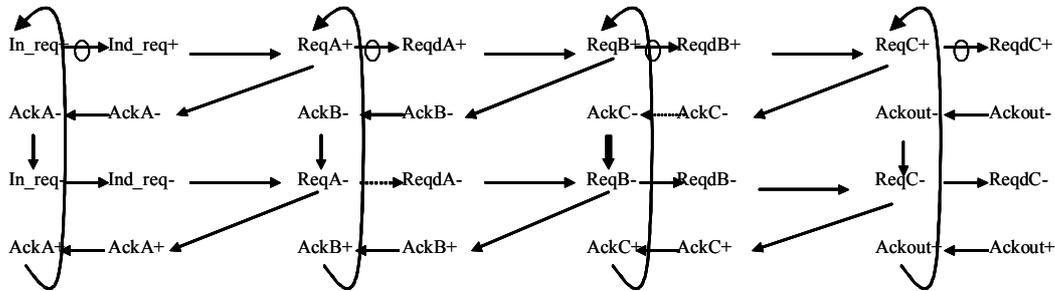


Figure 6-46 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 3'

La phase 3'' (Figure 6-47) est déclenchée à la fin de la phase 3' de l'étage A (*ReqA-*) et à la fin de la phase 3' de l'étage B (*ReqB-*). Pendant cette phase, l'acquittement d'entrée est remis à 1 (*AckB+*).

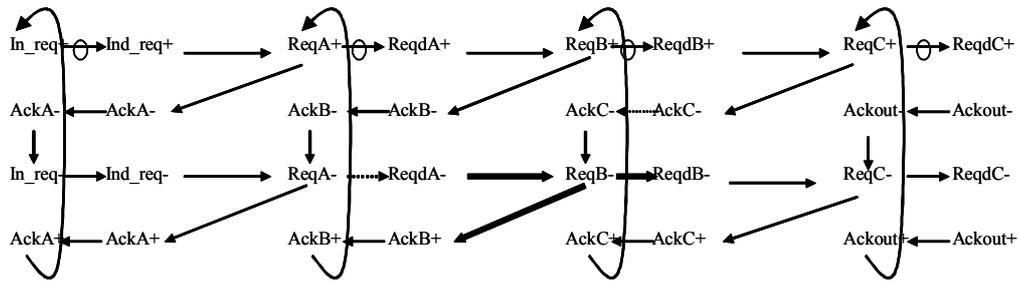


Figure 6-47 : STG des 3 opérateurs utilisant le protocole PCHB - Phase 3''

6.7.4 Phases de l'activité du courant

6.7.4a Modélisation par CDFG

L'étude du CDFG de la structure (Figure 6-48) permet de comprendre les dépendances entre les phases.

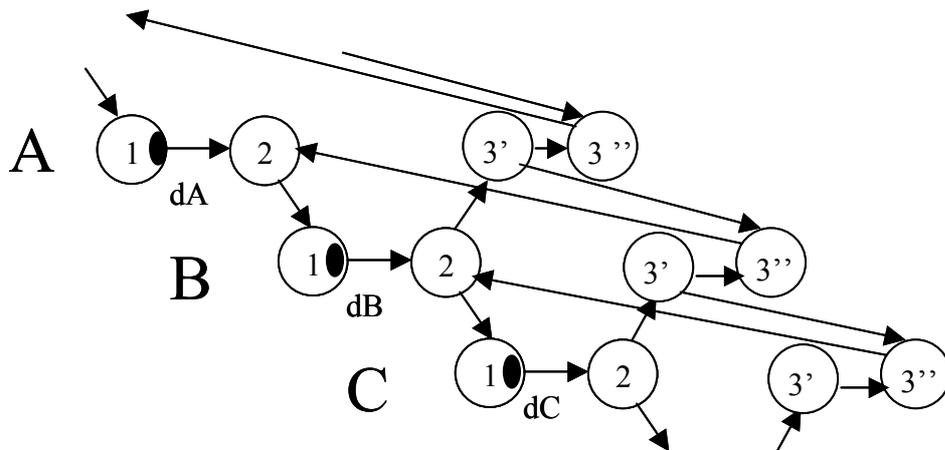


Figure 6-48 : CDFG des 3 opérateurs micropipeline utilisant le protocole PCHB

La décomposition de la troisième phase permet de désynchroniser la remise à 0 de la requête de sortie (phase 3') et la remise à 1 de l'acquittement d'entrée (Phase 3''). Cependant, la phase 3'' dépend directement de la phase 3', ce qui fait du protocole PCHB un protocole semi découplé.

6.7.4b Temps de cycle d'un opérateur

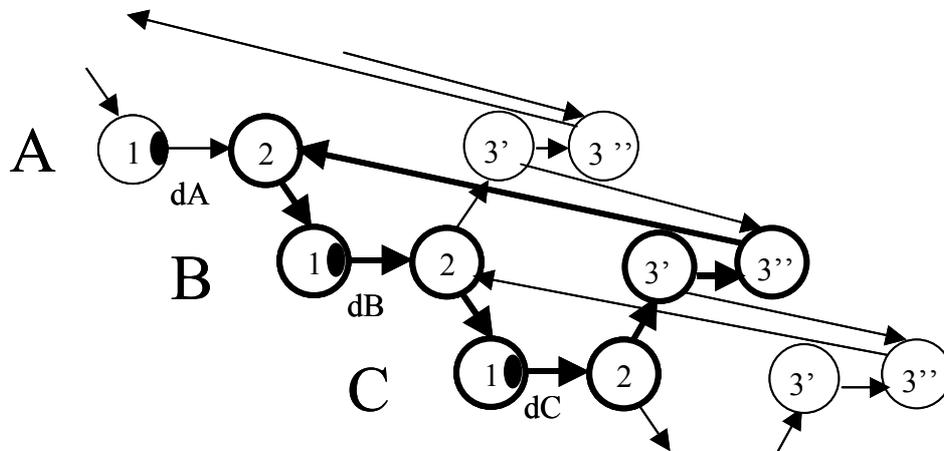


Figure 6-49 : Chemin critique pour l'activité en courant de l'opérateur B (protocole PCHB)

Le temps de cycle de l'opérateur B est représenté en gras sur la Figure 6-49. Il est égal à :

$$T = dB + \text{délai_phase_2B} + dC + \text{délai_phase_2C} + \text{délai_phase3B} + \text{délai_phase2A}$$

Équation 6-3

Comme pour le protocole WCHB, le temps de cycle du contrôleur PCHB dépend des autres phases de l'activité du courant des autres contrôleurs. A noter que le temps de cycle du contrôleur PCHB est du même ordre de grandeur que celui du WCHB.

6.7.5 Dépendances

6.7.5a Influence des évènements

La Figure 6-50 montre le CDFG déroulé du système représenté Figure 6-42. Les évènements relatifs aux dépendances ont été notés.

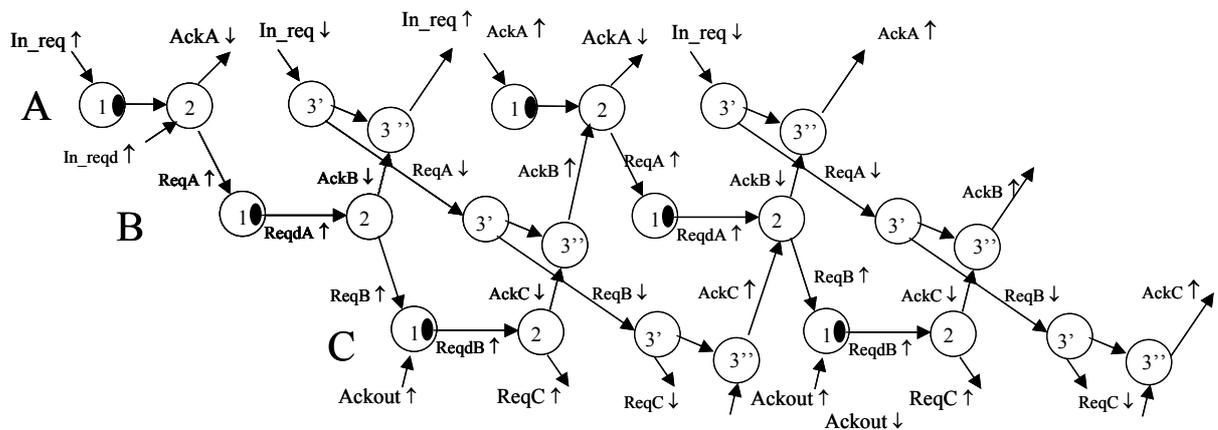


Figure 6-50 : CDFG déroulé des 3 contrôleurs A, B, C

Par souci de clarté, les liaisons entre les phases 3'' et 1 d'un même contrôleur (implicites dans la représentation CDFG) n'ont pas été représentées dans ce CDFG déroulé.

6.7.5b Les types de dépendances

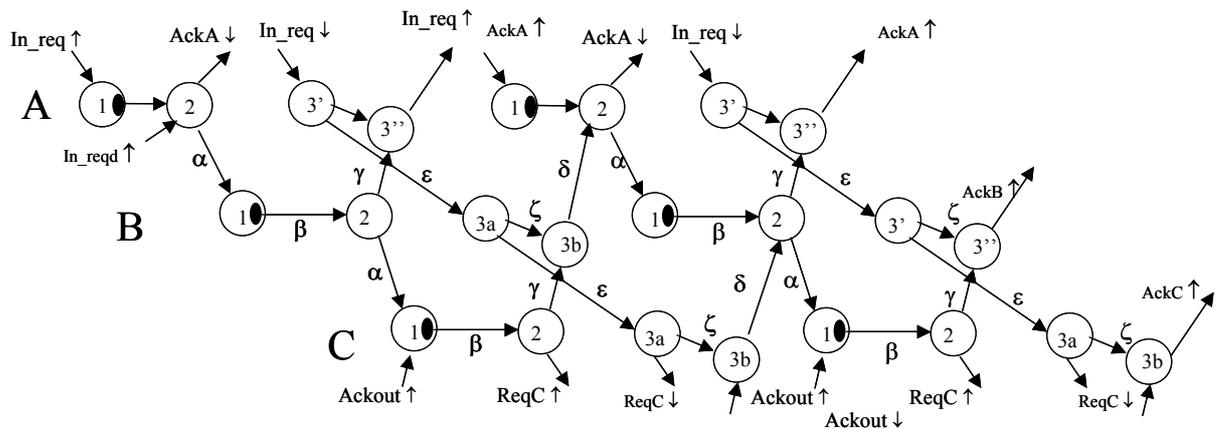


Figure 6-51 : Dépendances dans le protocole PCHB

Les dépendances α , β , δ , ε et γ du protocole WCHB (6.6.5b) se retrouvent dans le protocole PCHB. Cependant, une dépendance ζ a été ajoutée, elle correspond à la liaison entre les phase 3' et 3'' (Figure 6-51).

6.7.5c Dépendances lâches et fixes

Comme pour le protocole WCHB, les dépendances β , γ et δ sont lâches tandis que la dépendance α est fixe.

ζ , Correspond à la mise à 0 de la requête B ($ReqB$ -) qui est contrôlée par un signal interne au contrôleur. ζ est donc une dépendance fixe.

6.7.6 Contrôle des dépendances

Maintenant que les dépendances et leur nature sont identifiées, l'insertion des délais dans les liaisons peut être faite.

6.7.6a Dépendances lâches

Pour contrôler β et ε , la solution consiste comme pour le protocole WCHB à insérer un délai dissymétrique sur la requête de sortie et un sur la requête d'entrée (Figure 6-52).

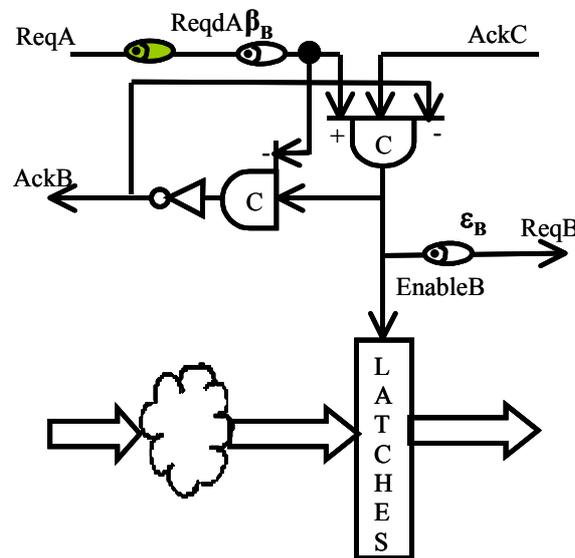


Figure 6-52 : Contrôle de β et ε

La Figure 6-53 représente le cas où γ et δ sont modifiées. La solution consiste comme pour le protocole WCHB à mettre un délai dissymétrique dans l'acquittement d'entrée du contrôleur. Il permet ainsi de gérer la mise à 1 et à 0 de ce signal.

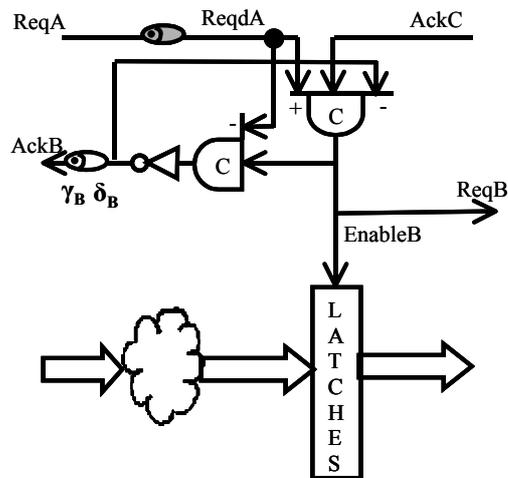


Figure 6-53 : Contrôle de γ et δ

6.7.6b Dépendances fixes

Le contrôle de α et ζ passe par une modification interne du contrôleur.

La Figure 6-54 montre le délai inséré pour contrôler α . Il permet de retarder la requête en sortie ainsi que le déverrouillage des *latches* de *B*.

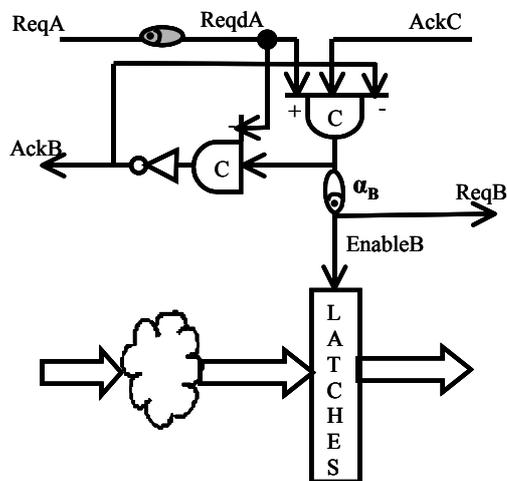


Figure 6-54 : Contrôle de α

Dans la Figure 6-55, un délai est placé sur *AckB*, à l'intérieur du contrôleur, afin de contrôler ζ . Il contrôle le signal interne qui permet de désynchroniser les phases 3' et 3''.

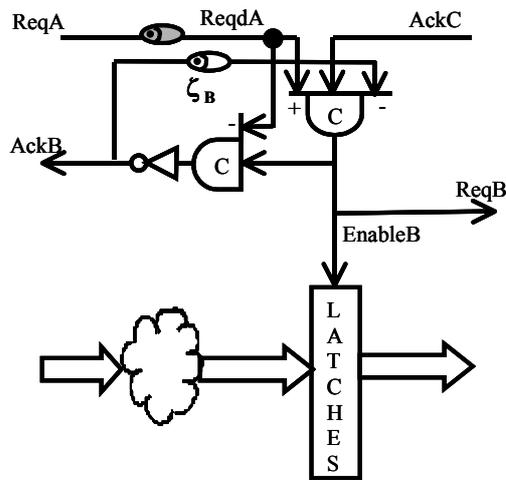


Figure 6-55 : Contrôle de ζ

6.7.6c Contrôleur générique

La Figure 6-56 montre comment à partir de l'implémentation de la Figure 6-53; le contrôle des dépendances γ et δ peut s'effectuer à l'intérieur du contrôleur.

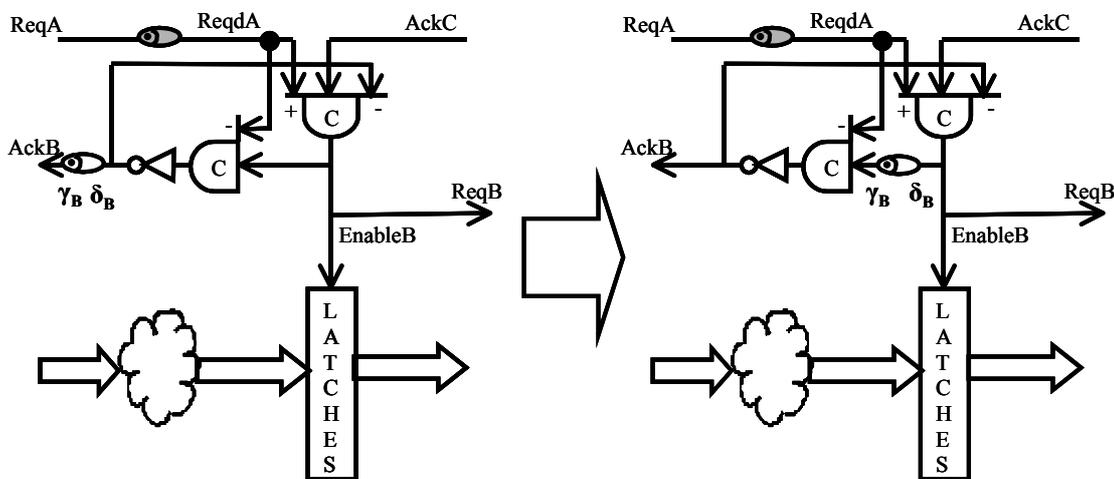


Figure 6-56 : Implémentation équivalentes

Le contrôleur générique proposé pour le contrôle de l'ensemble des dépendances est donné Figure 6-57. Seulement 3 délais sont à insérer. Comme pour le protocole WCHB, les propriétés de l'architecture des délais dissymétriques sont exploitées pour gérer plusieurs dépendances.

Pour le duplicateur, les liaisons communes au protocole WCHB sont gérées de la même manière :

$\gamma B1$ et $\gamma B2$, $\delta B1$ et $\delta B2$ sont fixes l'une par rapport à l'autre. Elles ne peuvent pas être traitées indépendamment. Cependant, les couples $\{\gamma B1, \gamma B2\}$ et $\{\delta B1, \delta B2\}$ forment des dépendances lâches.

$\alpha A1$ et $\alpha A2$ déclenchent respectivement les phases 1 des contrôleurs $B1$ et $B2$. En allongeant les délais dans ces dépendances, ces déclenchements peuvent être retardés. La remarque est la même pour $\varepsilon B1$ et $\varepsilon B2$.

Les dépendances $\beta B1$ et $\beta B2$ sont indépendantes et peuvent être modifiées.

Par ailleurs, les dépendances ζ , spécifiques au PCHB, sont lâches.

Pour le Rendez-vous, les liaisons communes au protocole WCHB sont gérées de la même manière :

Les couples $\{\alpha A1, \alpha A2\}$ et $\{\varepsilon A1, \varepsilon A2\}$ forment des dépendances lâches mais contiennent des liaisons fixes l'une par rapport à l'autre.

$\gamma B1$, $\gamma B2$, $\delta B1$, $\delta B2$, sont des dépendances lâches.

Par ailleurs, les dépendances ζ sont également lâches.

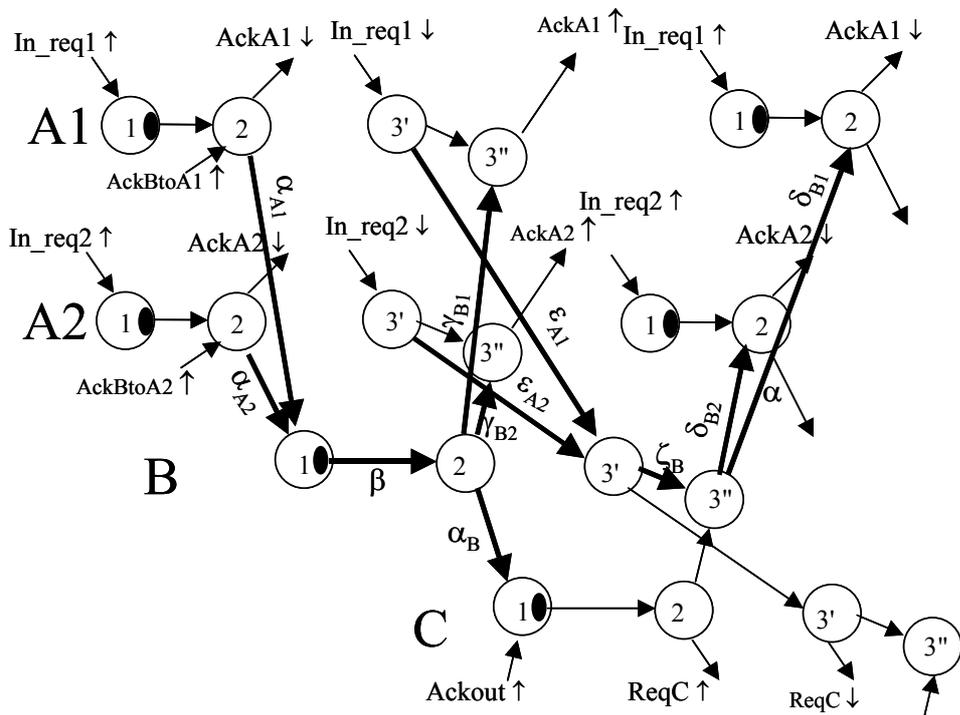


Figure 6-59 : Rendez-vous – PCHB

6.8 Protocole PCFB

Le protocole PCFB (*Precharge logic Full Buffer*) découple la remise à 0 de la requête de sortie et la remise à 1 de l'acquittement d'entrée. C'est donc un protocole découplé.

6.8.1 Formalisme

Le HSE qui décrit le contrôleur PCFB est le suivant :

$$*[[Sacq \wedge Ereq]; Sreq+; Eacq-; En-; ([\neg Sacq]; Sreq-); ([\neg Ereq]; Eack+); En+[En]]$$

Le contrôleur proposé dans [RIGA02] étudié est donné Figure 6-60.

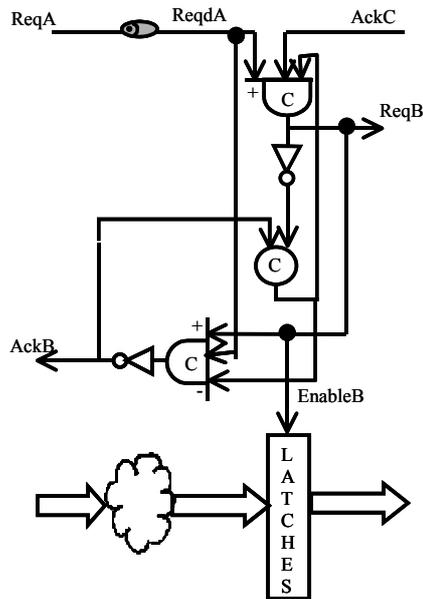


Figure 6-60 : Contrôleur PFCB

Comme pour le protocole WCHB et PCHB, *EnableB* est directement lié à la requête de sortie *ReqB*. A noter que la structure de ce contrôleur permet d'ajouter une variable interne (*En*) dans le HSE.

6.8.2 Architecture du système étudié

La Figure 6-61 représente le système *A, B, C* des 3 contrôleurs PFCB en série.

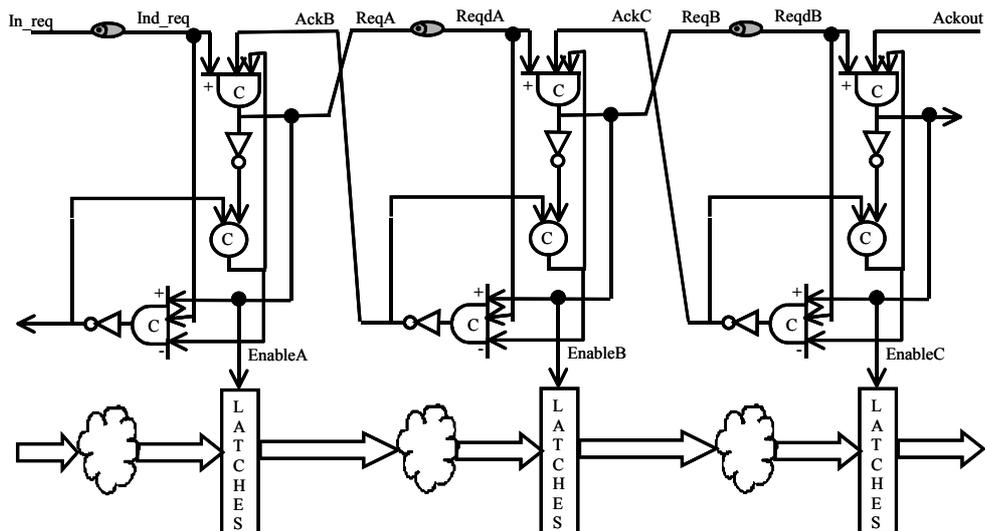


Figure 6-61 : 3 Contrôleurs PCFB en série

6.8.3 Fonctionnement du protocole

Comme pour le protocole PCHB, dans le protocole PCFB, la phase 3 peut être décomposée en phase 3' et phase 3'' (6.7.3d).

6.8.3a A l'initialisation

Le STG de la Figure 6-62 montre le système à l'état initial.

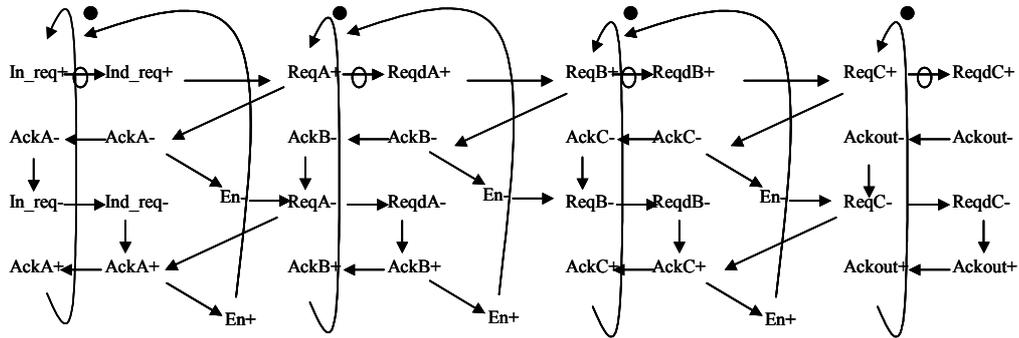


Figure 6-62 : STG des 3 opérateurs utilisant le protocole PCFB – Etat initial

6.8.3b Phase 1 de l'opérateur B

Comme le montre la Figure 6-63, la phase 1 de l'opérateur B est déclenchée à la fin de la phase 2 de l'étage A ($ReqA+$), comme pour les protocoles WCHB et PCHB. Cette requête est retardée par le délai intrinsèque. Les latches du half-buffer A sont déverrouillés ($EnableA+$) et les parties combinatoires de l'opérateur B en amont des latches sont actives.

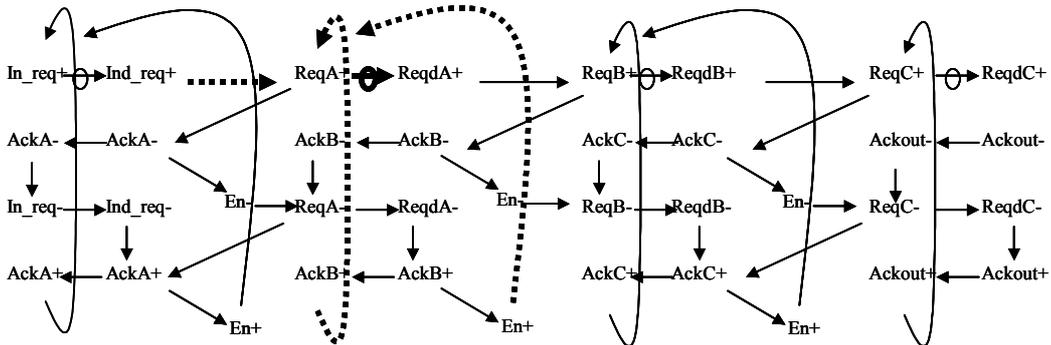


Figure 6-63 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 1

6.8.3c Phase 2 de l'opérateur B

La phase 2 de l'étage B (Figure 6-64) est déclenchée à la fin de la phase 1 de B (*ReqdA+*). Comme pour le protocole PCHB, elle est déclenchée à la fin de la phase 3'' de l'étage C (*AckC+*). Cette phase génère l'activation de l'acquiescement en entrée (*AckB-*) et celle de la variable interne (*En-*). Le contrôleur C passe en phase 1.

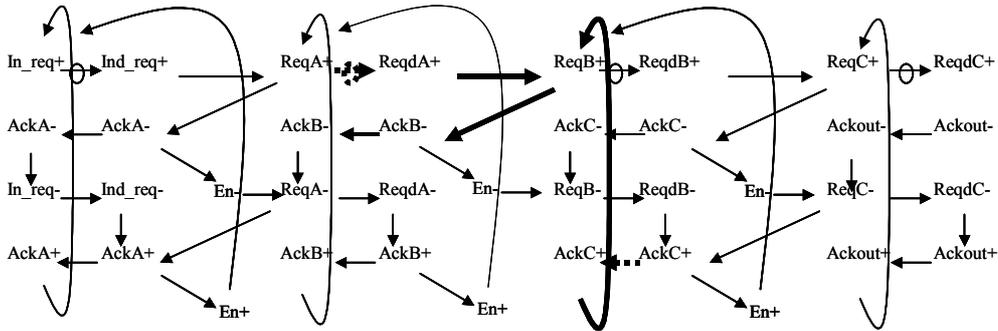


Figure 6-64 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 2

6.8.3d Phase 3 de l'opérateur B

Comme pour le protocole PCHB, la phase 3 est décomposable en 2 phases : la phase 3' et la phase 3''.

La phase 3' de l'étage B est déclenchée à la fin de la phase 2 de l'étage C (*AckC-*) et à la fin de la phase 2 de l'étage B (*En-*) (Figure 6-65). Cette phase permet de générer la mise à 0 de la requête en sortie (*ReqB-*).

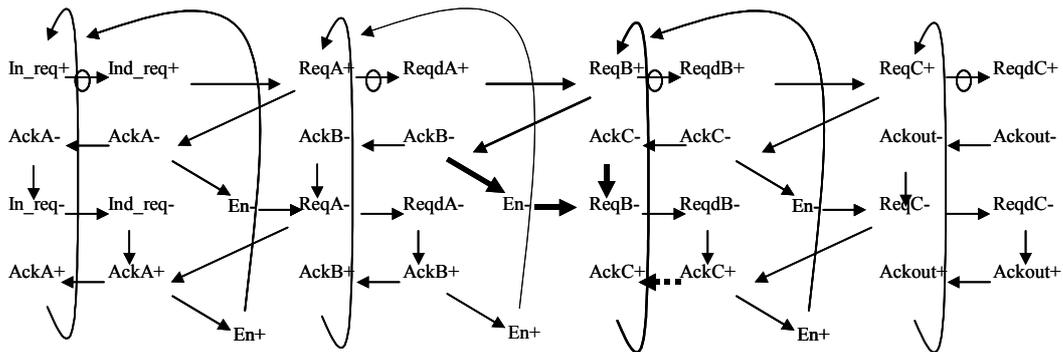


Figure 6-65 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 3'

La phase 3'' est déclenchée à la fin de la phase 3' de l'étage A (Figure 6-66). L'acquiescement d'entrée est remis à 1 (*AckB+*). Par conséquent, les phases 3' et 3'' sont désynchronisées, ce qui fait le protocole découplé.

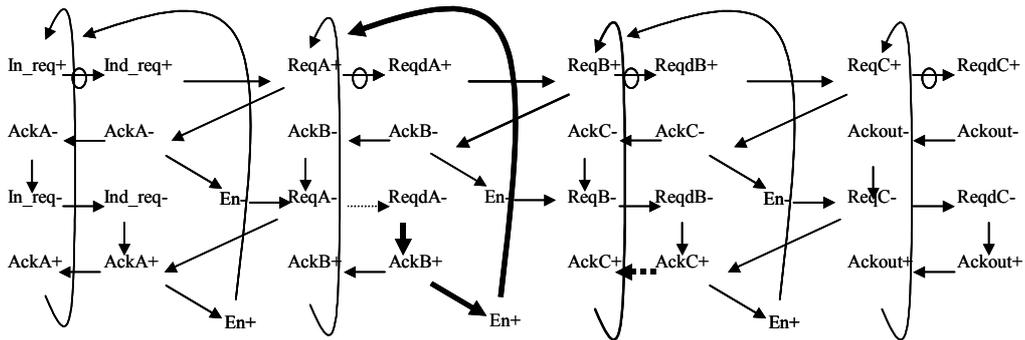


Figure 6-66 : STG des 3 opérateurs utilisant le protocole PCFB - Phase 3''

6.8.4 Les phases de l'activité du courant

6.8.4a Modélisation par CDFG

Le CDFG Figure 6-67 représente le fonctionnement du système.

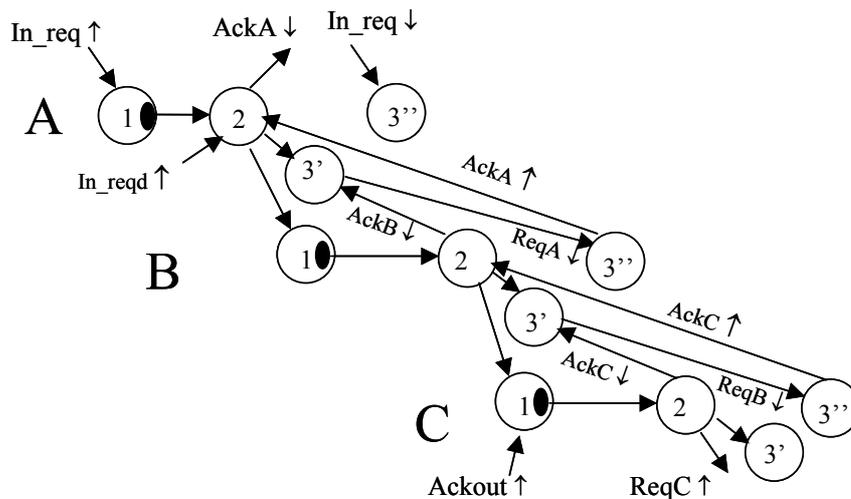


Figure 6-67 : CDFG des 3 opérateurs micropipeline utilisant le protocole PCHB

La décomposition de la troisième phase permet de désynchroniser la mise à zéro (phase 3') de la requête de sortie et la mise à 1 de l'acquittement d'entrée (Phase 3''). Contrairement au protocole PCHB, la phase 3'' et la phase 3' ont des fonctionnements concurrents grâce à l'utilisation de la variable interne (*En*), ce qui fait du protocole PCFB un protocole découplé. Les étages seront alors appelés *Full-Buffer*.

6.8.4b Temps de cycle d'un opérateur

Pour l'opérateur *B*, le temps pour effectuer un cycle est égal à :

$$T = dB + \text{délai_phase_2B} + \text{délai_phase_3'A} + \text{délai_phase_3''B} + \text{délai_phase_2A}$$

Équation 6-4

Il est représenté en gras sur la Figure 6-68.

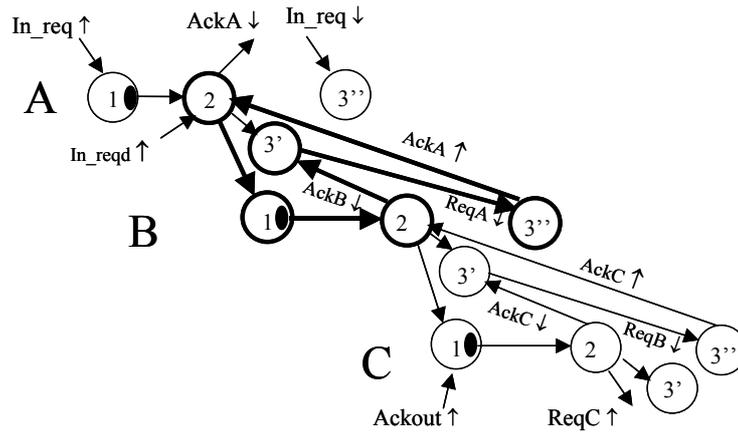


Figure 6-68 : Temps de cycle de l'opérateur B dans le protocole PCFB

6.8.5 Dépendances

6.8.5a Influence des évènements

Comme pour le protocole PCHB, le CDFG déroulé est étudié pour identifier les influences des évènements. La Figure 6-69 représente donc le comportement du système dans le temps.

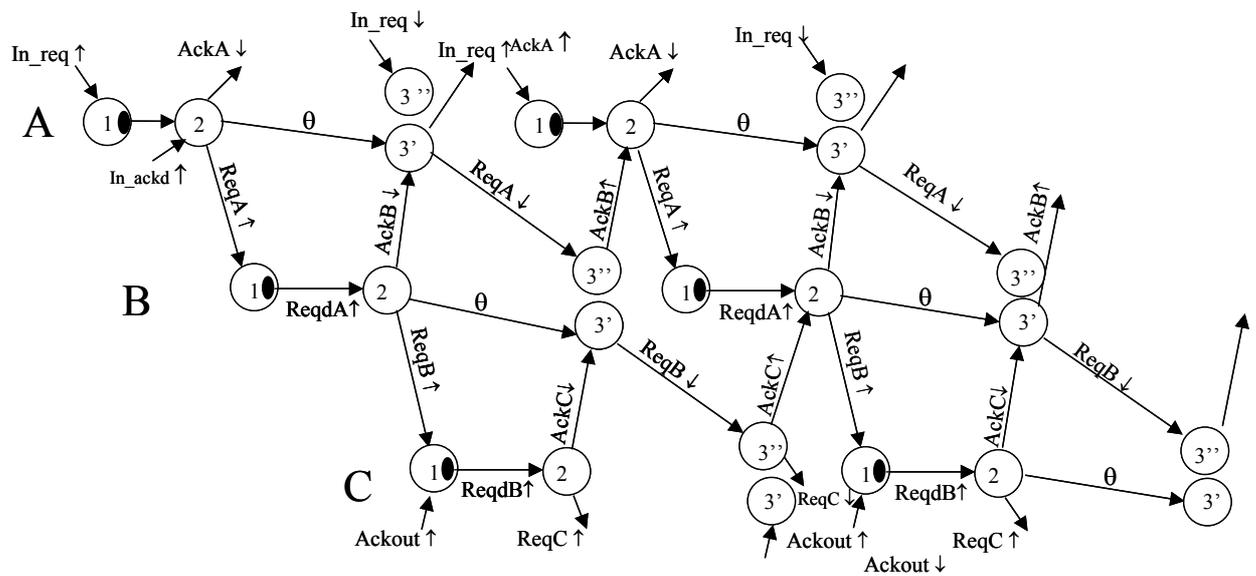


Figure 6-69 : CDFG déroulé

Par souci de clarté, les liaisons entre les phases 3', 3'' et 1 d'un même contrôleur (implicites dans la représentation CDFG) n'ont pas été représentées dans ce CDFG déroulé.

6.8.5b Type de dépendances

Les dépendances du protocole WCHB (α , β , δ , ε et γ) se retrouvent dans le protocole PCFB (Figure 6-70). Contrairement au protocole PCHB, il n'existe pas de dépendance ζ entre les phases 3' et 3'' d'un même contrôleur. Une liaison entre la phase 2 à la phase 3 (θ) ainsi que la dépendance ε (allant de la phase 3' du contrôleur précédent à la phase 3'') permettent de désynchroniser l'entrée de la sortie et de garder la séquence des évènements.

6.8.5c Dépendances fixes et lâches

Comme pour le protocole PCHB, les 5 liaisons principales du protocole WCHB sont identifiées. De plus, θ permet de faire la désynchronisation entre les sorties et les entrées.

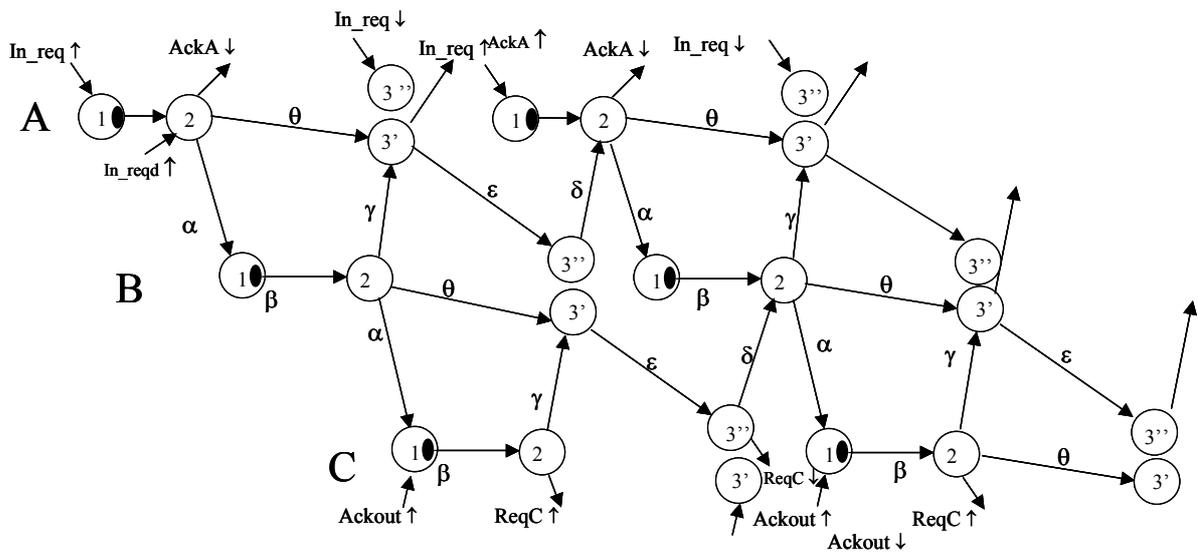


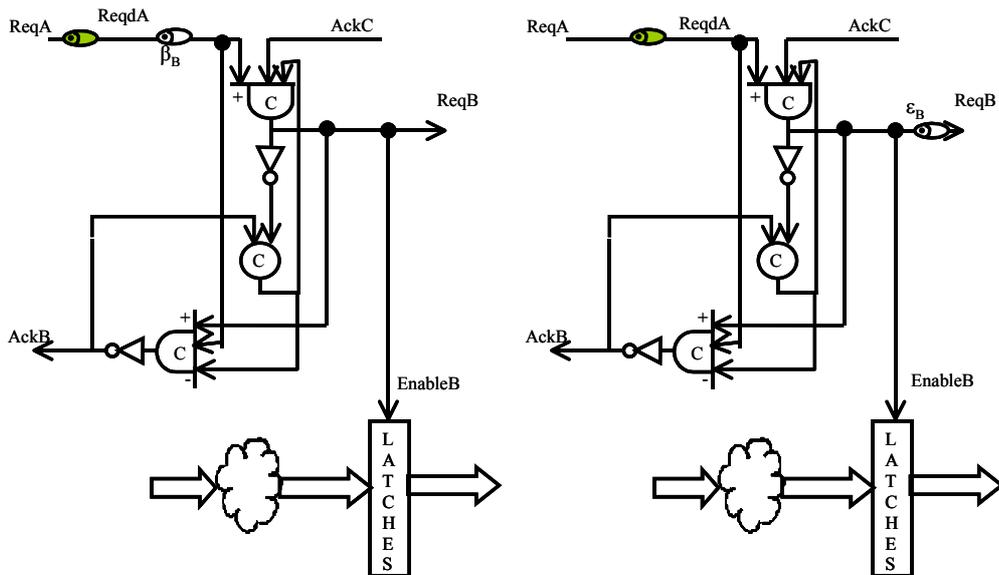
Figure 6-70 : Les dépendances dans le protocole PCFB

Comme pour le protocole WCHB, les dépendances β , γ et δ sont lâches tandis que la dépendance α est fixe.

θ est gérée à l'intérieur du contrôleur (par l'utilisation de la variable interne En) donc ne peut pas être retardée sans modification du contrôleur.

6.8.6 Contrôle des dépendances

Comme pour le protocole PCHB, le contrôle de β et ε (Figure 6-71) se fait par l'insertion de délais dissymétriques sur la requête de sortie et sur la requête d'entrée.



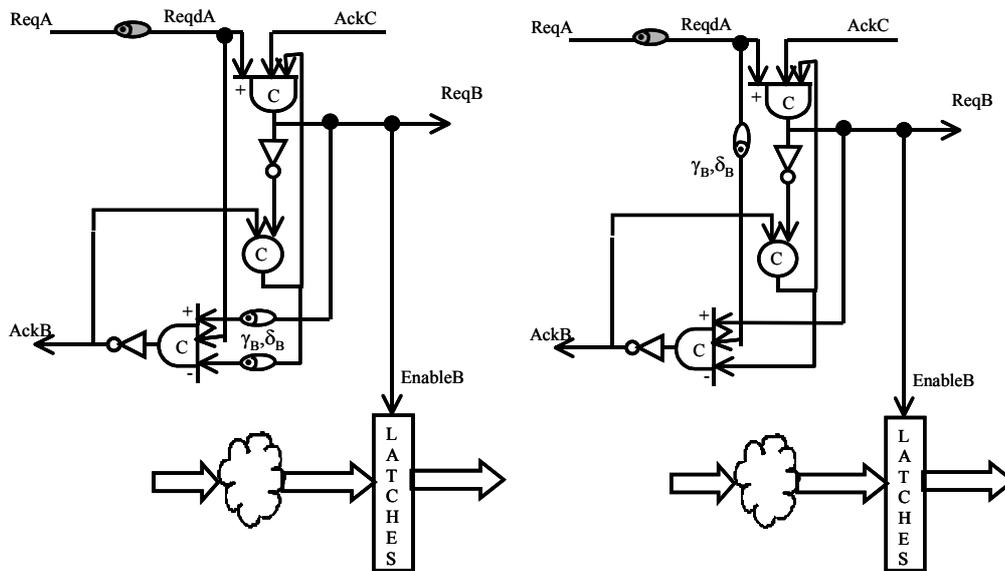


Figure 6-73 : Contrôle de γ et δ

Pour contrôler la variable interne (θ), 2 solutions sont présentées Figure 6-74. Ces solutions sont équivalentes. Le choix sera effectué par le concepteur lui-même.

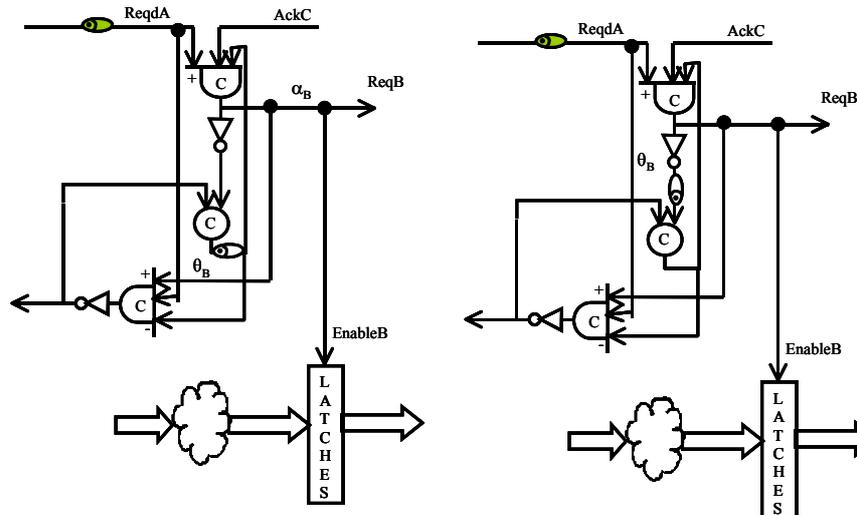


Figure 6-74 : Contrôle de la liaison θ dans le contrôleur PCFB

A noter qu'une troisième solution est possible en déplaçant le délai de la solution de droite avant l'inverseur.

Contrôleur générique

Comme pour les protocoles WCHB et PCHB, un contrôleur générique est proposé. Il utilise également les propriétés des délais dissymétriques pour gérer plusieurs dépendances.

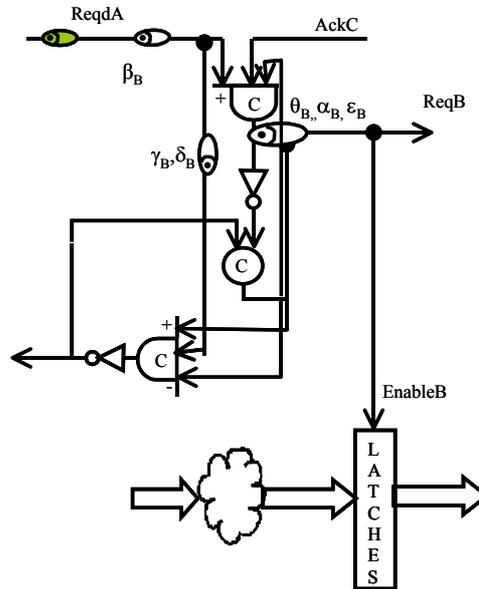


Figure 6-75 : Contrôleur générique pour le protocole PCFB

6.8.7 Autres structures

Cette étude de cas se limite aux structures RV et Duplicateur. Comme pour les protocoles précédents, ces structures sont symétriques. Pour les dépendances α , β , δ , ϵ et γ , les caractéristiques dans les structures RV et Duplicateur sont identiques à celles des structures des protocoles PCHB et WCHB.

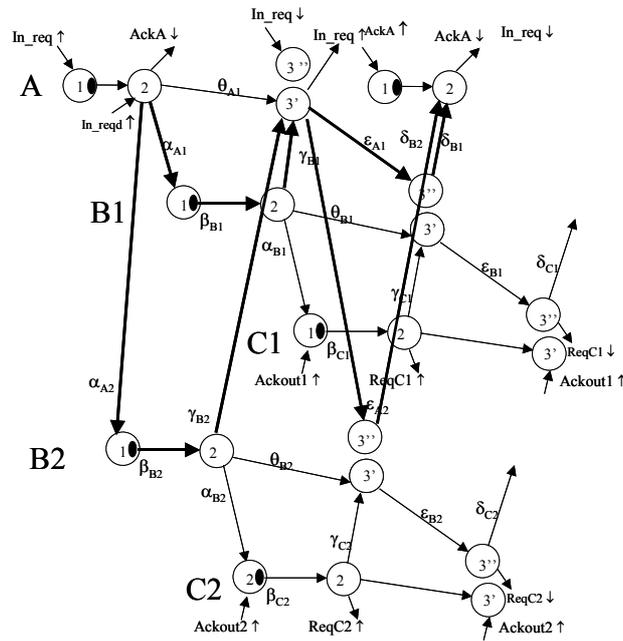


Figure 6-76 : Duplicateur PCFB

Pour le duplicateur (Figure 6-76), les liaisons communes aux protocoles WCHB et PCHB sont gérées de la même manière :

$\gamma B1$ et $\gamma B2$, $\delta B1$ et $\delta B2$ sont fixes l'une par rapport à l'autre. Elles ne peuvent pas être traitées indépendamment. Cependant, les couples $\{\gamma B1, \gamma B2\}$ et $\{\delta B1, \delta B2\}$ forment des dépendances lâches.

$\alpha A1$ et $\alpha A2$ déclenchent respectivement les phases 1 des contrôleurs $B1$ et $B2$. En allongeant les délais dans ces dépendances, ces déclenchements peuvent être retardés. La remarque est la même pour $\varepsilon B1$ et $\varepsilon B2$.

Les dépendances $\beta A1$ et $\beta A2$ sont indépendantes et peuvent être modifiées.

Par ailleurs, les dépendances θ , spécifiques au protocole PCFB, sont lâches.

Pour le Rendez-vous (Figure 6-77), les liaisons communes aux protocoles WCHB et PCHB sont gérées de la même manière :

Les couples $\{\alpha A1, \alpha A2\}$ et $\{\varepsilon A1, \varepsilon A2\}$ forment des dépendances lâches mais contiennent des liaisons fixes l'une par rapport à l'autre.

$\gamma B1$, $\gamma B2$, $\delta B1$, $\delta B2$ sont des dépendances lâches.

Par ailleurs, la dépendance θ est également lâche.

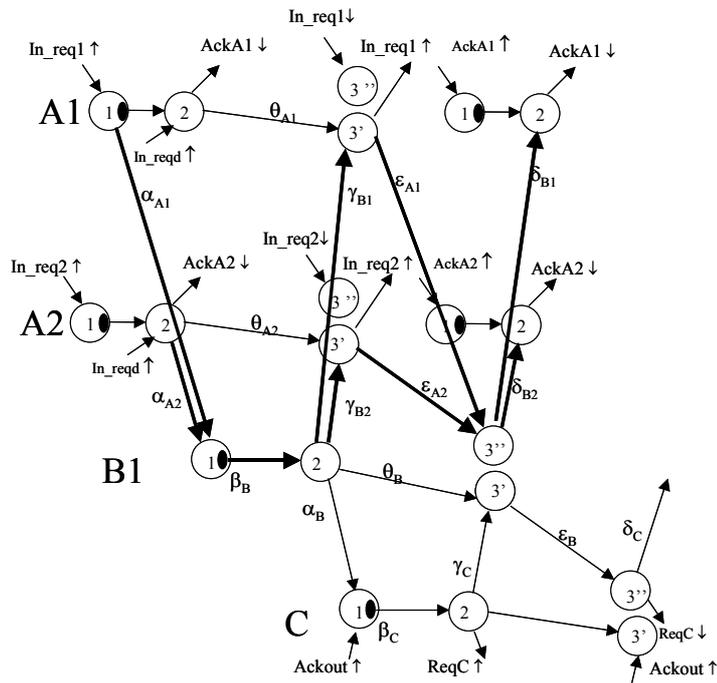


Figure 6-77 : Rendez-vous - PCFB

6.9 Synthèse de l'étude des protocoles

La table suivante résume les dépendances identifiées dans les protocoles WCHB, PCHB, PCFB et les contrôleurs génériques proposés pour la méthode "mise en forme du courant".

protocole	phases identifiées	dépendances identifiées	contrôleur proposé
WCHB	phase1, phase2, phase3	α , β , δ , ϵ et γ (5 dépendances)	

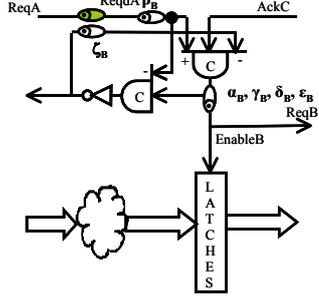
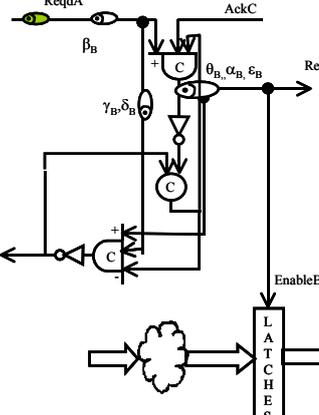
PCHB	phase1, phase2, phase3', phase3''	$\alpha, \beta, \delta, \varepsilon, \gamma$ et ζ (5 dépendances)	
PCFB	phase1, phase2, phase3', phase3''	$\alpha, \beta, \delta, \varepsilon, \gamma$ et θ (6 dépendances)	

Tableau 6-5 : Synthèse de l'étude des protocoles

6.10 Utilisation de l'étude

Ce paragraphe présente les utilisations de l'étude qui a été faite précédemment.

6.10.1 Périodicité des protocoles

La périodicité des protocoles peut être déduite de l'étude faite précédemment. Connaître la périodicité d'un protocole permet de prédire les événements concurrents qui peuvent survenir lors du fonctionnement du circuit. Plus particulièrement, la détection du recouvrement des phases des contrôleurs permettra d'effectuer l'optimisation des pics de courant.

Le protocole WCHB est pris pour exemple pour illustrer la technique (Figure 6-78).

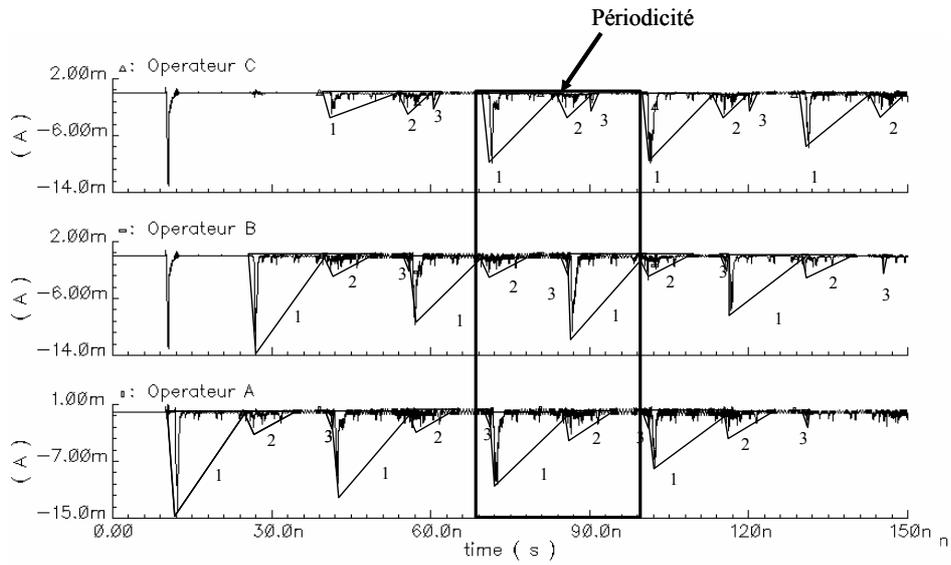


Figure 6-80 : Simulation électrique de A,B,C - protocole WCHB

La Figure 6-80 montre les résultats obtenus de la simulation électrique du système A,B,C quand $dA=dB=dC$. Les différentes phases ont été identifiées (triangle) et les résultats sont conformes à la prédiction précédente utilisant le HSE temporel.

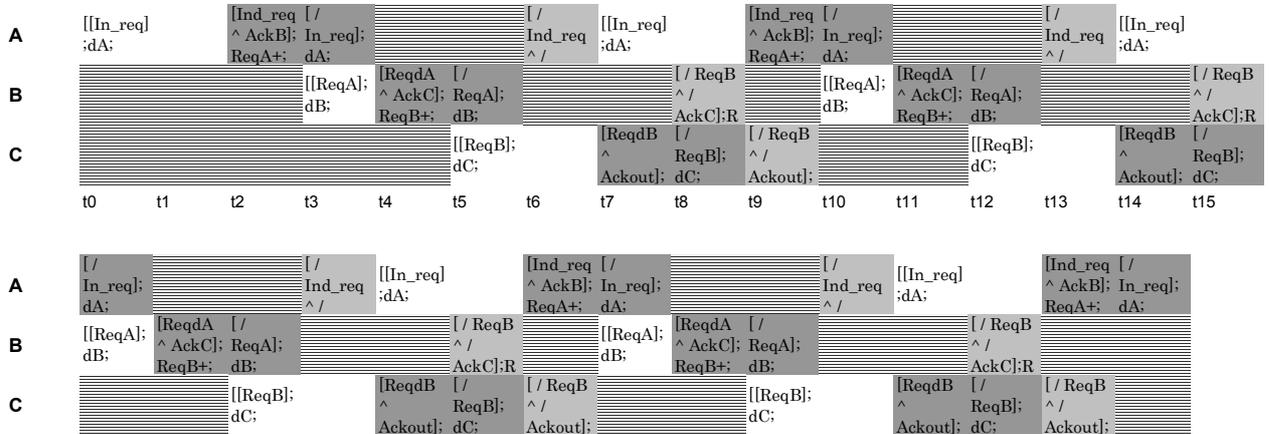


Figure 6-81 : WCHB cas où $dA=dC \neq dB$

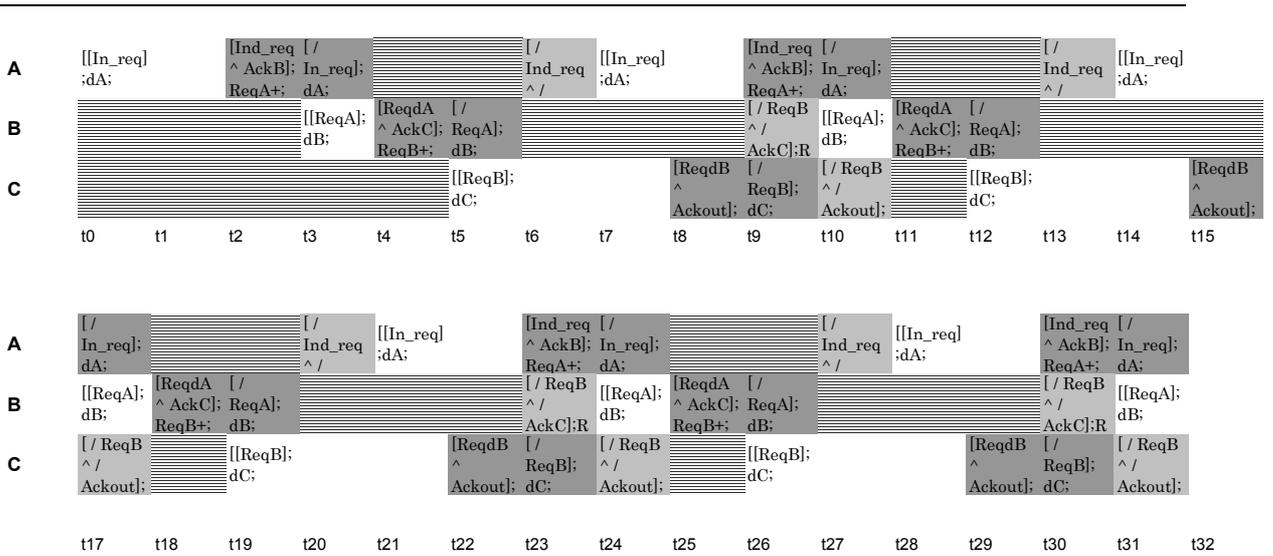


Figure 6-82 : WCHB cas où $dA \neq dC \neq dB$

Pour identifier les recouvrements de phases, le régime transitoire ne doit pas être pris en compte (avant $t6$). Dans le cas où les 3 délais sont identiques, 5 motifs de recouvrement de phase sont identifiés : $[3A,2B,1C]$; $[1A,1C]$; $[1A,2C]$; $[2A,3B,2C]$ et $[2A,1B,3C]$.

Le nombre de motifs de recouvrement N_{mr} est fonction du temps de cycle de chacun des opérateurs et de la durée de chacune de leurs phases :

$$N_{mr} = f(TA, TB, TC, dph1A, dph1B, dph1C, dph2A, dph2B, dph2C, dph3A, dph3B, dph3C)$$

Équation 6-5

TA , TB et TC , représentent les temps de cycle qui dépend de la latence des différentes phases. Dans le cas du protocole WCHB, les durées des phases 1 sont égales au délai intrinsèque et donc $dph1A = dA$, $dph1B = dB$ et $dph1C = dC$:

$$N_{mr} = f(TA, TB, TC, dA, dB, dC, dph2A, dph2B, dph2C, dph3A, dph3B, dph3C).$$

Équation 6-6

La périodicité des motifs rencontrés est égale au temps de cycle de l'opérateur le plus long:

$$P = \max(TA, TB, TC)$$

Équation 6-7

6.10.2 Protocole de communication et répartition du courant

De l'étude précédente et connaissant la périodicité des phases de l'activité du courant ainsi que les dépendances entre ces phases, une représentation graphique CDFG et CDFG déroulée modélisant les communications peut être utilisée par la méthode "mise en forme du courant".

La méthode "mise en forme du courant" peut donc répartir le courant dans le circuit en adressant le protocole de communication : les pas en temps et le poids des phases en courant de chaque opérateur seront déterminés comme décrit en 5.4.2. La méthode répartit ensuite les phases en privilégiant les motifs dont la somme des poids des phases concurrentes est faible. Cependant, la complexité du calcul augmente car le nombre de paramètres ajoutés augmente. Le travail du concepteur est de maintenant choisir quelles dépendances du circuit il doit traiter pour réduire la complexité et le temps du calcul.

6.11 Conclusion

L'étude des protocoles 4 phases WCHB, PCHB et PCFB pour la faible émission électromagnétique porte sur l'analyse des dépendances entre les phases de l'activité en courant.

Trois phases principales en courant ont été identifiées dans les protocoles 4 phases. Des "sous-phases" sont ajoutées à la décomposition du modèle selon la synchronisation des signaux d'entrée et de sortie. Cela correspond aux phases 3 des protocoles PCHB et PCFB qui sont décomposées en 2 phases (3' et 3").

Les types de dépendances ont été identifiés et classifiés (fixes ou lâches). Des délais sont alors insérés dans les liaisons et leurs valeurs sont calculées par la méthode "mise en forme du courant" afin de répartir les activités dans le temps. Le temps est contraint par le cycle des opérateurs et la périodicité du circuit.

L'optimisation pourra s'effectuer soit partiellement en adressant certains types de dépendances (ce qui allège le calcul), soit globalement en utilisant les contrôleurs génériques proposés.

Ce choix dépendra de divers facteurs tels que le temps de calcul, la consommation, la surface...

Conclusion

Dans un circuit numérique synchrone, la principale source d'émission électromagnétique est l'activité du signal d'horloge. Des milliers de portes commutent à la cadence de l'horloge et provoquent des appels de courant brusques sur l'alimentation créant ainsi de fortes ondes électromagnétiques. Celles-ci sont susceptibles d'interférer avec d'autres ondes de même fréquence et peuvent être aussi à l'origine du dysfonctionnement du circuit ou d'autres circuits dont les lignes et les boucles sont réceptrices.

Pour diminuer le rayonnement électromagnétique d'un circuit, des méthodes existent et s'appliquent à plusieurs stades d'élaboration du circuit. En particulier, dès la conception, les méthodes actuelles proposent de répartir l'activité électrique dans le circuit afin que les variations du courant soient moins élevées. A différents niveaux de la conception, ces méthodes agissent sur l'étalement du spectre de l'horloge, sur le déphasage de l'horloge ou par ordonnancement des actions dans le circuit.

Ce travail de thèse s'adresse aussi au circuit dès sa conception. Il apporte des méthodes basées sur les circuits asynchrones dont le rayonnement électromagnétique est naturellement inférieur au circuit synchrone. En effet, dans ce type de circuit, le courant est réparti grâce à la synchronisation locale des actions par des communications de type poignée de mains. Plusieurs catégories de circuits asynchrones existent. Le choix se fera selon les hypothèses en temps et l'environnement du circuit. Ce travail de thèse s'est porté essentiellement sur les circuits de type micropipeline où des délais dans les canaux de communications, équivalents aux latences des parties combinatoires permettent d'assurer la validité des données.

Une méthode, appelé *Asynchronisation*, a été élaborée et validée sur l'exemple pédagogique d'un filtre à Réponse Impulsionnelle Finie. Elle consiste à remplacer le circuit synchrone par un circuit asynchrone dont le fonctionnement est identique mais dont le rayonnement est inférieur. L'analyse de la description RTL (*Register Transfer Level*) du circuit synchrone permet d'identifier des dépendances entre les registres. Ces dépendances forment différents types de structures identifiées. Les parties de contrôles composées de l'horloge et des composantes séquentielles qui lui sont sensibles sont ensuite remplacées par des verrous commandés par des contrôleurs communiquant par le biais de canaux. La méthode conserve la compatibilité cycle à cycle du circuit. Ceci permet en particulier d'utiliser le circuit obtenu dans

un mode synchrone et de le valider avec l'environnement de test du circuit d'origine. Cependant, le circuit obtenu possède une surface plus grande que le circuit d'origine. Car un ensemble de bascules et un fil de transmission du signal d'horloge sont remplacés par des *latches*, un contrôleur et des canaux de communications. Des interfaces pour gérer la synchronisation avec l'extérieur sont aussi ajoutées. Pour gagner en surface, des architectures utilisant différents protocoles et des optimisations logiques ont été proposées.

Les perspectives de ce travail d'*asynchronisation* sont nombreuses :

- La méthode doit être automatisée. Ce travail d'automatisation passe par l'analyse du code RTL du circuit synchrone, et l'identification des dépendances. C'est alors que la méthode pourra prouver son efficacité sur des circuits plus complexes.
- L'ajout de surface est un réel inconvénient. Une étude pour la réduction de la surface du circuit est donc indispensable. Celle-ci peut être menée à plusieurs niveaux : portes (optimisation logique, utilisation de portes dédiées), architecture (utilisation de différents protocoles, de différentes portes) et comportement (factorisation des actions).
- Enfin, l'*Asynchronisation* doit être insérée dans le flot de conception TAST (*TIMA Asynchronous Synthesis Tools*).

Pour diminuer l'émission électromagnétique dans les circuits asynchrones, une méthode, appelée "mise en forme du courant" (*Current Shaping*) a été élaborée et aussi validée sur un filtre RIF. Elle consiste à contrôler la forme du courant en réduisant le nombre de traitements parallèles dans le circuit dès sa conception. Des délais sont ajoutés dans les communications afin de retarder des traitements par rapport à d'autres. Le temps est contraint pour ne pas ralentir le fonctionnement global du circuit. Pour connaître les délais à insérer, l'architecture du circuit et l'activité du courant sont tout d'abord modélisées. Ensuite, le modèle de l'architecture est analysé pour identifier les traitements concurrents. Les délais sont calculés par une technique d'ordonnancement qui utilise comme algorithme le *Force Directed Scheduling* et comme paramètres les modèles du courant et les concurrences des traitements. Les perspectives de ce travail sont nombreuses :

- Pour améliorer les performances en terme d'émission électromagnétique, le nombre de paramètres peut être augmenté en augmentant le niveau de granularité. Ainsi, des concurrences peuvent être trouvées dans les communications elles-mêmes. Le chapitre 6 étudie à ce sujet les protocoles de communications 4 phases.
- La méthode doit être automatisée. Ceci passe par l'analyse de la description du circuit, le calcul des pas en temps définis pour la répartition du courant, l'application de l'algorithme et l'insertion de délais dans l'architecture existante.
- Par ailleurs, ces délais doivent être optimisés d'une part en surface et d'autre part pour être le moins possible dépendants de la technologie utilisée. Dans un souci d'automatisation, ils

peuvent être aussi bien insérés comme paramètres génériques dans les contrôleurs comme le propose le chapitre 6.

Les performances obtenues sur un filtre RIF sont intéressantes malgré le faible nombre de concurrences. Ainsi, une fois automatisée, la méthode pourra être appliquée sur des circuits plus complexes dont le nombre de concurrences des traitements sera plus élevé. Les gains seront alors plus significatifs.

Enfin, tout comme l'*Asynchronisation*, la méthode "mise en forme du courant" pourra être insérée dans l'environnement TAST.

Cette thèse s'est terminée sur l'étude de protocoles de communications 4 phases utilisés dans les circuits asynchrones. Ceci permet d'affiner le modèle du courant dans le circuit et d'augmenter les paramètres utilisables pour l'ordonnancement. Pour les protocoles WCHB (*Weak Condition Half Buffer*), PCHB (*PreCharge Logic Half Buffer*) et PCFB (*PreCharge Logic Full Buffer*), les liaisons entre les phases de l'activité du courant dans les communications ont été déterminées. Des contrôleurs pour la faible émission électromagnétique ont été proposés. Ces contrôleurs utilisent des délais dissymétriques dont l'architecture permet de les optimiser en surface et consommation. Les perspectives de ce travail sont les suivantes :

- Travailler sur les délais pour améliorer leurs performances en surface et en consommation.
- Inclure les modèles des protocoles dans le modèle de l'architecture du circuit pour appliquer la méthode "mise en forme du courant".
- Créer une librairie avec les modèles des protocoles en vue d'une automatisation.
- Etudier d'autres protocoles de communication.

Ainsi cette thèse apporte des solutions pour réduire les émissions électromagnétiques dans les circuits numériques par optimisation de l'activité électrique dès le début de leur conception

D'une manière générale, les perspectives d'utilisation des circuits asynchrones avec ces méthodes d'optimisation sont nombreuses dans les systèmes à forte contrainte électromagnétique.

BIBLIOGRAPHIE

- [AIRI98] R. Airiau et al., "VHDL : du langage à la modélisation", Presses Polytechnique et Universitaires Romandes, 1990.
- [AGRA99] V. D. Agrawal, M. L. Bushnell, G. Parthasarathy, R. Ramadoss, "Digital Circuit Design for Minimum Transient Energy and a linear Programming Method", 12th International Conference on VLSI Design, January 1999.
- [ALLI03] E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "A new class of asynchronous A/D converters based on time quantization", Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC'03), Vancouver, Canada, 12-16 May , 2003.
- [ARAGO2] X. Aragonès, J.L. Gonzalez, F. Moll, A. Rubio, "Noise Generation and Coupling Mechanisms in Deep-Sbmicron IC's", IEEE Design & Test of Computers , 19 (5), pp.27-35, 2002.
- [AZEVO1] A. Azevedo et al., "Architectural and Compiler Strategies for Dynamic Power Management in the COPPER Project", International Workshop on Innovative Architecture (IWIA 2001), Maui, Hawaii, January, 2001.
- [BARD00] A. Bardsley, D. Edwards, "Synthesising an Asynchronous DMA controller with Balsa", Journal of Systems Architecture, 46, pp.1309-1319, 2000.
- [BEND03] S. Bendhia, E. Sicard, E. Lamoureux, B. Vrignon, "Un système de mesure sur puce pour la caractérisation de l'émission parasite d'un circuit intégré ", TELECOM 2003, 3rd JFMMA, Morocco, 15-17 October, 2003.
- [BENI97] L. Benini, P. Vuillod, A. Bogliolo, G. De Michelli, "Clock skew Optimization for Peak Current Reduction", Journal of VLSI Signal Processing, 16, pp.117-130, 1997.
- [BERK99] K. V. Berkel, M. B. Josephs, S. M. Nowick, "Scanning the technology: Applications of asynchronous circuits", proceedings of the IEEE, 87(2), pp.223-233, February 1999.

- [BLUN02] I.Blunno, F.Gregoretti, C.Passerone, D.Peretto, L.M.Reyneri, "Designing Low Electromagnetic Emissions Circuits through clock skew optimization", 9th IEEE International Conference on Electronics Circuits and Systems, Croatia, September 2002.
- [BLUN04] I. Blunno, J. Cortadella, A. Kondratyev, K. Lwin and C. Sotiriou. "Handshake Protocols for De-synchronization", 10th International Symposium on Asynchronous Circuits and Systems (ASYNC04), Greece, April 2004.
- [BOUE02] G.F. Bouesse, L. Fesquet, M. Renaudin, "QDI Circuits to Improve Smartcard Security", Second ACID Workshop (ACID02), Munich, Germany, 28-29 January, 2002.
- [BOUE04] G.F. Bouesse, G. Sicard, A. Baixas, M. Renaudin, "Quasi Delay Insensitive Asynchronous Circuits for low EMI", EMC Compo04, Angers, France, March 31st- April 1st, 2004.
- [BREJ03] C. Brej, "Obtaining Asynchronous benefits from synchronous design flow", Third ACiD Workshop (ACID03), Heraklion, Greece, 27-28 January, 2003.
- [BURN91] S.M. Burns, A.J. Martin, "Performance Analysis and Optimization of Asynchronous Circuits", Advanced Research in VLSI Conference, Santa Cruz, CA, March 1991.
- [CANA00] F.G. Canavero, "Modélisation des circuits intégrés digitaux pour la simulation CEM", EMC compo2000, Toulouse, June 2000.
- [CARR97] I. M. Carrig, A. M. Chu, F. D. Ferraiolo, J. G. Petroviek, P. A. Scott, and J. Weiss, "A Clock Methodology for High Performance Microprocessors," Journal of VLSI Signal Processing, 16, pp.217-224; 1997.
- [CATA01] J.P. Catani, "La compatibilité électromagnétique : du système au composant", Revue de l'Electricité et de l'Electronique, Janvier 2001.
- [CHAR99] E. Charbon, "Substrate Noise : Analysis, model and optimization", VLSI 99, Libon, Portugal, 1-4 December, 1999.
- [CHAR92] A. Charoy, "Parasites et perturbations des électroniques", Dunod, 1992.

- [CHAU92] F. Chauvet, "Introduction to electromagnetic compatibility", Rapport LAAS No92342 Université d'Eté Technologique "Composants et Systèmes", pp.1-10, Arcachon (France), 15-17 Septembre, 1992.
- [CHO93] J. D. Cho, M.Sarrafzadeh, "A Buffer Distribution Algorithm for High Performance Clock Routing", 30th IEEE/ACM Design Automation Conference, pp.537-543, 2003.
- [CHOV00] A. Chovet, "Interface Characterization Techniques in Deep Submicron MOSFET's-Noise Spectroscopy", Notes de cours, Ecole Doctorale d'Electronique Electrotechnique Automatique et Traitement du Signal de Grenoble, France, 2000-2001.
- [CHIL00] J. Chilo, "Bruit dans les interconnexions submicronique" Notes de cours, Ecole Doctorale d'Electronique Electrotechnique Automatique et Traitement du Signal de Grenoble, 2000-2001.
- [CORT03] J. Cortadella, A. Kondratyev, L. Lavagno, "Slicing Synchronous/Asynchronous trade-off curves", Third ACiD Workshop, Heraklion, Greece, 27-28 January, 2003.
- [DAVI97] A. Davis, S.Nowick, "An Introduction to Asynchronous Circuit Design" Technical Report UUCS-97-013, Departement of Computer Science, University of Utah, September 1997.
- [DECO99] F. De Coulon, "Théorie et Traitement des signaux", Presses Polytechniques et Universitaires Romandes, 1999.
- [DIAZ01] M. Diaz, "Les Réseau de Petri – Modèles Fondamentaux", Edition Hermès, 2001.
- [DINH02] A-V Dindhuc, "Synthèse automatique de circuits asynchrones QDI", Thèse de l'Institut National Polytechnique de Grenoble, France, Mars 2003.
- [DOUG02] Wai-Ching Douglas Lam, and Cheng-Kok Koh, and Chung-Wen Albert Tsao, "Power Supply Noise Suppression via Clock skew Scheduling," International Symposium on Quality Electronic Design, pp. 355-360, March 2002.
- [ESSA02] M. Essalhiene, L. Fesquet, M. Renaudin, "Dynamic voltage scheduling for real time asynchronous systems ", Twelfth International Workshop on Power And

Timing Modeling, Optimization and Simulation (PATMOS'02), Sevilla, Spain, 11-13 September, 2002.

- [FRAG03] J. L. Fragoso , G. Sicard, M. Renaudin, "Generalized 1-of-M QDI Asynchronous Adders", Third Asynchronous Circuits Design Workshop (ACID'03), Heraklion, Greece, February 2003.
- [FURB93] S.B. Furber, "Breaking Step - The Return of Asynchronous Logic" IEEE Colloquium on Design and Test of Asynchronous Systems, 1996.
- [FURB99] S. B. Furber, J. D. Garside, P. Riocreux, S. Temple, "Amulet2e: An asynchronous Embedded Controller", Proceedings of the IEEE, 87(2), pp.223-233, February 1999.
- [GAGE98] H. v. Gageldonk, D. Baumann, K. van Berkel, D. Gloor, A. Peeters, and G. Stegmann, "An asynchronous low-power 80c51 microcontroller", IEEE International Symposium on Advanced Research in Asynchronous Circuits and Systems, pp. 96-107, 1998.
- [GARE00] O. Garelli, "compatibilité électromagnétique", Notes de cours, Institut National Polytechnique de Grenoble, Ecole de Radioélectricité (ENSERG), 2000.
- [GOVI95] S. Govindarajan, "Scheduling Algorithms for High level Synthesis", Disponible sur : <http://www.eecs.uc.edu/~ddel/src/papers.html>.
- [GUF97] J. Gufeng Xi, W. Wei-ming Dai, "Useful-Skew Clock Routing with Gate Sizing for Low Power Design", Journal of VLSI Signal Processing, Kluwer Academic Publisher 16, pp.163-169, 1997.
- [GUPT01] S.Gupta, S. Katkooi, "Force-Directed Scheduling for Dynamic Power Optimization", IEEE Computer Society Annual symposium on VLSI, 25-26 April 2002, Pittsburg, Pennsylvania.
- [HARD94] K.B. Hardin, J.T. Fessler, D.R. Bush, "Spread Spectrum Clock Generation for the Reduction of Radiated Emissions", Proc. of IEEE International Symposium on Electromagnetic Compatibility, Chicago (IL) , 1994, pp. 227-231.
- [HAUC95] S. Hauck, "Asynchronous design methodologies: An overview", Proc. of the IEEE, 83(1), pp.69-93, January 1995.

- [HEAT02] M. Heath, "A Survey of Clocking Strategies from Synchronous to Asynchronous", available at: http://www-unix.ecs.umass.edu/~mheath/gals/clocking_survey.pdf.
- [HSIE97] H. Hsieh, W. Liu, P. Franzon, R. Cavin, "Clocking Optimization and Distribution in Digital Systems with Scheduled Skews", *Journal of VLSI Signal Processing*, 16, pp.131-147, 1997.
- [HUQ00] S. B. Huq, "Effective Signal Integrity Analysis using IBIS Models", *DesignCon 2000*, January 31 - February 3, 2000.
- [IBIS] IBIS Standard website, available at : http://www.mentor.com/pcb/ibis_modeling.html
- [IEC61] std IEC61967 - EMC of Ics : Conducted and Radiated Emission Methods.
- [IEC62] std IEC62014-3 -EMC for Ics : An Integrated Circuit Emission Model (ICEM).
- [IEEE10] IEEE Std 1076.6, IEEE Standard for VHDL Register Transfer Level (RTL) Synthesis, 1999.
- [KEAT98] M. Keating, P. Bricaud, "Reuse Methodology Manual for System-On-A-Chip Designs", Kluwer Academic Publisher, 1998.
- [KIM01] Ki-Wook Kim, S.-M. Kang "Crosstalk Noise Minimization in Domino Logic Design," *IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(9), pp.1091-1100, September 2001.
- [KNAP96] D. W. Knapp, "Behavioral Synthesis - Digital System Design Using the Synopsys Behavioral Compiler", Prentice Hall, 1996
- [KOIK98] S. Koike and K. Kaizu, "A Low-Cost Technique for Reducing the Simultaneous Switching Noise in Sub-Board Packaging Configurations", *IEEE Transactions on Components, Packaging, and Manufacturing Technology Part B: Advanced Packaging*, , 21(0), November 1998.
- [KOL 97] R. Kol and R. Ginosar, "A Doubly-Latched Asynchronous Pipeline", *IEEE International Conference on Computer Design (ICCD)*, October 1997.

- [LANK97] M. Lankford, "Electromagnetic compatibility design and performance of multichip module", International Journal of Microcircuit and Elec Packaging 1997
- [LARS96] P. Larsson, "di/dt Noise in CMOS Integrated Circuits", Journal of Analog Integrated Circuits and Signal Processing, 14, pp.113-129, 1997.
- [LI99] H.-S. Li, Y.-C. Cheng, D. Puar, "Dual-Loop Spread-Spectrum Clock Generator", IEEE International Solid State Circuits Conference, 1999.
- [LIEN97] J. Lienig, "Physical Design of VLSI circuits and the Application of Genetic Algorithms", Evolutionary Algorithms in Engineering Applications, Springer Verlag, Berlin, pp.277-292, 1997.
- [LINE95] A.M.Lines, "Pipelined Asynchronous Circuits", CS-TR-95-21, 1995
- [MAIO99] I.A. Maio, "Signal integrity and behavioural models of digital devices", Symposium on EMC, 1999
- [MARD03] M. Mardiguian, "Manuel pratique de compatibilité électromagnétique", Hermès Science, 2003.
- [MART96] R.S Martin, J.P.Knight, "Optimizing power in ASIC behavioral synthesis", IEEE Design and Test of Computer, 13(2) , pp. 59-70, 1996.
- [MEIN98] T. Meincke, A. Hemani, P. Ellervee, J. Öberg, S.Kumar, D. Lindqvist, H. Tenhunen, A. Postula, "Evaluating benefits of Globally Asynchronous Locally Synchronous VLSI Architecture.", Proceedings of the 16th IEEE NORCHIP Conference, Lund, Sweden, pp. 50-57, November 1998.
- [MONN04] Y. Monnet, M. Renaudin, R. Leveugle, "Asynchronous circuits sensitivity to fault injection", 10th IEEE International On-Line Testing Symposium, Madeira Island, Portugal, July 12-14, 2004.
- [MOON99] Y. Moon, et.al., "Clock Dithering for Electromagnetic Compliance using Spread Spectrum Phase Modulation," ISSCC Digest of Tech. Papers, pp.186-187, 1999.

- [MORI98] R.K. Morizawa, Y. Ueno, H. Nakamuray, T. Nanyay, "A Scheduling Method for Asynchronous VLSI System Design", available at: <http://citeseer.nj.nec.com/141931.html>
- [MUSS96] E. Musoll, J.Cortadella."On Reducing Electromagnetic Interference in High-Speed Digital Systems", Technical Report UPC-DAC-96-42, Departament Arquitectura de Computadors (UPC), 1996.
- [NOWI02] S.M. Nowick, "MINIMALIST: A CAD Environment for the Synthesis and Optimization of Burst Mode Asynchronous Controllers", Second Asynchronous Circuit Design Workshop, Munich, Germany, 28-29 January, 2002.
- [OBER01] J.Y. Oberle, "Simulation of electromagnetic radiations produced by integrated circuits", IFIP International Conference on Very Large Scale Integration, The Global System on Chip Design & CAD Conference, Montpellier, France, December 3-5, 2001
- [PAUL86] P.G. Paulin, J. P. Knight, E.F. Girczyc, "HAL: A multi-paradigm approach to automatic data path synthesis", 23rd Design Automation Conference, Las Vegas, Nevada, USA, 1986.
- [PAUL89] P.G. Paulin, J.P. Knight. "Force-Directed Scheduling for the Behavioral Synthesis of ASIC's", IEEE trans. on CAD, 8 (6), pp. 661-679, june 1989.
- [PAVE98] N. C. Paver, P. Day, C. Farnsworth, D. L. Jackson, W. A. Lien, and J. Liu, "A low-power, low-noise configurable self-timed DSP", International Symposium on Advanced Research in Asynchronous Circuits and Systems, pp. 32-42, 1998.
- [PERR96] T. Perry, L.Geppert, "Do Portable Electronics Endanger Flight? The Evidence Mounts," IEEE Spectrum, September, 1996.
- [PIGU01] C. Piguet, "Design of Dynamic Asynchronous Flip-Flops and Counters based on Dynamic STG", Workshop on Asynchronous Circuit Design (ACID01), Neuchâtel, 2001.
- [RAHM97] M. Rahmouni, "Ordonnancement et Optimisations pour la synthèse de Haut Niveau des Circuits de Contrôle", Thèse de l'Institut National Polytechnique de Grenoble, France, Février 1997.

- [RAIC02] D. Raic, "Reduction of Switching Noise and power Supply Currents in Digital circuits with Directed Data Flow", *Informacije MIDEM*, Ljubljana, 32(4), 2002.
- [RENA00] M.Renaudin, "Asynchronous circuits and systems: a promising design alternative", *Microelectronic Engineering Journal*, 54 (1-2), pp.133-149, December 2000.
- [RENA02] M. Renaudin, J.-B. Rigaud, A. Dindhuc, A. Rezzag, A. Sirianni, J. L. Fragoso, "*TAST CAD Tools* ", Tutorial at the ACiD Summer School, Grenoble, France, July 2002.
- [RENA04] M. Renaudin, J. Fragoso, "Asynchronous Circuits Design: an Architectural Approach", *V Escola de Microeletrônica da SBC-Sul*. Edited by José Guntzel, Ricardo, Reis.Universidade Federal de Pelotas, 2004.
- [RICH02] O. Richard, "Les boîtiers pour les circuits intégrés, Etat de l'art", Rapport interne du Laboratoire LEMO, Grenoble, 2000.
- [RIGA02] J.B. Rigaud, "Spécification de Bibliothèques pour la Synthèse de circuits Asynchrones", Thèse de Institut National Polytechnique de Grenoble, Décembre 2002.
- [ROSS02] B. Ross, "Overview of standards in electromagnetic compatibility for integrated circuits", *EMC compo02*, Toulouse, 2002.
- [SADA02] H. Sadamura, T. Daimon, T. Shindo, H. Kobayashi, T. Myono, T. Suzuki, S. Kawai, T. Iijima, "Spread-Spectrum Clocking in Switching Regulators to Reduce EMI," 2002 IEEE Asia-Pacific Conference on ASICs, Taipei, Taiwan, pp.141-144, August 2002.
- [SHA00] I-T. Sha, A. Chen, K.Y. Chen, J. Keip, "Spread spectrum and PLL technology, combine to reduce EMI", *RF Design*, April 2000.
- [SHA01] I-T. Sha, K.-Y. Chen, A. Chen and J.Keip, "An adaptive spread-spectrum clock generator", *RF Design*, August 1st, 2001.
- [SHIU00] W.-T. Shiue, "High Level Synthesis for Peak Power Minimization Using ILP", *IEEE International Conference on Application-Specific Systems Architectures*

and Processors (ASAP'00), Arizona State University Boston, Massachusetts, p. 103, July 10-12, 2000.

- [SLIM03] K. Slimani, Y. Remond, A. Sirianni, G. Sicard, L. Fesquet, M. Renaudin, "Estimation et optimisation de la consommation d'énergie des circuits asynchrones", 4èmes journées d'études Faible Tension, Faible Consommation (FTFC'03), Paris, France, pp. 59-64, 15-16 May, 2003.
- [SICA99] E.Sicard "Le couplage dans les circuits intégrés CMOS Sub-microniques", Habilitation à Diriger des Recherches, Université Paul Sabatier, spécialité Electronique, 1999.
- [SICA02] E. Sicard, S. Baffreau, S. Bendhia, S. Calvet, C. Huet, C. Marot, "A Standard Model for Predicting the Parasitic Emission of Micro-controllers", Paper for EMC Sorreto, Italy, Sept 9-11, 2002.
- [SICA03] E. Sicard, "The Key Role of Integrated circuits in EMC", TELECOM 2003, 3rd JFMMA, MARRAKECH, Morocco, 15-17 Oct 2003.
- [SOTI03] C. P. Sotiriou "De-synchronisation : Asynchronous Circuits from Synchronous Specifications", Third ACiD Workshop, Heraklion, Greece, 27-28 January, 2003.
- [SPAR01] J. Sparso, S.Furber, "Principles of Asynchronous Circuit Design – A system perspective", Kluwer Academic, 2001.
- [SPUR01] B. Spurgeon, "As Chips Reach Speed Limit, Makers Tap Into 'Clockless' Logic", International Herald Tribune, December 17th, 2001.
- [SUGA00] Z. Sugar, "Synthèse comportementale basée sur l'ordonnancement", Thèse de l'Institut National Polytechnique de Grenoble, France, Mai 2000.
- [SUTH89] I. Sutherland, "*Micropipelines*", Communication of the ACM, Volume 32, No 6, June 1989.
- [STEI98] T.Steinecke, M. Anfang,"Reducing Electromagnetic Emissions On-chip" available at : http://www.spacetools.com/site/pdf/V1_3p36.pdf, 1998.
- [STON96] D.A. Stone, B. Chambers, N.J. Ryan "Radiated EMI reduction from microcontroller system using spread-spectrum modulation of the system

- clock", Proc. of IEEE International Symposium on Electromagnetic Compatibility, Rome, Italy, 1, pp. 249-252, September 17-20, 1996.
- [TERR97] D. L. Terrel, R. Kenneth Keenan, "Digital Design for Interference Specifications- A practical Handbook for EMI Suppression", TKC, 1997.
- [TIMA] TIMA laboratory, Technics of Informatics and Microelectronics for computer Architecture, official website available at : <http://tima.imag.fr/>.
- [TORK02] K. Torki, F. Ciontu, "IC Thermal Map From Digital and Thermal Simulations", 8th THERMINIC Workshop, Madrid, 1-4 October 2002.
- [TRIS01] C. Tristram, "Des puces trop en avance sur leur temps"; Courrier International, 25 octobre 2001.
- [VASI99] G. Vasilescu, "Bruit et signaux parasites", Dunod, 1999.
- [VIVE01] P. Vivet, "Une méthodologie de conception de circuits intégrés quasi-insensibles aux délais : application à l'étude et la réalisation d'un processeur RISC 16bits asynchrone", Thèse de l'Institut National Polytechnique de Grenoble, France, Juin 2001.
- [WALK95] R. A. Walker, S. Chaudhuri. "Introduction to the Scheduling Problem", IEEE design and test of computers, pp. 60-68, winter 1995.
- [WILL99] T. Williams, "Compatibilité Electromagnétique, de la conception à l'homologation", Publitrone/Elektor, 1999.
- [WILS03] R. Wilson, "Les concepteurs de puces s'éloignent de la fine pointe technologique", EE times, August 26 2003.
- [YANG02] L. Yang, J. S. Yuan, M. Hagedorn, "Analyzing the simultaneous switching noise due to internal gate switching", 45th IEEE International Midwest Symposium on Circuits and Systems, Tulsa, Oklahoma, August 4-7, 2002.
- [YUN99] Young-Hwan Yun, Ho-jung You, Young-jun Kim, Seog-Heon Ham, and Yong-Hee Lee , "A Filter for Low EMI and Low Noise", AP-ASIC'99, August 1999.

Bibliographie de l'auteur

- [PANY01] D. Panyasak, "Conception de Circuits Asynchrones à faible Emission Electromagnétique", Journée des doctorants de l'école EEATS (JDEEATS), Grenoble, France, June 15, 2001.
- [PANY02a] D. Panyasak, G. Sicard, M. Renaudin, "Shaping Current Profile of Asynchronous Circuits", Communication to Asynchronous Circuit Design workshop 2002 (ACID'02), Munich, Germany, January 28-29, 2002.
- [PANY02b] D. Panyasak, G. Sicard, M. Renaudin, "A Current Shaping Methodology for Low EMI Asynchronous Circuits", 3rd International Workshop on Electromagnetic Compatibility of Integrated Circuits (EMC Compo'02), Toulouse, France, November 14-15, 2002.
- [PANY03a] D. Panyasak, G. Sicard, M. Renaudin, "From RTL code description to low EMI Asynchronous Circuits", Communication to Asynchronous Circuit Design workshop 2003 (ACID'03), Heraklion, Greece, January 27-28, 2003.
- [PANY03b] D. Panyasak, G. Sicard, M. Renaudin, "Conception de Circuits Asynchrones à Faible Emission Electromagnétique", Journées Nationales du Réseau Doctoral en Microélectronique 2003 (JNRDM'03), Toulouse, France, 14-16 Mai, 2003.
- [PANY04a] D. Panyasak, G. Sicard, M. Renaudin, "CEM des Circuits Intégrés : vers l'alternative Asynchrone", 12^{ème} colloque international de Compatibilité Electromagnétique (CEM04), Toulouse, France, 16-18 Mars, 2004.
- [PANY04b] D. Panyasak, G. Sicard, M. Renaudin, "Asynchronous Design for improved EMC behavior of IC's", 4rd International Workshop on Electromagnetic Compatibility of Integrated Circuits, EMCCompo04, Angers, France, 31 March–2 April, 2004.
- [PANY04c] D. Panyasak, G. Sicard, M. Renaudin, "A Current Shaping Methodology for Lowering EM Disturbances in Asynchronous Circuits", Microelectronics Journal, Issue Spéciale sur EMCCompo02, volume 35 Issue 6, 2004.

GLOSSAIRE

Circuit numérique asynchrone : circuit dont la séquence des traitements est assurée par des communication de type poignée de main (*handshake*) entre les opérateurs.

Circuit synchrone : circuit dont la séquence des traitements est régie par un signal global appelé couramment horloge. A chaque front de ce signal, des données sont libérées par des registres à l'intérieur du circuit.

CEM : compatibilité électromagnétique. Aptitude d'un système électrique à fonctionner dans un environnement donné sans perturber son environnement ni être lui-même perturbé. La compatibilité électromagnétique s'adresse aujourd'hui aussi au circuit intégré.

CHP : *Communicating Hardware Process*. Langage de description haut niveau dérivé du CSP et enrichi pour la synthèse et la simulation.

CSP : *Communicating Sequential Process*. Langage de description haut niveau permettant de décrire les communications dans les canaux.

(C)DFG : *(Control) Data Flow Graph*. Graphe permettant de décrire le flot de donnée d'un système. Les nœuds représentent les opérations dans le système. Les arcs représentent les dépendances entre les opérations.

di/dt : Variation du courant qui est la principale cause des émissions électromagnétiques du circuit lors de son passage dans les éléments inductifs et capacitifs. Plus les variations du courant sont brusques, plus le circuit génère des émissions électromagnétiques.

Discrétisation : action de découper en échantillons régulier un signal soit en temps soit en amplitude. Cette opération est très utile en particulier pour pouvoir appliquer un algorithme de répartition pour les ressources d'un circuit asynchrone dans le temps.

FDS : *Force Directed Scheduling*. Algorithme de répartition introduit par Pierre Paulin [PAUL86]. Cet algorithme associe des forces aux concurrences entre les ressources du circuit et tend à réduire ces forces. Les traitements concurrents sont donc réduits.

HSE : *Handshaking Expansion*. Syntaxe permettant de décrire les communications dans les canaux.

ICEM : Standard de modélisation d'émission du circuit intégré. Il contient en particulier le paramètre *Ib* qui représente l'activité du cœur du circuit, principale origine des émissions électromagnétiques.

IEM : Intérférences électromagnétiques. Dues aux émissions électromagnétiques des systèmes, elles peuvent survenir suite à des phénomènes de couplages ou une interprétation des ondes électromagnétiques parasites comme signal utile par des circuits récepteurs.

Micropipeline : Catégorie de circuit asynchrones où la partie de contrôle est distincte de la partie chemin de donnée. Des délais sont utilisés dans les canaux de communication pour assurer la validité des données dans le circuit.

Opérateur : Appelé aussi ressource dans ce mémoire, l'opérateur est un élément du circuit qui effectue un traitement.

Ordonnancement : Action de répartir et d'ordonner dans le temps les traitements de différents opérateurs. L'ordonnancement s'effectue sous différentes contraintes (temps, nombre de ressources...). Cette action est généralement effectuée à la synthèse comportementale du circuit.

PCFB : *PreCharge Logic Full Buffer*. Protocole de communication utilisé dans les circuits asynchrone. La remise à 0 de ses signaux de sortie est complètement découplée de celle de ses signaux d'entrée.

PCHB : *PreCharge Logic Half Buffer*. Protocole de communication utilisé dans les circuits asynchrone. La remise à 0 de ses signaux de sortie est partiellement découplée de celle de ses signaux d'entrée.

RTL : *Register Transfer Level* : Description au niveau registre d'un circuit.

SSC : *Spread Spreactrum Clock* : Etalement du spectre d'horloge introduit par Hardin en 1994 [HARD94]. Utilisé pour la modulation des signaux lors des transmissions. Il est peut être aussi utilisé pour réduire l'émission électromagnétique d'un circuit.

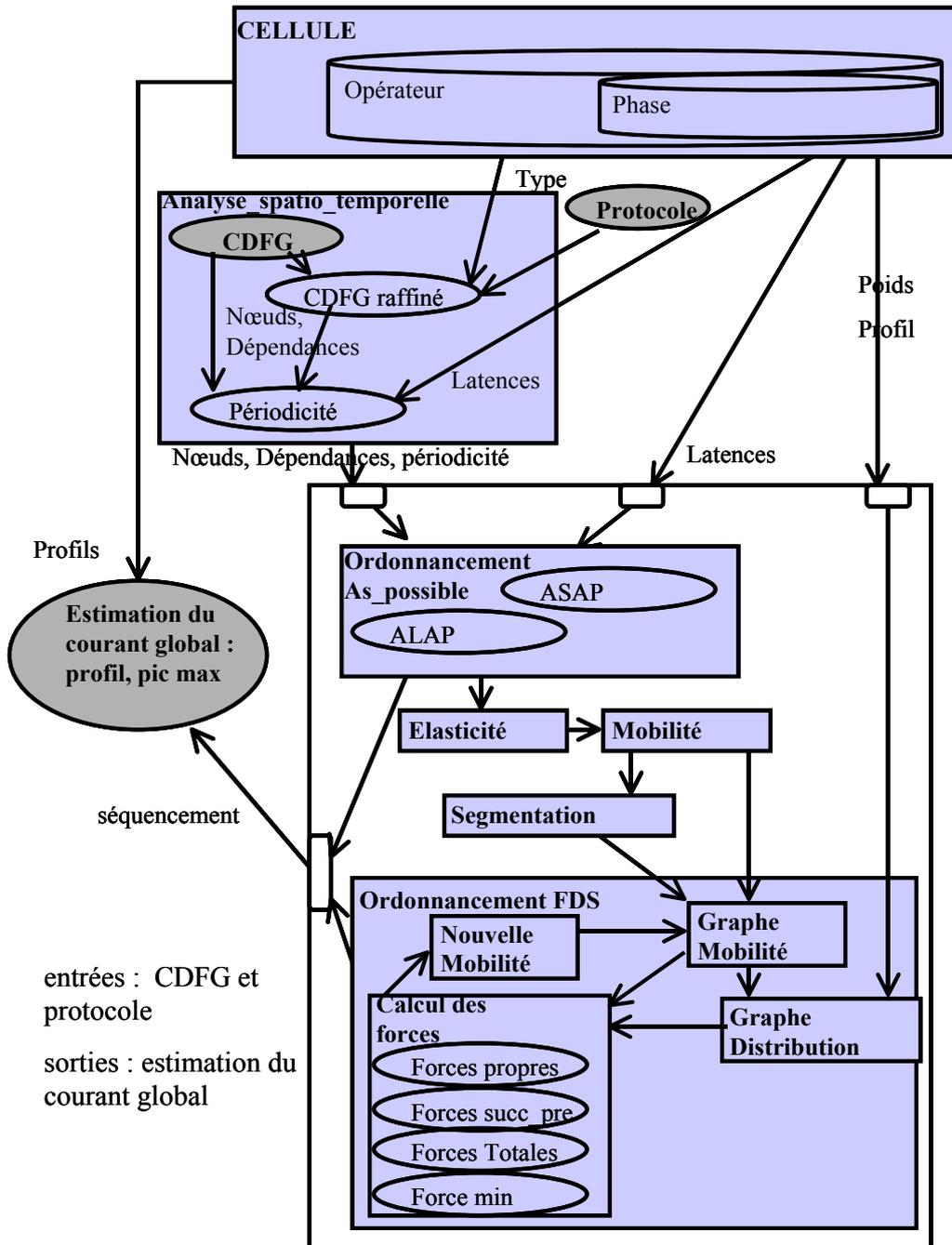
STG : *Signal Transition Graph*. Graphe permettant de décrire les évènements des signaux dans les canaux de communication.

VHDL : *Very High Speed Integrated Circuits Hardware Description Langage*. Langage de description haut niveau des circuits très utilisé et standard. Il a aussi comme équivalent le langage Verilog.

WCHB : *Weak Condition Half Buffer*. Protocole de communication utilisé dans les circuits asynchrone appelé aussi protocole standard.

Annexe

Spécification pour la programmation de l'algorithme de répartition : **Schéma d'ensemble**



Description des éléments pour la programmation

La cellule

On appelle ici cellule, un élément qui a une activité en courant lorsqu'il est actif. Dans cette étude, une cellule peut être un opérateur (multiplication, addition, identité...) ou une phase d'activité en courant d'un opérateur. Dans le CDFG, chaque nœud représente une cellule.

Cellule

Entrées :

Type_cellule

Sorties :

latence *tps*

profil tableau [*valeur;tps*]

poids *valeur*

La cellule se reconnaît grâce à son type. Type_cellule actuellement utilisés : mult, add, id, ph1_mult, ph2_mult, ph3_mult, ph1_add, ph2_add, ph3_add, ph1_id, ph2_id, ph3_id.

Elle se caractérise par sa latence, son profil en courant et le poids de son profil.

L'analyse spatio-temporelle

Elle fait la synthèse des informations (nœuds, dépendances, périodicité) contenues dans la représentation spatio-temporelle.

Analyse du graphe d'entrée sur une période de fonctionnement

Dans notre étude, cette analyse peut s'effectuer soit sur un CDFG ou chaque nœud représente un opérateur, soit sur un CDFG ou chaque nœud représente une phase d'activité en courant d'un opérateur (appelé CDFG raffiné). En effet, le modèle d'activité en courant d'un opérateur peut être raffiné en utilisant les modèles des phases en courant de l'opérateur. Dans le modèle raffiné, les dépendances entre les nœuds dépendent du protocole de communication.

Analyse_du_Graphe	
Analyse_CDFG Entrées : aucune Sorties : nœuds [<i>intitulés, dépendances_précédentes, dépendances_suivantes, type_cellule</i>] dépendance [<i>intitulés, nœuds_précédents, nœuds_suivants, type_dépendance</i>]	Analyse_CDFG_raffiné Entrées : Analyse_CDFG protocole Fonction utilisée : Remplacer chaque nœud représentant un opérateur par la composition des phases en activité de courant de l'opérateur. Sorties : nœuds [<i>intitulés, dépendances_précédentes, dépendances_suivantes, type_cellule</i>] dépendances [<i>intitulés, nœuds_précédents, nœuds_suivants, type_dépendance</i>]

Types de dépendances connues et utilisées: α , β , γ , δ , ε . D'autres types de dépendances sont susceptibles d'apparaître suite à l'étude sur les autres protocoles. Dans le CDFG non raffiné, les dépendances sont uniquement de type α .

Périodicité du fonctionnement du système

Elle caractérise le fonctionnement cyclique du circuit. La latence de la périodicité est le temps de cycle et la composition est la représentation spatio-temporelle du chemin critique (nœuds et dépendances).

Périodicité Entrées : Nœuds Dépendances Fonction : déterminer le chemin le plus long (temps pour repartir d'un nœud et y revenir) Sorties : Latence_périodicité <i>tps</i> Chemin_critique [<i>intitulés_nœuds;intitulés_dépendances</i>]

L'ordonnement As_possible

Permet de déterminer la séquence soit des opérateurs dans un CDFG, soit des phases dans un CDFG raffiné. Ceci de 2 façons : en activant les cellules le plus tôt possible (ASAP) ou le plus tard possible (ALAP).

L'ordonnancement As_possible	
<p>ASAP Entrées : nœuds dépendances périodicité Fonction utilisée : positionner le nœud suivant immédiatement après Sorties : séquencement_ASAP [nœuds, tps]</p>	<p>ALAP Entrées : nœuds dépendances périodicité Fonction utilisée : Positionner le nœud précédent immédiatement avant Sorties : séquencement_ALAP [nœuds, tps]</p>

L'élasticité

C'est la durée maximum que peut avoir une dépendance. Cette durée respecte la contrainte en temps imposée.

<p>Elasticité Entrées : ASAP ALAP Fonction utilisée : séquence ALAP – séquence ASAP Sorties : Elasticité [intitulé_dépendance, tps]</p>
--

La mobilité

Elle caractérise l'intervalle de temps sur lequel on peut déclencher l'activité d'une cellule. La représentation ASAP indique quand on peut effectuer ce déclenchement au plus tôt et la représentation ALAP quand on peut l'effectuer au plus tard tout en respectant la période de fonctionnement du circuit.

<p>Mobilité Entrées : Elasticité Calcul la somme des élasticités. Sorties : mobilité [intitulé_noeud, tps]</p>
--

La segmentation (slicing)

La segmentation définit les pas (step) qui vont servir à effectuer l'ordonnancement FDS. On discrétise sur la périodicité de fonctionnement du système.

Segmentation

Entrées :

mobilité

latence_périodicité

Fonction : moyenne mobilité nœud, nbre de nœuds, latence

Sorties :

durée_pas *tps*

pas [*intitulés pas*]

Ordonnement FDS

Graph de mobilité (TimeFrame)

Appelé Time Frame, ce graphe représente la probabilité que chaque cellule soit présente sur chaque pas.

Graph de mobilité

Entrées :

mobilité

pas

Sorties :

Graphe_mobilité [*intitulé_nœud, intitulé_pas, valeur_probabilité*]

Graph de Distribution (Distribution Graph)

Ce graphe représente la somme des probabilités d'avoir un pic pour chaque step. Dans ce graphe, on prend en compte les poids des cellules lorsque le pic est présent dans un pas.

Graphe de Distribution

Entrées :

Graphe_mobilité

Poids

Fonction : Dans le Graphe_de_mobilité, multiplier la probabilité par le poids de la cellule lorsque le pic peut se trouver dans le pas. Faire la somme des probabilités dans chaque pas.

Sorties :

Graphe_Distribution [*intitulés pas, valeur_somme_proba*]

Calcul des forces

Lorsqu'un nœud est affecté à un pas, cela induit une force propre.

Cette force propre est la somme du produit à chaque pas de la variation de la probabilité par la distribution de la probabilité dans le pas. L'affectation du nœud à un pas, change la probabilité des forces des nœuds aux autres pas. Il faut donc calculer les forces des successeurs et prédécesseurs.

Calcul_forces

Entrées :

graphe_distribution

graphe_mobilité

Fonction : Pour chaque possibilité {nœud,pas}, on calcul les forces induites :

Self_force [*intitulé_nœud, intitulé_pas, valeur_self_force*]

Succ_pred_force [*intitulé_nœud, intitulé_pas, valeur_succ_pred_force*]

Sorties :

Total_force [*intitulé_nœud,intitulé_pas*]

Dans le tableau des forces totales, considère le couple {nœud,pas} associé à la force la plus faible. On affecte le nœud au pas.

Force_minimale

Entrées :

Total_force

Sorties :

force_minimale [*intitulé_nœud, intitulé_pas*]

Nouvelle mobilité

On prend la force totale la plus faible et on affecte le nœud au pas considéré. Le graphe de mobilité et le graphe de distribution sont changés par la nouvelle mobilité.

Nouvelle_Mobilité

Entrées :

Mobilité

force_minimale

Fonction : affecte 0 de mobilité au nœud affecté.

Sorties :

New_Mobilité [*intitulé_pas,tps*]

Boucle principale

Tant que les éléments tps de la nouvelle mobilité ne sont pas égaux à 0 (c-a-d tant qu'on n'a pas affecté tout le nœuds): on met à jour le graphe de mobilité et le graphe de distribution. On calcule les forces totales pour affecter le nœud au pas indiqué par la force minimale et ainsi de suite.

Boucle_principale

Entrées :

pas

mobilité

poids

profil

Sorties :

Séquencement_FDS [*intitulé_pas,tps*]**L'estimation du courant global**

On fournit l'allure du courant global, la position et la valeur du pic de courant maximum.

Estimation du courant global

Entrées :

Séquencement

Sorties :

profil_global[*tps,valeur*]pic_max (*tps,valeur*)

RESUME

Ce travail de thèse s'attache à la réduction du phénomène d'interférence électromagnétique dans les circuits intégrés numériques. Les méthodes développées s'adressent aux descriptions haut-niveau des circuits afin de réduire les modifications rétroactives dans le flot de conception et de circonscrire ainsi le temps et les coûts dédiés à l'élaboration du circuit. Une première méthode appelée "Asynchronisation" propose de s'affranchir de la principale source d'émission dans les circuits numériques : l'horloge. Ce signal de contrôle est remplacé par des communications locales d'une manière permettant de conserver la compatibilité cycle à cycle du circuit. Les communications locales sont ensuite exploitées par la seconde méthode "mise en forme du courant" ("Current Shaping") afin de réduire davantage son émission électromagnétique. La concurrence des traitements dans le circuit est minimisée par ordonnancement. Les pics de courant résultant de la simultanéité des actions dans le circuit sont ainsi minimisés. De plus, une étude a été menée sur les protocoles 4 phases en vue de déterminer comment réduire davantage leur émission électromagnétique.

Mots Clés : Réduction des Emissions Electromagnétiques, Répartition de l'activité du courant, Circuits Numériques Asynchrones, Description au niveau registres (RTL), Protocole de Communication.

TITLE

ELECTROMAGNETIC EMISSION REDUCTION IN INTEGRATED CIRCUITS : THE ASYNCHRONOUS ALTERNATIVE.

ABSTRACT

This PhD thesis work aims to reduce electromagnetic interference phenomenon in digital integrated circuits. The developed methods address high-level descriptions of circuits in order to reduce the retroactive modifications in the design flow and to thus circumscribe the time and the costs dedicated to the the circuit development. A first method called "Asynchronisation" suggests to suppress the principal source of electromagnetic emission in the digital circuits: the clock. This signal of control is replaced by local communications whit a technique which preserves cycle compatibility with cycle of the circuit. The local communications are then exploited by the second method ("Current Shaping") in order to reduce further electromagnetic emission. The treatment concurrencies in the circuit are minimized by scheduling. The peaks of current resulting from the simultaneity of the actions in the circuit are thus minimized. Moreover, the 4 phases protocols were studied in order to determine how their electromagnetic emission can be more reduced.

Key-words: Reduction of the Electromagnetic Emissions, Distribution of the activity of the current, Asynchronous Digital Circuits, Register Transfer Level Description, Communication protocol.

INTITULE ET ADRESSE DU LABORATOIRE

Laboratoire TIMA (Techniques de l'Informatique et de la Microélectronique pour l'Architecture des ordinateurs), 46, avenue Félix Viallet, 38031 Grenoble Cedex, France.

ISBN : 2-84813-018-0
