



HAL
open science

Systèmes intégrés asynchrones et de traitement des signaux non uniformément échantillonnés

Laurent Fesquet

► **To cite this version:**

Laurent Fesquet. Systèmes intégrés asynchrones et de traitement des signaux non uniformément échantillonnés. Micro et nanotechnologies/Microélectronique. Institut National Polytechnique de Grenoble - INPG, 2008. tel-00280679

HAL Id: tel-00280679

<https://theses.hal.science/tel-00280679>

Submitted on 19 May 2008

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

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

Remerciements

Je ne sais pas très bien quelle forme doit prendre une page de remerciements pour une habilitation à diriger des recherches tant on est redevable à autrui. Néanmoins, je vais m'efforcer de remercier l'ensemble des acteurs, nombreux, qui ont permis la production de ce document.

Je remercie en premier lieu les rapporteurs de cette thèse, Michel Robert, Olivier Sentieys et Frédéric Pétrot qui ont accepté d'analyser mon manuscrit et une petite décennie de recherche sans horloge. Je remercie également Eric Martin et Marc Renaudin d'avoir accepté de participer à ce Jury.

Je suis arrivé au laboratoire TIMA en septembre 1999 et depuis le temps s'est arrêté à moins qu'il se soit écoulé trop vite. Je ne sais plus très bien car, en franchissant les portes du laboratoire, tout est bizarrement devenu asynchrone. Rien n'était plus pareil. Plus de tic tac, plus de signaux pour cadencer les circuits : bref une révolution ! Cette révolution, je la dois à Marc Renaudin qui m'a initié à cette logique à contre-courant. Rapidement, je me suis aussi approprié cette idée un peu folle et j'ai eu la chance de pouvoir la partager avec Marc bien sûr mais aussi avec mon fidèle compagnon (désynchronisé lui aussi, cela va de soit) Gilles Sicard. Je les remercie tous les deux très chaleureusement pour cette aventure un peu hors du temps !

Bien évidemment, cela n'aurait pas été possible sans nos anciens doctorants et désormais docteurs : Jean-Baptiste Rigaud, Anh Vu Dinh Duc, Emmanuel Allier, Mohammed Es Salhiene, Bruno Galilée, Laurent Alacoque, Jérôme Quartana, Kamel Slimani, Amine Rezzag, Antoine Sirianni, Dhanistha Panyasak, Damien Caucheteux, Fraidy Bouesse, João Fragoso, Fabien Aeschlimann, Bertrand Folco, Yannick Monnet, Vivian Brégier, Estelle Labonne. Fort heureusement la relève est assurée, les doctorants sont toujours aussi jeunes et motivés, contrairement à leurs encadrants, moins jeunes, mais tout aussi motivés ! Merci donc à Aurélien Buhrig, Julien Goulier, Cédric Koch-Hoffer, David Rios, Eslam Yahya, Jérémie Hamon, Khaled Alsayeg, Taha Beyrouthy, Livier Lizarraga, Saeed Mian Qaisar, Hatem Zakaria, Hakim Zimouche, Fadi Abouzeid et Oussama Elissati. Le groupe a aussi bénéficié de l'aide d'ingénieurs qui nous aident au quotidien et je tiens à les remercier ici pour leur travail qui est indispensable à nos recherches. Je remercie Alejandro Chagoya, Robin Roland, Bernard Bosc, Arnaud Baixas, Salim Renane, Yann Rémond, Nicolas Ninon, Grégory Lopin, Sophie Dumont et Franck Pagnat qui ont su soutenir les efforts du groupe durant de nombreuses années. Enfin, je remercie les nombreux stagiaires du groupe qui ont apporté leur pierre à l'édifice.

Même si mon activité a été intense dans le groupe, j'ai néanmoins lié des collaborations scientifiques au sein du laboratoire. Je pense notamment à Katell Morin-Allory, Dominique Borionne et Nacer Zergainoh. Je les remercie d'avoir partagé avec moi leur expertise et je m'excuse encore une fois d'avoir débranché le signal d'horloge !

J'ai également noué de nombreuses relations d'amitié avec des collègues qui m'ont soutenu et encouragé dans mes travaux, la gestion quotidienne d'un groupe de recherche et la rédaction d'un rapport appelé « Habilitation à Diriger des Recherches ». Merci à Alain Guyot, Skandar Basrou, Dominique Borionne, Katell Morin-Allory, Lorena Anghel, Paul Amblard, Pierre Gentil et Christian Schaeffer.

Enfin, comment ne pas remercier mes trois femmes ! Clara et Iris, 3 ans toutes les deux, Brigitte mon épouse qui, malgré les précautions prises pour limiter mes absences et réduire mes journées de travail, ont tout de même été un peu victime de mon intense activité. Je les remercie pour leur patience, leur compréhension et leur douceur.

Sommaire

Partie I : Activités de Recherche	9
I. Introduction	9
II. Conception de systèmes intégrés.....	11
II.1 Introduction	11
II.2 Les défis de la conception des systèmes intégrés.....	11
II.2.1 Les difficultés d'une conception complètement synchrone	11
II.2.2 Spécificités des technologies submicroniques profondes	12
II.2.3 Contrôle de la puissance dans les SoCs	13
II.2.4 Vérification, test et surveillance des systèmes intégrés	13
II.3 Les systèmes globalement asynchrones	14
II.3.1 Communications dans les systèmes sur puce.....	14
II.3.2 Synchronisation dans les GALS.....	15
II.3.3 Arbitrage et synchronisation	15
II.3.4 Réseaux de communication sur puce flexibles et extensibles.....	22
II.4 La gestion de l'énergie	26
II.4.1 Contrôle local de l'énergie	26
II.4.2 Gestion de l'énergie au niveau système	27
II.5 Les systèmes reconfigurables.....	31
II.5.1 Nécessité des systèmes configurables	31
II.5.2 Les circuits programmables et la logique asynchrone	32
II.5.3 Les circuits programmables dédiés à l'asynchrone.....	35
II.5.4 Les systèmes configurables sécurisés	38
II.6 Conclusion.....	39
III. Flot de conception des circuits asynchrones	41
III.1 Les différentes méthodes de conception	41
III.1.1 Méthodes de spécification.....	41
III.1.2 Spécifications basées sur des graphes	42
III.1.3 Spécifications basées sur un langage de description	42
III.1.4 Tour d'horizon des méthodes de synthèse	43
III.2 Modélisation des circuits asynchrones en CHP	47
III.2.1 Le langage CHP	47
III.2.2 Les extensions de CHP.....	48
III.2.3 DTL : une forme synthétisable de CHP	49
III.3 La synthèse des circuits QDI.....	49
III.3.1 Le format intermédiaire PN-DFG	49
III.3.2 Transformations et vérifications du réseau de Petri	50
III.3.3 Structure du circuit cible QDI	51
III.3.4 Méthode 1 : Synthèse QDI avec des équations de dépendances.....	51
III.3.5 Méthode 2 : Synthèse QDI avec des graphes de données multi-valués	55
III.4 La synthèse des circuits micropipeline.....	59
III.4.1 Modèle de circuits cibles.....	59
III.4.2 Flot de conception des circuits micropipeline.....	59
III.5 Génération de moniteurs à partir d'une expression logico-temporelle	60
III.5.1 Les langages de spécification de propriétés	61
III.5.2 Etat de l'art	62
III.5.3 Génération des moniteurs asynchrones	62
III.6 Conclusion.....	64
IV. Traitement des signaux non uniformément échantillonnés.....	65

IV.1	A propos de l'échantillonnage non uniforme !	65
IV.2	La conversion de signaux analogiques avec un schéma d'échantillonnage non uniforme en temps	66
IV.2.1	L'échantillonnage par traversée de niveaux	66
IV.2.2	Principe d'un convertisseur à échantillonnage par traversée de niveaux	67
IV.2.3	Rapport signal sur bruit d'un CANA	68
IV.2.4	Implémentation d'un convertisseur analogique numérique asynchrone	70
IV.3	Exploitation des signaux échantillonnés non uniformément	71
IV.3.1	Le filtrage RIF	71
IV.3.2	Le filtrage RII	76
IV.3.3	Algorithme de sélection d'activité (ASA)	81
IV.3.4	Résultats expérimentaux	82
IV.4	Conclusion	86
V.	Conclusion et perspectives scientifiques	87
V.1	Systèmes intégrés asynchrones	87
V.2	Outils de synthèse de circuits asynchrones	88
V.3	Systèmes à échantillonnage non uniforme	89
V.4	Perspectives scientifiques	89
V.4.1	Conception dans les technologies décanométriques	90
V.4.2	Synthèse de systèmes complexes et sûrs	91
V.4.3	Optimiser en sortant du cadre restrictif de la microélectronique	92
	Bibliographie	93
	Partie II : Curriculum Vitæ	101
I.	Cursus universitaire et professionnel	101
II.	Thèmes de recherche	101
II.1	Traitement des signaux non uniformément échantillonnés	101
II.2	Algorithmes et architectures des systèmes	102
II.3	Outils et Méthodes de synthèse pour les circuits asynchrones	102
III.	Responsabilités administratives et scientifiques	103
III.1	Responsabilités scientifiques	103
III.2	Organisation de manifestations	103
III.3	Participation à des projets	106
III.3.1	Projets européens	106
III.3.2	Projets nationaux	106
III.3.3	Projets collaboratifs	107
III.4	Responsabilités administratives	108
IV.	Publications	108
IV.1	Thèse	108
IV.2	Brevet	108
IV.3	Edition d'ouvrages	108
IV.4	Contributions à des ouvrages	108
IV.5	Journaux	109
IV.6	Papiers de conférences sélectionnés	109
IV.7	Conférences invitées	109
IV.8	Conférences internationales avec actes	110
IV.9	Conférences internationales sans actes	112
IV.10	Conférences nationales	112
IV.11	Séminaires	113
IV.12	Autres publications	113
V.	Encadrement de thèses	114

V.1	Thèses en cours	114
V.2	Thèses soutenues	115
VI.	Activités d'enseignement	119
VI.1	Conception microélectronique	119
VI.1.1	INPG - ENSERG.....	119
VI.1.2	Formation continue INPG	120
VI.1.3	Ecole Nationale Supérieure de Physique de Grenoble.....	120
VI.1.4	MASTER CSINA ET MNE	120
VI.2	Informatique	120
VI.2.1	INPG – ENSIMAG et Département télécoms.....	120
VI.2.2	INPG-ENSERG.....	120
Annexes	121
I.	Modélisation et conception d'arbitres pour NoC	
II.	Réseau de communication sur puce asynchrone	
III.	Synthèse logique de circuits QDI.....	
IV.	Convertisseur analogique numérique asynchrone.....	
V.	Filtrage RIF de signaux non uniformément échantillonnés	

Partie I : Activités de Recherche

I. Introduction

De nos jours, l'usage de systèmes intégrés occupe une place prépondérante dans les objets qui nous entourent : téléphones mobiles, PDAs, appareils photographiques, ordinateurs, voitures, ... La liste n'est bien évidemment pas exhaustive mais elle est là pour rappeler l'existence insidieuse de ces systèmes souvent très complexes. En effet, les quelques grammes de ces systèmes (System on Chip en anglais ou SoC) intègrent à la fois des blocs matériels (processeurs, coprocesseurs dédiés aux calculs ou aux communications, blocs analogiques) architecturés autour d'un réseau de communication intégré (Network on Chip en anglais ou NoC) et sur lesquels s'exécutent des applications supportées par un système d'exploitation. Ces quelques mots suffisent à prendre la mesure de la tâche des concepteurs de ces systèmes quand on sait la complexité qui en résulte et le nombre de degrés de liberté laissés à l'arbitrage des concepteurs.

La tâche nécessite beaucoup de rigueur et de méthode pour obtenir un résultat fonctionnel et opérationnel. Elle ne se limite plus aujourd'hui uniquement à adopter un flot de conception approprié mais aussi à user d'un flot de vérification assurant la meilleure détection des erreurs de conception. A l'heure actuelle, on estime que le temps de conception d'un système ne représente plus que 25 à 30% du temps de développement, le reste du temps étant consacré aux vérifications. L'évolution de ces temps montre clairement une tendance vers l'augmentation des temps de vérification des systèmes. Il apparaît donc nécessaire de définir de nouvelles méthodes dans les flots de conception et de vérification afin de réduire les temps de développement, de mieux gérer la complexité de ces systèmes et d'augmenter la productivité des ingénieurs concepteurs.

Les recherches menées depuis plusieurs années au sein du groupe CIS vise à réduire les difficultés de conception et de vérification de ces systèmes. Ces travaux s'appuient notamment sur une expertise forte dans le domaine de l'asynchrone. C'est donc tout naturellement que ces recherches ont été consacrées à l'étude et à la conception de systèmes intégrés asynchrones. Nous avons recherché des moyens pour réduire la consommation des circuits intégrés, abaisser leur rayonnement électromagnétique, augmenter leur robustesse et leur modularité. Ces travaux nous ont aussi amenés à concevoir des outils de CAO spécifiques pour l'asynchrone. Nous avons ainsi abordé l'étude des outils de synthèse tant au niveau algorithmique qu'au niveau de la sémantique des formalismes permettant de décrire efficacement nos circuits asynchrones. Enfin, nous avons recherché des moyens, plus éloignés de la microélectronique, pour aborder les délicats problèmes de maîtrise et de réduction de l'énergie dans les systèmes intégrés. Une première approche a été de contrôler l'énergie grâce à un superviseur qui est typiquement le système d'exploitation. La seconde, probablement moins connue, vise à réduire l'activité du système en utilisant des techniques d'échantillonnage des signaux analogiques alternatives aux classiques convertisseurs analogiques numériques dont l'échantillonnage s'effectue à fréquence fixe. En remettant en cause les chaînes classiques de traitement du signal, nous avons été obligés de reconsidérer la théorie du traitement du signal (avec l'aide de collaborateurs mathématiciens). Ainsi le présent document se décompose naturellement en trois parties :

- la conception de systèmes intégrés asynchrones,
- la synthèse des circuits asynchrones,
- le traitement numérique des signaux non uniformément échantillonnés.

La première partie traite l'étude de dispositifs asynchrones appliqués aux systèmes « Globalement Asynchrones Localement Synchrones » (GALS), à la gestion de l'énergie et aux systèmes (re)configurables. La partie GALS aborde les délicats problèmes des communications intrapuces, de synchronisation et d'arbitrage et démontre la pertinence d'une conception asynchrone pour ces dispositifs. Le paragraphe consacré à la gestion de l'énergie, quant à lui, montre les gains substantiels possibles en exploitant des circuits asynchrones tant au niveau matériel qu'au niveau logiciel (système d'exploitation). Enfin, la partie consacrée aux systèmes reconfigurables montre qu'il est non seulement possible d'utiliser les composants programmables du commerce pour prototyper nos circuits asynchrones mais aussi de concevoir des composants dédiés offrant de meilleures performances (vitesse, sécurité, ...).

La seconde partie du manuscrit traite de la synthèse des circuits asynchrones. Le premier point abordé concerne les langages et formalismes qui offrent une sémantique appropriée à la description et à la synthèse des circuits quasi-insensibles aux délais (QDI) et micropipelines. Les techniques de synthèse et les optimisations sont également décrites. Enfin, avec un formalisme approprié, il est possible de prouver mathématiquement que la synthèse produit des circuits quasi-insensibles aux délais. Cette propriété est très importante car elle confère aux circuits des qualités de robustesse remarquables. Enfin, l'étude de moniteurs QDI prouvés corrects et générés à partir d'une expression logico-temporelle sont étudiés.

La troisième et dernière partie aborde le traitement des signaux numériques non uniformément échantillonnés. Cette approche, initialement développée sur le principe d'un échantillonnage par traversée de niveaux, a montré qu'elle permettait, pour bon nombre de signaux, de réduire considérablement l'activité d'un système de traitement du signal. Par ailleurs, le concept, qui échantillonne le signal uniquement quand ce dernier varie, s'accommode très bien de la nature événementielle des circuits asynchrones. Ce travail ouvre également de nombreuses perspectives de recherches dont certaines revêtent un caractère très fondamental comme la remise à plat de la théorie du traitement du signal. Il faut en effet reconsidérer la théorie à chaque fois que l'on reconsidère la technique d'échantillonnage. Les travaux présentés ici décrivent la conception d'un convertisseur analogique numérique asynchrone (et à échantillonnage non uniforme) et de filtres RIF et RII exploitant des signaux non uniformément échantillonnés. Des éléments de la théorie du signal associée sont également donnés.

II. Conception de systèmes intégrés

II.1 Introduction

L'augmentation de la surface des puces, d'une part, et de leur complexité, d'autre part, amène les concepteurs à imaginer des systèmes qui ne sont plus complètement synchrones : on parle alors de systèmes GALS (Globally Asynchronous Locally Synchronous). En effet, il est de plus en plus difficile de construire des systèmes totalement synchrones car les distances intra puce et les fréquences d'horloge nous éloignent de l'approximation des régimes quasi-permanents et de l'isochronisme. De plus, la métastabilité liée à l'échantillonnage de signaux complètement asynchrones vis-à-vis de l'horloge rendent les systèmes peu fiables, induisant un temps moyen avant panne de plus en plus court. Ces difficultés s'ajoutent à la nécessité de réduire la consommation des puces, ces dernières étant souvent embarquées dans des systèmes autonomes. La tâche du concepteur n'est plus aujourd'hui seulement contrainte par la simple construction d'un dispositif fonctionnel à un coût raisonnable (c'est-à-dire la surface de la puce) mais aussi par des contraintes de consommation, voire de rayonnement électromagnétique. Enfin, avec la réduction des dimensions et l'émergence des technologies décanométriques, il est nécessaire de trouver des voies pour garantir un rendement acceptable en sortie des unités de fabrication. Les technologues s'efforcent d'augmenter le rendement en ajustant les paramètres de leur procédé de fabrication, mais aussi en recherchant de nouvelles architectures de composants. Les concepteurs apportent également des solutions sur ce point en développant de nouvelles techniques de layout (layout régulier) et en implémentant des systèmes parallèles redondants, auto-testables et configurables en fonction des besoins des utilisateurs. Nous sommes probablement à l'aube d'une utilisation massive de systèmes multiprocesseurs et de FPGA embarqués architecturés autour de réseaux de communication.

L'ensemble de ces points constitue autant d'axes de recherche à explorer. Le présent chapitre propose d'en explorer quelques uns. Durant ces dernières années, les travaux menés au sein du groupe nous ont conduit à étudier les mécanismes de synchronisation, d'arbitrage et de construction d'un réseau de communication sur puce asynchrone. Ces travaux ont fait l'objet de la thèse de Jérôme Quartana [QUA 04]. Les aspects consommation électrique ont également été explorés. Nous nous sommes beaucoup intéressés dans le groupe au bénéfice d'une conception asynchrone pour réduire la consommation et le rayonnement électromagnétique des puces. Au niveau système, en collaboration avec Mohammed Es Salhiène, nous avons étudié les gains potentiels que nous pourrions obtenir avec un système d'exploitation tournant sur un microprocesseur asynchrone. Enfin, un dernier axe d'étude a été de mettre au point une technique pour implémenter de la logique asynchrone sur des structures programmables classiques ou spécifiques. Dans le cadre des travaux de thèse de Taha Beyrouthy, nous développons une structure programmable dédiée à des circuits asynchrones sécurisés.

II.2 Les défis de la conception des systèmes intégrés

II.2.1 Les difficultés d'une conception complètement synchrone

La plupart des systèmes numériques modernes sont synchrones. Ils sont organisés autour d'un signal d'horloge global, et les activités du système sont séquencées par ce signal. Ce mécanisme d'activation global introduit de nombreuses contraintes de plus en plus difficiles à maîtriser avec l'avènement des technologies décanométriques et l'usage de fréquences d'horloge de l'ordre du GHz. Les problèmes les plus couramment évoqués sont :

- La distribution de l'horloge dans un circuit complexe est délicate à concevoir, coûteuse en surface et en consommation.
- Les instants d'occurrence de certains signaux critiques dans les NoCs (Network on Chip) et les Systèmes GALS (Globally Asynchronous Locally Synchronous) sont susceptibles de provoquer des états métastables dans les éléments de mémorisation.
- Un fonctionnement basé sur une analyse pire cas du chemin critique du système limite la fréquence d'horloge.
- La modularité des systèmes synchrones est faible car il est difficile d'ajouter ou de remplacer un composant sans que cela ait une incidence sur le fonctionnement global.
- La sensibilité de plus en plus forte aux procédés de fabrication due à l'usage de technologies décanométriques induit des variations temporelles et de consommation importantes à l'échelle du circuit lui-même (et pas seulement au niveau du Wafer).

C'est sans doute à cause de l'arrivée de ces obstacles majeurs que la logique asynchrone a gagné en popularité ces dernières années. Solution naturelle à bon nombre de ces limitations, la méthodologie asynchrone ouvre la voie à de nouvelles approches de conception. Au lieu de fonctionner avec une horloge globale, les systèmes asynchrones fonctionnent avec un contrôle localement distribué, ce qui permet d'éviter les problèmes liés à une horloge globale.

II.2.2 Spécificités des technologies submicroniques profondes

Les technologies submicroniques profondes ou décanométriques offrent la perspective de densités d'intégration très importantes (plusieurs millions de transistors par mm^2), mais dans le même temps soulèvent des défis majeurs dans le domaine de la conception des circuits. Il semble qu'une rupture profonde se dessine avec les technologies 45 nm et 32 nm [ITR 07]. Si jusqu'alors il était possible de concevoir des ASICs basés sur des bibliothèques de cellules standard pré-caractérisées, en utilisant les flots de conception usuels, il apparaît qu'à partir de 32 nm, la nature des composants changeant et des phénomènes nouveaux (variabilité, interaction entre composants ...) apparaissant, les méthodes classiques deviennent en partie inapplicables. A ce jour, les contraintes identifiées sont :

- Les règles de dessin (layout) deviennent complexes car les performances d'un composant dépendent des composants voisins. Ce point met en exergue la nécessité de se tourner vers des structures régulières bien maîtrisées et caractérisées.
- Les phénomènes de variabilité des procédés de fabrication au sein de la puce deviennent prépondérants. Ainsi au sein d'un même circuit, les caractéristiques des transistors peuvent varier dans des proportions considérables. Les approches « pire cas » et « meilleur cas » utilisées pour caractériser les circuits ne sont plus viables. Il est nécessaire de mettre en œuvre des techniques de conception robustes (statistic design, conception asynchrone, ...) s'accommodant de l'imprédictibilité des performances.
- Les rendements (nombre de puces fonctionnelles) seront proches de zéro si des techniques spécifiques ne sont pas appliquées. En effet, la conception des masques et les techniques de lithographie optique avec des motifs dont les dimensions sont proches de la longueur d'onde utilisée (deep UV)), cumulées aux incertitudes de fabrication des composants, font que de nombreux transistors sur la puce ne seront pas fonctionnels. Il apparaît de plus en plus indispensable de prévoir de la redondance architecturale pour garantir des rendements acceptables.

Avec les nœuds technologiques à venir, 32 nm et inférieurs, les difficultés rencontrées dans les précédentes générations, qui étaient considérées comme indépendantes, se verront corrélées. La variabilité des procédés de fabrication est susceptible d'induire des différences sur les fréquences de fonctionnement des puces pouvant aller jusqu'à un ordre de grandeur entre deux puces d'une même plaquette. On estime par ailleurs que l'on pourrait observer un facteur d'environ quatre entre deux blocs identiques d'une même puce. La conception d'un système intégré devient alors un problème complexe qui doit prendre en compte les procédés de fabrication, les caractéristiques des composants, des circuits et du système. Il est nécessaire de changer les paradigmes de conception et d'imaginer des architectures et des techniques nouvelles pour aborder cette évolution drastique des technologies.

II.2.3 Contrôle de la puissance dans les SoCs

Un autre point délicat est la puissance de calcul qui croît exponentiellement. Il apparaît de plus en plus nécessaire de trouver des solutions architecturales et matérielles répondant conjointement aux besoins de puissance de calcul et de maîtrise de la consommation. Par ailleurs, compte tenu de la dispersion des paramètres technologiques et de la difficulté de prédire la performance d'un nœud de traitement sur un SoC, il est opportun d'envisager des mécanismes locaux d'optimisation fonction des performances mesurées sur la puce. Les systèmes vont se doter de plus en plus de mécanismes pour isoler les éléments défectueux ou à faibles performances et les redéployer par des éléments redondants ou des dispositifs d'adaptation tension/fréquence différenciés par unité de traitement. En outre, l'usage de ces structures devrait favoriser l'émergence massive de techniques de régulation de l'énergie dans les systèmes intégrés. Ces mécanismes peuvent s'implémenter à plusieurs niveaux dans le système :

- Au niveau de la logique, il est possible de minimiser la consommation d'énergie des circuits en contrôlant les tensions d'alimentation et les polarisations (back-biasing par exemple).
- Au niveau d'un bloc de traitement, en sus d'une régulation sur les alimentations, le contrôle de sa fréquence de fonctionnement peut être mis en œuvre, soit pour fixer une fréquence appropriée à l'usage du bloc, soit pour réduire sa consommation quand le traitement ne requiert pas la puissance de calcul maximale.
- Enfin, une régulation au niveau système est également envisageable. Elle permet d'une part de répartir les tâches du système sur les nœuds de traitement en fonction des performances mesurées des éléments de calcul et d'autre part d'assurer un contrôle de l'énergie efficace en ordonnant les tâches en fonction de leur échéance et leur priorité sur des blocs dont la vitesse de traitement est ajustée aux besoins.

Il est à noter que le dernier point mentionné nécessite une intégration entre les mécanismes de contrôle matériel de l'énergie et le système d'exploitation. Par ailleurs, l'usage de signaux de contrôle, nécessaire à la mise en œuvre de ces dispositifs, favorise l'introduction des circuits asynchrones qui sont usuellement déjà pourvus de ce type de signalisation. Enfin, la tendance amorcée ces derniers temps pour maîtriser la consommation d'énergie profite bien évidemment aux systèmes portables sur batteries, mais aussi aux systèmes qui n'ont pas vocation au nomadisme car la gestion de la puissance dissipée devient aujourd'hui un enjeu majeur pour garantir leur bon fonctionnement.

II.2.4 Vérification, test et surveillance des systèmes intégrés

Un dernier point essentiel à la réalisation et à l'exploitation de systèmes complexes est la possibilité de les vérifier lors de leur conception, de les tester après leur fabrication et de

surveiller leur fonctionnement tout au long de leur vie. Pour la vérification, à ce jour, les techniques par simulation sont majoritairement employées. Elles incluent les simulations, mais aussi tout le panel d'analyse statique temporelle (Static Timing Analysis) et de puissance (Power Analysis). A cela s'ajoute la possibilité de prototyper avec des systèmes exploitant des circuits programmables (FPGA). Ces approches ne sont pas exhaustives dans leur analyse et la difficulté pour les utiliser à bon escient est de trouver un jeu de test approprié. Ce dernier est en général difficile à définir. C'est pourquoi il apparaît aujourd'hui des techniques faisant appel à des approches plus formelles telles que la simulation symbolique, la preuve de théorème (aidée en général par des logiciels « theorem prover ») ou l'analyse d'assertions logico-temporelles. Cette dernière catégorie peut être utilisée pour extraire les informations les plus pertinentes d'une simulation ou comme moniteur de surveillance intégré dans la puce. Ce point sera abordé en détail au paragraphe III.5.

II.3 Les systèmes globalement asynchrones

II.3.1 Communications dans les systèmes sur puce

La conception synchrone présente l'avantage de fournir un contrôle d'exécution largement éprouvé. Toutefois le rôle central joué par l'horloge nécessite d'importants efforts dans la conception, l'optimisation et la vérification du réseau de distribution d'horloge. Les perspectives dressées dans les "Roadmaps" de la *Semiconductor Industry Association* [SIA 07] consacrés à la conception de systèmes intégrés utilisant des technologies submicroniques avancées, mentionnent la nécessité d'interfacer des fonctions intégrées animées par des horloges différentes. La distribution d'une horloge unique, ou d'un ensemble d'horloges synchronisées entre elles, dans ce type de systèmes s'avère très difficile pour plusieurs raisons.

- D'un point de vue électrique, la distribution d'une horloge dans un système intégré pose des problèmes de temps de propagation, d'intégrité du signal, de bruit et de consommation.
- D'un point de vue fonctionnel, la synchronisation de l'ensemble des fonctions d'un système complexe par une horloge unique complexifie significativement le développement de la puce et constitue un frein important à la réduction du temps de développement.

Dans une perspective GALS, le système de communication a alors pour rôle de découpler les composants du système et de les rendre modulaires et extensibles. Cette difficulté de conception du système de distribution d'horloge croît très rapidement avec la taille des circuits. C'est pour cette raison que les techniques de conception de systèmes asynchrones suscitent l'intérêt depuis une dizaine d'années. L'engouement fut suffisant pour que certains prédisent même l'affranchissement total de la tyrannie de l'horloge, tandis que les défenseurs du tout synchrone poursuivent l'amélioration du système de distribution de l'horloge [XAN 01].

Le travail de thèse de Jérôme Quartana s'inscrit dans ce contexte [QUA 04]. Ainsi, une étude complète a été menée sur le concept de systèmes globalement asynchrones et localement synchrones (GALS). Les systèmes GALS induisent la conception d'un réseau de communication spécifique interconnectant les blocs de calcul et les mémoires entre eux. Ce réseau devient un élément essentiel dans le système. La synchronisation par communication entre domaines d'horloge indépendants doit être abordée avec une attention toute particulière pour garantir au système fiabilité et performances. La synchronisation aux interfaces des domaines et la conception du réseau de communication sont les deux principaux enjeux à relever par la conception de ces systèmes GALS.

II.3.2 Synchronisation dans les GALS

La contrainte de conception d'une architecture GALS vient de sa spécificité : la synchronisation par communication entre domaines d'horloge indépendants. Cette synchronisation entre blocs est uniquement assurée par des signaux de contrôle locaux. Ces signaux de contrôle sont asynchrones vis-à-vis des entrées des blocs qu'ils synchronisent. Les instants d'apparition de ces signaux ne sont ni prédictibles, ni corrélés avec les signaux d'horloge des blocs. En effet, ils sont échantillonnés par l'horloge locale des blocs synchrones. L'ordre relatif des transitions de ces signaux d'entrées et des transitions d'horloge n'est pas déterministe.

Cet échantillonnage d'un signal asynchrone au moyen du signal d'horloge d'un bloc synchrone est à l'origine de deux problèmes connexes, la métastabilité et le non déterminisme :

- Le phénomène de métastabilité caractérise un état dont la durée est indéterminée et non bornée [REN 04]. Un circuit peut entrer dans un état métastable lorsque un signal est échantillonné à une valeur intermédiaire qui ne permet pas de déterminer son niveau logique ;
- Le non déterminisme est la caractéristique, généralement indésirable, que possède un système à fournir des séquences de sortie différentes en réponse à une même séquence d'entrée. Ainsi, si un des blocs synchrone possède de multiples entrées asynchrones, les délais relatifs entre les différents signaux de contrôle de ces communications rendent la séquence d'entrée, et par conséquent la séquence de sortie, non déterminée.

Des circuits spécifiques, dits synchroniseurs, sont utilisés pour résoudre ces problèmes et permettre la synchronisation de domaines d'horloge différents. Toutefois ces circuits ne sont pas parfaitement fiables et sont susceptibles d'introduire une latence importante dans les communications. Ces circuits synchroniseurs sont nécessaires dans les architectures GALS chaque fois qu'un bloc synchrone échantillonne au moyen de son signal d'horloge un signal asynchrone. En revanche, un réseau de communication sans horloge s'affranchit de l'usage de synchroniseurs pour échantillonner les entrées si l'on utilise une logique insensible aux délais. Ce point est discuté en détail au paragraphe suivant.

II.3.3 Arbitrage et synchronisation

Le paragraphe précédent souligne les difficultés pour s'affranchir des conséquences de la métastabilité. Le problème est qu'un échantillonneur constitué d'une ou plusieurs bascules D n'offre aucune garantie quant à sa fiabilité. La probabilité pour que ce dispositif soit défaillant est d'autant plus grande que l'on travaille à des fréquences élevées [DIK 99][GIN 02]. En effet, il n'est pas simple de déterminer de façon sûre l'ordre d'arrivée de deux signaux (dont les transitions sont temporellement très proches) ou d'affirmer qu'à un instant donné un signal a franchi un seuil (dont la transition est proche de cet instant). Ces deux problèmes équivalents sont généralement surmontés par un synchroniseur dit à « double bascule » dont la fiabilité est acceptable [SEM 03]. Le principe est de disposer entre le signal de contrôle asynchrone et le bloc synchrone deux bascules en série synchronisées par l'horloge du bloc (cf. Figure 1). En contrepartie, ce dispositif introduit une latence de deux cycles d'horloge.

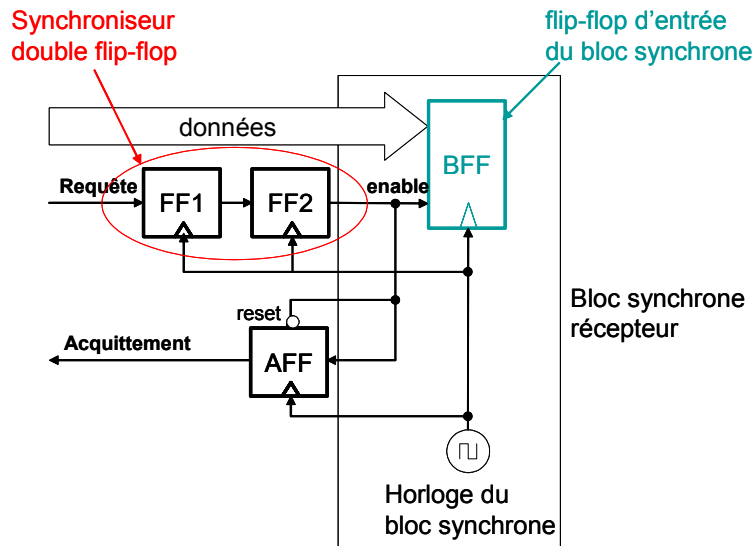


Figure 1 : Synchroniseur à double bascule

II.3.3.1 Une approche fiable pour la synchronisation

L'idée est d'aborder le problème de la synchronisation en considérant que la métastabilité est inévitable et qu'elle n'est pas seulement un phénomène peu probable. Comme le temps de résolution de la métastabilité est non borné, il n'est plus envisageable de synchroniser les signaux à une fréquence fixe. Par conséquent, aucun système synchrone ne peut garantir un échantillonnage fiable à 100%. Il faut donc songer à exploiter les synchroniseurs avec des circuits insensibles aux délais. Ainsi et seulement ainsi, il sera possible d'échantillonner un signal par un autre signal sans que cela n'ait de conséquence sur le circuit exploitant le résultat de cet échantillonnage. En effet, un circuit insensible aux délais reste fonctionnel quelque soit le temps que mettra le synchroniseur à lui fournir la donnée si celui-ci passe par un état métastable.

L'exclusion mutuelle

L'exclusion mutuelle est un circuit qui permet de faire le choix entre deux requêtes asynchrones concurrentes potentiellement vraies au même instant. La synthèse de cette fonction d'exclusion mutuelle est détaillée par Jean-Baptiste Rigaud dans sa thèse [RIG 02] et donne le circuit donné Figure 2.

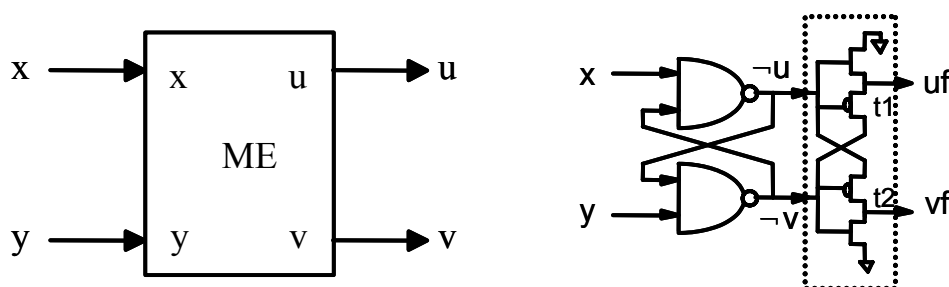


Figure 2 : Circuit d'exclusion mutuelle

La réalisation de cet élément en technologie CMOS utilise les tensions de seuil à son avantage :

- le canal du transistor t1 ne conduit que si $(u \wedge \neg v)$ est vrai,
- le canal du transistor t2 ne conduit que si $(v \wedge \neg u)$ est vrai.

Lorsque le bloc d'exclusion mutuelle est dans un état métastable, le deuxième étage se comporte comme une mémoire et conserve sur ses sorties (*uf* et *vf*) des valeurs stables jusqu'à ce que le premier étage commute de façon franche dans l'une des deux situations précédemment citées [MAR 93].

Le circuit synchroniseur

Le synchroniseur permet de déterminer la valeur logique d'un signal asynchrone vis-à-vis d'un signal d'échantillonnage. La réalisation la plus élémentaire du synchroniseur peut être considérée comme la juxtaposition d'un bloc de mutuelle exclusion avec un étage de logique booléenne. Si la sortie *u* passe à 1, cela signifie que *z* a échantillonné la valeur 1 sur *b* et si *v* passe à 1, que *z* a échantillonné la valeur 0 sur *b*.

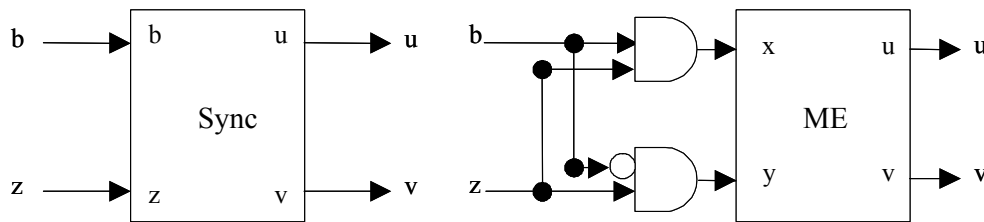


Figure 3 : Circuit synchroniseur

L'usage d'éléments d'exclusion mutuelle et de synchroniseurs associés à de la logique asynchrone insensible aux délais permet de construire des systèmes de communication fiables à 100%. Cela permet notamment de garantir un échantillonnage correct des signaux des blocs synchrones d'un système GALS par son réseau de routage (ce dernier étant asynchrone). En revanche, l'échantillonnage de l'information depuis un tel réseau de routage vers un bloc synchrone ne pourra être réalisé que par des synchroniseurs traditionnels (tels que les synchroniseurs à double bascule). Par ailleurs, les éléments d'exclusion mutuelle et les synchroniseurs présentés ci-dessus sont tout à fait adaptés à la construction d'arbitres asynchrones qui pourront être utilisés dans les réseaux de routage asynchrones des NoCs.

II.3.3.2 Les arbitres à base d'exclusions mutuelles et de synchroniseurs

Les systèmes intégrés complexes (GALS), architecturés autour d'un réseau de communication, nécessitent l'usage de dispositifs de synchronisation mais aussi d'arbitres afin de gérer les contentions lors de l'accès à une même ressource. Il est donc nécessaire de prévoir des arbitres performants et fiables afin de garantir un bon fonctionnement au réseau de communication. Nous avons étudié dans le cadre des thèses de Jérôme Quartana et de Jean-Baptiste Rigaud des arbitres asynchrones sûrs offrant de bonnes performances en vitesse [RIG 01][RIG 02][RIG 02b][RIG 02c]. En effet, des arbitres répondant avec des latences et des temps de cycle de quelques nanosecondes (entre 2 et 6 ns) avec une technologie CMOS 0.18 μm ont été étudiés. Par ailleurs, cette étude ne s'est pas simplement limitée à la conception de ces arbitres, mais aussi à leur modélisation fine par un langage. Ce point sera présenté plus loin dans le document (cf. III.2). Enfin le paragraphe présente l'utilisation de ces arbitres dans les nœuds de contrôle de routage de l'information d'un réseau sur silicium sans horloge.

Afin d'illustrer le propos, voici un arbitre simple réalisé à partir d'une exclusion mutuelle. Sa fonction est de garantir que seul le client C1 ou C2 accède à une ressource commune. L'arbitre présenté Figure 4 respecte un protocole 4 phases.

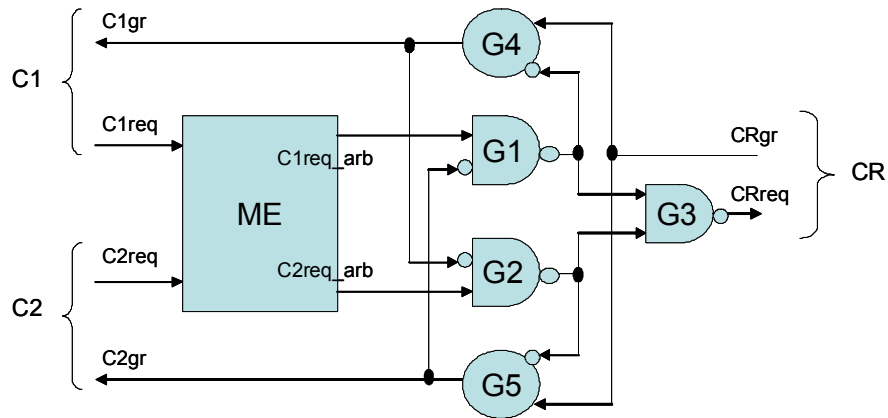


Figure 4 : Un arbitre simple utilisant une exclusion mutuelle

Il est possible de construire des arbitres d'une complexité quelconque en utilisant des éléments d'exclusions mutuelles ou des synchroniseurs. L'approche proposée (qui est associée à un modèle formel, cf. §III.2) permet de construire de façon systématique tout type d'arbitres. Pour des raisons de simplicité et de concision, un arbitre asynchrone à priorités fixes est pris en exemple. L'arbitre asynchrone à priorités fixes, présenté Figure 5, est composé de trois parties. Le bloc nommé « loop control » a la charge de réarmer l'arbitre dès qu'une requête a été servie et que la ressource partagée a été accédée. Le bloc de déclenchement (« trigger block » sur le schéma) a pour fonction de surveiller les entrées et d'activer l'arbitre à l'apparition d'une requête. La seconde partie est constituée de synchroniseurs en charge de l'échantillonnage des requêtes. Enfin, la troisième partie constitue un module de sélection de la requête la plus prioritaire. Cette dernière est également réalisée en logique quasi insensible aux délais.

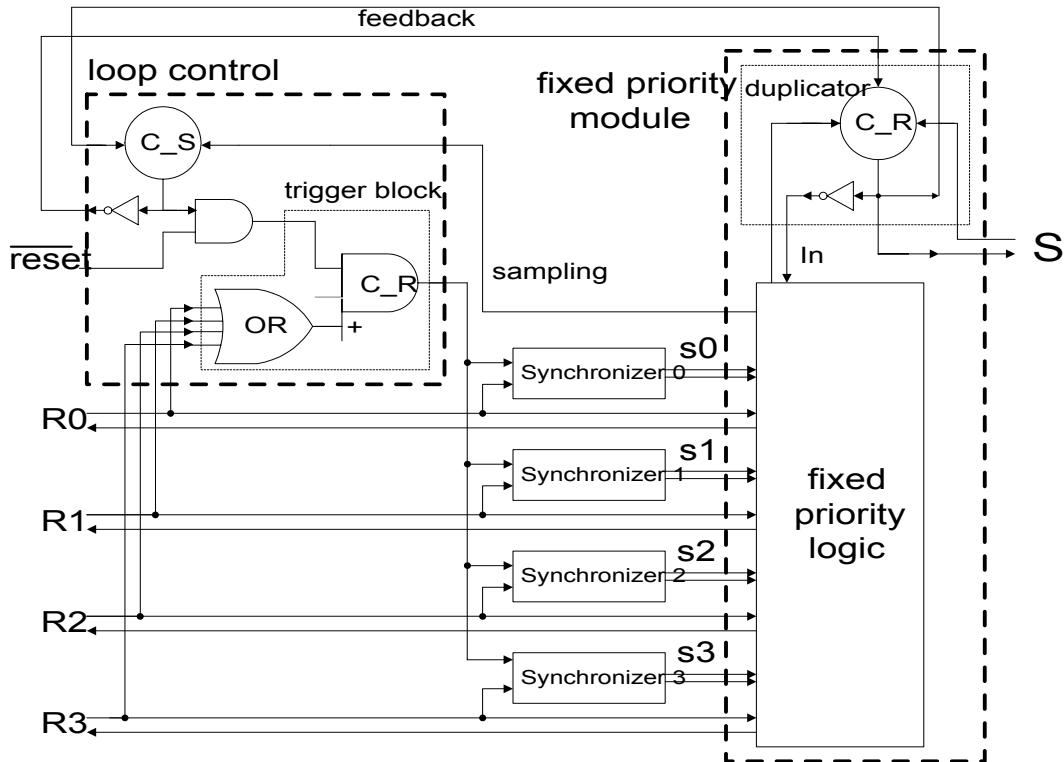


Figure 5 : Arbitre asynchrone à priorités fixes

Il est possible à partir de synchroniseurs, de logique asynchrone et d'une architecture adaptée de réaliser des circuits d'arbitrage fiables et modulaires de n'importe quelle

complexité et dont la politique d'arbitrage peut-être spécifiée avec un haut niveau d'abstraction (cf. III.2). Ce point est extrêmement important pour spécifier et concevoir de façon systématique des arbitres et plus généralement des réseaux de communication sur puce (NoCs).

II.3.3.3 Adaptation des vitesses de fonctionnement entre blocs

Une autre difficulté rencontrée lors de la conception des GALS est le fait que les blocs synchrones constituant le système fonctionnent avec des horloges différentes qui n'ont pas nécessairement une même fréquence. Le réseau de communication a ici aussi la charge de gérer et d'adapter les différentes vitesses d'exécution rencontrées dans le système.

FIFOs à double horloge

Usuellement l'adaptation entre deux domaines d'horloges est réalisée avec des FIFOs dites à double horloge [GIN 02]. Il existe de nombreuses implémentations de FIFOs à double horloge (*Dual-Clock FIFO*) qui adaptent deux domaines d'horloge pour des systèmes plésiochrones ou hétérochrones. L'émetteur écrit les données dans une mémoire RAM à double port à sa fréquence de fonctionnement et le récepteur lit les données à sa propre vitesse. Le principe est présenté Figure 6 :

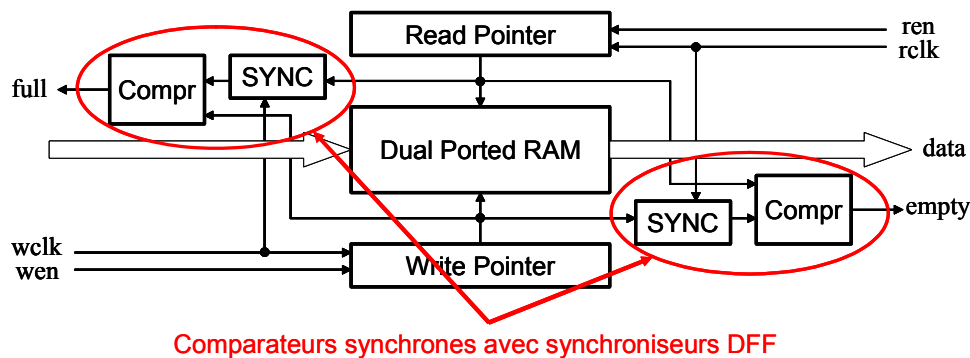


Figure 6 : Principe d'une FIFO à double horloge

Ces FIFOs à double port nécessitent une conception soignée afin de garantir une lecture et une écriture concurrente dans la mémoire. La synchronisation entre les domaines de fonctionnement n'est nécessaire que pour gérer les états plein et vide de la FIFO grâce au contrôleur composé d'un comparateur et d'un synchroniseur à double bascule (cf. Figure 6). La principale limitation de ce dispositif est qu'il nécessite de nombreuses synchronisations (non fiables) si les différences entre les fréquences d'horloge sont très importantes. En effet, la FIFO sera souvent pleine ou vide, selon que le récepteur est le plus lent ou le plus rapide [DAL 98]. Par ailleurs, il est aisé d'adapter ce mécanisme pour réaliser une interface de synchronisation mixte entre un bloc synchrone et un bloc asynchrone [CHE 01][CHE 04].

FIFOs asynchrones

Une autre approche consiste à utiliser des FIFOs asynchrones pour réaliser cette fonction de synchronisation de deux domaines d'horloge. La synchronisation n'est réalisée dans ce cas qu'aux extrémités de la FIFO sur le signal d'acquittement du côté de l'émetteur et sur le signal de requête du côté du récepteur. La FIFO, quant à elle, est constituée d'un pipeline asynchrone. La Figure 7 présente ce dispositif en explicitant les signaux de contrôle (de requête et d'acquittement). Cependant, il est tout à fait envisageable d'implémenter le pipeline en logique QDI (Quasi Delay Insensitive) et avec un codage de type *1 parmi n*. En outre, il apparaît ici de façon évidente que ce dispositif s'adapte aisément pour synchroniser un bloc synchrone et un bloc asynchrone.

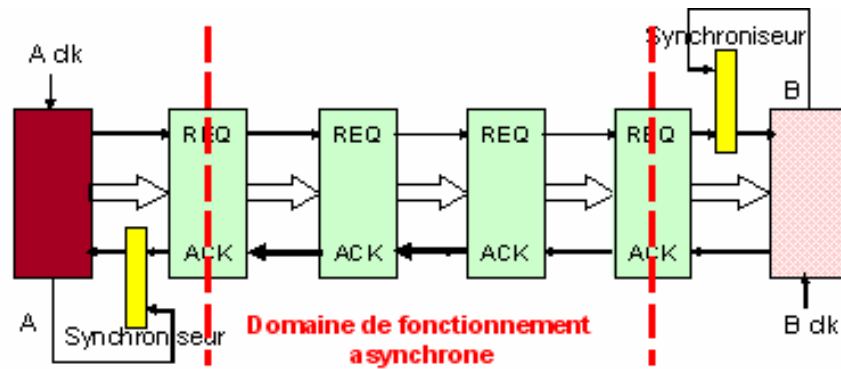


Figure 7 : Pipeline de synchronisation asynchrone

Plusieurs réalisations ont été proposées sur ce principe, mais ces solutions conservent cependant le défaut de déplacer les données, ce qui induit en général une latence et une consommation électrique élevées [BRU 95][SUT 01]. Cela nous a conduit à envisager une autre classe de FIFOs, dites circulaires, basées sur les travaux de Chelcea et Nowick [CHE 00]. Le principe en est donné Figure 8 .

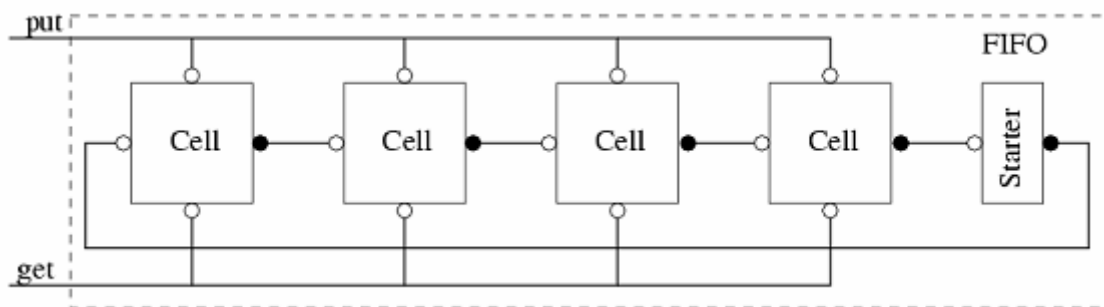


Figure 8 : Structure d'une FIFO asynchrone et circulaire

Ce sont deux jetons qui contrôlent le comportement de la FIFO : un jeton Put Token (PT) circule de cellule en cellule pour remplir la FIFO et le jeton dual Get Token (GT) circule pour autoriser la lecture dans les cellules de la FIFO. Le jeton PT précède toujours le jeton GT et les deux jetons ne peuvent être sur une même cellule. Cela correspond au fonctionnement usuel d'une FIFO : la cellule doit être remplie avant d'être vidée. Le Starter est une cellule spéciale qui permet d'initialiser la FIFO en injectant deux jetons PT et GT dans l'anneau.

L'intérêt de cette architecture est double :

- le contrôle distribué permet une excellente modularité et extensibilité de la FIFO,
- les données sont immobiles, ce qui offre une faible consommation et permet de courtes latences.

En revanche, la communication avec l'environnement se fait au moyen de bus qui se révèlent pénalisants en consommation lorsque le nombre de cellules augmente. La « bufferisation » des bus doit être redimensionnée pour chaque taille de FIFO. Une amélioration possible de ce dispositif est de remplacer les bus par des arbres de multiplexage. Pour réaliser cette FIFO asynchrone, nous utilisons deux composants appelés One-to-Two Sequential switch (OTS) et Two-to-One Sequential switch (TOS). Ces composants sont assemblés en arbres binaires sur le principe de la Figure 9 qui représente une FIFO asynchrone circulaire. Chaque composant OTS est un démultiplexeur de un vers deux sans contrôle. A l'initialisation, il aiguille la première communication à droite (sortie marquée par un jeton sur la Figure 9), puis bascule ensuite automatiquement d'une sortie sur l'autre à chaque communication. Le composant TOS fournit la fonctionnalité réciproque de

multiplexeur à deux canaux vers un, sans contrôle, en commençant de même toujours par la droite.

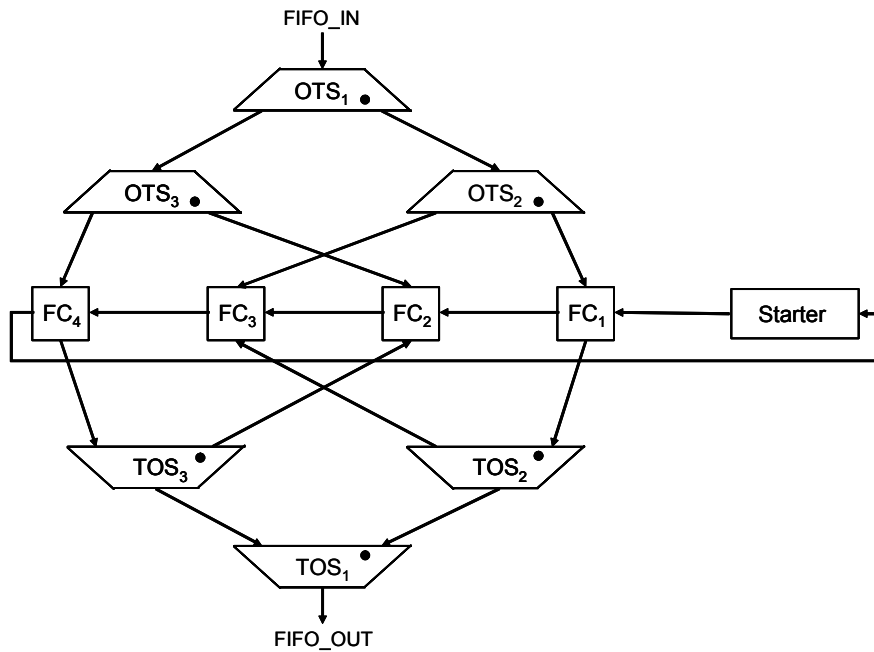


Figure 9 : FIFO circulaire asynchrone avec des arbres de multiplexage

Cette architecture toute simple présente les intérêts suivants :

- La séquence est garantie correcte et sans blocage. L'ordre d'arrivée des données sur les cellules suit exactement celui du jeton PT. Il en va de même pour l'ordre de multiplexage des données par le réseau de composants TOS qui suit exactement l'ordre de circulation du jeton GT.
- Le contrôle distribué et les composants parfaitement identiques assurent une excellente modularité et extensibilité de la FIFO.
- La consommation d'énergie est réduite par rapport à celle utilisant des bus (cf. Figure 10).

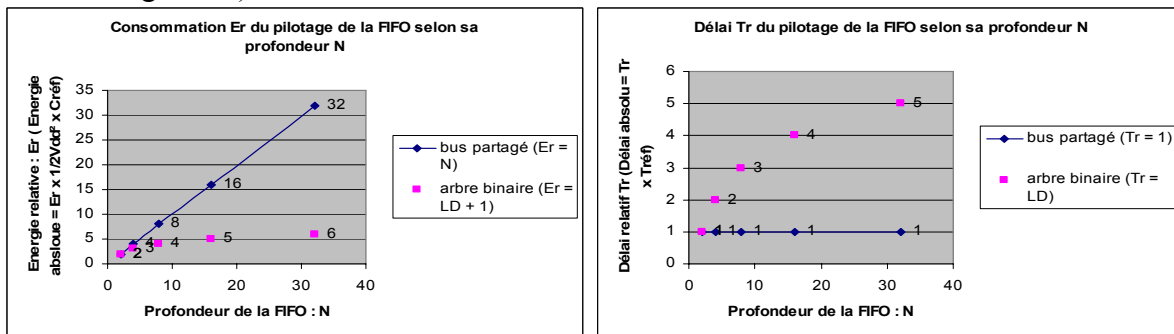


Figure 10 : Consommation et délai du pilotage de la FIFO en fonction de sa profondeur

En revanche, le délai augmente avec la profondeur de l'arbre. Le temps d'accès avec un bus est théoriquement indépendant du nombre de cellules de la FIFO, mais il ne faut pas oublier que la charge du bus augmente et qu'il est nécessaire de « bufferiser » correctement ce dernier. Il est à noter qu'il est possible d'envisager des sélecteurs $1 \text{ vers } n$ et de $n \text{ vers } 1$, si l'on souhaite implémenter des profondeurs de FIFO qui ne sont pas des multiples de deux. En outre, ces sélecteurs $1 \text{ vers } n$ sont combinables avec tous sélecteurs $1 \text{ vers } p$. Cela permet d'adapter à volonté les profondeurs des FIFOs et la profondeur de l'arbre de multiplexage. En

d'autres termes, cela offre la possibilité de limiter dans une certaine mesure les latences d'accès aux cellules mémoires.

Les travaux effectués dans ce domaine ont montré que l'usage d'une FIFO circulaire asynchrone offrait fiabilité de fonctionnement, modularité et extensibilité ainsi qu'une consommation réduite. En outre, le raccordement d'une telle structure à un bloc synchrone ne pose pas de problème particulier. Cette structure peut parfaitement convenir pour construire un réseau de communication asynchrone qui serait le cœur d'un système GALS par exemple. Une démonstration de ce type a été faite sur FPGA dans le cadre de la thèse de Jérôme Quartana [QUA 04].

II.3.4 Réseaux de communication sur puce flexibles et extensibles

Ce paragraphe décrit une méthode générique pour spécifier et construire un réseau de communication (NoC) flexible et extensible. L'idée est de proposer une approche qui est totalement indépendante de la topologie. Ce travail a été mené durant la thèse de Jérôme Quartana [QUA 04] et a permis d'identifier les blocs constituant tout réseau de communication. Les éléments de synchronisation et d'adaptation de vitesse d'un système GALS, décrits au paragraphe II.3.3, font partie de ces blocs spécifiques aux systèmes de communication. La connaissance de toutes les briques permet d'envisager une approche générique pour spécifier un réseau sur puce (NoC) en explicitant uniquement ses communications.

Cela revient à dire que l'on sait implémenter la synchronisation par communication entre différents domaines d'horloge tel que cela a été présenté au paragraphe II.3.2. Toutefois, une telle approche impose de définir formellement la nature d'une communication. Ainsi, il est souhaitable de séparer les transactions déterministes qui sont synthétisables avec des outils standard et les transactions non déterministes qui, elles, ne le sont pas. Ces dernières nécessitent le recours aux éléments synchroniseurs vu précédemment. Cette distinction est indispensable pour synthétiser de façon systématique les communications. Une modélisation formelle de ce point est présentée au paragraphe III.2. Elle montre notamment comment on peut expliciter ces nuances par un langage.

La distinction faite sur les communications ne suffit pas pour concevoir un NoC générique. La Figure 11 montre l'ensemble des blocs utiles à la conception du réseau d'un système GALS. Il faut être capable de spécifier correctement les interfaces de synchronisation (qui incluent potentiellement les FIFOs présentées au paragraphe II.3.3.3), la topologie du réseau et sa politique de routage et d'arbitrage. Il faut également définir les blocs qui constitueront les interfaces entre les périphériques et le réseau (interfaces de communication) en charge de l'adaptation des signaux du périphérique à ceux du réseau. Enfin, l'interface de performance est une interface de synchronisation (comme défini ci-dessus) si le réseau est synchrone et simplement une FIFO et un adaptateur de protocole si le réseau est asynchrone.

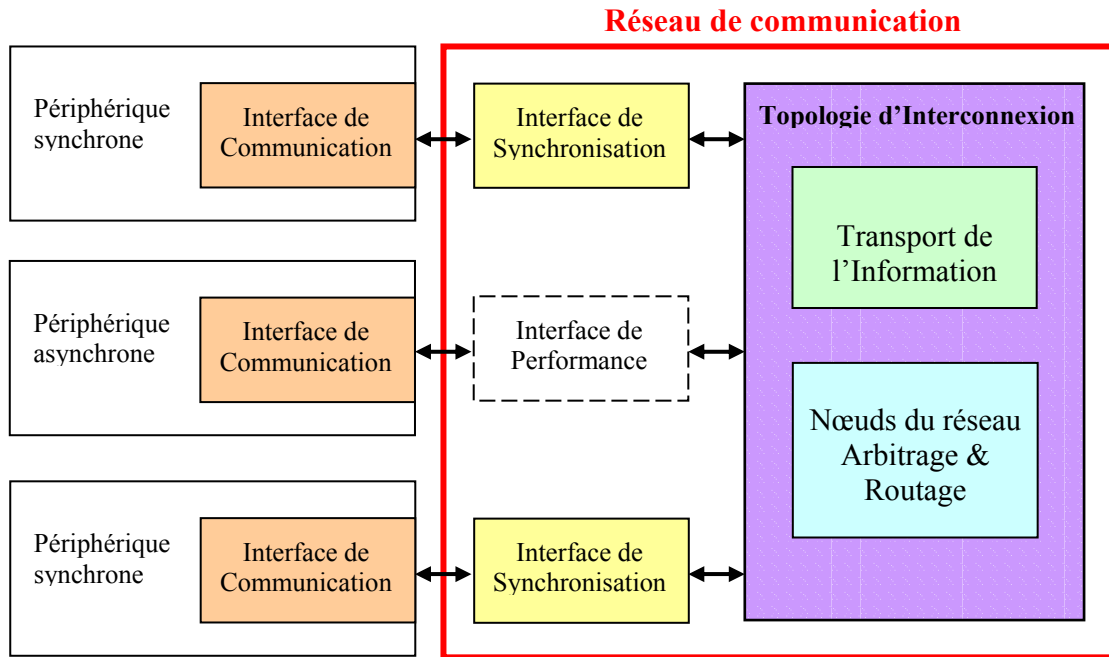


Figure 11 : Un réseau de communication générique

II.3.4.1 Etude d'un réseau synchrone

Afin de mieux cerner les avantages d'une conception asynchrone pour le réseau de communication, considérons en premier une communication entre deux domaines d'horloge avec une conception synchrone pour le NoC. Cela se traduit, comme on peut le voir sur la Figure 12, par l'usage de 3 horloges différentes et de 4 synchroniseurs à double bascule. En effet, le réseau d'interconnexion synchrone possède sa propre horloge (asynchrone de celle des périphériques). Chaque périphérique échange avec le réseau un signal asynchrone (la requête ou l'acquittement) vis-à-vis de son horloge. De même, le réseau reçoit des signaux asynchrones provenant des périphériques. Les communications doivent donc être synchronisées au moyen de circuits synchroniseurs afin d'échantillonner correctement les signaux de requête et d'acquittement. Les données, quant à elles, n'ont pas besoin d'être synchronisées car le protocole garantit en général que les données sont prêtes lorsque la requête est émise. La synchronisation des communications entre périphériques synchrones au travers d'un réseau synchrone nécessite quatre synchroniseurs, soit deux synchroniseurs par canal de communication.

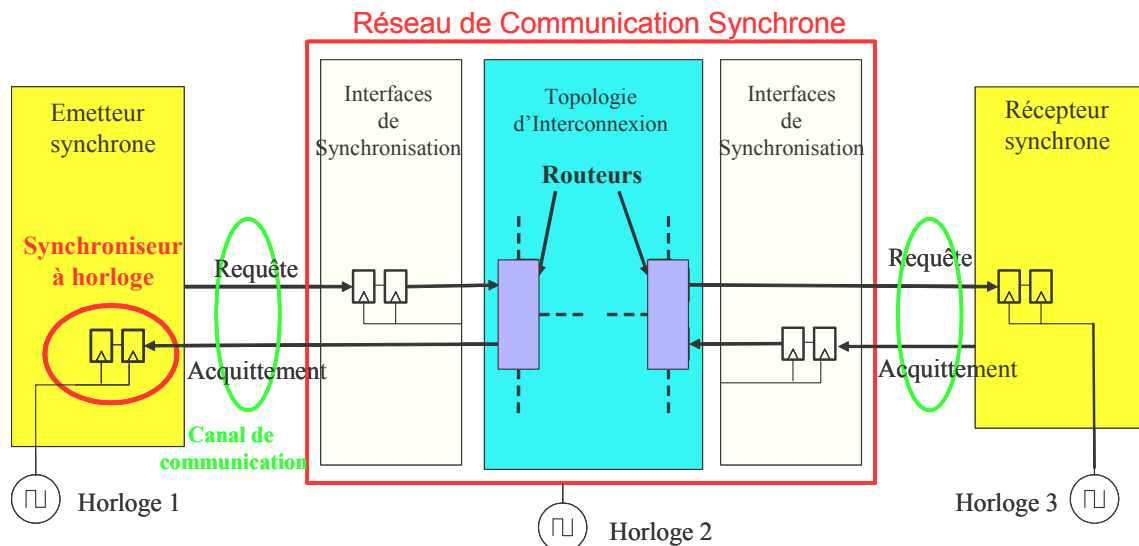


Figure 12 : Synchronisation avec un réseau synchrone

II.3.4.2 Étude d'un réseau asynchrone

Si l'on compare avec un système similaire dont le réseau serait réalisé par une technologie asynchrone, nous aurions la structure donnée sur la Figure 13. Les périphériques synchrones nécessitent toujours des synchroniseurs à double bascules pour échantillonner les signaux de requête et d'acquittement émis par le réseau de communication. En revanche, le réseau de communication asynchrone ne nécessite pas l'usage de synchroniseurs. La synchronisation de communication entre les périphériques synchrones au travers d'un réseau asynchrone nécessite donc deux synchroniseurs à horloge. Ainsi le nombre total de synchroniseurs à horloge aux interfaces de communication entre les composants et le réseau de communication est le double si ce dernier est synchrone.

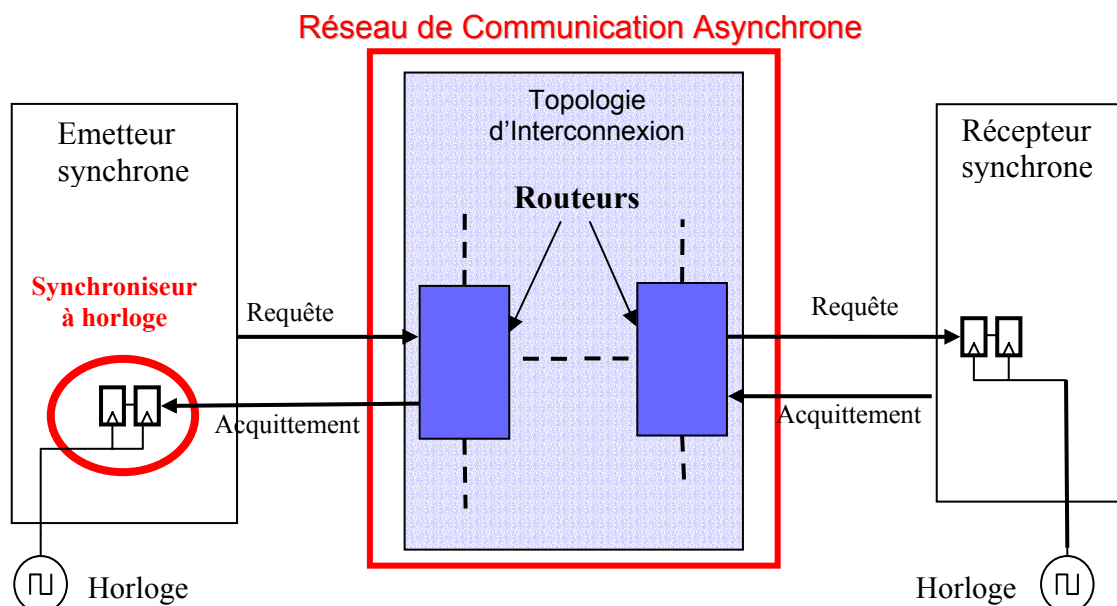


Figure 13 : Synchronisation avec un réseau asynchrone

Cette étude montre les avantages d'un réseau de communication asynchrone pour résoudre les problèmes de synchronisation d'une architecture GALS :

- Les risques de non déterminisme sont fortement réduits. La fiabilité est accrue car on s'affranchit du coût de la synchronisation dans le domaine d'horloge du réseau qui est remplacé par un réseau asynchrone entièrement fiable.
- La latence due aux synchroniseurs est réduite de moitié. Cela n'exclut pas l'usage des FIFOs pour gagner en performance.
- L'activité et la consommation due aux synchroniseurs sont divisées par deux.

II.3.4.3 Un exemple de réseau asynchrone : la plateforme PACMAN

Ce travail sur les réseaux asynchrones a donné lieu dans le cadre de la thèse de Jérôme Quartana à une expérimentation sur FPGA [QUA 05]. Un système GALS équipé d'un réseau de communication asynchrone a été implémenté sur un FPGA (Stratix, Altera) possédant deux domaines d'horloge. Ce système a été nommé PACMAN pour *Programmable And Configurable Multiprocessor Asynchronous Network* (Figure 14).

Ce démonstrateur comporte :

- Un réseau de communication asynchrone (ANoC) de type *crossbar* 5x5 équipé d'un lien parallèle permettant d'échanger directement des données avec l'extérieur. L'ANoC gère deux modes de trafic : un mode *burst* et un mode simple. Il dispose également d'un système d'arbitrage avec deux niveaux de priorité. A niveau de priorité égale, le premier paquet arrivé est le premier servi (*First In First Granted policy*). Si un paquet de priorité haute arrive, il peut interrompre la transmission de paquets de priorité basse sur un canal de communication déjà établi, sauf si ces derniers sont transmis en mode burst.
- Deux processeurs synchrones avec une mémoire RAM et un lien série. Le premier est cadencé à une vitesse de 10 MHz et le second à 50 MHz.
- Un module de cryptographie DES (Data Encryption Standard) asynchrone QDI.
- Un banc de mémoire partagée.

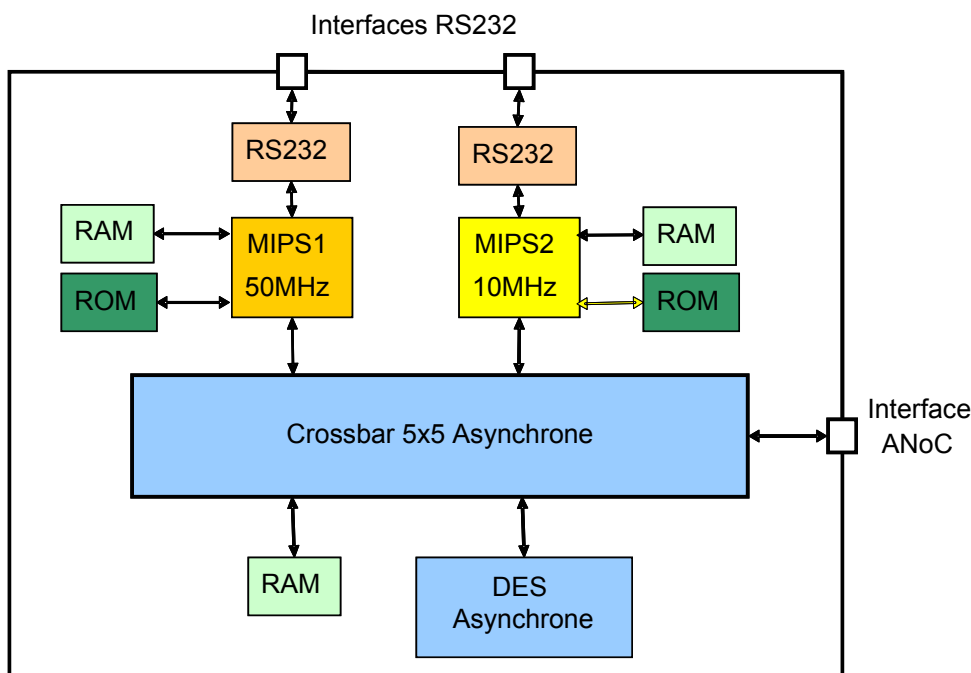


Figure 14 : La plateforme PACMAN

Il est à noter que cet ANoC a été construit en utilisant un outil de génération automatique (conçu spécifiquement pour cette thèse). L'outil permet notamment de générer

des réseaux complètement interconnectés, des architectures de type Octagon [KAR 01] ou des modules de routage qui peuvent être réutilisés dans des topologies plus complexes. L'objectif étant ici d'automatiser la génération de topologies d'interconnexion asynchrones et régulières. Cela est d'autant plus aisé que les blocs asynchrones (qui se synchronisent localement) s'assemblent comme des Legos. L'outil permet en outre de paramétrer la taille du réseau et la largeur des liens de communication.

La plateforme PACMAN a été testée et validée avec succès sur un FPGA Stratix d'Altera. Les simulations *post-place and route* ont montré (étant donné que l'on n'a pas glissé la sonde de l'oscilloscope dans le FPGA !) une latence de 5 ns pour traverser l'ANoC. Cela correspond à une partie du système réalisée complètement en logique asynchrone (La logique asynchrone est implantable sur des FPGAs du commerce, voir §II.5.2). Lorsqu'on ajoute les modules d'adaptation de protocole entre les processeurs et l'ANoC, la latence de processeur MIPS à processeur MIPS atteint alors 78 ns, ce qui permet d'atteindre potentiellement un débit de 13 MFlits/s.

Ainsi, nous avons démontré qu'il était possible de prototyper sur FPGA un système multi-horloges avec des modules synchrones et des modules asynchrones architecturé autour d'un réseau de communication sans horloge. Ces résultats montrent que les ANoCs sont des réseaux modulaires, rapides et fiables et qu'ils constituent des alternatives intéressantes aux réseaux synchrones.

II.4 La gestion de l'énergie

La gestion de l'énergie est devenue un problème crucial dans la conception des systèmes intégrés (cf. II.2.3). Dans ce domaine, la logique asynchrone offre des opportunités pour gérer plus efficacement la consommation des systèmes intégrés par rapport à son homologue synchrone. En effet, il est possible d'aborder cette problématique à trois niveaux différents : au niveau de la logique, au niveau d'un bloc de traitement et au niveau système.

II.4.1 Contrôle local de l'énergie

II.4.1.1 Gestion de l'énergie au niveau des portes

La logique asynchrone possède un certain nombre d'atouts pour maîtriser la consommation d'énergie au niveau des portes logiques. En effet, la présence de nombreux signaux de requête et d'acquiescement permet d'envisager de les utiliser pour déconnecter les portes et blocs logiques inutilisés ou d'agir sur la polarisation du substrat (back-biasing) afin de réduire les courants de fuite inhérents aux technologies fortement submicroniques. Comme les circuits asynchrones fonctionnent avec des signaux de signalisation, il est aisé de connaître le lieu où les données sont traitées à un instant précis. Cette connaissance permet de connecter ou de polariser à volonté le chemin de données en fonction de l'avancement des calculs. Comparativement aux circuits synchrones où il est impossible de déterminer l'état d'avancement des données dans le circuit (à moins de les instrumenter spécifiquement), les circuits asynchrones présentent ici un avantage intrinsèque.

II.4.1.2 Les états de veille faible consommation

Mode veille des processeurs synchrones

Il est bien établi que les processeurs synchrones utilisent un temps et une énergie non négligeables pour se mettre en sommeil et se réveiller. Ce point constitue un sévère handicap pour passer d'un état à l'autre rapidement et avec un coût énergétique raisonnable. En effet, les temps de transition nécessaires pour la mise en veille d'un microprocesseur varient de quelques dizaines de microsecondes (pour les microprocesseurs lpARM [PER 00] et Crusoe [FLE 01][TRA 07] à plusieurs dizaines de millisecondes pour le réveil d'un microprocesseur

tel que le StrongARM [KUM 00]. Les surcoûts temporels et énergétiques ne permettent donc pas de démarrer et arrêter continuellement les processeurs synchrones. Dans le but de minimiser ces inconvénients, les processeurs synchrones disposent de plusieurs états de sommeil. Par exemple, un état de sommeil léger serait de stopper uniquement l'horloge afin de redémarrer au plus vite le processeur en cas de nécessité alors qu'un état de sommeil profond correspondrait à couper l'alimentation du processeur à l'exception du circuit de veille.

Mode veille des processeurs asynchrones

Contrairement aux processeurs synchrones, les processeurs asynchrones sont bien adaptés pour exploiter des mises en sommeil et des réveils fréquents et temporellement proches. En effet, ces derniers ne consomment de l'énergie que s'ils ont des données à traiter. Quand aucune donnée n'est disponible, les processeurs se positionnent automatiquement en mode veille et se réveillent dès qu'une interruption survient ou que des données apparaissent. Ce point traduit en fait les propriétés intrinsèques des circuits asynchrones qui possèdent un fonctionnement de type « flot de données ». Le coût temporel pour le réveil des processeurs asynchrones est équivalent à celui du traitement d'une interruption. Ces propriétés remarquables des circuits asynchrones leur confèrent l'aptitude de passer du mode actif au mode veille très rapidement. On parle alors de commutations à « grain fin » entre état de veille et activité. Ce point est illustré sur la Figure 15.

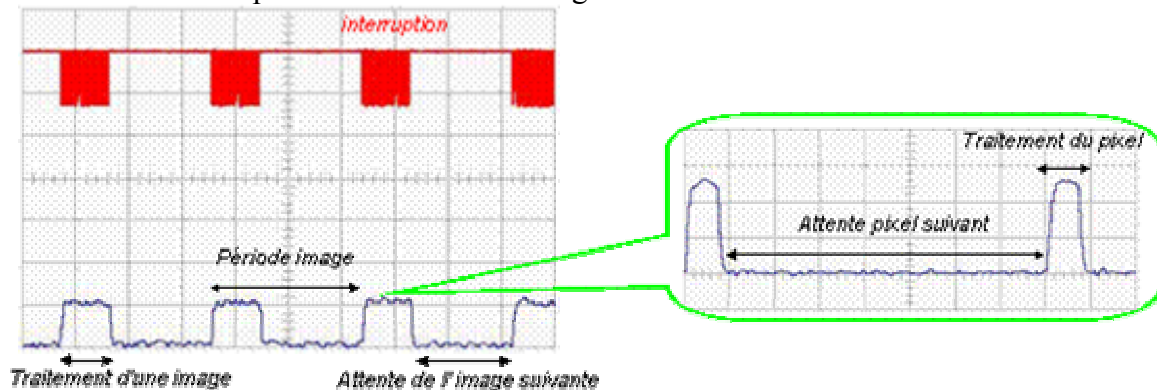


Figure 15 : Le processeur asynchrone ASPRO est arrêté entre le traitement de deux pixels d'une image.

La figure montre des mesures faites sur une application vidéo réalisée avec le processeur ASPRO [REN 98][REN 99]. L'application envoie les images d'une caméra numérique au processeur ASPRO par un lien série haut débit. Entre 2 trames consécutives, le processeur passe en mode veille car il n'a plus de données à traiter. En effectuant un zoom (dans la bulle de la Figure 15), il est possible de constater que nous observons le même phénomène mais avec un grain cette fois-ci bien plus fin. En effet, le processeur se met en veille également entre deux pixels et se réveille dès qu'un nouveau pixel est disponible sur le lien série. Le rapport cyclique correspondant au temps d'activité du processeur par rapport à une période image est de 0.333. Le rapport cyclique entre le temps d'activité du processeur et la période pixel est de 0.125. Au final, la réduction d'énergie est de 95% comparativement à un système ne disposant d'aucune technique de mise en veille. Cette technique n'est malheureusement pas réalisable avec des processeurs synchrones !

II.4.2 Gestion de l'énergie au niveau système

De nombreuses techniques logicielles et matérielles ont été développées au cours de la dernière décennie pour contrôler la consommation électrique des systèmes électroniques. Néanmoins, avec l'augmentation des performances des composants, la consommation électrique des systèmes intégrés ne fait que croître [GOW 98]. Ces raisons ont conduit les

concepteurs à étudier des solutions pour réduire la consommation tant au niveau logiciel qu'au niveau matériel [ESS 02][LI 03]. Dans ce contexte de systèmes intégrés, pour lesquels les performances sont nécessaires, il est souvent difficile de concilier énergie et performances malgré des techniques de mise en veille et de contrôle de la tension d'alimentation, qui ne sont souvent pas très réactives. En effet, il apparaît clairement que les temps de commutation en mode veille, de variations de fréquence et de tension se font sur des durées non négligeables. Le temps et le coût énergétique associés à ces opérations empêchent d'exploiter ces techniques de façon intensive. De ce point de vue, la technologie asynchrone peut constituer une alternative intéressante.

Dans la littérature spécialisée, il existe un grand nombre d'articles sur la gestion de l'énergie exposant des solutions matérielles [BUR 95][CHA 92][PED 97] ou logicielles [KUM 00][WEI 94], mais très rarement des solutions combinant les deux aspects. Les travaux effectués avec Mohammed Es Salhiene sur ce thème [FES 03] nous ont conduit à une méthode intégrant à la fois logiciel et matériel. L'étude a porté sur l'association d'un microprocesseur asynchrone et d'un système d'exploitation dédié à la faible consommation. En adaptant à un microprocesseur asynchrone une politique de gestion de l'énergie dirigée par le système d'exploitation, il est possible de contrôler la vitesse d'exécution du processeur. Cette technique exploite notamment la propriété remarquable des circuits asynchrones de réguler leur vitesse en fonction de leur tension d'alimentation [ABR 01][NIE 94][REN 99].

II.4.2.1 Modèle temporel pour l'adaptation de vitesse des microprocesseurs

Processeurs asynchrones

La vitesse d'exécution des processeurs asynchrones suit une loi de commande dont le seul paramètre d'entrée est la tension d'alimentation. Il en résulte que le temps de passage d'une vitesse d'exécution à une autre, t_v , ne dépend que du convertisseur DC-DC et de la capacité de charge du processeur. Le temps t_v peut être modélisé comme une fonction de V_{dd1} et V_{dd2} , les tensions d'alimentation avant et après le changement de vitesse. En première approximation, il est possible de considérer t_v comme une simple expression linéaire : $t_v = k \cdot |V_{dd1} - V_{dd2}|$ où k est un paramètre d'échelle qui dépend du convertisseur DC-DC et de la capacité de charge du processeur. De plus, il est important de noter que dans le cas asynchrone, le processeur n'est pas arrêté durant cette phase de transition entre deux vitesses. Il continue à exécuter les instructions malgré la variation de la tension d'alimentation. Enfin, la vitesse peut être ajustée continûment et très finement. La Figure 16 montre que les tâches τ_2 et τ_3 ne sont pas stoppées durant la variation de vitesse.

Processeurs synchrones

Dans le cas synchrone, la vitesse de variation dépend de la tension d'alimentation mais aussi de la fréquence de l'horloge. Le temps de variation de la vitesse, t_v , dépend ici du convertisseur DC-DC mais aussi de la PLL qui contrôle la fréquence d'horloge. Dans ces conditions, il s'exprime comme suit : $t_v = t_{DC-DC} + t_{PLL}$ où t_{DC-DC} est le temps de transition de la tension initiale à la tension finale et t_{PLL} est le temps nécessaire au changement de fréquence. Il est à noter que ces temps sont additifs. En effet, si l'on baisse la tension sans avoir au préalable réduit la fréquence, les hypothèses temporelles de fonctionnement du circuit peuvent être violées. De même, un accroissement de la fréquence sans une augmentation préalable de la tension risque de provoquer également des violations de « timing ». Il est possible d'exprimer $t_v = k_1 |V_{dd2} - V_{dd1}| + k_2 |f_2 - f_1|$ où k_1 et k_2 sont respectivement des paramètres d'échelle pour le convertisseur DC-DC et la PLL. Les fréquences f_1 et f_2 correspondent à la fréquence initiale et la fréquence finale du processeur. Ce changement de fréquence est en général synonyme d'arrêt du traitement des instructions. Ce point contribue

grandement à diminuer l'efficacité de la variation de vitesse de ces processeurs. La Figure 16 montre le modèle temporel retenu pour effectuer une variation de vitesse d'exécution d'un processeur synchrone.

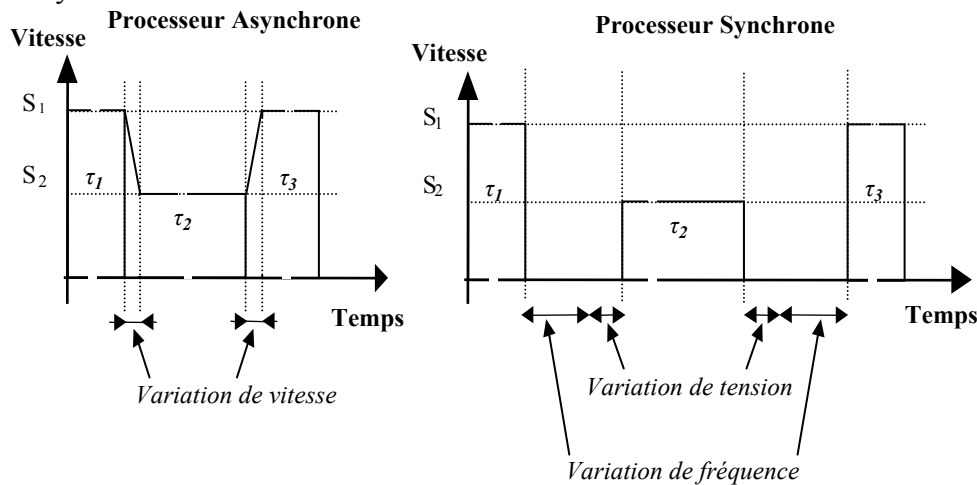


Figure 16 : Variation de la vitesse d'exécution des processeurs synchrones et asynchrones

On observe que le processeur est stoppé durant la phase de transition et qu'aucune instruction n'est exécutée. Pendant ce laps de temps, les processeurs synchrones continuent à consommer de l'énergie comparativement aux processeurs asynchrones qui, eux, poursuivent le traitement des instructions tout en assurant la modification de leur tension d'alimentation. Il est possible d'exprimer quantitativement ce coût. Il s'exprime comme la somme du coût lié à la variation de la tension plus le coût induit par la modification de la fréquence : $E = E_v + E_f$. Le coût énergétique du DVS s'exprime comme suit :

$$E = \alpha C f \int_{t_0}^{t_1} V^2(t) dt + \alpha C V^2 \int_{t_1}^{t_2} f(t) dt \text{ où } \alpha \text{ est un facteur d'échelle, } C \text{ est le}$$

facteur de charge du circuit et les intervalles $[t_0, t_1]$ et $[t_1, t_2]$ sont respectivement le temps nécessaire à la variation de tension à fréquence fixe et le temps de modification de la fréquence à tension fixe. Cette formule montre bien que le changement de la vitesse d'exécution à un coût énergétique alors qu'aucune instruction n'est traitée par le processeur.

II.4.2.2 Algorithmes d'adaptation dynamique de la vitesse d'exécution des processeurs asynchrones

Un système temps réel est souvent amené à gérer des tâches périodiques et des tâches sporadiques. En effet, les tâches périodiques sont généralement dédiées à la lecture périodique de données sur des capteurs ou à la mise à jour de l'état du système. Pour cette raison, la plupart des politiques de gestion de la tension d'alimentation proposées dans la littérature ne considèrent que des tâches périodiques. Les tâches sporadiques, quant à elles, sont utilisées pour gérer les événements asynchrones. Elles ont été relativement peu étudiées [PER 00]. Dans nos travaux, nous avons étudié ces deux situations [ESS 02][FES 03]. Pour des raisons de concision, seul le cas périodique sera présenté dans la suite.

Définition du modèle d'une tâche

Chaque tâche peut être caractérisée par un triplet $\langle NI_i, D_i, T_i \rangle$ où NI_i est le nombre d'instructions de la tâche, D_i son échéance et T_i sa période ou son occurrence la plus forte. Nous supposons que :

- les tâches sont indépendantes et que leurs paramètres sont connus dès le début de leur exécution,

- les tâches périodiques ont des échéances égales à leur période et que les tâches ont des périodes différentes,
- qu'à tension maximale (ou à vitesse maximale d'exécution), toutes les tâches périodiques et sporadiques peuvent être traitées,
- que le surcoût temporel et énergétique dû aux changements de contexte entre tâches est négligeable.

De plus, comme l'exécution des programmes n'est pas suspendue durant le changement de tension, nous supposons que l'impact du changement de tension est également négligeable.

Algorithmes d'adaptation de la tension pour des tâches périodiques

Dans ce paragraphe, nous considérons un système où toutes les tâches sont périodiques. Nous supposons qu'au début (instant $t = 0$), il y a n tâches périodiques prêtes dans la file d'attente des tâches prêtes ordonnées par priorité selon l'ordonnancement EDF (Earliest Deadline First). La vitesse du processeur est alors fixée à :

$$S = \sum_{j=1}^n \frac{NI_j}{D_j}$$

Lorsqu'une nouvelle tâche périodique τ_i est créée, elle est insérée dans la file d'attente des tâches prêtes, puis l'algorithme d'adaptation de la tension réévalue la vitesse du processeur afin que toutes les tâches dans la file d'attente soient exécutées avant leurs échéances. La nouvelle vitesse est donnée par :

$$S = \frac{NI_i}{D_i} + \sum_{j=1}^n \frac{\overline{NI}_j}{td_j - t}$$

où \overline{NI}_j représente le nombre d'instructions de la tâche τ_j restant à exécuter, td_j son échéance, NI_i le nombre d'instructions de la tâche τ_i , D_i son délai critique et t le temps courant. De même, si une tâche périodique est supprimée, la vitesse de processeur est mise à jour par l'équation :

$$S = \sum_{j=1}^{n-1} \frac{\overline{NI}_j}{td_j - t}$$

Pour illustrer la mise en œuvre de cet algorithme, considérons les trois tâches de la Table 1 pour lesquelles une date d'activation, un nombre d'instructions et une échéance ont été attribués. La valeur S_{MAX} de la vitesse du processeur et la valeur P_{MAX} de la puissance consommée sont celles correspondant à la tension d'alimentation maximale.

Tâches	NI_j	D_j	T_j	Date d'activation
τ_1	$0.25 \cdot 10^6$	2	2	0
τ_2	$1 \cdot 10^6$	5	5	0
τ_3	$0.5 \cdot 10^6$	3	3	4

Table 1 : Paramètres des tâches

Quand le processeur fonctionne à vitesse maximale, S_{MAX} , il existe des périodes d'attente entre la fin de certaines tâches, avant leurs échéances, et le début de l'exécution des tâches suivantes (cf. Figure 17.a). L'énergie consommée pendant cette attente active constitue une perte pure pour le système puisqu'elle ne participe à la production d'aucun résultat. Par conséquent, la vitesse du processeur peut être réduite en diminuant la tension d'alimentation de sorte que les délais impartis à l'exécution des tâches soient pleinement utilisés. Dans la Figure 17.b, la vitesse du processeur est diminuée afin de réduire la puissance dissipée. Au début, elle est fixée à 33% de S_{MAX} selon l'équation 4. A $t = 4$, la tâche 3 devient prête et la vitesse de processeur est réévaluée, selon l'équation ci-dessus, à 49% de S_{MAX} . La puissance moyenne consommée est alors 14% de P_{MAX} .

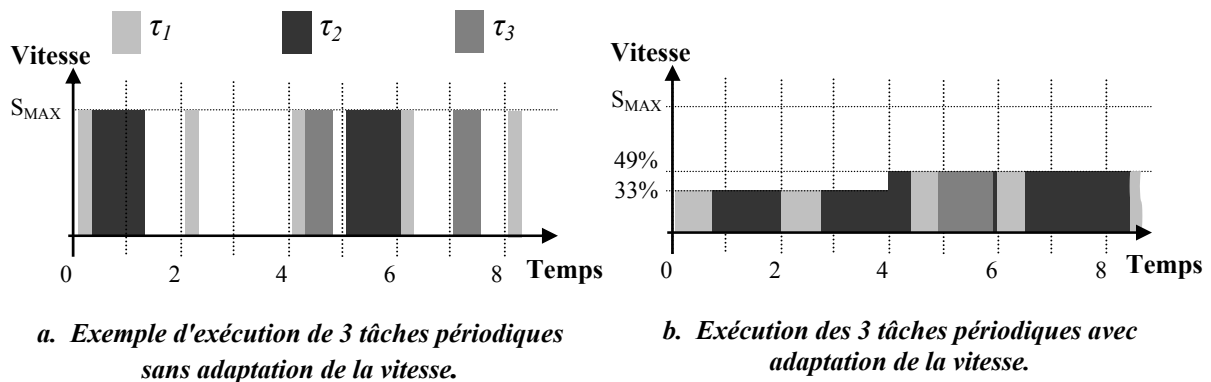


Figure 17 : Illustration de l'adaptation de la vitesse d'exécution pour des tâches périodiques.

Les résultats obtenus (avec un ordonnancement dynamique) s'approchent asymptotiquement de ceux de Bruno Gaujal [GAU 03] qui a proposé un algorithme pour ordonnancer statiquement un ensemble de tâches soumises à des contraintes temps réel en minimisant la consommation d'énergie du processeur (avec du « Dynamic Voltage Scaling»). L'algorithme de Gaujal est linéaire avec le nombre de tâches, ce qui constitue une amélioration par rapport à l'algorithme classique de Yao [YAO 95]. Cette amélioration a été possible en considérant le problème d'ordonnancement des tâches comme un problème de plus court chemin. Gaujal a également proposé un algorithme pour traiter le cas où le processeur possède un nombre fini de vitesses (Vdd hopping, Frequency hopping). Il a ensuite étendu son algorithme pour minimiser le nombre de changements de vitesse, ce qui est important puisque le coût d'un changement de vitesse ne peut être négligé. Ces résultats sont remarquables car tous les algorithmes sont linéaires avec le nombre de tâches si les instants d'arrivée et d'échéance sont pré-triés, et en $O(N \log N)$ sinon. De plus, ces complexités sont prouvées être optimales.

Conclusion

Ces travaux montrent les bonnes dispositions des systèmes asynchrones à mieux gérer l'énergie que leurs homologues synchrones. Comparativement à ces derniers, ils présentent l'avantage de se mettre automatiquement en veille dès qu'il n'y a plus de données à traiter. De plus, cette mise en sommeil se fait avec des coûts temporels et énergétiques insignifiants par rapport aux dispositifs de mise en veille des circuits synchrones. Enfin, la gestion de la vitesse d'exécution d'un microprocesseur est simplifiée à l'extrême car seule la tension d'alimentation d'un circuit asynchrone influe sur sa vitesse. Tous ces avantages démontrent le grand intérêt des systèmes asynchrones, notamment pour des systèmes intégrés portables et autonomes, car ils conduisent à des réductions d'énergie plus importantes, à des systèmes matériellement plus simples et à des politiques de gestion de l'énergie simplifiées.

II.5 Les systèmes reconfigurables

II.5.1 Nécessité des systèmes configurables

Les systèmes configurables sont aujourd'hui très répandus. Ils offrent en effet aux concepteurs un moyen efficace et peu onéreux de valider par prototypage les circuits numériques complexes et des systèmes intégrés. Ils sont également très utilisés pour réaliser de petites séries de circuits numériques pour lesquels le ticket d'entrée pour la fabrication d'un ASIC est trop élevé. Par ailleurs, on assiste depuis quelques années à une montée en puissance des systèmes (re)configurables. Ils sont presque toujours associés à un système complexe, intègrent souvent un microprocesseur et sont de plus en plus intégrés au système

sur une même puce. Les circuits configurables ne sont plus seulement des composants externes, ils deviennent aussi des composants embarqués ! Ils offrent en effet de nombreux avantages. La (re)configurabilité apporte flexibilité et souplesse d'emploi : elle permet de corriger aisément les petites erreurs de conception à moindre coût et de faire évoluer un produit durant son cycle de vie. Elle permet aussi de répondre dans une certaine mesure aux demandes de dernières minutes des clients qui souhaitent ajouter des fonctionnalités à leur système. Avec un ASIC, de telles requêtes ne sont pas concevables sans un surcoût conséquent et un retard de fabrication. Par ailleurs, leurs structures répétitives en font de bons candidats à l'intégration dans les technologies décanométriques où la variabilité des procédés de fabrication devient problématique et cruciale pour le rendement. En effet, il est plus facile de valider une brique unique, répétée un grand nombre de fois, que de valider un circuit complexe où les composants et les blocs le constituant diffèrent. En contrepartie, le prix à payer pour bénéficier de la souplesse d'un système programmable est une surface notablement accrue et une consommation plus élevée que pour un système dédié.

Par ailleurs, on assiste à l'émergence de systèmes programmables où les nœuds du système sont des structures programmables indépendantes de type FPGA interconnectés entre elles par un réseau de communication sur puce (NoC). Chaque nœud est indépendant et peut être instrumenté afin de déterminer son niveau de performances (vitesse, consommation, ...). Cette information est alors exploitée pour répartir les tâches dans le système en fonction des besoins. Ainsi, une tâche gourmande en ressources de calcul sera placée sur un nœud rapide. Il devient alors possible d'imaginer des systèmes intelligents capables de gérer la fréquence de fonctionnement des nœuds et leur consommation en fonction d'objectifs globaux visant une optimisation des performances ou une réduction de la consommation par exemple. Le projet ARAVIS (pôle de compétitivité Minalogic) propose une architecture SoC de ce type, entièrement symétrique et re-configurable basée sur des îlots de calcul (DSP Fabric re-configurable) localement synchrone et globalement asynchrone dans une technologie avancée (32 nm). Il est notamment proposé de combiner trois technologies clés pour concevoir et fabriquer un système sur puce (SoC) en 32 nm et apporter ainsi des éléments de réponse aux défis submicroniques. Ces trois technologies sont : les structures re-configurables à grain grossier, la technologie asynchrone, et la gestion dynamique de l'énergie et de l'activité basée sur des techniques d'automatique avancées. L'infrastructure logicielle du SoC ARAVIS sera également conçue pour ces nouvelles technologies en tenant compte des besoins de contrôle et de gestion de l'activité dans le circuit pour maîtriser la consommation et les variations de la technologie. Enfin, il est à noter que le projet ARAVIS cible deux domaines applicatifs avec un fort potentiel industriel et commercial : le multimédia et les télécoms.

II.5.2 Les circuits programmables et la logique asynchrone

Les circuits programmables sont très intéressants pour le prototypage et c'est donc tout naturellement que nous avons cherché à les utiliser pour prototyper nos circuits asynchrones. Toutefois, le mapping d'un circuit asynchrone sur un FPGA standard n'est a priori absolument pas évident. En effet, les circuits asynchrones nécessitent une conception plus rigoureuse que leurs homologues synchrones du fait qu'ils sont conçus tout ou partiellement sans aléa [REN 00]. Cette propriété de ne pas générer d'aléa doit être conservée lors du mapping des fonctions sur les éléments logiques programmables. Il faut donc s'attacher à concevoir les circuits de sorte qu'ils ne produisent pas d'aléa. Les aléas apparaissent généralement lors de transitions multiples sur les entrées d'une LUT ou d'un mapping inapproprié du circuit sur le FPGA (toutes les simplifications et optimisations ne sont pas autorisées). Par ailleurs, les hypothèses temporelles faites avec ces circuits sont généralement moins fortes que celles des circuits synchrones. Elles ne posent pas de problème particulier, mais peuvent nécessiter de simples précautions lors de la phase de placement-routage.

II.5.2.1 Mapping des circuits asynchrones

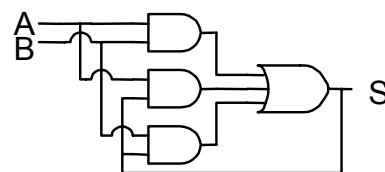
Enfin d'illustrer l'approche, les circuits quasi-insensibles aux délais (QDI) seront pris en exemple, mais l'approche peut être généralisée. L'outil TAST (TIMA Asynchronous circuit Synthesis Tools), utilisé pour la synthèse de circuits QDI, est présenté au chapitre III. Il suit une méthode générale pour la synthèse de circuits asynchrones qui, à partir d'une spécification HDL, génère une netlist utilisant les cellules d'une bibliothèque de portes de Muller et de portes AND, INV et OR (Les verrous et bascules peuvent également être utilisés pour des circuits micropipelines par exemple). Les portes AND et OR sont normalement implémentées dans le FPGA. Les portes de Muller sont implémentées en instanciant la fonctionnalité directement sur une LUT. Cette approche permet notamment de réaliser toutes les cellules de Muller symétriques et dissymétriques. La bibliothèque de portes de Muller contient des portes à 2, 3 ou 4 entrées (selon la cible utilisée) avec ou sans reset. Afin de garantir qu'il n'y aura pas d'aléa généré par les cellules de Muller, on impose qu'elles soient mappées sur une seule LUT. Cela empêche les courses de signaux qui pourraient exister entre plusieurs blocs mais ne suffit pas pour supprimer toutes les causes d'aléa. Toutefois, cela est suffisant pour l'usage qui en est fait. Les circuits QDI par construction respectent des règles de fonctionnement qui assurent que les transitions sur les entrées des portes de Muller sont monotones et qu'elles autorisent uniquement des transitions multiples non dangereuses [HO 02]. Les entrées monotones impliquent qu'il n'y a pas de transitions successives sur un même signal sans qu'il y ait eu de transitions sur les autres entrées. Par exemple avec une MULLER2, les transitions multiples $AB=01 \rightarrow 10$ et $AB=10 \rightarrow 01$ sont interdites où A et B sont les entrées de la porte.

Suppression des aléas

La méthode standard consiste à synthétiser les fonctions logiques en un réseau à deux niveaux composés de AND et de OR. La théorie des aléas pour les transitions uniques (Single Input Change ou SIC) a été développée initialement par Huffman, Unger et McCluskey [UNG 71][WAK 02]. Ils ont notamment montré que seuls les aléas statique-1 (une transition qui fait « passer » la sortie de 1 à 1!) sont susceptibles de générer des aléas. Il faut donc supprimer uniquement ces aléas lors de la synthèse. La Figure 18 montre l'implémentation sans aléa dans le cas SIC de la porte MULLER2 avec des portes. La sortie est rebouclée sur une entrée. L'état précédent de la sortie S est dénoté S^{-1} .

		AB			
		00	01	11	10
S	S	0	0	1	0
	S^{-1}	0	0	1	0

a. table de vérité de la porte MULLER2
transitions interdites : $AB=01 \rightarrow 10$ et $10 \rightarrow 01$



b. décomposition de la porte MULLER2 en réseau à deux niveaux AND, OR.

Figure 18: Implémentation sans aléa de la porte MULLER2 avec des portes AND et OR.

La minimisation des mintermes canoniques en trois termes recouvrants AB , AS^{-1} , BS^{-1} permet d'éviter l'aléa statique-1 qui dépend de la distribution des délais lorsqu'une entrée change [DAV 97]. Or les réseaux AND-OR à deux niveaux sont sans aléa statique-0 ($0 \rightarrow 0$) et dynamique ($0 \rightarrow 1$, $1 \rightarrow 0$) dans le cas SIC. Ainsi cette implémentation est sans aléa lorsqu'une seule entrée change. Cette implémentation est aussi exempte d'aléas lorsque les deux entrées changent de $AB=00 \rightarrow 11$ ou $AB=11 \rightarrow 00$. Si les changements $AB=01 \rightarrow 10$ ou $AB=10 \rightarrow 01$ surviennent, il peut apparaître un aléa [DAV 97]. Dans le tableau de Karnaugh de la Figure 18.a, cette situation est matérialisée par la flèche. Or comme les circuits QDI, les

signaux d'entrée d'une porte de Muller sont monotones, on peut en conclure qu'il ne se produira pas d'aléa. Il est à noter que l'on sait éliminer ce type d'aléa [BEI 74], mais cela ne s'avère pas nécessaire dans notre cas.

La structure symétrique des LUTs (voir Figure 19) met en évidence qu'un changement sur une entrée ne provoque pas d'aléa. L'aléa statique-1 ne peut pas se produire ici. En revanche, des aléas peuvent apparaître dans le cas MIC. Toutefois, dans la situation qui nous intéresse, seules les transitions $AB = 01 \rightarrow 10$ ou $AB = 10 \rightarrow 01$ sont susceptibles de provoquer un aléa. En effet, si les signaux A et B (S1 et S2 sur la figure) changent simultanément, ils sont susceptibles de provoquer une course entre les signaux du premier étage et du second étage de la LUT. Cette situation peut mener à la création d'un aléa qui peut potentiellement se propager via le troisième étage qui fait office de mémorisation (S0 étant connecté à la sortie).

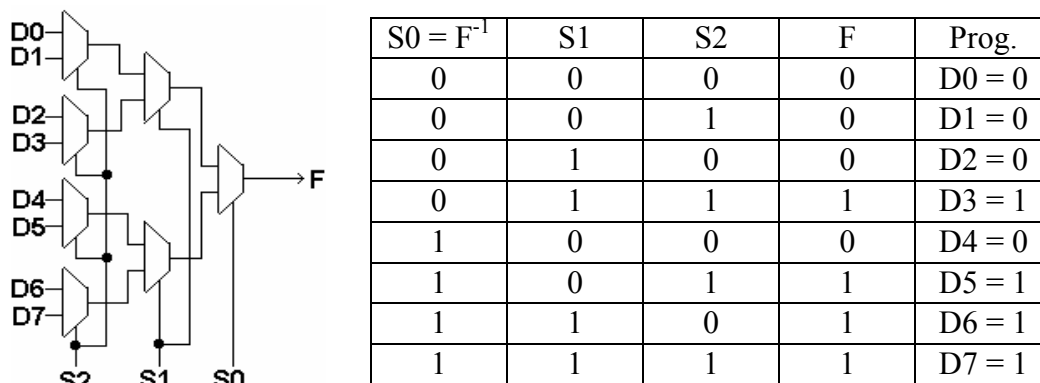


Figure 19 : Structure d'une LUT à 3 entrées et programmation d'une Muller à deux entrées

Mapping d'un circuit asynchrone sur un FPGA du commerce

L'étude des principes de base des circuits asynchrones et des circuits programmables montre qu'il est possible d'implémenter des circuits asynchrones QDI sur des FPGA du commerce [HO 02]. Pour que les circuits asynchrones fonctionnent correctement, il est tout de même nécessaire de respecter deux contraintes :

- Une logique exempte d'aléa.
- Un temps équilibré dans les fourches isochrones.

Il est à noter que seules certaines fourches exigent un fonctionnement isochrone. Pour réaliser ce mapping, il est nécessaire de construire une bibliothèque de cellules asynchrones (Mullers) qui sera utilisée avec la netlist en portes de Muller du circuit à implémenter. Il ne reste alors plus qu'à procéder au mapping du circuit avec les outils du fabricant de FPGA. Toutefois, afin de garantir que l'implémentation se fera correctement sur le FPGA, il faut prendre les mesures suivantes qui sont réalisables aisément :

- La résolution du problème d'aléas impose de mapper chaque cellule de la bibliothèque TAL (TIMA Asynchronous Library, la bibliothèque de cellules de Muller !) dans une seule cellule logique. Selon l'architecture du FPGA ciblé, le nombre de cellules de Muller possible peut varier.
- L'équilibrage des fourches isochrones exige de forcer le routage avec des contraintes. Cela est réalisé en imposant un placement équidistant des cellules constituant une fourche et en routant ces cellules en premier.

Par extension, il est aussi possible de prototyper d'autres types de circuits asynchrones comme les circuits micropipeline par exemple, le contrôleur étant QDI et le chemin de données synchrone. Il faudra prendre le soin de bien caractériser les structures de retards nécessaires à ces circuits.

II.5.2.2 Le flot de conception asynchrone sur FPGA standard

Afin de conclure cette partie sur le prototypage de circuits asynchrones sur des FPGA du commerce, le flot de conception développé au laboratoire est représenté sur la Figure 20.

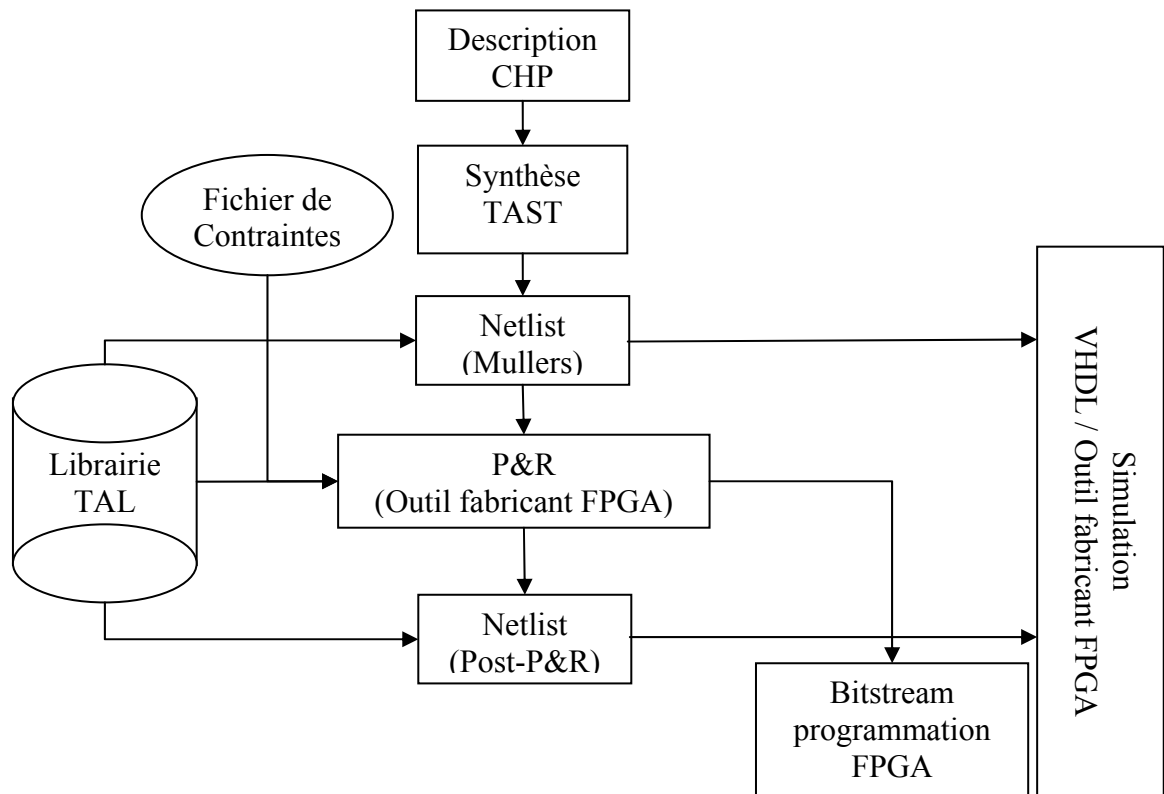


Figure 20 : Flot de conception asynchrone pour les FPGA du commerce

Le flot de conception est finalement très classique. L'effort principal a été de concevoir une bibliothèque de cellules asynchrones (TAL). La bibliothèque présente en général des vues technologiques qui précise le mapping à appliquer sur les cellules programmables, mais aussi un modèle comportemental VHDL des cellules de Muller qui sert à la simulation. L'outil de synthèse TAST est présenté dans le document au paragraphe III.3. Cette méthode a été employée avec succès sur des circuits de filtrage numérique, de chiffrement DES et sur la plateforme GALS présentée au paragraphe II.3.4.3.

II.5.3 Les circuits programmables dédiés à l'asynchrone

Comme nous l'avons vu dans le paragraphe précédent, il est possible d'implémenter des circuits asynchrones sur des FPGA standard. Néanmoins cette approche n'est pas optimale dans la mesure où de nombreuses ressources du FPGA restent inexploitées. C'est le cas notamment des verrous et bascules placés les blocs programmables (CLB chez Xilinx et LC chez Altera). En effet, il n'est pas toujours possible, ni souhaitable de les utiliser pour construire les éléments des cellules de bibliothèques. Il en résulte une sous-utilisation massive de ces cellules de mémoires et comme les ressources combinatoires sont pleinement exploitées, il en résulte un taux de remplissage du FPGA élevé. Afin de réduire ces inconvénients, nous avons développé des structures programmables plus adaptées au monde asynchrone. La première étape a été de définir une architecture de FPGA qui soit la plus indépendante du style de logique asynchrone utilisée. Cette caractéristique nous semble particulièrement intéressante pour avoir des éléments de comparaison entre les différents styles de logiques asynchrones existants. Ils sont d'autant plus nombreux que l'on fait plus ou

moins d'hypothèses temporelles et que l'on varie les protocoles. On peut par exemple concevoir des circuits aussi différents que des circuits micropipeline avec des contrôleurs deux phases qu'un circuit QDI quatre phases avec un protocole WCHB. Il est à noter que les nombreux essais effectués sur des FPGA asynchrones étaient tous fortement liés à un style de logique. Par exemple, GALSA [GAO 96] et STACC [PAY 97] sont en fait des structures GALS et ne sont donc pas réellement de vrais FPGA asynchrones. En revanche, le circuit PAPA [TEI 03][TEI 04], conçu par John Teifel et Rajit Manohar, est un FPGA totalement asynchrone dédié à la construction de structures pipelinées haut débit. Il existe aussi des FPGA en logique QDI comme celui réalisé par Alain Martin et Catherine Wong [WON 03].

II.5.3.1 Architecture d'un FPGA asynchrone multi-style

Cette réflexion nous a conduit à une architecture finalement assez classique pour notre FPGA asynchrone. Le motif constituant le FPGA est en effet composé d'une logique de connexion au réseau de routage (Connexion Box), de cellules de routage (Switch Box) et de cellules programmables (Programmable Logic Bloc) (cf. Figure 21).

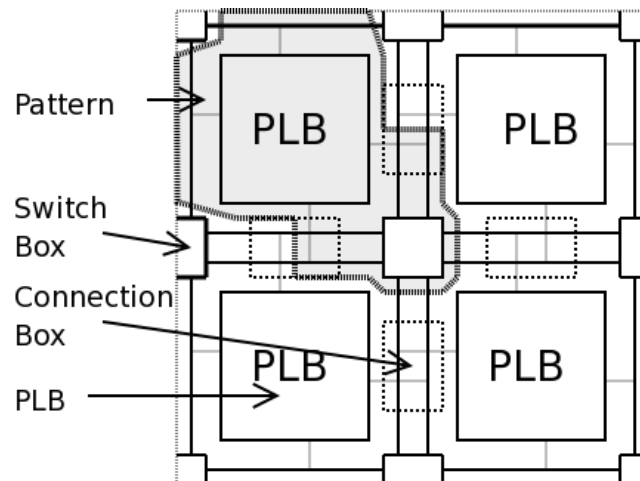


Figure 21 : Motif élémentaire constituant un FPGA asynchrone

En fait, l'innovation a été faite principalement sur la structure de la logique programmable [FES 05]. Le bloc de logique programmable (PLB) a été conçu de façon à accepter au mieux de la logique asynchrone multivaluée (codage en 1 parmi n). Ce type de codage est favorable à la construction de circuits insensibles aux délais. Par ailleurs, nous avons dimensionné les éléments programmables (Look-Up Table ou LUT) afin de faciliter la constitution des structures élémentaires qui sont le plus souvent rencontrées avec les circuits asynchrones. Ainsi une structure du type LUT7-3 (7 entrées et 3 sorties – voir le paragraphe qui suit pour la définition) couplée à une LUT2-1 (2 entrées) utilisée pour la génération des acquittements remplit parfaitement ce rôle (voir Figure 22). Typiquement, une construction logique telle que deux Mullers suivies d'une porte OR, qui est classique en asynchrone, s'intègre parfaitement dans ce dispositif.

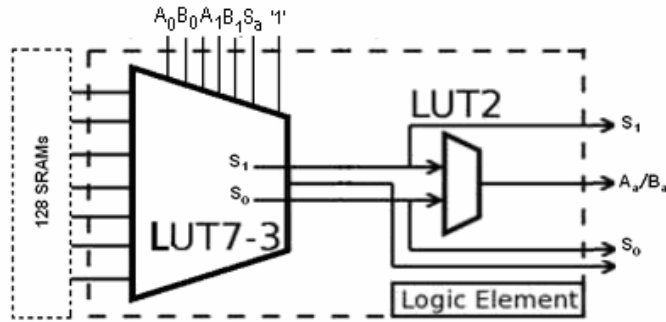


Figure 22 : La structure élémentaire pour des FPGA asynchrones

II.5.3.2 Des LUT mémorisantes avec plusieurs sorties

Comme cela est indiqué au paragraphe précédent, l'usage de LUT7-3 est requis pour concevoir ces éléments programmables. Une LUT7-3 est composée de 2 LUT à 6 entrées suivies d'un multiplexeur à 2 entrées (voir Figure 23). Les sorties des deux LUT à 6 entrées ainsi que celle du multiplexeur forment les trois sorties de la LUT7-3, la notation 7-3 signifiant 7 entrées et 3 sorties.

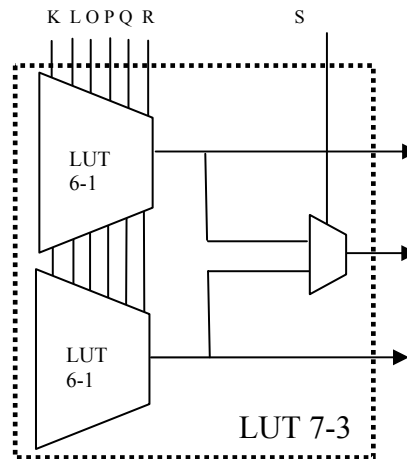


Figure 23 : Structure d'une LUT7-3

Par ailleurs, un des points clés de ces LUT sera leur capacité à disposer d'une boucle combinatoire entre la sortie et une entrée afin d'implémenter les mémorisations (qui constituent les portes de Muller). Ainsi, l'architecture de ces LUT a été optimisée afin de réaliser au mieux ces fonctions [FES 06]. Cette fonctionnalité peut être réalisée en ajoutant un multiplexeur supplémentaire à la LUT comme cela est représenté sur la Figure 24. Cette figure montre une LUT3-1 avec l'extension nécessaire à la mémorisation. On remarque qu'en fonction de M, le dernier étage de la LUT se comporte soit comme une mémoire, soit comme une entrée de la LUT. Il est possible d'utiliser cette LUT3-1 comme une LUT classique à 3 entrées ou bien comme une LUT à 2 entrées avec une mémoire. Ce dernier cas correspond typiquement à l'implémentation d'une porte de Muller à 2 entrées. Il suffit alors de reprendre la programmation donnée au paragraphe II.5.2.1.

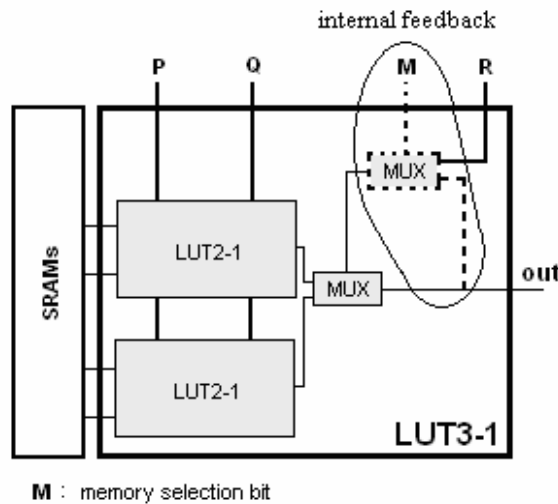


Figure 24 : Structure d'une LUT mémorisante

II.5.4 Les systèmes configurables sécurisés

Les circuits asynchrones comptent de nombreuses propriétés intéressantes que l'on peut exploiter à diverses fins. L'une d'elles est la possibilité de concevoir des puces sécurisées. On entend par puce sécurisée la possibilité de concevoir des circuits (typiquement de chiffrement mais pas nécessairement) dont la résistance à des attaques par canaux cachés (Side-Channel Attacks) est supérieure à celle des circuits ordinaires. Les travaux menés dans le groupe ont permis de mettre en évidence d'excellentes dispositions des circuits asynchrones QDI à cet égard. Dans le microcosme de la carte à puces, comme les standards de chiffrement évoluent, que les usages des cartes changent au cours du temps, il semble opportun de nos jours de concevoir des cartes reconfigurables au gré des besoins de l'utilisateur et de la technologie. Cela ne se fait pas sans considérer les délicats problèmes liés à la sécurité des données contenues dans ces cartes. Dans le cadre d'un projet ANR, SAFE, effectué en collaboration avec l'ENST Paris, la conception d'un FPGA asynchrone sécurisé a été entreprise. Le cahier des charges vise donc la conception d'un FPGA embarqué prévu pour supporter de la logique asynchrone et capable d'une bonne tolérance aux attaques en puissance (Differential Power Analysis et Simple Power Analysis) et en temps (Timing Attacks). Afin de parvenir à cette fin, il est nécessaire de construire un FPGA qui ne signe pas en courant, ni en temps. Il est donc nécessaire de concevoir une architecture de FPGA parfaitement équilibré ainsi qu'une méthode de programmation adaptée qui produit des fonctions logiques dont la profondeur est constante. L'ensemble de ces solutions est en cours d'expérimentation, mais les résultats préliminaires obtenus sont extrêmement encourageants.

Afin d'équilibrer les temps de réponse des LUT et de rendre le courant consommé le plus indépendant des données, une architecture de LUT à base de mémoires statiques a été réalisée [REN 07]. L'architecture est présentée Figure 25. Elle est constituée d'une FIFO asynchrone (qui sert également à la configuration du FPGA), d'un réseau d'interrupteurs CMOS et d'un décodeur d'adresse symétrique. Contrairement à la LUT réalisée avec multiplexeurs, il est ainsi possible d'obtenir des temps de propagation et des consommations pratiquement indépendants des données.

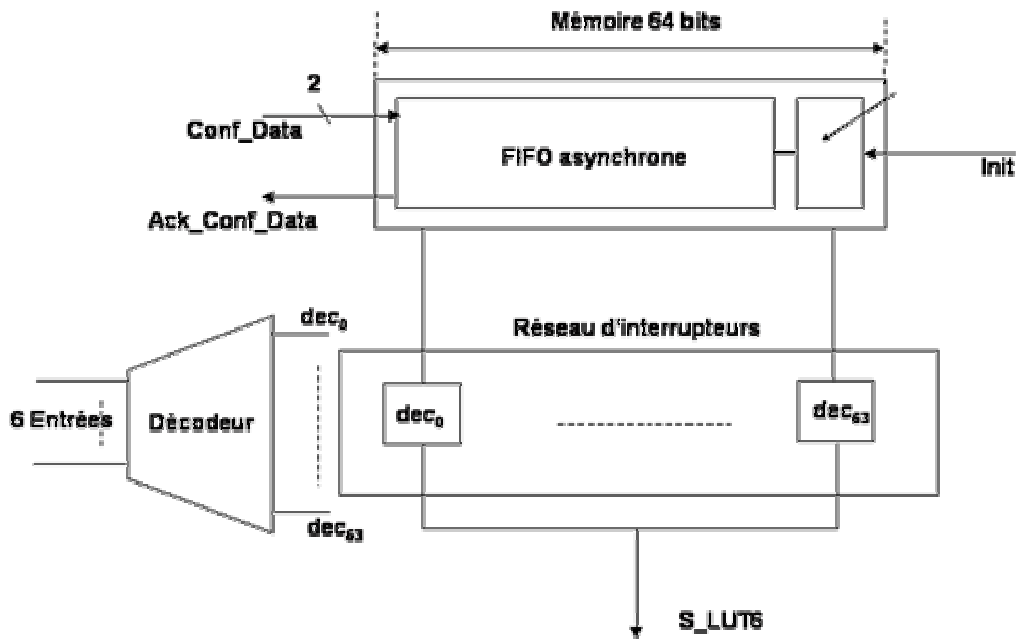


Figure 25 : Architecture d'une LUT6-1 sécurisée

II.6 Conclusion

Nous avons vu que l'approche asynchrone offre de nombreuses perspectives en terme de conception de circuits. Elle offre notamment des potentiels dans les domaines suivants :

- la maîtrise de la consommation électrique en désactivant les zones du circuit n'ayant pas de données à traiter,
- le rayonnement électromagnétique en assurant une meilleure répartition temporelle de l'activité du circuit et en supprimant les pics d'activité dus à l'horloge,
- la réduction des fuites d'information dans les systèmes sécurisés en maîtrisant les consommations dépendantes des données,
- la construction de systèmes complexes en profitant d'un assemblage de type *LEGO* (modularité) pour peu que les protocoles de communication soient identiques,
- la synchronisation et l'arbitrage de signaux en offrant des modes de fonctionnements sûrs,
- la minimisation de l'impact des variations des procédés de fabrication sur la fonctionnalité même du circuit du fait de leur insensibilité aux délais.

Cette liste n'est probablement exhaustive mais montre que les niches susceptibles d'exploiter la technologie asynchrone sont nombreuses. Ainsi il devient possible d'imaginer des systèmes complexes intégrant des réseaux de communication asynchrones, des zones reconfigurables, des dispositifs de gestion de l'énergie que l'on peut assembler à la manière d'un *Mécano* géant. En effet, les difficultés d'intégration liées à l'horloge sont ainsi relâchées, la correction fonctionnelle étant assurée par construction car le fonctionnement global du système n'est plus soumis à une conception pire cas (détermination du chemin critique de la puce entière). Ainsi l'approche asynchrone constitue une réponse possible aux problèmes de conception de systèmes complexes et de variabilité des paramètres de fabrication d'une puce, en relâchant les contraintes de communication des signaux entre des points distants.

III. Flot de conception des circuits asynchrones

Les circuits asynchrones possèdent un grand nombre d'atouts tels que faible consommation, faible rayonnement électromagnétique, calcul en temps moyen, modularité (pas d'horloge), robustesse, facilité de migration technologique, ... Cependant, ils souffrent d'une lacune importante : il n'existe pas aujourd'hui d'outils de CAO suffisamment matures pour permettre une large diffusion de ces techniques, même si l'on commence à voir apparaître des flots commerciaux. Il existe par exemple des flots capables de produire des circuits micropipelines tel que TiDE de Handshake Solutions ou des flots traduisant une synthèse synchrone sous la forme d'un circuit asynchrone tel que les outils de Theseus Logic. Le problème à mon sens est que ces produits n'offrent pas la quintessence de la logique asynchrone et que les circuits produits avec ces outils sont soit trop proches de leurs homologues synchrones (trop d'hypothèses temporelles), soit pas assez agressifs en termes de performances. De plus, les bonnes pratiques pour un design synchrone ne sont pas forcément celles correspondant à un design asynchrone. Il y a également un effort de formation à produire auprès des concepteurs.

Toutes ces raisons nous amènent à penser que l'élaboration d'une méthode de conception des circuits asynchrones permettant de les spécifier à haut niveau par un langage et de les synthétiser au niveau porte est indispensable. En outre, il est à noter que les méthodes synchrones usuelles ne peuvent pas être utilisées pour principalement deux raisons :

- Les langages de description de matériel utilisés pour les circuits synchrones, tels que VHDL et Verilog, ne permettent pas de décrire aisément des communications dans la philosophie asynchrone.
- Les outils de CAO standard ne produisent pas une synthèse exempte d'aléas. Les aléas provoquent en effet un fonctionnement incorrect des circuits asynchrones.

Dans ce chapitre, les principales méthodes de conception des circuits asynchrones sont brièvement présentées. Les méthodes de spécification et de synthèse développées au laboratoire sont ensuite abordées. L'ensemble des travaux relatifs à la synthèse correspond aux activités de recherches menées avec Anh Vu Dinh Duc, Amine Rezzag et Vivian Brégier. Enfin, une dernière partie, plus personnelle et effectuée en collaboration avec Katell Morin-Allory et Dominique Borrione, traite de la surveillance en ligne des circuits intégrés et de la génération de moniteurs de surveillance asynchrones depuis un langage de spécification de propriétés.

III.1 Les différentes méthodes de conception

III.1.1 Méthodes de spécification

Les méthodes de spécification et de représentation des circuits asynchrones se scindent en deux groupes :

- les méthodes basées sur une description en graphes telle que les réseaux de Petri, les graphes d'états, les STG, ...
- les méthodes basées sur un langage de description de haut niveau tel que CHP, Tangram, Balsa, SystemC, ...

Ces deux groupes diffèrent par leurs méthodes de conception et produisent en général des implémentations de circuit différentes. De manière générale, les méthodes basées sur un langage de description de haut niveau sont expressives et adéquates pour décrire les systèmes complexes avec une structure hiérarchique et modulaire, alors que les méthodes basées sur un graphe sont préférées pour la synthèse de contrôleurs et de petits circuits. Les méthodes de spécification basées sur un langage facilitent la tâche du concepteur (en décrivant structurellement et hiérarchiquement le circuit), mais le circuit synthétisé à partir de ces

spécifications n'est pas optimal. Si les méthodes de spécification basées sur un graphe s'avèrent ardues et pénibles pour le concepteur (notamment pour des circuits complexes), le circuit synthétisé est souvent plus rapide et plus compact.

III.1.2 Spécifications basées sur des graphes

Les méthodes de spécification basées sur les graphes spécifient le comportement des circuits asynchrones avec un niveau d'abstraction faible (fréquemment au niveau signal). Les graphes utilisés pour ces approches incluent les réseaux de Petri [PET 62], les graphes de transitions de signaux (STG ou « Signal Transition Graph » en anglais) [CHU 87], les diagrammes de changements [VAR 90], les machines à états asynchrones [YUN 92][DAV 93] et les graphes d'états [MUL 59].

Les réseaux de Petri permettent une représentation graphique des événements concurrents autant que séquentiels. Cette notation est très adéquate pour spécifier le comportement des circuits asynchrones. Par conséquent, de nombreuses méthodes de spécification des comportements des circuits asynchrones utilisent ce type de réseaux [ROS 85][MOL 85][YAK 00][MOR 06]. Une autre représentation est largement répandue dans la communauté asynchrone : le graphe de transitions de signaux (STG pour Signal Transition Graph en anglais). Il existe de nombreuses méthodes de conception de circuits asynchrones qui sont basées sur ce type de graphes [CHU 87][COR 02][MYE 92][SEN 92]. Une méthode alternative exploite des machines à états asynchrones (ASM ou « Asynchronous State Machine » en anglais). L'ASM est en fait une machine à états pour laquelle chaque état peut recevoir des entrées, générer des sorties et modifier son état courant. Sa structure est similaire aux machines à états synchrones à l'exception de l'élément de stockage qui n'est pas piloté par une horloge.

Les spécifications obtenues avec ces approches sont généralement exprimées au niveau signal, ce qui rend leur usage inadéquat à la conception de circuits complexes. En effet, la spécification de circuits asynchrones même de taille faible devient rapidement ardue et sujette à des erreurs. Leur emploi sera donc plutôt réservé à la synthèse de contrôleurs ou à des éléments de circuits qui doivent être parfaitement optimisés.

III.1.3 Spécifications basées sur un langage de description

Les langages employés pour spécifier des circuits asynchrones incluent CSP [HOA 78], CHP [MAR 90], Occam [MAY 90], Tangram [BER 91], Balsa [BAR97], VHDL [ZHE 98] et Verilog [BLU 00]. Les méthodes de spécification des circuits asynchrones basées sur des langages présentent des avantages très importants. Depuis une spécification de haut niveau jusqu'à une description structurelle à grain fin, le circuit est décrit avec un unique langage. Cela garantit une continuité sémantique entre tous les niveaux de description. De plus, elles permettent de modéliser les environnements de test des circuits. elles constituent aussi des outils puissants pour faire de l'exploration architecturale et, à ce titre, elles méritent une attention particulière.

Les langages de description de matériel comme VHDL et Verilog, actuellement supportés par les outils commerciaux et largement adoptés par l'industrie, fournissent un niveau d'abstraction élevé et évitent la spécification des transitions de signaux comme c'est le cas avec les STG par exemple. Néanmoins, ils n'offrent pas le concept de canal de communication qui est particulièrement adapté à la description des circuits en général et plus particulièrement aux circuits asynchrones. Il faut donc spécifier des paquetages permettant au concepteur de définir la communication entre des processus concurrents communicants [REN 99b]. A l'opposé, les autres langages sont généralement dérivés du langage CSP. Ces langages utilisent des processus concurrents communicants par passage de messages via des canaux. Ceci offre au concepteur la facilité de décrire des blocs fonctionnels asynchrones

communiquant concurremment et séquentiellement entre eux. Cependant, même si de nombreux langages basés sur CSP sont largement utilisés pour modéliser des circuits asynchrones, il n'existe pas aujourd'hui de réel consensus sur un unique langage de spécification dédié à leur modélisation. L'université Caltech a proposé le langage CHP tandis que l'université de Manchester a développé le langage Balsa. De son côté, Philips a défini le langage Tangram, devenu Haste avec l'éclosion de la société « Handshake Solutions ». Enfin, au laboratoire TIMA, nous avons opté pour une version enrichie de CHP.

III.1.4 Tour d'horizon des méthodes de synthèse

L'automatisation de la synthèse des circuits asynchrones est généralement une tâche ardue mais sa complexité varie beaucoup en fonction de l'approche utilisée pour les spécifier. Cependant, il est possible de classer les méthodes de synthèse en trois grandes familles : synthèse logique « bas niveau », synthèse dirigée par la syntaxe et synthèse par compilation.

III.1.4.1 Synthèse logique « bas niveau »

La synthèse logique « bas niveau » exploite des spécifications décrivant le comportement des signaux du circuit. La spécification utilisée dans cette approche est donc ici basée sur des graphes. Cette méthode produit généralement des circuits efficaces et rapides. Toutefois, les méthodes de synthèse basées sur les graphes exigent souvent l'exploration complète de l'espace d'états pour trouver tous les états accessibles (C'est le cas avec les STG par exemple !). Le problème est que le nombre d'états accessibles de l'espace d'états explose rapidement quand la complexité et la taille de la spécification augmentent. De plus, comme les spécifications avec ces approches sont en général au niveau des signaux, l'écriture des spécifications est ardue et est sujette à des erreurs, surtout si la taille du circuit à concevoir est conséquente. Par ailleurs, des problèmes d'aléas existent avec certaines méthodes car certains outils de synthèse génèrent des circuits indépendants de la vitesse (Speed Independent). Des hypothèses temporelles sont donc faites sur ces circuits.

Plusieurs outils de synthèse utilisant cette approche sont aujourd'hui disponibles. Ils sont tous issus de la recherche académique. Les plus représentatifs, à mon sens, sont *Petrify* et *Minimalist*.

L'outil *Petrify* et la méthode proposée par Jordi Cortadella [COR 97][COR 02] traitent la conception de circuits de type « indépendant de la vitesse » (SI) spécifiés par un graphe de transitions de signaux (STG). La spécification STG décrit des relations de causalité entre les événements (les transitions de signaux). Pour calculer la fonction de chaque sortie du circuit, l'espace des états atteignables doit être calculé. Le nombre d'états de l'espace d'états augmente exponentiellement avec le nombre de signaux du STG. C'est le principal point faible de cette méthode qui en pratique limite le nombre de signaux à une centaine. Par ailleurs, la description d'un circuit complexe par des STG serait particulièrement ardue. Son utilisation ne peut donc se limiter à mon sens qu'à la synthèse de contrôleurs.

La méthode proposée par Steven Nowick [NOW 93] traite la synthèse de contrôleurs asynchrones fonctionnant en « burst mode ». La spécification est fournie par une machine à états asynchrone (ASM). L'outil de synthèse *Minimalist* [FUH 99] a été développé pour prototyper cette méthode. Tout comme avec les STG, il est difficile de décrire un système complexe par cette méthode mais la spécification se prête bien à la description de systèmes dont la taille est petite ou moyenne. En outre, le fonctionnement sophistiqué et complexe des circuits générés exige une vérification soigneuse des délais et de la logique des différents chemins afin de s'assurer que les hypothèses temporelles sont bien respectées.

III.1.4.2 Synthèse dirigée par la syntaxe

Les méthodes de synthèse dirigée par la syntaxe utilisent des spécifications de circuits décrites par un langage à base de processus concurrents communicants. La structure du langage permet une traduction directe du langage en blocs du circuit. L'avantage de l'approche est la simplicité de description à haut niveau de systèmes concurrents, complexes et hiérarchiques. On évite ici les problèmes d'explosion d'états en traduisant directement les structures de langage sous forme de blocs de circuits. La méthode permet en outre la synthèse de circuits possédant des comportements non déterministes comme des arbitres ou des synchroniseurs. Cependant, les circuits générés ainsi sont difficiles à optimiser car les transformations sont uniquement locales. Cela se traduit par un manque de flexibilité puisque les optimisations globales ne sont pas possibles par cette approche. De plus, il faut que le concepteur ait une très bonne connaissance de la conception de circuits asynchrones pour produire des circuits bien conçus. Enfin, il est important de noter que la synthèse dirigée par la syntaxe préserve la correction par construction, mais n'assure pas que globalement le circuit soit correct (deadlock possible).

Il existe plusieurs méthodes développées selon ce principe. Nous pouvons citer par exemple les travaux réalisés avec le langage *Occam* par Eric Brunwand [BRU 89], les travaux de Jo Ebergen [EBE 91] pour générer des circuits insensibles aux délais et ceux de Kees Van Berkel [BER 91] avec le langage *Tangram*.

Les travaux de Van Berkel ont conduit au développement de l'outil de synthèse *Tangram* (même nom que le langage !) qui a donné naissance aujourd'hui à l'outil *TiDE* (Timeless Design) de *HandShake Solutions* et au langage *Haste* [PEE 04]. La méthode de synthèse *Tangram/TiDE* utilise une structure intermédiaire basée sur des circuits de type « poignée de main » [BER 93]. Les circuits de type « poignée de main » communiquent avec les autres circuits via des canaux en utilisant un protocole de communication et des données groupées (bundle data). Le back-end du flot de conception implique de disposer d'une bibliothèque de circuits de type « poignée de main » que le compilateur cible de même que l'analyseur de performance et le simulateur fonctionnel. Les bibliothèques développées permettent des implémentations utilisant différents protocoles (4 phases données groupées, 4 phases double-rails, ...) et différentes technologies cibles (cellules standard CMOS, FPGA).

Avec une approche similaire, l'outil *Balsa* a vu le jour à l'Université de Manchester [BAR97][BAR 00]. Tout comme *Tangram*, il exploite un format intermédiaire *Breeze* qui définit le réseau de circuits « poignée de main ». *Balsa* intègre également un simulateur fonctionnel, la simulation post-synthèse étant réalisée avec des outils commerciaux. Bien que le succès de cette méthode de conception ait été illustré par la conception du microprocesseur *Amulet3i*, il apparaît que, comme les autres méthodes basées sur la synthèse dirigée par la syntaxe, l'implémentation des circuits générée n'est pas suffisamment agressive.

III.1.4.3 Synthèse par compilation

L'approche proposée par Alain Martin [MAR 90] traduit une spécification, décrite en langage CHP, en circuit par une série de transformations préservant la sémantique du programme. Cette méthode est reconnue dans la communauté asynchrone comme étant l'une des voies de développement asynchrone les plus probantes. Basée également sur des processus concurrents communicants et le langage CSP [HOA 78], la modélisation CHP garantit du début à la fin du développement (approche top-down) la préservation des clauses d'insensibilité aux délais imposées par le modèle : les processus dialoguent entre eux sans jamais faire d'hypothèse concernant la propagation des signaux le long des canaux. Le langage CHP a été principalement développé pour décrire un circuit correct (en termes d'hypothèses temporelles autant que de fonction implémentée) et préserver cette correction tout au long des transformations appliquées. Cette approche a également été retenue par notre

équipe [DIN 02b]. Le principal obstacle à l'adoption de cette méthode, qui présente beaucoup d'avantages, est la difficulté de mettre en œuvre son automatiser.

Le principe de la méthode repose sur la modélisation de chaque processus en termes de communication : au plus haut niveau, le protocole est implicite, seules les actions de communication apparaissent. Le raffinement successif du code permet de faire intervenir différents types de protocoles, de codages et de conventions (choix du processus actif sur le canal, séquençement des communications) pour aboutir finalement à une description explicite des signaux. Sophistiqués et complexes, les étapes de raffinement préservent la sémantique. Les étapes suivies par le flot de synthèse (cf. Figure 26) développé par l'équipe du Pr. Alain Martin à Caltech sont les suivantes :

- **Décomposition des processus** : chaque processus est itérativement affiné en une collection de processus interactifs simples.
- **Expansion des communications** (HSE ou Handshake Expansion) : chaque canal de communication est remplacé par les fils explicites conformément au codage choisi.
- **Expansion des règles de production** : une règle de production (PR ou Production Rule) se compose d'une condition (garde) et d'une action : l'action est exécutée quand la condition est vraie. Dans cette étape, chaque expansion de communication est remplacée par un ensemble de règles de production. A titre d'exemple, $Ereq \wedge Sacq \vdash Sreq \uparrow$ et $\neg Ereq \wedge \neg Sacq \vdash Sreq \downarrow$.
- **Réduction des opérateurs** : c'est l'étape dans laquelle les règles de production sont regroupées et associées pour former des composants matériels comme les C-éléments généralisés. Par exemple, les deux règles de production données ci-dessus peuvent être rassemblées pour former une porte de Muller à deux entrées.

Plusieurs circuits ont été conçus avec succès par cette méthode à Caltech. Les circuits générés sont QDI avec un protocole de communication 4 phases. Toutefois, l'approche comporte encore des étapes manuelles et l'optimisation des circuits n'est pas à notre sens optimale. En effet, elle génère des circuits plus optimisés que ceux générés par les méthodes de synthèse dirigée par la syntaxe car des optimisations sont réalisées lors de la génération des portes.

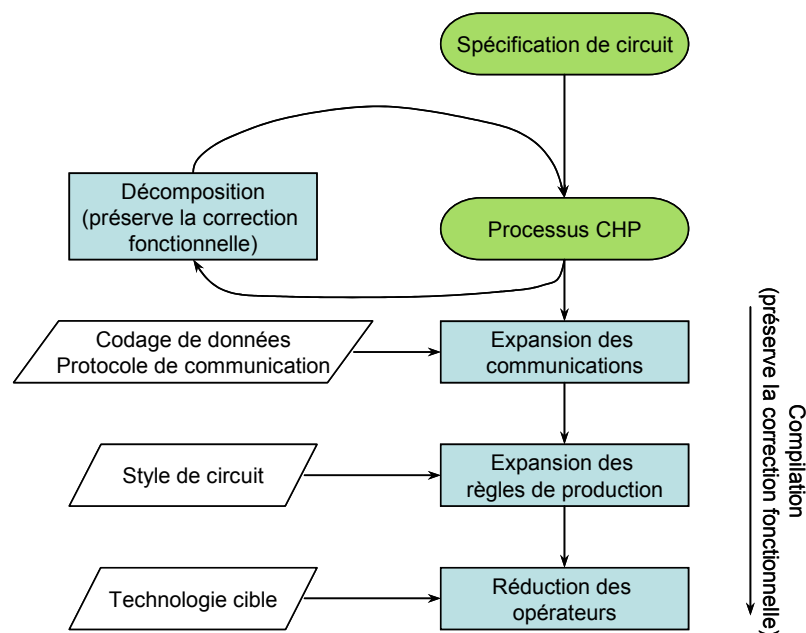


Figure 26 : Flot de conception de la méthode de Caltech

A TIMA, nous avons adopté une approche similaire mais avec un leitmotiv constant depuis de plusieurs années : l'automatisation complète et l'optimisation du processus de

synthèse. Les flots de conception que nous avons adoptés sont schématisés sur la Figure 27. En effet, trois méthodes différentes ont été développées au laboratoire. Elles ont donné lieu à trois thèses : deux ciblant les circuits QDI [DIN 03][BRE 07] et une ciblant les circuits micropipeline [REZ 04]. Les méthodes sont présentées plus en détails dans la suite de la thèse.

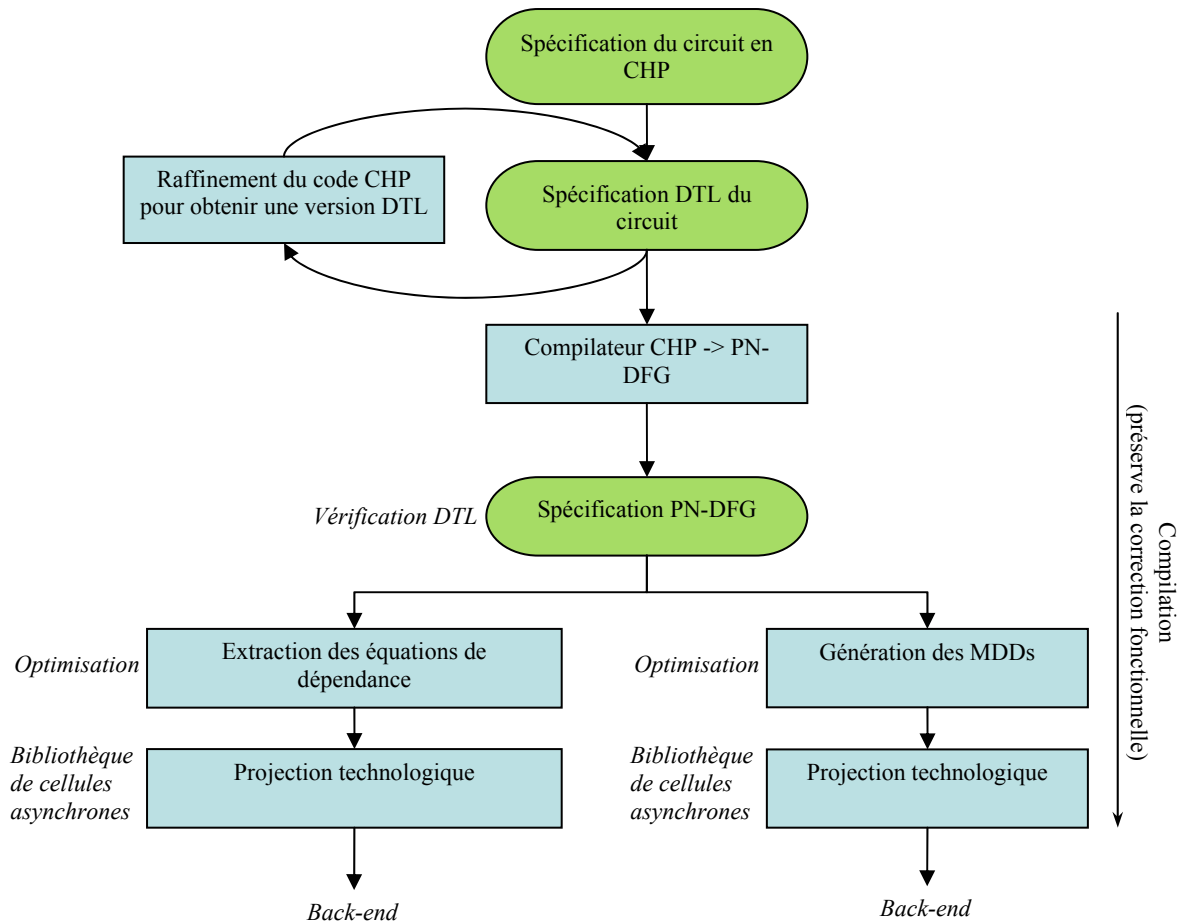


Figure 27 : Flot de conception TAST

III.1.4.4 Autres techniques de synthèse

Afin d'être complet sans être exhaustif, il est à noter qu'il existe des méthodes qui permettent de partir d'une description ou d'une netlist synchrone et de la transformer en circuit asynchrone. C'est notamment le cas des méthodes d'asynchronisation comme celle proposée par Linder [LIN 96], ou l'approche proposée par la société Theseus Logic. La logique propriétaire [FAN 96] brevetée par Thesus Logic, se base sur un codage trois états appelé NCL3, qui est comparable à une représentation dual-rail. En NCL3, chaque fil peut avoir trois valeurs possibles : 0, 1 et Null. La sortie des opérateurs logiques est Null si une entrée est à Null. Dans les autres cas, la sortie est comparable à celle que l'on obtient avec les opérateurs booléens.

La philosophie de la méthode NCL est d'exploiter au maximum les outils existants, qui ont été développés pour synthétiser des circuits synchrones. Ainsi, le codage à 3 états permet d'utiliser les circuits combinatoires synchrones, en remplaçant les portes logiques booléennes par les portes logiques NCL3 équivalentes. Le circuit NCL3 obtenu est sans aléa. Seul le chemin de données est synthétisé ainsi. La gestion des acquittements et de la validité des sorties étant faite manuellement. La synthèse est réalisée en deux étapes :

- Le circuit est tout d'abord synthétisé à partir d'une description VHDL par un outil de synthèse commercial (Synopsys), en n'utilisant que des portes en logique NCL3.
- Le circuit NCL3 est ensuite transformé en un circuit logique booléen. Chaque fil NCL3 est remplacé par deux fils booléens, en codage 1-parmi-2, et chaque porte NCL3 est mappée sur une bibliothèque de portes à seuil (Mullers généralisées).

Le principal intérêt de cette méthode réside dans l'utilisation d'outils existants ainsi qu'une spécification en VHDL. L'idée est ici de réutiliser ce qui existe déjà afin de faciliter le passage aux technologies asynchrones pour les concepteurs. Contrairement à *Haste*, *Balsa* et *CSP*, la spécification n'est pas basée sur la description des communications. Le principe de la synthèse exploite une « forme de communication simplifiée » dans laquelle le concepteur a pour charge d'implémenter manuellement les acquittements. Cela constitue un manque de flexibilité et de robustesse vis-à-vis de la spécification haut niveau. En effet, il n'est nulle part possible de vérifier si la génération des signaux d'acquiescement est correcte ou qu'elle réalise le schéma de communication souhaité. De plus, on réalise manuellement un protocole dont la description serait relativement aisée avec un modèle haut niveau.

III.2 Modélisation des circuits asynchrones en CHP

Les circuits asynchrones sont constitués de blocs communicant entre eux par des canaux. Une approche intégrant explicitement le concept de canal de communication est donc particulièrement appropriée. Le formalisme choisi pour décrire les circuits asynchrones est basé sur la notion de processus concurrents communicant entre eux par des passages explicites de messages via des canaux et des assignations explicites de variables. Le formalisme le plus proche de ces exigences était le langage CSP (« Communicating Sequential Processes »), conçu et spécifié par Hoare [HOA 78]. Il est donc tout naturel que la communauté asynchrone se soit tournée vers des langages similaires à CSP.

III.2.1 Le langage CHP

CHP est le langage introduit par Alain Martin [MAR 90], inspiré de CSP, mais aussi des commandes gardées de Dijkstra [DIJ 76]. Le langage CHP apporte quelques extensions mais aussi quelques limitations pour s'adapter à la spécification des circuits asynchrones. Les restrictions du CHP d'Alain Martin sont liées à l'implémentation physique des VLSI. Ce langage est pourtant loin de fournir toutes les facilités d'un langage de haut niveau comme C ou VHDL, notamment au niveau des types et des opérateurs arithmétiques. Ainsi, il existe un unique type de donnée : le booléen. Les autres types sont créés par l'association de variables booléennes et l'usage d'opérateurs sur ces types (non booléens) fait implicitement référence à des fonctions qui traitent des booléens ! La description de systèmes complexes à l'aide de ce langage, comparativement à l'utilisation de VHDL ou Verilog, est donc plus difficile.

La spécification en CHP d'un composant matériel est constituée d'une interface de communication (les ports d'entrées/sorties), d'une partie déclarative et de processus concurrents. La description de chaque processus se décompose en trois parties : l'interface de communication (liste des ports), une section déclarative pour les variables locales et les instructions. Une instruction peut être une affectation de variable, une structure de choix, une opération logique, une lecture ou une écriture dans un canal de communication. Ainsi, un processus apparaît comme une composition séquentielle et parallèle d'instructions comportant des structures de choix avec des commandes gardées. En guise d'illustration, l'exemple donné ci-dessous décrit une ALU.

```
process alu
port ( Op      : in  DI mr[4];
      A       : in  DI bit ;
```



```

        B      : in  DI bit ;
        Cout   : out DI bit ;
        S      : out DI bit
    )
variable i : mr[4];
[
Op?i; @[ i='0' => A?a, B?b; S!a and b; break
        i='1' => A?a, B?b; S!a or b; break
        i='2' => A?a, B?b; S!a xor b; break
        i='3' => A?a, B?b;
            @[ a='0' and b='0' => Cout!'0', [Cin?c; S!c]; break
              a='1' xor b='1' => Cin?c;
                @[ c='0' => Cout!'0', S!'1'; break
                  c='1' => Cout!'1', S!'0'; break
                ]; break
              a='1' and b='1' => Cout!'1', [Cin?c; S!not c]; break
            ]; break
]; loop
]

```

III.2.2 Les extensions de CHP

Les extensions apportées à CHP visent à faciliter la tâche du concepteur et à le rendre plus puissant et plus lisible pour la simulation et la synthèse. La syntaxe choisie s'inspire des langages de description de matériel : elle est simple et régulière. Les principales améliorations apportées au langage concernent la modélisation hiérarchique, l'arithmétique, l'instrumentation et la modélisation du non déterminisme.

- Les processus étant déclarés à « plat » dans la proposition initiale de CHP, la structuration hiérarchique des processus a été introduite afin de gérer efficacement la complexité des VLSI. La hiérarchie s'applique aux processus et aux composants, la connectivité entre blocs reposant uniquement sur des canaux ; il n'est pas possible de déclarer des signaux.
- La possibilité d'effectuer des calculs arithmétiques avec les opérandes de type signé et non signé en multi-rails (codage en 1 parmi N) a été ajouté.
- Dans le but de faciliter l'analyse des circuits et systèmes complexes, le langage CHP a été instrumenté. Il est notamment possible de déterminer l'activité et la consommation des circuits asynchrones, l'observation se faisant au niveau d'instructions labellisées, des processus, des composants et du circuit complet.
- Le CHP étendu permet également de spécifier les comportements non déterministes.

Il est à noter que ce dernier point est extrêmement important pour spécifier le comportement des circuits de synchronisation et d'arbitrage que l'on rencontre par exemple dans les réseaux de communication sur puce (NoC). Concrètement, le CHP propose deux structures de contrôle : le choix et la répétition qui peuvent être spécifiés comme non déterministes ou non. Les structures CHP possibles sont donc le choix déterministe (1) et la répétition déterministe (2), le choix non déterministe (3) et la répétition non déterministe (4). Leur syntaxe est la suivante :

```

(1)  @[G1 => S1 ; break ...           Gn => Sn; break]
(2)  @[G1 => S1 ; loop ...            Gn => Sn ; loop]
(3)  @@[G1 => S1 ; break ...          Gn => Sn ; break]
(4)  @@[G1 => S1 ; loop ...           Gn => Sn ; loop]

```

Dans un choix déterministe (1), au plus une garde est vraie. L'exécution est suspendue jusqu'à ce qu'une garde soit vraie. La commande S, correspondant à la garde qui est vraie, est exécutée et le choix se termine. Dans une répétition déterministe (2) (noté par « loop »), au plus une garde est vraie. La répétition continue à exécuter les commandes associées à la garde qui est vraie. Quand aucune garde n'est vraie, la répétition se termine. Dans le cas où

l'exclusion mutuelle ou la stabilité des gardes ne peuvent être garanties, le choix (3) et la répétition non déterministes (4) sont utilisés. Cela se traduit au niveau circuit par un problème d'arbitrage. Dans ce cas plusieurs gardes peuvent être vraies en même temps, mais une seule, arbitrairement choisie est sélectionnée et la commande correspondante est exécutée.

III.2.3 DTL : une forme synthétisable de CHP

De même qu'il existe des règles dites RTL (« Register Transfer Level ») pour la synthèse des circuits synchrones, dans le cadre des travaux de thèse d'Anh Vu Dinh Duc nous avons défini des règles d'écriture, dites DTL (« Data Transfer Level »), pour la conception de circuits asynchrones [DIN 02]. Les descriptions RTL spécifient les transferts de données entre registres. De même, les programmes DTL indiquent comment les données sont transférées entre processus. Cette spécification DTL résulte de l'analyse des formes synthétisables de CHP en circuits asynchrones. On peut donc considérer les règles DTL comme une spécification synthétisable de circuits asynchrones. Elles comprennent des restrictions dans l'écriture des programmes décrivant des circuits asynchrones et fixent le style de description des circuits asynchrones en vue de la synthèse. Un circuit asynchrone, qui remplit les règles de la spécification DTL, est automatiquement synthétisable. Si la spécification d'un circuit asynchrone n'est pas conforme aux règles DTL, elle peut être transformée par décomposition en un ensemble de processus conformes à la spécification DTL et donc synthétisables. La décomposition considère deux types de processus : les processus ayant une ou des variables de mémorisation et les processus ayant un ou des opérateurs séquentiels. Il s'agit en fait de supprimer de la description les variables mémorisantes des processus ou les mémoires résultants d'opérateurs séquentiels en les remplaçant par un processus explicitant la construction d'un registre. Cette décomposition facilite la synthèse car elle évite de transformer les opérateurs séquentiels en machine à états. Ainsi, il est possible de garantir la synthèse de tout programme CHP à condition de lui appliquer les règles de transformation DTL.

III.3 La synthèse des circuits QDI

III.3.1 Le format intermédiaire PN-DFG

Les premiers travaux sur la synthèse des circuits QDI ont été menés dans le cadre de la thèse d'Anh Vu Dinh Duc [DIN 03]. Une des premières tâches a été de déterminer un format intermédiaire qui serait la structure pivot de notre technique de synthèse mais qui puisse également supporter d'autres fonctions comme la simulation ou la vérification.

La méthode de synthèse proposée opère à partir d'une description comportementale. Cette dernière est exprimée sous une forme intermédiaire qui sert à exprimer l'ensemble des opérations manipulant des données et l'enchaînement ces opérations. Ceci nous a conduit à discerner :

- la partie opérative : un graphe de flot de données (DFG) permet de la représenter. Cette représentation donne les dépendances fonctionnelles entre les données et les opérations et autorise d'éventuelles optimisations.
- la partie contrôle : elle est chargée de piloter l'unité d'exécution (c'est-à-dire la sélection des données à utiliser, des commandes et le contrôle de la validité des données). Une description de type réseau de Petri permet de caractériser efficacement son comportement.

Cette représentation intermédiaire permet de séparer le front-end et le back-end de l'outil. L'avantage de cette approche est qu'elle garantit une indépendance entre les

différentes parties de l'outil. On peut ainsi facilement ajouter un compilateur pour un autre langage de description dans le front-end ou un logiciel prototypant un autre style d'implémentation de circuits dans le back-end en passant par cette forme intermédiaire.

III.3.2 Transformations et vérifications du réseau de Petri

Afin de s'assurer que la description est synthétisable, il est nécessaire de se livrer à des vérifications qui garantiront que la description est conforme au modèle DTL. Ces vérifications sont faites directement sur le réseau de Petri. En outre, pour parvenir à la synthèse automatique depuis une forme intermédiaire PN-DFG, il est également nécessaire d'abaisser le niveau d'abstraction du réseau de Petri afin de faciliter sa synthèse. Pour ce faire, il est utile de procéder à l'expansion des données et des structures de choix.

III.3.2.1 Vérification DTL

Comme nous l'avons vu au paragraphe III.2.3, les règles DTL définissent une forme synthétisable pour les circuits asynchrones. La vérification DTL a alors pour but de garantir que les circuits asynchrones modélisés par des réseaux de Petri peuvent être synthétisés. La vérification des règles DTL s'effectue sur le réseau de Petri et assure que :

- les valeurs envoyées sur les canaux de sortie sont des fonctions combinatoires des valeurs lues dans les canaux d'entrée,
- il n'y a pas d'état implicite, sauf l'état nécessaire à l'initialisation des canaux,
- il existe une implémentation QDI en portes standard du circuit.

A cela, il faut ajouter les vérifications suivantes :

- vérification des opérateurs non synthétisables tels que la division, le modulo, ...
- détection des segments inatteignables du réseau de Petri. Par exemple, une garde ayant la condition toujours « faux » implique que le segment du réseau qui suit la transition est inatteignable,
- vérification de l'exclusion mutuelle des conditions des gardes. Les gardes dans l'opérateur de sélection/répétition déterministe doivent être mutuellement exclusives. Cela signifie qu'il n'y a pas de valeurs qui rendent deux conditions vraies.

A la fin de ces vérifications, le réseau de Petri obtenu est synthétisable et il existe a priori une implémentation QDI.

III.3.2.2 Expansion des structures de choix

Dans la génération des équations de dépendances qui seront décrites au paragraphe III.3.4, l'expression logique de chaque canal de sortie doit être produite. En général, un réseau de Petri d'un circuit asynchrone a plusieurs branches de choix dans lesquelles un même canal de sortie est affecté. La fonction de sortie pour ce canal est une réunion exclusive des fonctions de sortie des différentes branches. Pour faciliter le calcul de ces fonctions, le réseau de Petri est déployé en plusieurs réseaux de Petri. Les nouveaux réseaux de Petri n'ont plus de branche de choix. En effet, ils correspondent à une branche d'exécution du programme initial (et donc à un choix particulier). Par conséquent, cette transformation permet de calculer une fonction de sortie pour chaque branche d'exécution donnée. L'expression finale pour chaque canal est obtenue par l'union exclusive des expressions de chacune des branches du choix.

III.3.2.3 Expansion des types de données

Les données manipulées par le langage CHP se présentent le plus souvent sous la forme d'un vecteur de digits de type multi-rails ($MR[B][L]$ est un vecteur de L digits de base B).

Cette deuxième transformation consiste mettre à plat sous forme de digits individuels les canaux de communication et les variables locales dans toutes les expressions des DFG. Cette forme permet de générer facilement les équations de dépendances pour chaque digit des canaux de sortie.

III.3.3 Structure du circuit cible QDI

Dans notre approche, un circuit asynchrone est toujours implémenté par un bloc de calcul et un bloc de contrôle (cf. Figure 28). Le bloc de calcul, qui est complètement combinatoire, prend en charge le calcul des sorties et des états suivants. Le bloc de contrôle, chargé de la synchronisation entre le circuit et les autres modules du système, est appelé buffer. Le contrôle implémente un protocole de communication de type « poignée de main » (séquentiel, WCHB, PCHB, PCFB) avec les modules auxquels ce circuit est connecté. Du point de vue de l'architecture, un circuit asynchrone est clairement séparé en deux parties. Ces deux parties sont mises en cascade. Il est à noter que le buffer peut-être placé indifféremment en entrée ou en sortie.

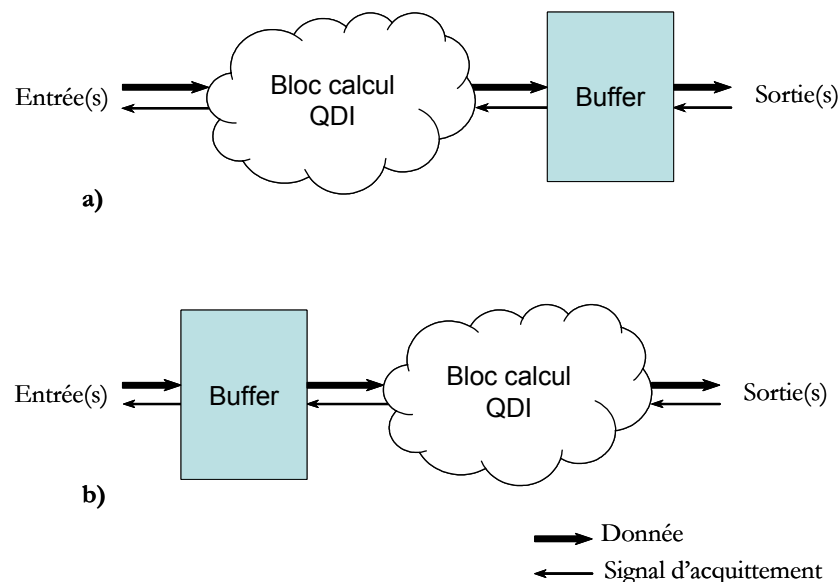


Figure 28 : Structure d'un circuit QDI

III.3.4 Méthode 1 : Synthèse QDI avec des équations de dépendances

L'équation de dépendances est une équation conventionnelle utilisée pour formaliser le fonctionnement d'un circuit asynchrone et qui donne les relations entre les sorties et les entrées du circuit. Ainsi, les équations de dépendances d'un circuit asynchrone expriment la sémantique des canaux de communication. Elles contiennent les informations qui font changer les sorties du circuit. Ces paramètres incluent :

- la validité des données sur les canaux d'entrée dont dépend la sortie,
- les conditions qui « gardent » l'émission de la donnée sur la sortie,
- la disponibilité du canal de sortie,
- et la fonction de calcul du canal de sortie en fonction des canaux d'entrée.

Pour les circuits asynchrones encodés avec un codage insensible aux délais, il existe deux types de sortie : les données des canaux de sortie et les acquittements des canaux d'entrée. De ce fait, il y a deux types d'équations de dépendances : l'un pour les rails de données des canaux de sortie et l'autre pour les signaux d'acquittement des canaux d'entrée.

III.3.4.1 Extraction des équations de dépendance

Nous présentons dans ce paragraphe la génération des équations de dépendances à partir des réseaux de Petri qui sont conformes à la spécification DTL [DIN 02]. Les circuits asynchrones, représentés par les réseaux de Petri conformes aux règles DTL, se trouvent sous la forme « consommation; production ». Cette structure est équivalente au programme CHP « E?; S! » dans lequel il y a d'abord consommation des canaux d'entrée en parallèle, puis production des canaux de sortie. Les sorties du circuit sont uniquement calculées en fonction des entrées courantes.

Comme explicité au paragraphe III.3.2, le programme décrivant un circuit asynchrone est compilé en un ensemble de réseaux de Petri. Chaque réseau de Petri correspond à une seule branche d'exécution (i.e. un choix) du programme original. Les équations de dépendance de chaque branche du réseau de Petri sont d'abord générées. Les équations de dépendances des canaux sont ensuite obtenues en réunissant exclusivement les équations issues de chacune des branches. Cette opération est illustrée par l'Équation 1 avec S_c le canal de sortie correspondant au choix c et N le nombre de choix dans le PN-DFG. E_c/acq symbolise le signal d'acquiescement du canal E qui correspond au choix c .

$$\text{Équation 1} \quad \begin{cases} E^{acq} = F_{union}^{acq} \left(\sum_{c=0}^{N-1} E^{c/acq} \right) \\ S = F_{union}^{données} \left(\sum_{c=0}^{N-1} S^c \right) \end{cases}$$

Les équations de dépendances du circuit asynchrone, représentées par l'ensemble de réseaux de Petri, sont ainsi générées.

III.3.4.2 Optimisation des équations de dépendances

A priori les équations de dépendances générées peuvent posséder des redondances. Lors de la génération des équations de dépendances, les tests de validité des canaux, garantissant la présence de données sur les entrées, sont systématiquement générés sans prendre en compte les relations entre les canaux testés, les expressions des gardes et les fonctions de calcul des sorties. Ceci a pour objectif d'assurer la correction des équations de dépendances et certains tests de validité sont redondants. Par conséquent, ces tests peuvent être supprimés.

Les redondances concernent aussi les protocoles de communication. Certains protocoles de communication comme le protocole séquentiel ou WCHB (Weak Condition Half Buffer) sont propices à des optimisations car la génération des équations de dépendances des signaux d'acquiescement dépend uniquement du test des canaux de sortie.

III.3.4.3 Synthèse

Ce paragraphe présente succinctement la génération de circuits en réseau de cellules génériques. Dans ce paragraphe, les portes génériques utilisées pour la synthèse sont les portes élémentaires AND, OR, INV plus les portes de Muller (symétriques et dissymétriques, avec ou sans signal de reset) [RIG 02]. Ces portes sont fonctionnellement définies de telle sorte qu'à ce stade il n'y a aucune contrainte électrique (*fan-in*, *fan-out*). Les portes de *fan-in* élevé sont décomposées ultérieurement. Durant cette phase, le choix des protocoles de communication (séquentiel, WCHB, PCHB, PCFB) influe sur le modèle de circuit qui est véritablement ciblé. Ainsi, l'approche offre la possibilité de générer plusieurs circuits fonctionnellement équivalents mais dont le protocole de communication utilisé est différent. Pour illustrer la génération de la *netlist* de portes génériques, un petit exemple exploitant un

protocole WCHB est donné. La description du circuit est donnée par le programme CHP (épuré) suivant :

```
[I ? instr ;
  @[ instr = 0 =>    A ? x , B ? y ; S ! x or y ; break
    instr = 1 =>    A ? x ; S ! x ; break
    instr = 2 =>    A ? x ;
                  @[ x = 0 => B ? y ; S ! y ; break
                    x = 1 => B ? y ; S ! x ; break
                  ] ; break
  ] ; loop
]
```

Dans cette description, en fonction de l'instruction (la valeur du canal I), les actions de communication correspondantes sont exécutées. De façon plus concrète, la première instruction correspond au calcul de l'opération « OR » sur les deux canaux d'entrée tandis que la deuxième consiste à envoyer la donnée issue du canal A vers la sortie. La troisième instruction est en fait une autre façon d'écrire la première instruction, mais permet d'illustrer la construction de gardes imbriquées.

Si on choisit le protocole de communication WCHB, les équations de dépendances optimisées de ce circuit sont les suivantes :

$$\text{Équation 2} \quad \left\{ \begin{array}{l} S^0 = F_{WCHB}^{données}(I = 0; A \text{ or } B; S^{acq}) \\ S^1 = F_{WCHB}^{données}(I = 1; A; S^{acq}) \\ S^2 = F_{WCHB}^{données}(I = 2, A = 0; B; S^{acq}) \\ S^3 = F_{WCHB}^{données}(V(B); I = 2, A = 1; A; S^{acq}) \\ S = F_{union}^{données}(S^0, S^1, S^2, S^3) \end{array} \right.$$

$$\text{Équation 3} \quad \left\{ \begin{array}{l} I^{0/acq} = A^{0/acq} = B^{0/acq} = F_{WCHB}^{acq}(V(S^0)) \\ I^{1/acq} = A^{1/acq} = F_{WCHB}^{acq}(V(S^1)) \\ I^{2/acq} = A^{2/acq} = B^{2/acq} = F_{WCHB}^{acq}(V(S^2)) \\ I^{3/acq} = A^{3/acq} = B^{3/acq} = F_{WCHB}^{acq}(V(S^3)) \\ A^{acq} = I^{acq} = F_{union}^{acq}(I^{0/acq}, I^{1/acq}, I^{2/acq}, I^{3/acq}) \\ B^{acq} = F_{union}^{acq}(I^{0/acq}, I^{2/acq}, I^{3/acq}) \end{array} \right.$$

Le résultat de la synthèse est donné sur la Figure 29. Sur le schéma, apparaissent les quatre choix correspondant respectivement aux chemins : « I=0 », « I=1 », « I=2 et A=0 » et « I=2 et A=1 ». Les buffers WCHB placés à la sortie de chaque branche, permettent d'acquitter les canaux d'entrée consommés. Les canaux I et A, utilisés dans les quatre branches, sont exclusivement acquittés par un des quatre signaux d'acquiescement des buffers (les choix sont mutuellement exclusifs). Par conséquent, leurs acquiescements sont réunis par une porte AND à 4 entrées (l'acquiescement est actif au niveau bas). Quant au canal B, comme il n'est pas consommé dans le deuxième choix (I=1), son signal d'acquiescement est seulement généré par trois signaux d'acquiescement (porte AND à 3 entrées).

Les signaux de données sont les unions des signaux générés par les 4 buffers correspondant à 4 choix. Les gardes imbriquées sont hiérarchiquement structurées. En particulier, dans la quatrième garde (« I=2 and A=1 »), le canal B est consommé mais jamais utilisé. Par conséquent, il faut s'assurer de sa validité avant d'autoriser l'écriture sur le canal de sortie. Comme nous l'avons énoncé précédemment III.3.4.2, ce test doit être implémenté de façon à garantir que le circuit généré soit toujours QDI. Il apparaît ici que des redondances

peuvent exister et qu'il sera nécessaire de procéder dans la suite à une phase d'optimisation. Enfin, lors de la décomposition des portes à fan-in élevé, il est important de bien prendre les précautions qui permettront de conserver les propriétés de quasi insensibilité aux délais (circuits QDI). En résumé, le circuit généré par la méthode de synthèse est par construction QDI. Le réseau de portes est générique à ce niveau. La netlist de portes doit maintenant subir les tâches d'optimisation et de projection technologique.

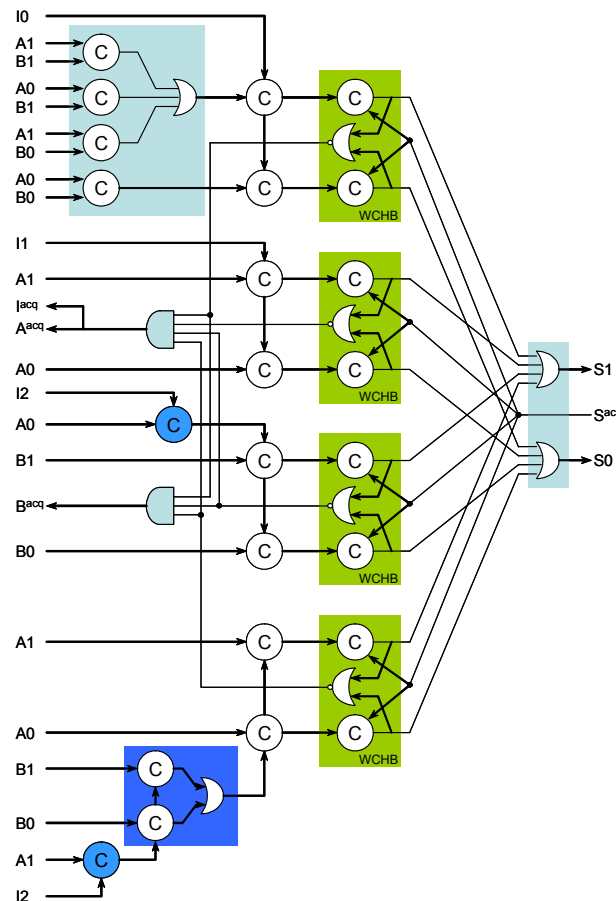


Figure 29 : Implémentation en protocole WCHB de l'exemple

III.3.4.4 Optimisation et projection technologique

Nous venons de présenter la synthèse de circuits asynchrones de type QDI en portes génériques. Cette synthèse part des équations de dépendances des circuits qui expriment les dépendances de sorties en fonction des entrées. Le protocole de communication et le modèle de circuit cible choisi sont pris en compte à ce niveau. Une des étapes les plus cruciales pour la synthèse QDI est l'optimisation et la projection technologique. En effet, il va falloir optimiser, décomposer certaines portes et les projeter sur une bibliothèque de cellules tout en préservant l'insensibilité aux délais.

A ce stade, il est possible d'exécuter des optimisations :

- Génération de signaux de validité globaux des canaux pour ne pas avoir à créer de fonctions redondantes.
- Elimination de la redondance entre les signaux de validité et les signaux de test d'une voie sur un multi-rails.
- Elimination des signaux « constants ».

- Elimination des portes ayant des entrées provenant des rails d'un même digit de canal.
- Factorisation des portes semblables lorsque deux portes ont exactement les mêmes entrées.

Ces optimisations sont répétées pour chaque itération d'optimisation afin de propager leurs effets vers les étages en aval. Dans l'outil de synthèse TAST, la décomposition de portes de *fan-in* élevé est effectuée dans les cas où la propriété d'insensibilité aux délais est garantie par l'exclusivité des signaux. En effet, la validité des règles de décomposition nécessite une analyse formelle non triviale qui a été faite par Steven Burns [BUR 96].

L'ultime étape avant le placement et le routage est la projection technologique (technology mapping). La projection technologique consiste à transformer le schéma logique optimisé en un schéma de portes existantes issues d'une bibliothèque de cellules prédéfinies. Les fonctions logiques complexes n'ayant pas d'équivalent dans cette bibliothèque, il s'agit de les décomposer avec des portes présentes dans la librairie. Indéniablement, le problème d'optimisation se pose à nouveau dans la mesure où la décomposition n'est jamais unique. Si en revanche la librairie de cellules contient des « sur-éléments » de portes (portes complexes) à implémenter, une factorisation est possible. Plusieurs bibliothèques de cellules asynchrones ont été étudiées développées au laboratoire pour mapper efficacement les circuits QDI sur des technologies ASIC mais aussi FPGA. Ces travaux sont relatés dans les thèses de Jean-Baptiste Rigaud [RIG 02] et de Bertrand Folco [FOL 07].

III.3.5 Méthode 2 : Synthèse QDI avec des graphes de données multi-valués

Une autre méthode pour réaliser la synthèse de circuits beaucoup plus optimisés que celle présentée au paragraphe III.3.4 a été étudiée et mise en œuvre dans le cadre de la thèse de Vivian Brégier [BRE 07]. Cette nouvelle approche offre en outre la possibilité de prouver formellement que le circuit généré respecte bien les propriétés de quasi insensibilité aux délais. Ce point est tout à fait crucial pour garantir que les circuits QDI générés seront fonctionnellement corrects et robustes.

III.3.5.1 Les modèles à base de MDD

Une autre approche exploitant des graphes de décision multi-valués (MDD) a été mise en œuvre dans la thèse de Vivian Brégier. La structure de MDD y est utilisée pour modéliser les circuits QDI. Cette structure présente l'avantage de mettre en exergue les propriétés de mutuelle exclusion des rails, propriété essentielle pour la construction de circuits quasi-insensibles aux délais. En fait, les MDD sont une généralisation des BDD (Binary Decision Diagram) qui sont largement répandus dans de nombreux outils de synthèse.

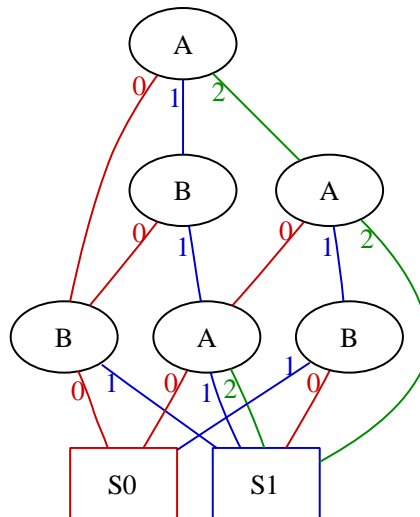


Figure 30 : Exemple de MDD

La Figure 30 montre un exemple de MDD. Un MDD spécifie, pour chaque combinaison des variables d'entrées, la valeur que la sortie doit prendre. La structure ressemble à un arbre si ce n'est que deux nœuds peuvent avoir le même fils. Pour décrire entièrement un circuit, le modèle spécifie la valeur de chaque sortie (et donc sa validité), ainsi que l'acquiescement de chacune des entrées. Le MDD est donc constitué de plusieurs composantes, qui sont utilisées conjointement, comme l'illustre la Figure 31 (ici un AND et un OR en logique ODI).

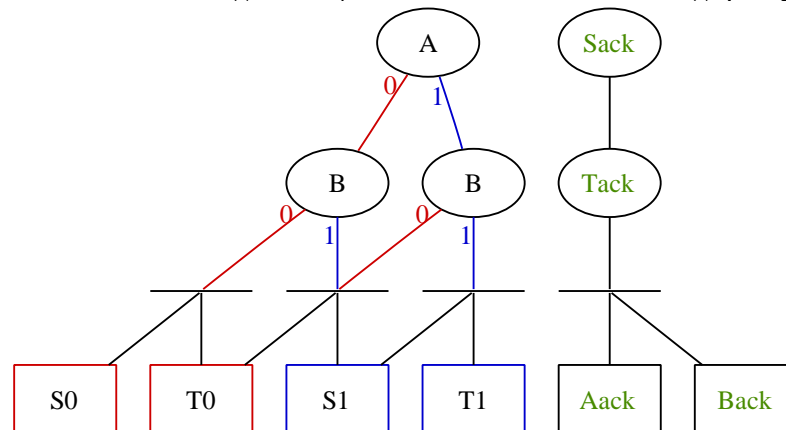


Figure 31 : MDD modélisant un circuit QDI (un AND et un OR)

On remarque que les composantes spécifiant chaque sortie et l'acquiescement des entrées ne sont pas indépendantes. De plus, il peut y avoir des expressions communes à ces composantes, qui pourront être factorisées lors des optimisations. On note aussi la présence de nœuds spéciaux, appelés fourches, qui diffusent un même signal sur plusieurs nœuds fils. Ces fourches sont nécessaires à la factorisation, c'est-à-dire à la mise en commun des branches du MDD. Par exemple, les composantes du MDD de la Figure 31, spécifiant les sorties S et T, partagent ici une partie de la structure située au-dessus des fourches.

III.3.5.2 Synthèse d'un MDD

L'intérêt et l'innovation de la méthode [BRE 04][BRE 07] résident dans le fait que les circuits synthétisés sont QDI. Il n'est donc pas nécessaire vérifier cette propriété après synthèse. En effet, toutes les étapes de transformation des circuits sont effectuées sur les MDD et il est prouvé formellement que les circuits générés depuis un MDD (optimisé ou non) sont QDI. Les explications relatives à la preuve d'insensibilité aux délais sont données au

paragraphe III.3.5.3. Afin de procéder à la synthèse du circuit, le MDD est généré à partir de la description CHP. Ensuite, le MDD subit une série d'optimisations dont le but est de supprimer les redondances en factorisant les parties communes. On réduit ainsi la taille du circuit. On y applique alors un algorithme de synthèse qui génère une netlist avec des portes à deux entrées. Cette netlist de portes à deux entrées est essentielle pour simplifier l'étape de projection technologique.

Optimisations

Après l'étape de génération des MDD, les optimisations sont directement appliquées au MDD. Cela est nécessaire pour garantir les propriétés QDI à la netlist générée mais aussi constitue une difficulté pour évaluer le résultat de l'optimisation : le circuit généré à partir du MDD transformé est-il plus petit que le circuit généré à partir du MDD initial ? Rien ne permet a priori de conclure, à moins de générer explicitement les deux. Cette difficulté peut être contournée sans générer explicitement le circuit à chaque étape. Pour ce faire, l'hypothèse simplificatrice suivante est utilisée : la taille d'un circuit est proportionnelle au nombre de nœuds du MDD. Ainsi, il est possible d'estimer l'efficacité des optimisations.

Les optimisations que nous effectuons sur les MDD sont les suivantes :

- L'algorithme de réduction permet de fusionner les nœuds du MDD qui sont fonctionnellement équivalents, c'est-à-dire les nœuds qui sont construits de manière identique à partir des terminaux. Ainsi le nombre de nœuds du circuit décroît puisque aucun nouveau nœud n'est créé et que des nœuds sont supprimés. De plus, cet algorithme est rapide, peu coûteux et optimise toujours un MDD.
- L'algorithme de factorisation permet de fusionner des nœuds similaires issus d'une même fourche. L'algorithme parcourt en profondeur le MDD. A chaque fourche, il analyse les nœuds fils et fusionne les nœuds similaires.
- L'algorithme d'unicité permet de s'assurer que sur chaque chemin du MDD, chaque variable est unique. D'après la sémantique définie sur les MDD, il n'y a aucun intérêt à étiqueter plusieurs nœuds avec une même variable sur un même chemin. En effet, pendant un cycle de calcul, une variable valide a une unique valeur.

Synthèse QDI en portes à deux entrées

Afin de garantir que le circuit conservera toujours ces propriétés d'insensibilité aux délais, l'algorithme de projection technologique ne doit pas décomposer les portes. En effet, cette opération serait locale et pourrait compromettre la nature QDI du circuit. Le problème est résolu en intégrant l'étape de décomposition des portes à la synthèse, qui est la seule façon de garantir formellement que la décomposition produit des circuits quasi insensibles aux délais. Le circuit est donc directement synthétisé sous sa forme décomposée en portes à deux entrées. Ainsi l'algorithme de projection technologique n'effectue plus que des fusions de portes. Cette approche est parfaitement sûre.

L'algorithme de synthèse du MDD en portes à deux entrées se base sur la structure de graphe du MDD, qu'il préserve : chaque nœud du graphe est synthétisé en un ensemble d'instances et chaque arc du MDD est synthétisé sous la forme d'un net. La Figure 32 montre un exemple de synthèse depuis un MDD.

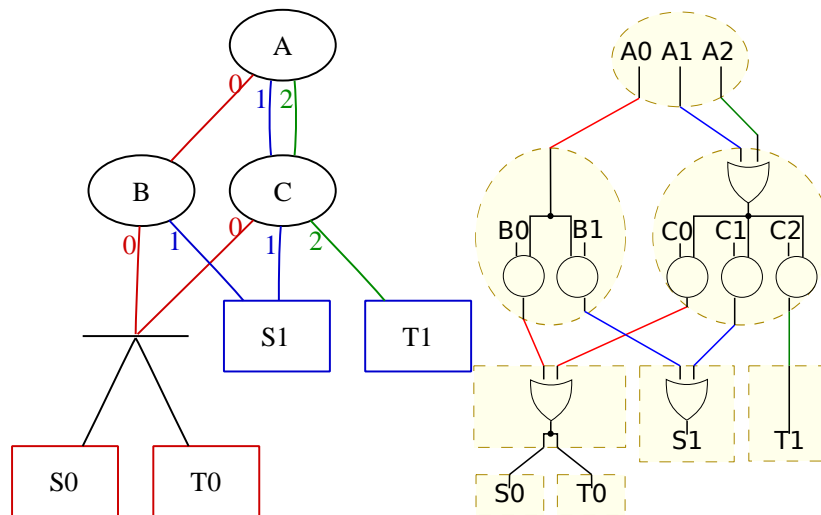


Figure 32 : le MDD (à gauche) et le résultat de la synthèse en portes à 2 entrées (à droite)

Les actions effectuées sur le MDD pour produire une synthèse peuvent se résumer comme suit :

- Les arcs entrants d'un nœud sont regroupés par un arbre de portes OR, pour obtenir un rail, appelé activation du nœud.
- Un terminal correspond à un rail d'un port primaire de la netlist.
- Une fourche correspond à une fourche dans la netlist.
- Un non-terminal est implémenté par une structure plus complexe, qui combine l'activation du nœud à chaque rail de la variable par une porte de Muller. La sortie de chaque porte de Muller correspond à l'un des arcs sortants du nœud.

Enfin il est à noter que les nœuds racine n'ont pas d'activation ; il n'y a donc pas de portes de Muller à créer, chaque rail de la variable correspond à l'un des arcs sortants.

III.3.5.3 Preuve d'insensibilités aux délais

La caractérisation formelle de la quasi insensibilité aux délais s'inspire des travaux menés par Alain Martin et Rajit Manohar à Caltech [MAN 96]. Ils ont notamment montré qu'un circuit modélisé par un ensemble de règles de production est quasi insensible aux délais si et seulement si les règles de production qui le décrivent sont stables et non interférentes (cf. la thèse de Vivian Brégier [BRE 07]). Ainsi la caractérisation de la quasi-insensibilité aux délais peut-être faite à partir des règles de production du circuit. En fait, la génération de ces règles depuis un MDD ne pose pas de problèmes particuliers puisque l'algorithme est très similaire à celui de la synthèse (les règles de production sont sensées modéliser le comportement du circuit synthétisé).

Afin de démontrer que le circuit synthétisé est quasi insensible aux délais, il est donc nécessaire d'énoncer les conditions équivalentes à celles des règles de production que le MDD doit respecter :

- La correction des acquittements

Un couple (entrée, sortie) est correctement acquitté si et seulement si l'entrée est consommée lors de la production de la sortie et que l'acquittement de l'entrée passe à 0 (resp. 1) avant que la sortie soit passée elle-même à 0 (resp. 1).

- La non-interférence

On démontre aisément que des règles de production issues d'un MDD sont non interférentes. En effet, les règles de production générées depuis le MDD sont par construction non interférentes.

- La stabilité

On démontre également, moyennant un développement un peu plus long, la stabilité des règles de production des chemins et des sorties du MDD.

Les preuves ne sont pas présentées ici pour des raisons de concision et j'invite le lecteur à se reporter aux documents [BRE 04][BRE 07] pour de plus amples détails. Néanmoins, le résultat important à retenir est que la propriété de quasi-insensibilité aux délais a pu être démontrée formellement pour tout circuit synthétisé depuis notre structure de MDD.

III.4 La synthèse des circuits micropipeline

III.4.1 Modèle de circuits cibles

Dans un esprit de concision, j'aborde ici les principes essentiels à la synthèse de circuits micropipeline. En effet, ils sont très faciles à appréhender car la partie chemin de données d'un circuit micropipeline est comparable à celle d'un circuit synchrone. Seul le circuit d'horloge est remplacé par un contrôleur QDI. Le modèle de circuit cible est donné Figure 33.

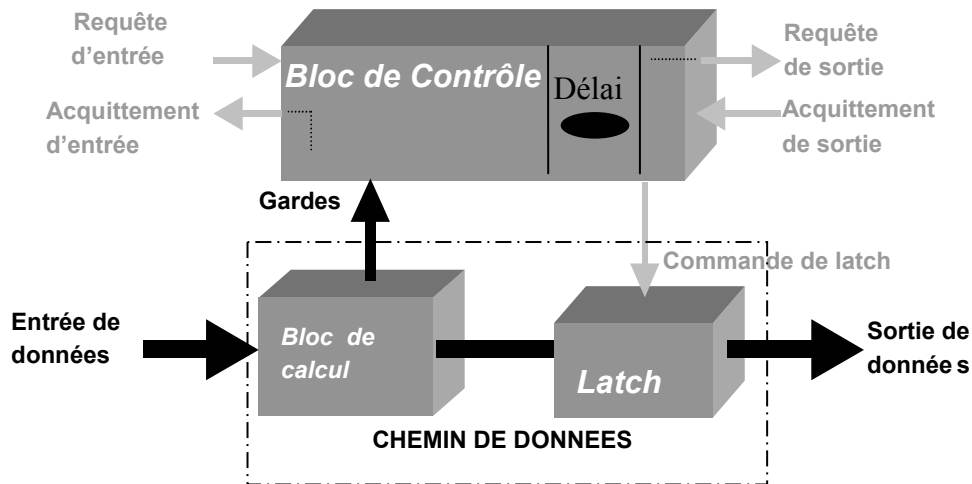


Figure 33 : modèle des circuits micropipeline

III.4.2 Flot de conception des circuits micropipeline

Les équations de dépendances définies au paragraphe III.3.4.1 constituent une forme de représentation des circuits asynchrones indépendante du protocole de communication et de la technologie cible. Elles peuvent donc être utilisées comme point de départ du flot de synthèse micropipeline. Dans l'approche proposée par Amine Rezzag dans sa thèse [REZ 04], le processus de synthèse se décompose en deux parties : synthèse du chemin de données et synthèse du contrôleur.

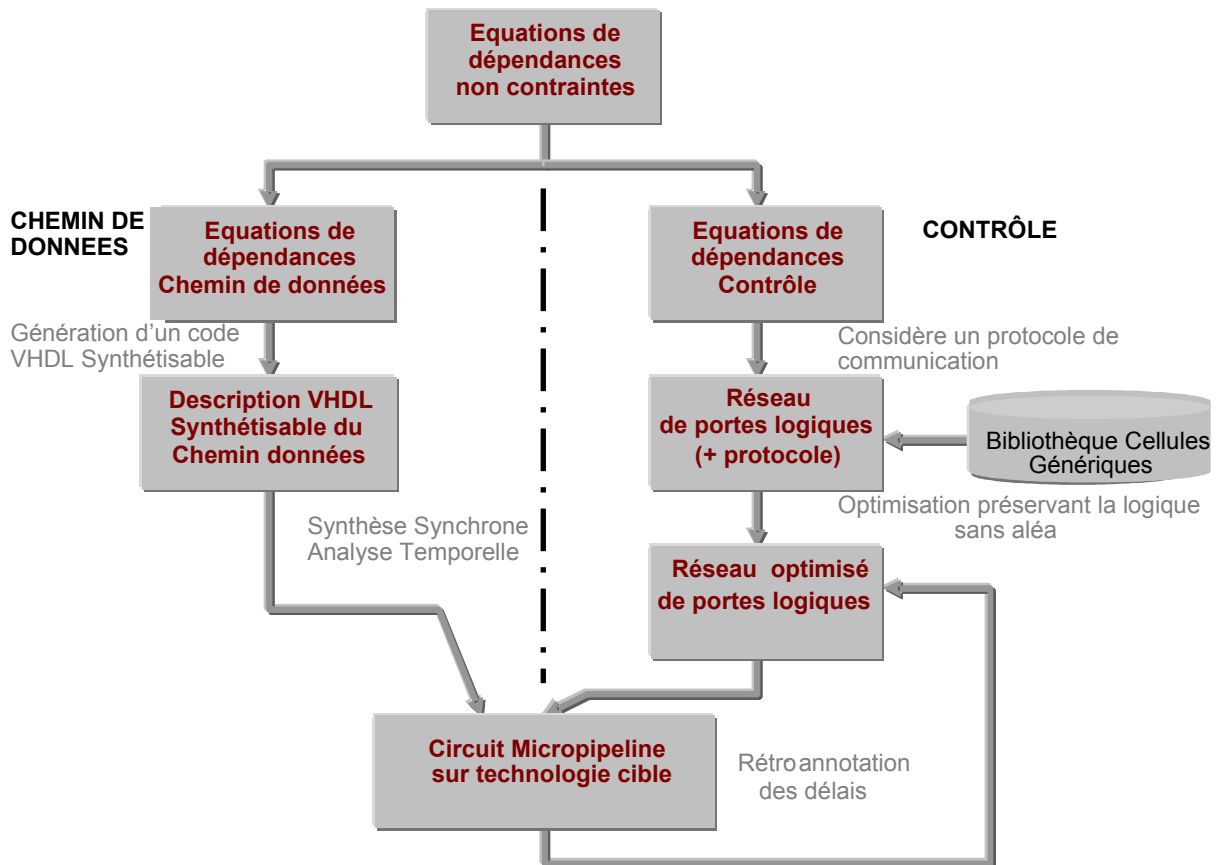


Figure 34 : Flot de synthèse d'un circuit micropipeline

Le flot de synthèse micropipeline illustré Figure 34 fait apparaître clairement les deux parties. La partie gauche du schéma montre que les équations de dépendance relatives au chemin de données sont extraites et utilisées pour produire un code VHDL RTL. Ce dernier sera synthétisé avec les outils du commerce. La partie droite de la figure permet de générer les équations de dépendance correspondant au contrôleur. Ce dernier est généré en logique QDI avant d'être optimisé. Il est à noter que le cadre de la thèse d'Amine Rezzag, la synthèse de contrôleurs exploitant différents protocoles – séquentiel, WCHB, PCHB et PCFB – a été étudiée. Enfin, il faut remarquer que dans ce flot l'extraction des temps de propagation de la partie calcul est très importante car elle sert à rétro-annoter les délais contenus dans les contrôleurs (et donc à choisir le délai approprié). En effet, ce style de circuit fait des hypothèses temporelles comparables à celles rencontrées en synchrone, si ce n'est qu'elles sont appliquées localement et non globalement.

III.5 Génération de moniteurs à partir d'une expression logico-temporelle

Les langages de spécification de propriétés logico-temporelles comme PSL [IEE 05] ou SVA [IEE 05b] permettent habituellement de vérifier formellement ou par simulation les propriétés des circuits numériques. Ces langages sont aujourd'hui largement répandus et intégrés dans les flots de conception commerciaux. Dans ce contexte relatif à la CAO des systèmes numériques, je mène au laboratoire en collaboration avec Dominique Borrione et Katell Morin-Allory des recherches visant la génération de moniteurs de surveillance en technologie asynchrones. Dans le cas présent, l'apport de l'asynchrone peut être crucial car il permet de générer des moniteurs bien plus robustes et fiables que les circuits qu'ils ont à

surveiller. Ces moniteurs sont donc parfaitement adaptés dès que l'on a ausculté des systèmes où la sécurité et la fiabilité sont les maîtres mots. Ce type de dispositif est susceptible d'apporter des solutions sûres pour l'électronique embarquée dans les avions (répondant à la norme DO 254) ou l'automobile. Ils peuvent aussi détecter des fonctionnements erronés dans systèmes complexes comme les GALS, notamment pour les transactions entre domaine d'horloge, et plus généralement dès que l'on a surveillé un évènement impromptu, c'est-à-dire asynchrone du dispositif de synchronisation.

III.5.1 Les langages de spécification de propriétés

Les langages de spécification de propriétés décrivent des assertions qui contiennent en général trois structures : les expressions booléennes, les expressions régulières (SERE) et des propriétés permettant de relier expressions booléennes et expression régulières. Afin de fixer les idées et surtout de cerner les subtiles différences existant entre l'interprétation d'une même assertion dans un environnement synchrone et dans un contexte asynchrone, un petit exemple est donné en langage PSL. Considérons donc la propriété suivante :

```
PSL Property P is
    Always A -> next (B until C);
```

La propriété P signifie qu'à chaque cycle d'évaluation quand A est à 1, au cycle d'après B devra être à 1 et se maintenir à 1 jusqu'à ce que C soit passé à 1. Dans un environnement synchrone, l'évaluation est faite sur le front de l'horloge (clock driven, voir la Figure 35) tandis que dans le cas asynchrone l'analyse de l'expression se fait en fonction des évènements sur les signaux (event driven, voir la Figure 36). On remarque sur les figures qu'au cycle 8 en mode synchrone, la propriété P devient fausse alors qu'au même endroit la propriété P est pendante (en cours d'évaluation) en mode asynchrone. Il peut donc y avoir des différences d'interprétation entre ces deux modes.

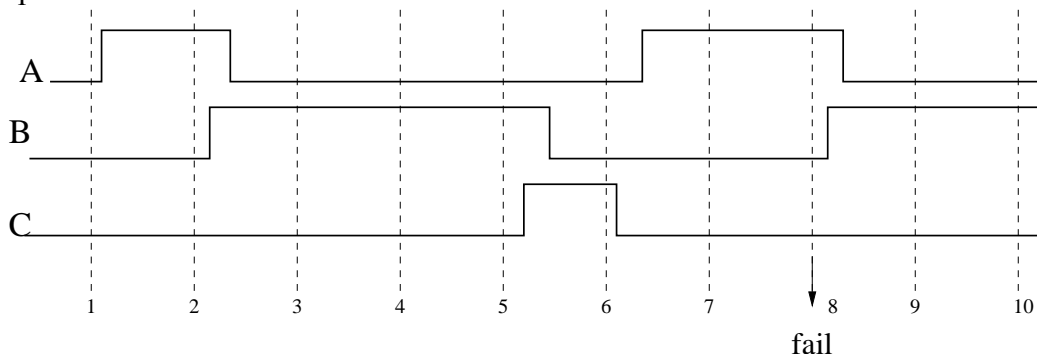


Figure 35 : Evaluation de la propriété P en mode synchrone

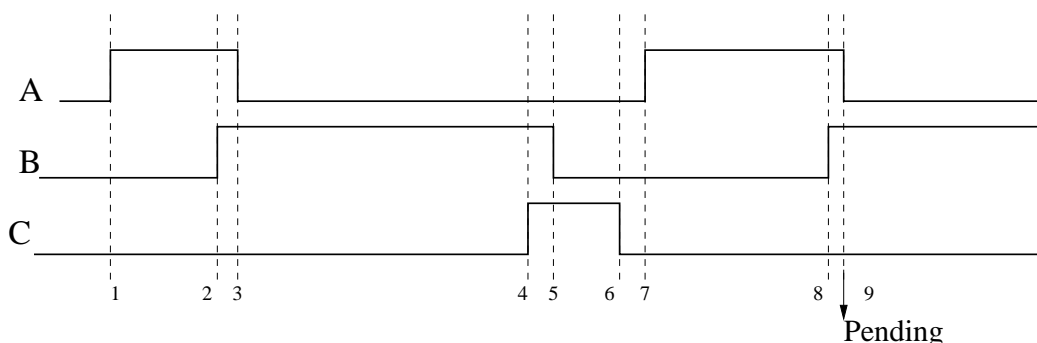


Figure 36 : Evaluation de la propriété P en mode asynchrone

III.5.2 Etat de l'art

Le premier outil à automatiser la synthèse de moniteurs depuis un langage de spécification de propriétés est probablement FoCs d'IBM [ABA 00]. FoCs traduit les assertions PSL sous la forme d'un code RTL synthétisable en VHDL ou en Verilog. Les moniteurs ainsi générés peuvent ensuite être connectés manuellement au design à vérifier. En général, ces outils construisent le graphe d'états de la spécification PSL avant de produire le code RTL. C'est le cas par exemple de l'outil Dialite de Temento Systems et probablement aussi la solution retenue par IBM pour FoCs. Le principal problème de ce type d'approche est que l'on peut se trouver dans la situation où le nombre d'états du graphe explose. Une alternative à cette méthode est de construire le moniteur en exploitant une approche type synthèse dirigée par la syntaxe. Cette méthode a été étudiée au laboratoire TIMA [BOR 05][BOR 06]. La force de cette approche est qu'elle a été prouvée formellement correcte par démonstration de théorème. Il est à noter qu'à l'université McGill au Canada une approche similaire est également suivie [BOU 06]. Enfin, il existe de nombreuses sociétés qui proposent dans leur flot des bibliothèques de moniteurs prédéfinis tels que CheckerWare [MEN 07] ou OVL [ACC 07].

Tout ce qui est décrit ci-dessus ne concerne que l'approche synchrone et, à notre connaissance, nous sommes aujourd'hui les seuls à proposer une méthode capable de générer des moniteurs asynchrones [MOR 06][MOR 07].

III.5.3 Génération des moniteurs asynchrones

Les moniteurs asynchrones générés sont capables de répondre à quatre niveaux de satisfaction d'une propriété : vraie fortement, vraie, pendante, fausse. Une propriété PSL est construite sous la forme d'un module matériel qui dispose d'un signal de reset, de signaux de synchronisation, d'un signal *start* pour lancer une évaluation de la propriété et de signaux issus du circuit à vérifier. Les sorties d'un moniteur sont au nombre de trois :

- le signal *checking* à 1 indique que l'évaluation est à considérer,
- le signal *valid* donne le résultat de l'évaluation (vraie ou fausse),
- le signal *pending* traduit le fait que le moniteur est en cours d'évaluation et qu'il ne peut pour l'instant dire si la propriété est vérifiée fortement ou pas (à utiliser uniquement avec les opérateurs forts).

La méthode de synthèse employée pour construire les moniteurs exploite une bibliothèque de moniteurs élémentaires (IP matérielle) correspondant aux opérateurs du langage PSL et à une méthode d'interconnexion systématique de ces moniteurs élémentaires pour la réalisation de moniteurs complexes. Les moniteurs asynchrones sont ainsi générés directement à partir de la construction syntaxique du langage. La Figure 37 montre le réseau d'interconnexion généré pour la propriété P donné au paragraphe III.5.1. Nous remarquons que le moniteur est constitué d'opérateurs élémentaires (Always, ImPLY, Next et Until), d'un bloc d'initialisation (Gen_Init) et de blocs de synchroniseurs ayant pour fonction de capturer l'activité des signaux (A, B et C) en entrée du moniteur.

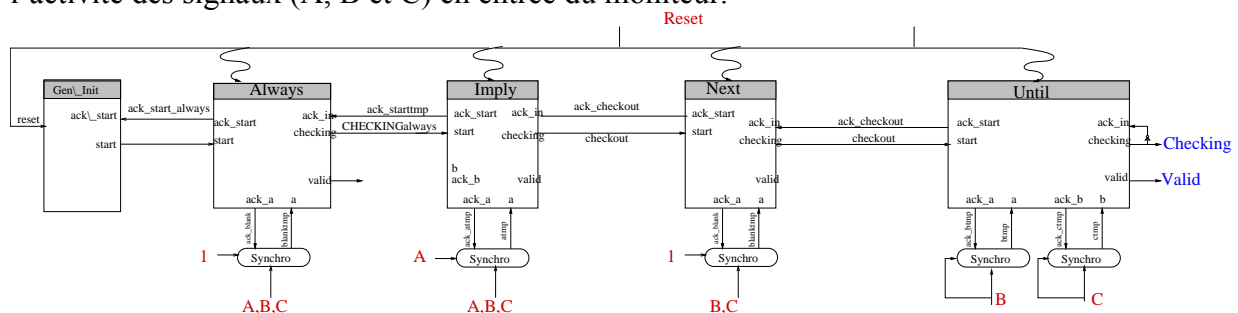


Figure 37 : L'interconnexion d'un moniteur complexe correspondant à la propriété P

Sans entrer dans les détails, la construction des moniteurs asynchrones est bien différente de leurs homologues synchrones. En effet, il n'est pas possible ici de générer un code VHDL synthétisable. La conception des moniteurs élémentaires est réalisée en exprimant la sémantique des opérateurs PSL sous la forme d'un réseau de Petri. Ce réseau de Petri est alors synthétisé en un circuit QDI par une technique de traduction systématique du réseau de Petri. Il est à noter qu'à ce stade des travaux, nous ne souhaitons pas faire d'optimisations sur les moniteurs élémentaires car nous souhaitons conserver strictement la sémantique de l'opérateur PSL et prouver formellement que la synthèse obtenue est correcte. Cette méthode va dans le sens de la génération de moniteurs dont le fonctionnement est sûr, robuste et prouvé correct (ce qui est une demande du milieu industriel pour les applications dans le domaine de l'aéronautique par exemple). La perte de surface occasionnée par cette approche est de toutes façons très modeste étant donné la taille des moniteurs en général. Afin d'illustrer le propos, la Figure 38 montre le réseau de Petri donnant la sémantique de l'opérateur PSL *ImPLY* (\Rightarrow). La Figure 39 donne le résultat de la synthèse en logique asynchrone de ce même opérateur.

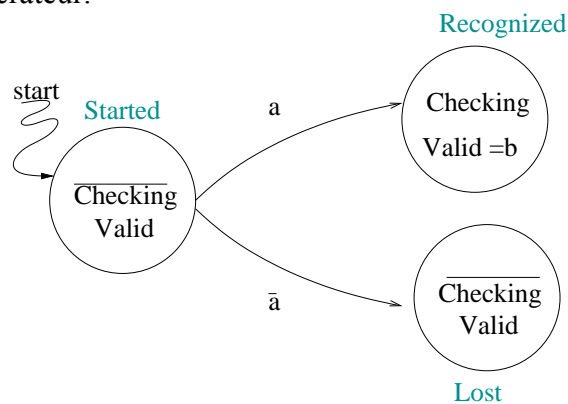


Figure 38 : Le réseau de Petri de l'opérateur ImPLY

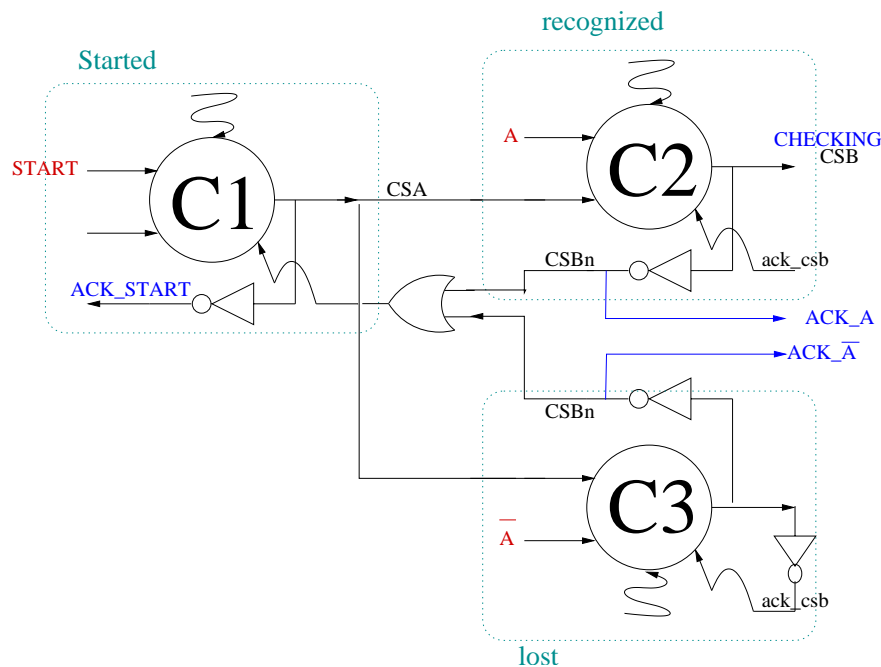


Figure 39 : Le moniteur élémentaire QDI correspondant à l'opérateur ImPLY

Enfin, le fonctionnement de ces moniteurs asynchrones a été validé sur une plateforme FPGA Altera Stratix II. Le système de monitoring mis en œuvre avait pour fonction de

vérifier les transactions sur le bus Avalon d'un processeur Nios et jouait le rôle d'un dispositif espion capable de surveiller les opérations effectuées par le logiciel sur le processeur.

III.6 Conclusion

La synthèse logique des circuits asynchrones a été étudiée pour les circuits QDI mais aussi pour les circuits micropipelines. L'ensemble de ces travaux a donné lieu à plusieurs thèses et a permis de définir un flot de conception générique s'appuyant sur des descriptions exploitant un langage de haut niveau (i.e. CHP). Par ailleurs, la sémantique du langage a été explicitée de sorte qu'il soit possible de coder des spécifications synthétisables. Les transformations et les formes intermédiaires du flot sont décrites en détails. Ainsi, il est important de noter qu'il est possible de réaliser des optimisations logiques sur les formes intermédiaires mais aussi, dans le cas des diagrammes de décision multi-valués, des preuves mathématiques. Il est particulièrement important de démontrer que les circuits générés sont quasi-insensibles aux délais. En effet, cette propriété, si elle est vérifiée, garantit un niveau de robustesse très élevé vis-à-vis de variations de tension ou de température. Ce point est donc primordial pour imposer la logique asynchrone comme une alternative sérieuse dans des environnements hostiles. Cette grande robustesse montre également l'intérêt que l'on peut avoir à utiliser ce type de logique pour surveiller des circuits synchrones qui auraient des fonctions sensibles en terme de sécurité par exemple. Enfin, même si cela n'est pas présenté, il est envisageable de contraindre la synthèse en fonction de la consommation, de la vitesse, de la sécurité ou du rayonnement électromagnétique. Les possibilités de synthétiser des circuits asynchrones étant multiples, il est naturel que les applications potentielles soient nombreuses et variées. Les travaux déjà effectués dans le groupe ouvrent donc de multiples perspectives autour de cette thématique de la synthèse de circuits asynchrones.

IV. Traitement des signaux non uniformément échantillonnés

Comme nous l'avons vu dans les chapitres précédents, la tendance en microélectronique est d'intégrer toujours plus mais en consommant toujours moins. Les nombreux efforts consentis par les concepteurs sur ce point offrent des perspectives d'améliorations substantielles. Toutefois, il me semble que ces approches sont limitées car elles ne prennent pas en compte la nature du système et les informations qu'elles manipulent. En général, il est plus intéressant de considérer l'ensemble d'un système pour l'optimiser plutôt que d'améliorer son flot de conception. Ce constat nous a amené à reconsidérer les approches usuelles et à rechercher des voies nouvelles. L'une d'entre elles nous a particulièrement séduit et je ne suis pas loin de penser qu'elle pourrait constituer un jour un élément de rupture technologique. Cette voie « séduisante » traite des chaînes de traitement du signal qui sont usuellement intégrés dans les systèmes complexes (intégrés ou non d'ailleurs). Un dispositif de traitement du signal est constitué de capteurs, de convertisseurs Analogique/Numérique (CAN, CNA), de circuits de traitement numérique du signal, de systèmes de transmission de données. Ces systèmes ont vu leur complexité croître très rapidement ces dernières années et conjointement la puissance de traitement des données. Paradoxalement, on exige de ces systèmes d'être peu coûteux, portables, faiblement rayonnants et surtout peu consommateurs. En effet, l'énergie consommée a été mise sous haute surveillance car ces dispositifs sont autonomes et très souvent alimentés par des batteries. La consommation d'énergie constitue la plupart du temps la pierre angulaire de leur conception. Par ailleurs, on constate que la plupart des systèmes traitent des signaux dont les propriétés statistiques sont intéressantes, mais aucune architecture de traitement du signal n'en tient vraiment compte. Des signaux, comme la température, la pression, les électrocardiogrammes, la parole, etc. restent longtemps constants avant de varier sur une courte période. L'échantillonnage classique à pas constant, que l'on appellera échantillonnage uniforme, ne profite pas de cette information et cela se traduit au niveau du système de conversion par de fortes contraintes liées au théorème de Shannon qui impose que la fréquence d'échantillonnage soit au moins égale au double de la fréquence maximale contenue dans le signal. L'information des échantillons prélevés dans les périodes stables est alors redondante, entraînant une suractivité du circuit et donc un surcoût en énergie.

Les travaux menés au sein de l'équipe visent à repenser complètement la chaîne classique de traitement numérique du signal afin de réduire son activité et du même coup la consommation électrique des circuits associés en exploitant une technique d'échantillonnage non uniforme. Ils ont fait l'objet de trois thèses (dont une est en cours) : la thèse d'Emmanuel Allier [ALL 03] qui a travaillé sur la conversion analogique-numérique asynchrone, celle de Fabien Aeschlimann [AES 06] qui a étudié l'utilisation de signaux non uniformément échantillonnés en temps et enfin celle de Saeed Mian Qaisar (en cours) qui traite de techniques de sélection de l'activité d'un signal et de techniques de rééchantillonnage.

IV.1 A propos de l'échantillonnage non uniforme !

Aujourd'hui, le traitement numérique du signal repose essentiellement sur un échantillonnage uniforme en temps des signaux analogiques. Cet échantillonnage produit un signal numérique dont les échantillons sont régulièrement espacés d'une durée T_e , la période d'échantillonnage. Le schéma d'échantillonnage est un peigne de Dirac uniforme en fréquence. Le spectre du signal échantillonné est donc périodisé à tous les entiers multiples de la fréquence d'échantillonnage. Il est ainsi possible de retrouver le théorème de Shannon qui découle de la périodisation : un signal à temps continu à bande limitée, de fréquence

maximale F_{max} , peut être reconstruit à partir d'observations régulières si l'occurrence de ces observations respecte la condition $F_e \geq 2F_{max}$. En respectant cette condition, la périodisation n'introduit pas de repliement du spectre du signal analogique. Le signal peut alors être reconstruit en filtrant le signal échantillonné par un filtre passe-bas idéal dont la réponse impulsionnelle est un sinus cardinal. Ce résultat fondamental montre qu'aucune information n'est perdue lors de l'échantillonnage si celui-ci est suffisamment dense. Dans le cas d'un échantillonnage non uniforme, les instants d'échantillonnage sont quelconques. Cela remet donc fondamentalement en cause les résultats connus avec un échantillonnage standard (même si la fonction d'échantillonnage reste donc sous la forme générale d'un peigne dont les dents sont irrégulièrement espacées !).

De nombreuses recherches ont déjà été menées sur les échantillonnages non uniformes, comme nous le montre l'imposant livre édité par Farokh Marvasti de l'université Sharif de Téhéran [MAR 01] ou les travaux théoriques du groupe de Hans Feichtinger et de Karlheinz Gröchenig en Autriche. Je ne donne pas ici de références (vu la profusion d'articles) mais je conseille au lecteur de recherche sur Internet avec le mot clé NuHAG (Numerical Harmonical Analysis Group). Il est également à noter que le traitement d'un signal échantillonné non uniformément diffère selon l'origine de son irrégularité. Des travaux ont été effectués par exemple dans le domaine des échantillonnages non uniformes aléatoires [TAR 05]. Ils traitent des irrégularités provenant de sources naturelles comme le jitter sur l'horloge d'échantillonnage, des pertes d'échantillons [MAR 05] ou bien des techniques d'échantillonnage avec un tirage aléatoire. Dans notre cas, qui est présenté dans la suite, nous nous intéressons à un échantillonnage non uniforme corrélé aux évolutions du signal à échantillonner. Il s'agit ici d'une technique d'échantillonnage dite par traversée de niveaux, mais il est possible de décliner cette approche sous d'autres formes comme la détection de la variation de la pente d'un signal ou la détection des extrema.

IV.2 La conversion de signaux analogiques avec un schéma d'échantillonnage non uniforme en temps

IV.2.1 L'échantillonnage par traversée de niveaux

Le principe recherché à l'origine avec l'échantillonnage par traversée de niveaux était de s'affranchir du signal d'horloge commandant l'échantillonnage dans les systèmes de conversion analogique numérique. Dans le cas usuel, la fréquence d'échantillonnage n'est liée au signal continu que par le théorème de Shannon. Les caractéristiques du signal ne sont nullement prises en compte. Dans les zones où le signal varie peu, un échantillonnage uniforme prélève des données redondantes. J. W. Mark a donc proposé une méthode où les échantillons sont capturés lorsque le signal croise les niveaux répartis uniformément sur la dynamique du signal d'entrée afin de compresser les données prélevées [MAR 81]. A la différence de l'échantillonnage uniforme où l'amplitude du signal est quantifiée, ce type d'échantillonnage introduit une quantification du temps liée à la mesure de la durée séparant deux échantillons successifs [SAY 96]. L'intérêt de cet échantillonnage réside dans le fait que l'activité du signal régule automatiquement l'activité du système de conversion. L'échantillonnage par traversée de niveaux est donc très attractif pour réduire l'activité de l'ensemble de la chaîne de traitement du signal et donc sa consommation d'énergie.

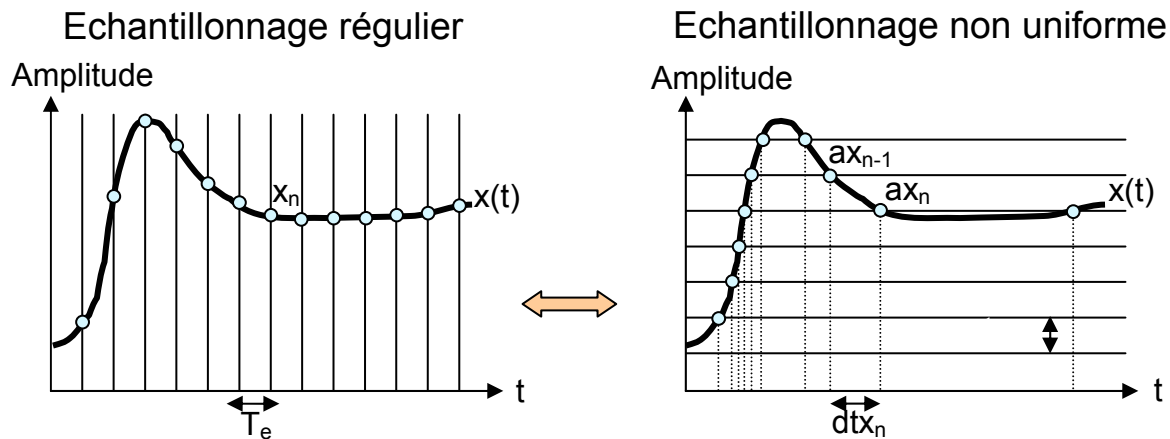


Figure 40 : Echantillonnages uniforme et non uniforme par traversée de niveaux

Sur la Figure 40, il est facile de se rendre compte des différences entre l'échantillonnage usuel et l'échantillonnage par traversée de niveaux. Les principales différences peuvent se résumer comme suit :

Echantillonnage régulier	Echantillonnage non uniforme
* Respect du théorème de Shannon	* "par traversée de niveaux"
* Quantification de l'amplitude	* Quantification du temps
* Instants connus	* Amplitudes connues
* Echantillons redondants	* Pas d'échantillons redondants
* En sortie: $\{ x_n, T_s \}$	* En sortie: $\{ Aq_n ; dtx_n \}$

On note dans ce tableau la dualité des deux approches. En effet, dans un cas on quantifie l'amplitude et dans l'autre c'est le temps qui est quantifié. De même, les instants sont parfaitement connus dans le cas régulier alors que ce sont les amplitudes qui le sont dans le cas non uniforme. Enfin en sortie, au lieu d'obtenir une nouvelle amplitude (x_n) à chaque période d'échantillonnage (T_s), on obtient (dtx_n), le temps entre deux échantillons, et (Aq_n), une information indiquant si l'amplitude a augmenté ou diminué d'un quantum.

IV.2.2 Principe d'un convertisseur à échantillonnage par traversée de niveaux

L'architecture retenue pour le convertisseur est une boucle de conversion composée de quatre éléments distincts : un comparateur, un compteur, un convertisseur N/A et un timer. Par ailleurs, le comportement « événementiel » de ce type de convertisseur se marie bien avec une conception asynchrone qui, elle aussi, fonctionne par événements. Ainsi dans la suite, nous parlerons de CANA pour « Convertisseur Analogique Numérique Asynchrone » pour désigner une implémentation usant de cette technique d'échantillonnage et conçue en logique asynchrone.

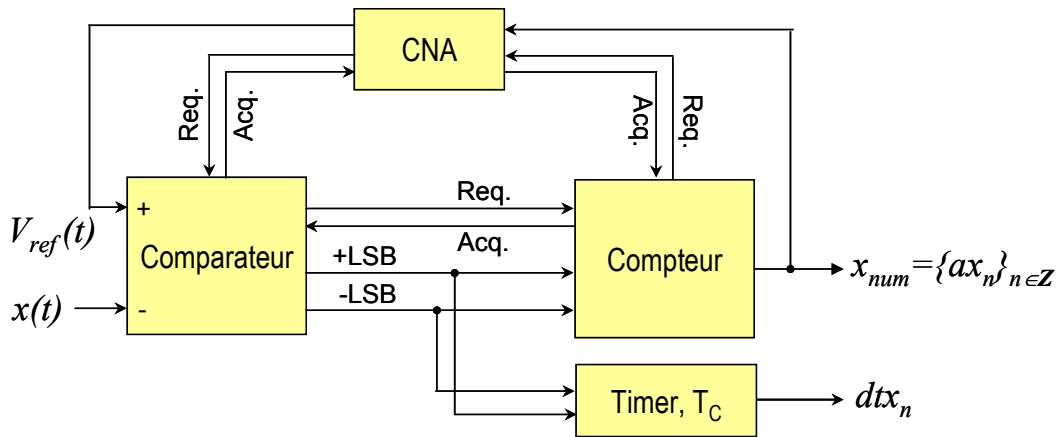


Figure 41 : Boucle de conversion asynchrone

Le système, présenté sur la Figure 41 est un asservissement dont la consigne est le signal analogique à convertir. Le signal analogique d'entrée $x(t)$ est converti en un signal numérique x_{num} , lui-même converti par le CNA en un signal analogique de référence $V_{ref}(t)$. Les signaux $x(t)$ et $V_{ref}(t)$ sont alors comparés : si la différence est plus grande qu'un quantum, q , alors le compteur est incrémenté ($+LSB = '1'$, $-LSB = '0'$) ; si elle est plus petite que q alors le compteur est décrémenté ($+LSB = '0'$, $-LSB = '1'$) ; dans tous les autres cas, la sortie reste constante ($+LSB = '0'$, $-LSB = '0'$), il n'y pas d'activité. Comme l'échantillonnage est adapté aux variations du signal d'entrée, donc non uniformément espacé dans le temps, une information temporelle est associée au signal numérique : le temps écoulé depuis l'occurrence précédente est mesuré par le timer. La sortie du convertisseur est ainsi composée de couples (amplitude, intervalle de temps) notés respectivement ax_n , et dtx_n pour spécifier la dépendance au signal d'entrée x . Les instants d'échantillonnage tx_n ne sont pas connus en pratique. Cependant pour simplifier la notation de certains algorithmes, nous pourrions facilement les reconstruire à partir de la relation $tx_n = tx_{n-1} + dtx_n$. Enfin, on peut constater que la structure présentée ne comporte aucun signal d'horloge global. Les blocs sont pilotés par des signaux de contrôle locaux : une requête (*Req.*) et un acquittement (*Acq.*). Leur synchronisation dépend du protocole de communication choisi lors de l'implémentation pour gérer les échanges. Par ailleurs, il est à noter que les caractéristiques du signal à convertir sont limitées par le délai borné δ introduit par la boucle. Ainsi, lorsque la conversion d'un échantillon est commencée (lorsqu'un niveau vient d'être croisé), le signal ne doit pas traverser un autre niveau (déclenchant une nouvelle conversion) avant que le signal de référence n'ait été mis à jour. Ceci limite donc la pente du signal d'entrée :

Équation 4
$$\left| \frac{dx(t)}{dt} \right| \leq \frac{q}{\delta}$$

La relation donnée par l'Équation 4 est appelée condition de poursuite. Si elle n'est pas vérifiée, les échantillons en sortie ne correspondront pas au signal analogique et on observera un effet de limitation de la pente du signal numérique.

IV.2.3 Rapport signal sur bruit d'un CANA

Avant de poursuivre sur les aspects relatifs à l'implémentation du convertisseur, il est bon de regarder le comportement du rapport signal sur bruit d'un tel dispositif. Dans un convertisseur A/N asynchrone, un échantillon est prélevé lorsque le signal croise un niveau ; la valeur de l'amplitude est exacte (elle correspond au niveau croisé). Cependant l'instant correspondant à ce croisement est quantifié à l'aide d'un timer qui introduit ici une erreur de quantification du temps. Cette erreur joue ici le même rôle que le quantum d'amplitude dans

le calcul du rapport signal sur bruit d'un convertisseur traditionnel. Avec une résolution T_C , le timer compte le nombre d'occurrences qui séparent deux conversions successives. Lorsque le signal traverse un niveau, l'état du timer est figé. L'instant d'échantillonnage est ainsi connu avec une précision égale à T_C . Le timer est ensuite réinitialisé pour la mesure suivante. En conséquence, la précision d'un intervalle de temps en sortie du timer dépend donc de la différence des deux erreurs de quantification successives. L'erreur de mesure du temps est donc comprise dans un intervalle $[-T_C ; T_C]$ (cf. Équation 5)

$$\begin{aligned} dx_n &= tx_n - tx_{n-1} \\ &= tx_{n,theo} - \delta t_n - tx_{n-1,theo} + \delta t_{n-1} \\ &= dx_{n,theo} + \delta t_{n-1} - \delta t_n \in [dx_{n,theo} - 2T_C, dx_{n,theo} + 2T_C] \end{aligned}$$

Équation 5

La remise à zéro est synchrone de l'horloge commandant le timer. Ce mode permet de générer des intervalles de temps de moyenne nulle. En reconstruisant les instants d'échantillonnage à partir des intervalles de temps, la quantification introduit un jitter de moyenne nulle (équivalent à un échantillonnage aléatoire variable). En outre, il est important de noter que si la remise à zéro était asynchrone, le jitter produit ne serait plus de moyenne nulle. Les instants d'échantillonnage reconstruits deviendraient rapidement erronés car chaque nouvelle erreur serait accumulée aux précédentes conduisant à une dilatation de l'axe temporel.

Ceci étant précisé, il est possible de déterminer le rapport signal sur bruit d'un convertisseur A/N asynchrone. En fonction de la pente instantanée, une erreur en amplitude δx résulte de l'erreur temporelle δt : $\delta x = (dx(t)/dt) \delta t$. En considérant les variables $dx(t)/dt$ et δx comme deux variables aléatoires indépendantes, la puissance du bruit de quantification est alors donnée par l'Équation 6.

Équation 6

$$P(\delta x) = P\left(\frac{dx}{dt}\right)P(\delta t)$$

En supposant que la densité de probabilité de δt est constante dans l'intervalle $[0; T_C]$, la puissance de δx est égale à $P(\delta x) = T_C^2/3$. Le rapport signal sur bruit (RSB) du convertisseur asynchrone est alors défini par l'Équation 7 :

Équation 7

$$RSB_{dB} = 10 \log\left(\frac{P(x)}{P(\delta x)}\right) = 10 \log\left(\frac{3P(x)}{P\left(\frac{dx}{dt}\right)}\right) + 20 \log\left(\frac{1}{T_C}\right)$$

Le rapport signal sur bruit (RSB) dépend donc à la fois du signal (et de sa dérivée) et de la résolution du timer. Or, pour une application donnée, c'est-à-dire pour une classe de signaux donnée, le premier terme devient constant. Le rapport ne dépend alors que de la résolution du timer T_C . En revanche, il ne dépend pas de la résolution N du convertisseur, c'est-à-dire du quantum, de la même manière que le RSB d'un convertisseur classique ne dépend pas de la fréquence d'échantillonnage. Pour une résolution matérielle donnée, le RSB du convertisseur peut être réglé en modifiant la période de l'horloge du timer. Afin de définir une résolution équivalente aux convertisseurs classiques, la notion de nombre effectif de bits (*ENOB*) est utilisée. Elle permet de comparer directement les performances entre les convertisseurs en égalisant les *RSB* dans les cas synchrone et asynchrone puis en remontant à la résolution dans le cas synchrone (cf. Équation 8).

Équation 8

$$ENOB = \frac{RSB_{dB} - 1,76}{6,02}$$

Cette technique a été employée pour caractériser notre convertisseur analogique numérique asynchrone et pour mettre en évidence que le rapport signal sur bruit du convertisseur suivait bien la loi prédite par notre modèle.

IV.2.4 Implémentation d'un convertisseur analogique numérique asynchrone

Ce paragraphe donne quelques indications générales sur la façon dont a été implémenté le convertisseur. Ce travail correspond en grande partie à la thèse d'Emmanuel Allier [ALL 03] et a donné lieu à de nombreuses publications [ALL 02][ALL 03b][ALL 05] ainsi qu'à un brevet [ALL 03c].

La structure asynchrone choisie pour implémenter la boucle de conversion est une structure de type micropipeline à trois étages qui réalise respectivement la comparaison, le comptage et la conversion numérique analogique. Chaque étage est donc composé de deux parties distinctes : un chemin de donnée et une partie contrôle QDI. Le schéma (sans le timer) est donné Figure 42. La partie contrôle est composée de portes de Muller, d'inverseurs et de retards. Ces derniers correspondent aux délais critiques de chaque étage et permettent d'éviter l'envoi d'une requête à l'étage suivant avant la fin de l'exécution de l'opération réalisée par l'étage. La partie contrôle permet de séquencer l'avancement des données. Après une phase d'initialisation, le premier étage devient actif et produit le résultat de ces comparaisons avec le signal analogique à convertir. Le résultat est transmis à l'étage de comptage qui joue ici le rôle d'un intégrateur. Enfin, la valeur numérique obtenue en sortie du compteur sert à générer un nouveau signal de référence avec le CNA pour les comparaisons qui seront faites au cycle suivant par le premier étage et ainsi de suite.

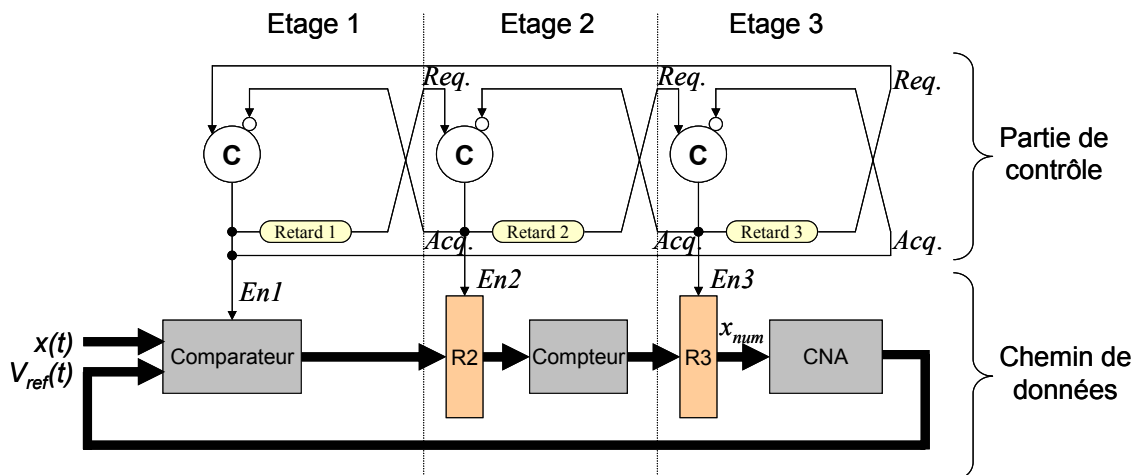


Figure 42 : Schéma de principe du CANA

A ce jour, un convertisseur a été réalisé [ALL 05] (voir Figure 43). Il s'agit d'un convertisseur 4 bits (15 niveaux) conçu en technologie CMOS 130nm, 1.2V ($q = 40mV$). Le temps de boucle est de 66ns permettant la conversion de signaux d'une fréquence maximale de 160kHz. Les intervalles de temps sont quantifiés sur 12 bits et la résolution du timer peut aller jusqu'à $T_C = 1\mu s$ permettant au CANA d'être équivalent à un convertisseur classique 8 bits. Les tests de consommation ont montré que la puissance P du convertisseur est toujours inférieure à $180\mu W$ et ce même lorsque le signal d'entrée atteint la fréquence maximale admise par le système qui correspond au pire cas pour la consommation. Le CANA a un

facteur de mérite, FoM (voir Équation 9), qui est généralement meilleur d’au moins un ordre de grandeur par rapport aux convertisseurs classiques (le gain n’était que de deux pour le meilleur CAN que nous ayons trouvé dans la littérature).

Équation 9
$$FoM = \frac{2^{ENOB} \cdot f_{max}}{P \cdot Surface}$$

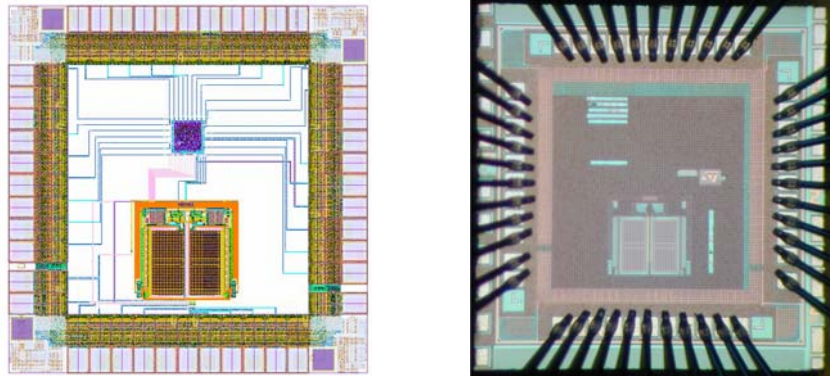


Figure 43 : Layout et photographie du CANA

IV.3 Exploitation des signaux échantillonnés non uniformément

Comme nous l’avons vu au paragraphe IV.2, la conversion par traversée de niveaux permet dans certaines situations de réduire le nombre d’échantillons et donc potentiellement l’activité du processeur de traitement de signal. Mais une fois la conversion faite, l’usage de ces signaux numériques n’est pas pour autant trivial. En effet, nous sommes passés d’un schéma d’échantillonnage régulier à un schéma non régulier dont l’irrégularité de surcroît dépend du signal lui-même. Il est nécessaire d’envisager des techniques appropriées de traitement de ces signaux qui diffèrent grandement de celles habituellement utilisées en traitement du signal. Dans ce qui suit, nous nous intéressons à différentes techniques de filtrage que l’on peut employer avec ces signaux numériques.

IV.3.1 Le filtrage RIF

La première proposition faite dans le domaine a été d’étudier un filtrage à réponse impulsionnelle finie (RIF). Ce travail constitue notamment une partie des travaux de thèse de Fabien Aeschlimann [AES 06].

Le filtrage est une opération de base en traitement du signal aussi bien sur des systèmes en temps continu qu’en temps discret. Nous nous sommes donc logiquement intéressés à cette opération dans le cadre spécifique de l’échantillonnage non uniforme par traversée de niveaux. Parmi les deux grandes catégories de filtres numériques – à réponse impulsionnelle finie RIF ou infinie RII, nous avons choisi dans un premier temps d’étudier les filtres RIF car ils possèdent la propriété fondamentale d’être inconditionnellement stables, la sortie courante ne dépendant que des entrées, l’algorithme ne peut donc pas diverger. Le calcul d’un filtrage RIF est donné par l’Équation 10.

Équation 10
$$y_n = \sum_{i=0}^{N-1} h_i x_{n-i}$$

Lorsque l'échantillonnage est non uniforme, il n'est plus possible d'utiliser l'Équation 10. Il faut donc élaborer une nouvelle formulation prenant en compte la spécificité de la non uniformité liée à l'échantillonnage par traversée de niveaux, c'est-à-dire la non uniformité des intervalles de temps. Comme l'existant n'est pas exploitable, il est nécessaire de revenir à la définition d'un filtre analogique afin d'interpoler le signal d'échantillonné et la réponse impulsionnelle.

IV.3.1.1 Produit de convolution « asynchrone »

Dans le domaine analogique, le signal issu du produit de convolution de deux signaux est défini par la relation suivante :

Équation 11
$$y(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t - \tau) d\tau$$

Théoriquement, le signal de sortie $y(t)$ est calculé pour tous t et τ réels mais peut aussi l'être pour n'importe quel temps t appartenant à un ensemble discret $\{t_n\}$. De plus, si la réponse impulsionnelle est de longueur finie T_h , le produit de convolution sera à durée limitée n'utilisant que des échantillons du signal d'entrée compris dans l'intervalle $[t_n; t_n - T_h]$. Cette propriété est utilisée pour définir le produit de convolution numérique pour des signaux échantillonnés non uniformément dans le temps à partir du produit de convolution analogique équivalent des signaux interpolés. Soient x et h , deux fonctions d'entrée définies à temps

continu par $x(t)$ et $h(t)$ et à temps discret par $\begin{cases} ax_n = x(tx_n) \\ dtx_n = tx_n - tx_{n-1} \end{cases}$ et $\begin{cases} ah_n = h(th_n) \\ dth_n = th_n - th_{n-1} \end{cases}$. La

fonction de sortie y représentée en temps discret par $\begin{cases} ay_n = y(ty_n) \\ dty_n = ty_n - ty_{n-1} \end{cases}$ est alors définie

comme le résultat du produit de convolution entre deux nouvelles fonctions interpolées à temps continu $\hat{x}(t)$ et $\hat{h}(t)$:

Équation 12
$$ay_n = y(ty_n) = \int_{-\infty}^{+\infty} \hat{x}(\tau) \cdot \hat{h}(ty_n - \tau) d\tau$$

Dans la pratique, l'intervalle de temps nécessaire au calcul est limité par la réponse impulsionnelle finie h d'une durée égale à T_h . Ainsi lorsque les N_h échantillons auront été utilisés, le calcul sera achevé. L'autre difficulté concerne les instants d'échantillonnage de la sortie. Il est possible d'envisager diverses solutions mais le choix des instants d'échantillonnage de l'entrée x permet de simplifier le problème et d'amener une méthode pratique pour implémenter ces filtres. Les instants d'échantillonnage de la sortie sont donc fixés égaux à ceux de l'entrée. Ainsi, il n'y a pas d'opération à effectuer sur les intervalles de temps de sortie puisqu'ils sont aussi égaux aux intervalles de temps de l'entrée :

Équation 13
$$\begin{cases} ty_n = tx_n \\ dty_n = dtx_n \end{cases}$$

Dans ces conditions, le calcul de l'aire résultante du produit des deux fonctions interpolées, dépend à la fois du signal entrant x et de la réponse impulsionnelle h . En effet, pour calculer le produit, il faut que les deux fonctions interpolées soient décomposables par morceaux de même taille. Ceci implique donc que x et h doivent être échantillonnés aux mêmes instants comme cela est montré sur la Figure 44. Les deux fonctions interpolées $\hat{x}(t)$ et $\hat{h}(tx_n - t)$ sont liées par les relations suivantes :

Équation 14

$$\left\{ \begin{array}{l} \hat{t}x_k = tx_{n-i} \\ a\hat{x}_k = \hat{x}(tx_{n-i}) = ax_{n-i} \end{array} \right. \quad \text{ou} \quad \left\{ \begin{array}{l} \hat{t}x_k = tx_n - th_j \\ a\hat{x}_k = \hat{x}(tx_n - th_j) \\ dt\hat{x}_k = \hat{t}x_k - \hat{t}x_{k-1} \end{array} \right.$$

et

$$\left\{ \begin{array}{l} \hat{t}h_k = tx_{n-i} \\ a\hat{h}_k = \hat{h}(tx_n - tx_{n-i}) \end{array} \right. \quad \text{ou} \quad \left\{ \begin{array}{l} \hat{t}h_k = tx_n - th_j \\ a\hat{h}_k = \hat{h}(tx_n - th_j) = ah_j \\ dt\hat{h}_k = \hat{t}h_k - \hat{t}h_{k-1} \end{array} \right.$$

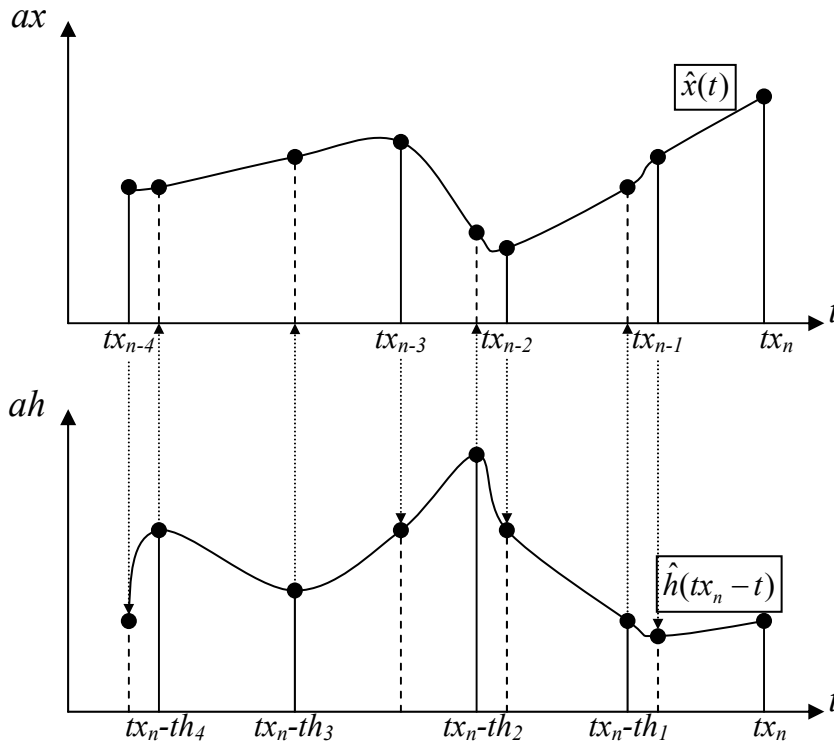


Figure 44 : Echantillonnage des points manquants par interpolation

Il est à noter que les valeurs $\hat{t}x_k$ et $\hat{t}h_k$ sont égales, les signaux étant échantillonnés aux mêmes instants et qu'il y a deux nouvelles interpolations à chaque nouveau calcul. Conformément à l'Équation 13, l'amplitude du signal de sortie est égale à la somme des surfaces du produit des fonctions dans chaque morceau élémentaire (cf. Figure 45).

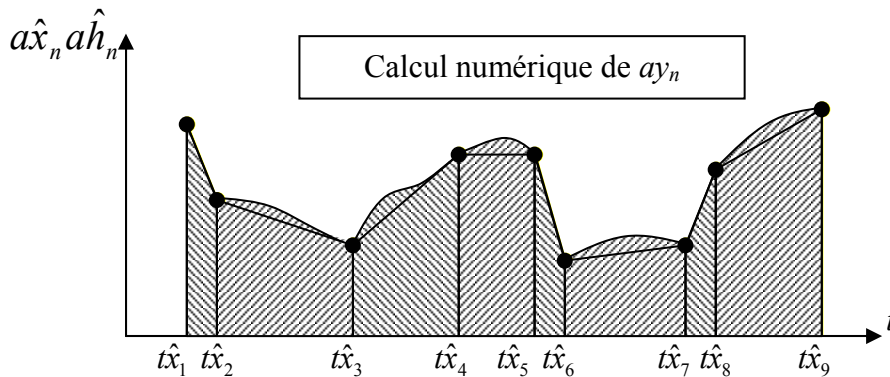


Figure 45 : Calcul de l'amplitude de sortie du signal ay_n

IV.3.1.2 Algorithme de convolution « asynchrone »

A ce stade, il est désormais possible de définir un algorithme itératif pour calculer le produit de convolution « asynchrone ». Ainsi, le calcul d'un échantillon de sortie suit les étapes suivantes [AES 04] :

1. Interpolation du signal x aux instants du signal h retourné et décalé de tx_n .
2. Interpolation du signal h retourné et décalé aux instants du signal x
3. Fusion des deux séries d'instants tx_{n-i} et $(tx_n - th_j)$ pour déterminer les instants $t\hat{x}_k$, où i et j sont des indices de boucles permettant de parcourir les signaux x et h .
4. Somme des surfaces élémentaires définies par les nouveaux intervalles $\{t\hat{x}_k - t\hat{x}_{k-1}\}$

Il est possible d'améliorer légèrement l'algorithme ci-dessus en suivant les étapes décrites dans la suite correspondant au schéma donné Figure 46. Lorsqu'un nouvel échantillon est prélevé en entrée, le calcul commence. Pour garantir lors de la première comparaison que les deux intervalles de temps sont synchronisés, les indices i et j sont mis à zéro. Parallèlement, les données du nouvel échantillon sont stockées en mémoire. Chaque itération débute par la recherche du minimum des temps entre tx_{n-i} et $(tx_n - th_j)$ afin de déterminer lequel des signaux entre x et h doit être échantillonné (Recherche de la plage élémentaire sur la figure). En fonction de ce minimum, l'aire est calculée et accumulée aux surfaces des itérations précédentes. Lorsque tous les échantillons de la fonction h ont été utilisés, l'amplitude de sortie est égale à la valeur des surfaces accumulées et le processus de calcul se termine. Cet algorithme présente deux avantages : il balaie une seule fois x et h et n'utilise que les données des signaux x et h . En effet, il ne nécessite pas le calcul explicite des instants $t\hat{x}_k$ pour déterminer les plages élémentaires.

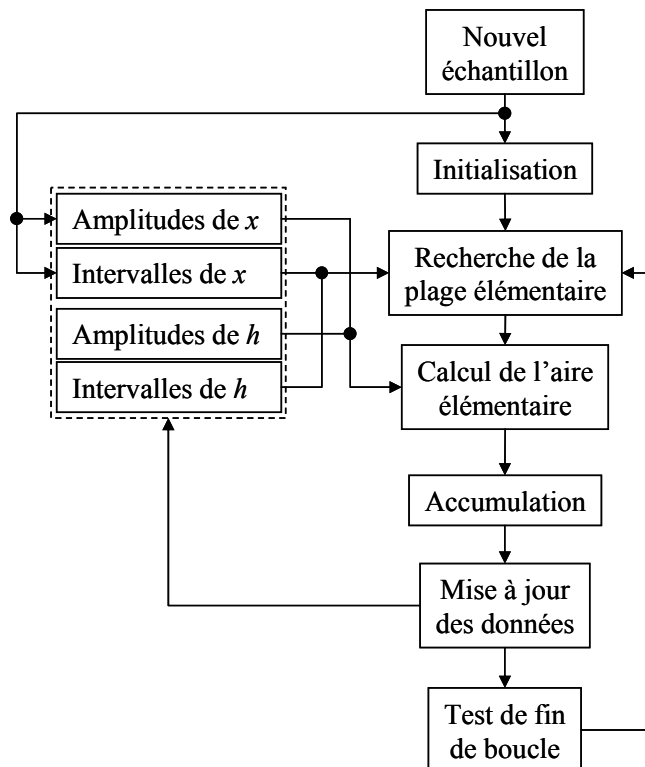


Figure 46 : Organigramme de l'algorithme itératif du calcul de la convolution « asynchrone »

Enfin, pour des raisons de concision, je n'aborde pas ici le choix du type d'interpolateur à utiliser. Néanmoins, il est important de bien le choisir en fonction de l'application. En effet,

les erreurs de calcul commises et la complexité de l'algorithme en dépendent. Il en est de même pour la méthode de calcul des surfaces élémentaires. Cette étude est détaillée dans la thèse de Fabien Aeschlimann [AES 06].

IV.3.1.3 Architecture matérielle

La structure implémentant l'algorithme de convolution « asynchrone » doit être suffisamment versatile pour s'accommoder des variations du nombre d'opérations dans le calcul de la sortie. Une structure directe, telle qu'elle serait implémentée dans le cas synchrone, ne respecterait pas ce critère car le nombre d'opérations est connu. La solution proposée est pourtant basée sur une architecture itérative comparable à celle que l'on peut rencontrer avec les filtres RIF synchrones (cf Figure 47).

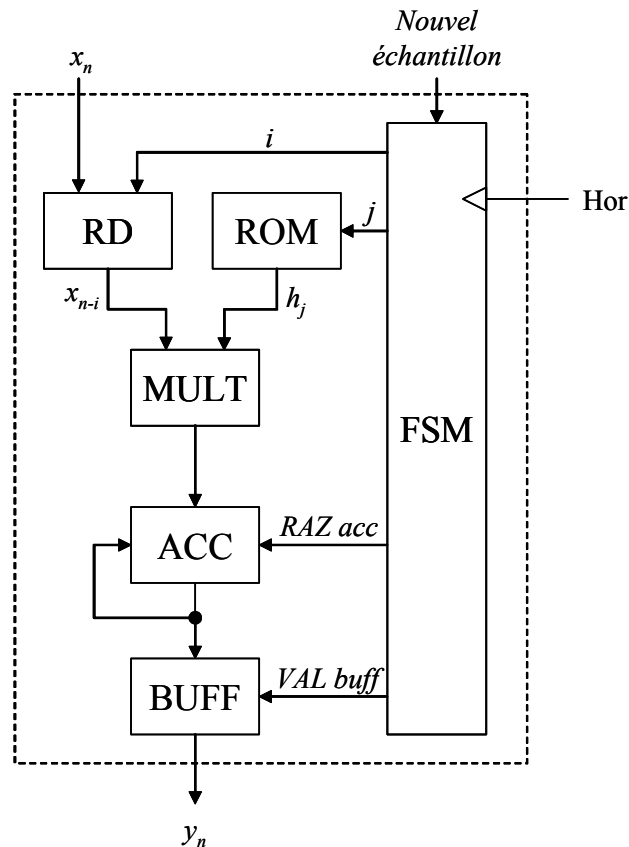


Figure 47 : Architecture itérative d'un filtre RIF synchrone

Dans le cas non uniforme, la structure itérative de la Figure 47 peut être modifiée pour implémenter le produit de convolution « asynchrone » avec une interpolation à l'ordre 0. L'architecture est présentée Figure 48 [AES 04]. On peut remarquer qu'il y a quatre registres à décalage au lieu d'un seul. En effet, deux sont destinés à stocker les amplitudes et les intervalles de temps des échantillons d'entrée ($RD ax$ et $RD dtx$) tandis que les deux autres remplacent la ROM pour mémoriser les coefficients de la réponse impulsionnelle et ses intervalles de temps ($RD ah$ et $RD dth$). Cette architecture très générale peut donc également exploiter des réponses impulsionnelles échantillonnées non uniformément. L'usage de filtre à réponse impulsionnelle non uniformément échantillonnée n'est pas abordé dans ce document, des études étant encore en cours. La structure comporte également deux multiplieurs (soit un de plus que dans le cas uniforme), un accumulateur (ACC) et un buffer ($BUFF$). Il est à noter que l'architecture du filtre asynchrone introduit trois nouveaux blocs : la mise à jour des intervalles de temps (MAJ) et des adresses de registres (ADD), et le calcul de l'intervalle de temps minimum (MIN). En revanche, il n'y a plus de machine à état pour piloter les blocs

mais un contrôle local asynchrone (non représenté sur la figure pour des raisons de clarté). Enfin, il est utile de préciser que le flot de calcul est comparable à celui d'une structure classique.

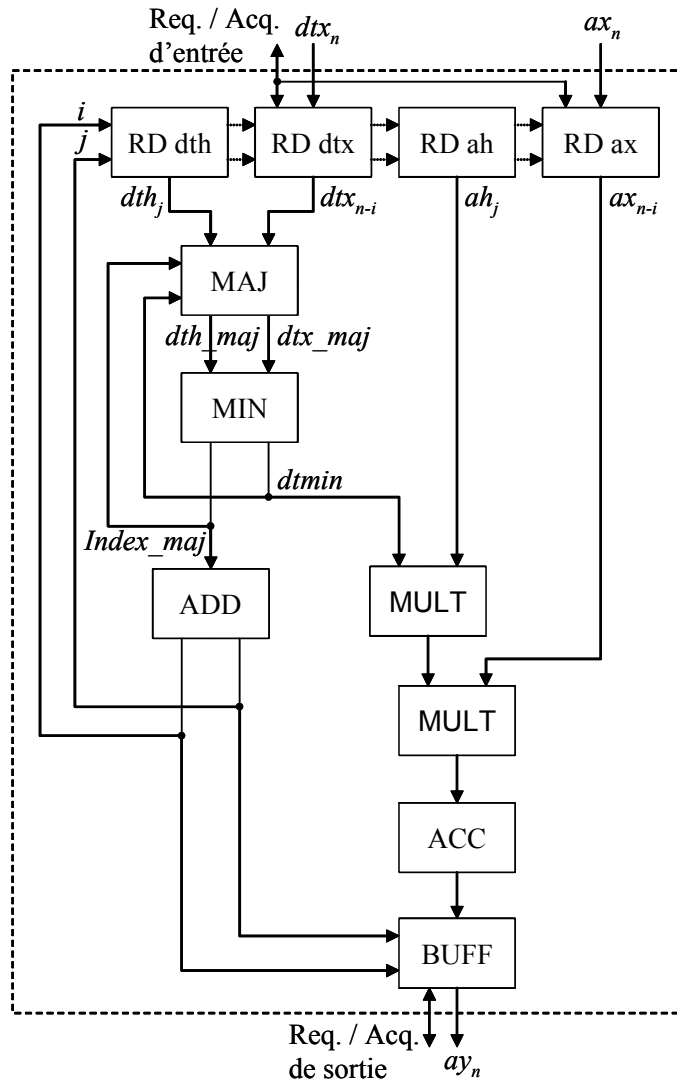


Figure 48 : Architecture itérative d'un filtre RIF « asynchrone »

IV.3.2 Le filtrage RII

IV.3.2.1 Introduction

Un filtre à réponse impulsionnelle infinie est toujours conçu selon le principe suivant : en premier lieu, un gabarit du module et de la phase en fonction de la réponse en fréquence est spécifié. Dans un second temps, la fonction de transfert d'un filtre analogique respectant le gabarit de départ est calculée à partir d'une fonction d'approximation telle que celles de Butterworth ou Chebyshev. La fonction de transfert du filtre numérique est alors déduite après approximation de la variable p , définissant le plan complexe de Laplace, par la variable z , définissant le plan complexe Z . Une équation récursive reliant les échantillons de sortie et d'entrée est ainsi obtenue permettant l'implémentation du filtre. Cependant, pour un même filtre analogique, plusieurs filtres numériques aux performances variables peuvent être conçus en fonction du choix du schéma numérique.

Quel que soit le type d'échantillonnage, le point de départ est la fonction de transfert du filtre analogique $H(p)$. En effet, c'est au moment de la discrétisation que la technique d'échantillonnage rentre en ligne de compte. La fonction $H(p)$, décrite par l'Équation 15, est une fraction rationnelle de degré N . Il est à noter que $M \leq N$.

Équation 15

$$H(p) = \frac{S(p)}{E(p)} = \frac{\sum_{i=0}^M a_i p^i}{\sum_{i=0}^N b_i p^i}$$

IV.3.2.2 Représentation d'un filtre numérique dans l'espace d'état

A ce stade, l'équation de récurrence entre les échantillons d'entrée et les échantillons de sortie ne peut être définie par la transformée en z . Afin de généraliser le schéma de discrétisation, il est nécessaire de passer sous la forme d'une représentation d'état. L'Équation 16 décrit la représentation d'état d'un système analogique où $x(t)$, $e(t)$ et $s(t)$ sont respectivement le vecteur d'état et les signaux d'entrée et de sortie.

Équation 16

$$\begin{cases} \frac{dx(t)}{dt} = Ax(t) + Be(t) \\ s(t) = Cx(t) + De(t) \end{cases}$$

Considérons les transformées de Laplace des signaux d'entrée et de sortie du système notés $E(p)$ et $S(p) = H(p)E(p)$ et posons un vecteur d'état de taille N noté dans le domaine de Laplace $X(p) = (X_0(p), \dots, X_{N-1}(p))^T$ où chaque élément est défini par :

Équation 17

$$X_k(p) = \frac{p^k}{\sum_{n=0}^N b_n p^n} E(p)$$

Pour les indices n allant de 1 à $N-1$, chaque élément du vecteur d'état $X_n(p)$ peut être calculé en fonction du précédent : $X_n(p) = pX_{n-1}(p)$. L'Équation 17 peut alors se mettre sous la forme suivante :

Équation 18

$$b_N p^N X_0(p) + \sum_{n=0}^{N-1} b_n p^n X_0(p) = E(p)$$

La transformée de Laplace du système s'écrit :

Équation 19

$$S(p) = \sum_{n=0}^N a_n X_n(p)$$

Soit :

Équation 20

$$\begin{aligned} S(p) &= a_N X_N(p) + \sum_{n=0}^{N-1} a_n X_n(p) \\ &= \sum_{n=0}^{N-1} (a_n - a_N b_n) X_n(p) + a_N E(p) \end{aligned}$$

En exploitant les Équation 18 et Équation 20, on déduit les matrices de la représentation d'état de l'Équation 16. La matrice A est appelée matrice d'état. Ses valeurs propres sont les pôles du filtre analogique. La matrice B est appelée matrice de commande, C matrice d'observation et D matrice de liaison. Comme le coefficient de plus haut degré du

dénominateur est non nul, la causalité du système est assurée (Le degré du dénominateur de la fonction de transfert doit être supérieur ou égal à celui du numérateur).

$$\text{Équation 21} \quad A_{(N \times N)} = \begin{pmatrix} 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \\ -b_0 & -b_1 & \dots & -b_{N-2} & -b_{N-1} \end{pmatrix}$$

$$\text{Équation 22} \quad B_{(N \times 1)} = (0 \quad 0 \quad \dots \quad 1)^T$$

$$\text{Équation 23} \quad C_{(1 \times N)} = (a_0 - a_N b_0 \quad \dots \quad a_{N-1} - a_N b_{N-1})$$

$$\text{Équation 24} \quad D_{(1 \times 1)} = (a_N)$$

La fonction de transfert est obtenue à partir de la représentation d'état grâce à la relation suivante :

$$\text{Équation 25} \quad H(p) = C(pI - A)^{-1} B + D$$

Pour obtenir un filtre numérique, il suffit alors de discrétiser l'espace d'état à l'aide d'un schéma numérique permettant d'approcher l'équation différentielle par une équation aux différences. Ce travail, mené avec Brigitte Bidégaray du laboratoire LJK (laboratoire de mathématiques appliquées), nous a amené à étudier de nombreux schémas numériques comme les schémas d'Euler, d'Euler rétrograde, bilinéaire et de Runge-Kutta. Par exemple, Poulton et Oksmann, qui ont étudié un cas similaire, ont adopté dans leurs travaux un schéma bilinéaire [POU 01]. Enfin, il est intéressant de noter que Fontaine et Ragot [FON 01] ont préféré résoudre directement l'équation différentielle en discrétisant la forme intégrale de l'espace d'état. En procédant de la sorte, l'approximation faite est que l'entrée est une fonction échantillonnée et bloquée, c'est-à-dire en marche d'escalier.

En utilisant le schéma numérique explicite de Runge-Kutta à l'ordre 4, nous obtenons la représentation d'état de l'Équation 26 qui est donnée à titre d'exemple.

$$\text{Équation 26} \quad \left\{ \begin{array}{l} x_n = \left(I + dt_n A + \frac{1}{2} dt_n^2 A^2 + \frac{1}{6} dt_n^3 A^3 + \frac{1}{24} dt_n^4 A^4 \right) x_{n-1} \\ \quad + dt_n \left(\frac{1}{2} I + \frac{1}{3} dt_n A + \frac{1}{8} dt_n^2 A^2 + \frac{1}{24} dt_n^3 A^3 \right) B e_{n-1} \\ \quad + dt_n \left(\frac{1}{2} I + \frac{1}{6} dt_n A + \frac{1}{24} dt_n^2 A^2 \right) B e_n \\ s_n = C x_n + D e_n \end{array} \right.$$

IV.3.2.3 Stabilité de la représentation d'état à temps discret

Lors du passage théorique à temps discret, une matrice e^{At} résulte de l'équation différentielle sans second membre. La condition de stabilité qui dépendait de la partie réelle des valeurs propres de la matrice A devient alors fonction du module des valeurs propres de e^{At} : celles-ci doivent en effet être de module inférieur à 1 c'est-à-dire être incluses dans le cercle unité. Or cette condition doit être respectée quel que soit le schéma utilisé pour

approcher e^{At} lors de la discrétisation de l'espace d'état. Ainsi, en généralisant, la représentation d'état à temps discret est donnée par l'équation suivante :

Équation 27
$$x_n = \Phi_n x_{n-1} + \Gamma_n e_{n-1}$$

où les valeurs propres de la matrice Φ_n qui approche e^{Adt_n} doivent appartenir au cercle unité. L'étude de stabilité dépend seulement d'une fonction T_n qui projette les valeurs propres λ de la matrice A vers celles de Φ_n . Ces dernières seront notées $\mu_n = T_n(\lambda)$.

Par exemple, pour la méthode d'Euler rétrograde, il est connu que l'image du demi plan gauche par la transformation définie par $T_n(\lambda) = (1-dt_n \lambda)^{-1}$ est un cercle de centre $1/2$ et de rayon $1/2$. Ainsi nous pouvons en déduire par avance que tout filtre analogique stable sera numériquement stable pour ce schéma. En revanche pour la méthode d'Euler progressive, la transformation est $T_n(\lambda) = 1+dt_n \lambda$; ainsi toute valeur propre à partie réelle négative est projetée dans une région dont les parties réelles sont inférieures à 1, incluant le cercle unité; la stabilité n'est donc pas assurée. Pour la méthode bilinéaire, la fonction $T_n(\lambda) = \frac{1+dt_n \lambda/2}{1-dt_n \lambda/2}$

projette le demi-plan gauche dans le cercle unité permettant de garantir lors de la conception une stabilité inconditionnelle. Ainsi, il apparaît qu'en fonction du schéma numérique retenu, un même filtre pourra être stable, conditionnellement stable ou instable. En d'autres termes, la zone de stabilité est inversement proportionnelle au pas de temps entre les échantillons. Ainsi, si le temps entre échantillons s'allonge, il se peut que la convergence de l'algorithme vers la solution analogique ne soit plus possible ou qu'elle le soit pour un ensemble de fonctions de filtrage dont les pôles sont bien localisés dans le plan complexe. Il faut donc prendre en compte cette spécificité dans la méthode de synthèse des filtres RII asynchrones.

IV.3.2.4 Flot de synthèse des filtres RII à échantillonnage non uniforme

Afin de prendre en compte la spécificité évoquée ci-dessus, les études menées conduisent à la définition d'un flot de conception permettant l'implémentation d'un filtre numérique stable. Le flot est représenté sur la Figure 49. Il y a deux entrées : le schéma numérique et le filtre analogique (c'est-à-dire les pôles obtenus à partir du gabarit). Pour un intervalle de temps donné, une transformation T_n est calculée ainsi que la région R_n du plan complexe analogique définie comme l'image inverse du cercle unité C_1 : $R_n = T_n^{-1}(C_1)$. Si le demi-plan gauche est inclus dans R_n alors le schéma numérique est une méthode stable pour tout filtre analogique. Dans le cas contraire, si les pôles du filtre analogique appartiennent à R_n , alors le schéma est stable pour ce filtre particulier. Sinon la condition de stabilité n'est pas respectée et le filtre sera instable. La solution consiste alors à réduire la longueur des intervalles de temps puisqu'elle agit sur la région R_n comme un facteur homothétique. Si la réduction de la longueur des intervalles n'est pas possible, il faut alors envisager de : soit modifier le filtre analogique en vérifiant que les pôles sont inclus dans R_n , soit de changer le schéma numérique. Cette méthode montre qu'il est possible de concevoir de façon systématique et sûre des filtres RII avec un échantillonnage non uniforme en temps, si l'on suit le flot de conception présenté sur la Figure 49.

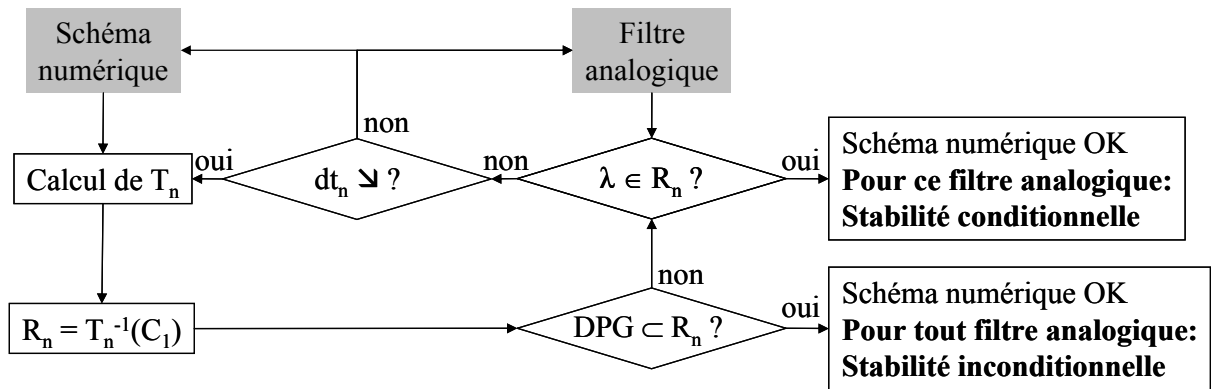


Figure 49 : Flot de conception d'un filtre RII « asynchrone » stable

IV.3.2.5 Architecture d'un filtre RII

Vu les nombreuses possibilités de choix de schémas numériques, il n'est pas surprenant que nous puissions trouver de nombreuses variantes d'implémentation de ce type de filtre. Si l'on considère par exemple un filtre utilisant la méthode bilinéaire, il est possible et pratique de le décomposer en blocs d'ordre 1 et 2. Ainsi, il est possible de réaliser un filtre à schéma bilinéaire en associant des blocs d'ordre 1 et 2 (Cf. Figure 50).

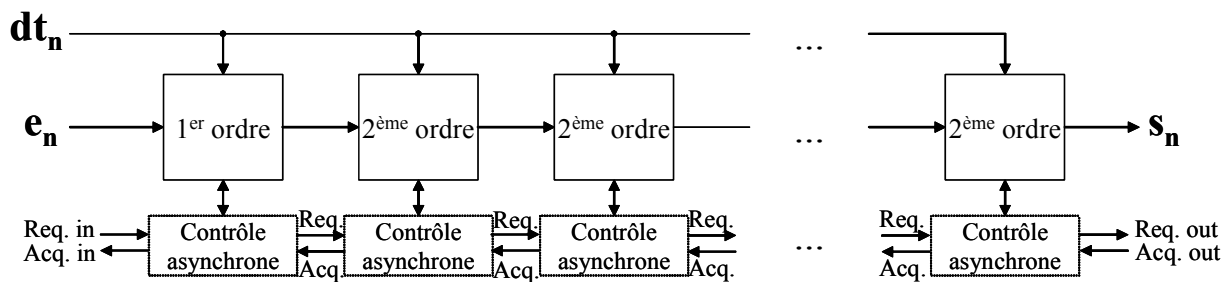


Figure 50 : Structure cascadée d'un filtre RII utilisant la méthode bilinéaire

Ce principe étant établi, il reste à définir les blocs élémentaires. Il est à noter qu'un contrôle local asynchrone pilote chaque bloc indépendamment. Lorsque le calcul d'une sortie de l'un des blocs est terminé, le contrôle envoie une requête au bloc suivant. Celui-ci peut alors recevoir les nouvelles données et acquitter. L'Équation 28 correspond à l'écriture de la représentation d'état du système numérique.

$$\text{Équation 28} \quad \begin{cases} x_{1,n} = \Phi_{11}x_{1,n-1} + \Phi_{12}x_{2,n-1} + \Gamma_1(e_n + e_{n-1}) \\ x_{2,n} = \Phi_{21}x_{1,n-1} + \Phi_{22}x_{2,n-1} + \Gamma_2(e_n + e_{n-1}) \\ s_n = c_1x_{1,n} + c_2x_{2,n} + De_n \end{cases}$$

Nous pouvons alors en déduire les deux structures élémentaires pour le 1^{er} et 2^{ème} ordre. Les deux architectures sont présentées sur la Figure 51 (Les éléments mémorisants sont notés L pour ne pas être confondus avec la matrice D).

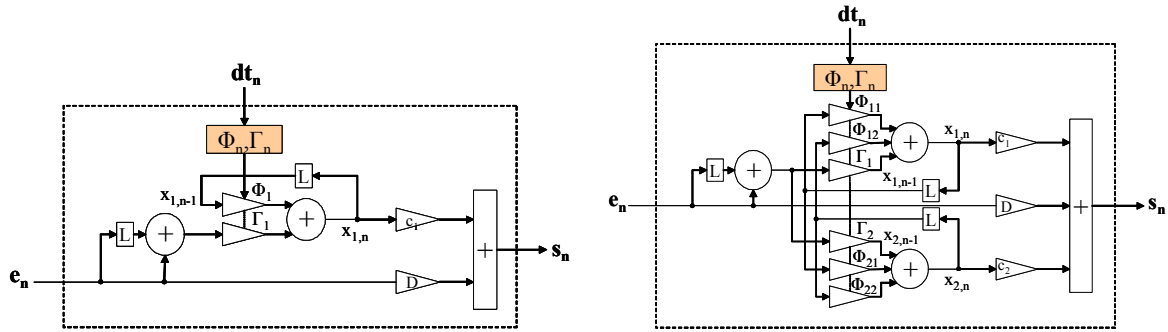


Figure 51 : Structures du 1er ordre et du 2ème ordre utilisant la méthode bilinéaire

Il est important de noter que le nombre de cycles nécessaires au calcul d'un échantillon en sortie est constant ce qui rend l'utilisation d'une représentation d'état numérique et de ces architectures dédiées très intéressantes pour le filtrage RII de signaux échantillonnés non uniformément. Cet exemple n'est absolument pas limitatif puisque d'autres architectures ont été étudiées avec des schémas numériques différents (Cf. [AES 06]).

IV.3.3 Algorithme de sélection d'activité (ASA)

Comme nous l'avons vu, l'utilisation d'un échantillonnage non uniforme peut être bénéfique pour réduire l'activité d'un système (et donc sa consommation) même si la complexité de calcul est légèrement plus complexe. L'algorithme de sélection d'activité (Activity Selection Algorithm ou ASA) [QAI 06][QAI 07a][QAI 07b][QAI 07c] contribue à en améliorer encore l'efficacité sur des signaux qui sont sporadiques. Le principe, très simple donné par l'algorithme ci-dessous, consiste à détecter les zones actives du signal et à les sélectionner avec une fenêtre de longueur variable.

```

Ti = 0
Ni = 0
While (dtn <= T0/2 and Ti <= T)
    Ti = Ti + dtn
    Ni = Ni + 1
end

```

où dt_n représente la distance temporelle entre deux échantillons et T₀ la période fondamentale du signal analysé. Le paramètre T indique la limite temporelle supérieure en secondes à ne pas dépasser pour la i^{ème} fenêtre dont la longueur est T_i. Enfin, N_i spécifie le nombre d'échantillons contenus dans la i^{ème} fenêtre. L'indice i est incrémenté à chaque nouvelle création de fenêtre. Il est à noter qu'il est aussi possible de limiter le nombre d'échantillons plutôt que la longueur temporelle de la fenêtre. En effet, il peut s'avérer intéressant lors d'une implémentation de maîtriser le nombre de mémoires.

L'intérêt de cet algorithme est d'adapter parfaitement la longueur des fenêtres en fonction de l'activité du signal. Ainsi, il est possible d'effectuer tout type de traitement sur les zones d'activité du signal sans avoir à effectuer des calculs inutiles sur les zones inactives. Il est à noter que cette méthode n'est pas transposable à l'échantillonnage uniforme car elle exploite l'information temporelle de distance entre les échantillons. Les travaux de thèse de Saeed Mian Qaisar montre que cette technique permet aussi de bénéficier de l'usage des outils usuels du traitement du signal (filtrage, FFT, ...) si l'on rééchantillonne uniformément les signaux à l'intérieur des fenêtres. Dans ces conditions, il devient possible d'adapter la cadence d'échantillonnage en fonction de l'application mais surtout du signal se trouvant dans la fenêtre. Ainsi, le rééchantillonnage devient adaptatif, chaque fenêtre pouvant définir sa propre fréquence d'échantillonnage [QAI 07b]. Par ailleurs, l'algorithme peut être amélioré afin de minimiser les effets des fuites spectrales qui apparaissent avec les signaux périodiques qui

sont capturés avec un nombre fractionnaire de cycles [QAI 06]. Cela évite notamment l'emploi de fenêtre de lissage qui ont un coût calculatoire non négligeable.

IV.3.4 Résultats expérimentaux

Avant de conclure ces travaux sur le traitement du signal à échantillonnage non uniforme et les architectures associées, il est important de donner quelques résultats quantitatifs. En effet, les filtres réalisés, bien qu'un peu plus complexes que leurs homologues synchrones, amènent tout de même à une réduction très significative de l'activité de calcul du fait du faible nombre d'échantillons prélevés (par rapport au cas régulier). Si l'on considère qu'il existe une relation quasi-linéaire entre activité et consommation, ces techniques offrent des possibilités nouvelles de réduction de la consommation des systèmes intégrés.

IV.3.4.1 Exemple avec un signal de parole

Pour fixer les idées, nous prenons une chaîne de traitement du signal comprenant un circuit A-ADC et un filtre numérique répondant au gabarit suivant :

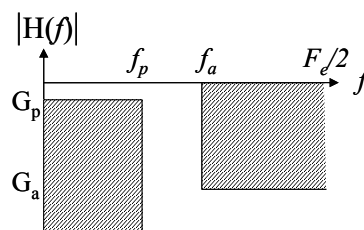


Figure 52 : Gabarit du filtre passe bas

- Fréquence de coupure : $f_p = 300\text{Hz}$,
- Fréquence minimale de la bande atténuée : $f_a = 400\text{Hz}$,
- Ondulation dans la bande passante : $G_p = 1\text{dB}$,
- Atténuation minimale de la bande atténuée : $-G_a = 40\text{dB}$.

Nous avons étudié le filtrage d'un signal de parole échantillonné non uniformément sur 15 niveaux. Les résultats, concernant une période d'une durée de 1,3 seconde, sont présentés sur la Figure 53. En étudiant l'évolution temporelle du nombre d'itérations de l'algorithme et le nombre d'échantillons en entrée utilisés pour le calcul d'une sortie, les relations entre activité du traitement et activité du signal peuvent être analysées. En règle générale, la charge de calcul suit le nombre d'échantillons en entrée dans les périodes d'activité. Dans les zones d'inactivité, on constate que la charge est constante. Ceci s'explique par la saturation du timer qui sert à mesurer les intervalles de temps entre les échantillons. Lorsque les variations du signal sont plus faibles que la valeur du quantum du convertisseur pendant une période plus longue que la plus grande valeur délivrée par le timer, il est nécessaire de rééchantillonner le signal d'entrée. On parle alors de saturation du timer. Durant les plages d'inactivité, le signal est donc échantillonné uniformément à une faible cadence. Sur la Figure 53, les zones d'inactivité, qui sont bien visibles, correspondent au silence (signal de parole). Il est à noter que l'approche exploitant la sélection d'activité (ASA) constitue ici un avantage car il évite d'échantillonner les zones inactives du signal.

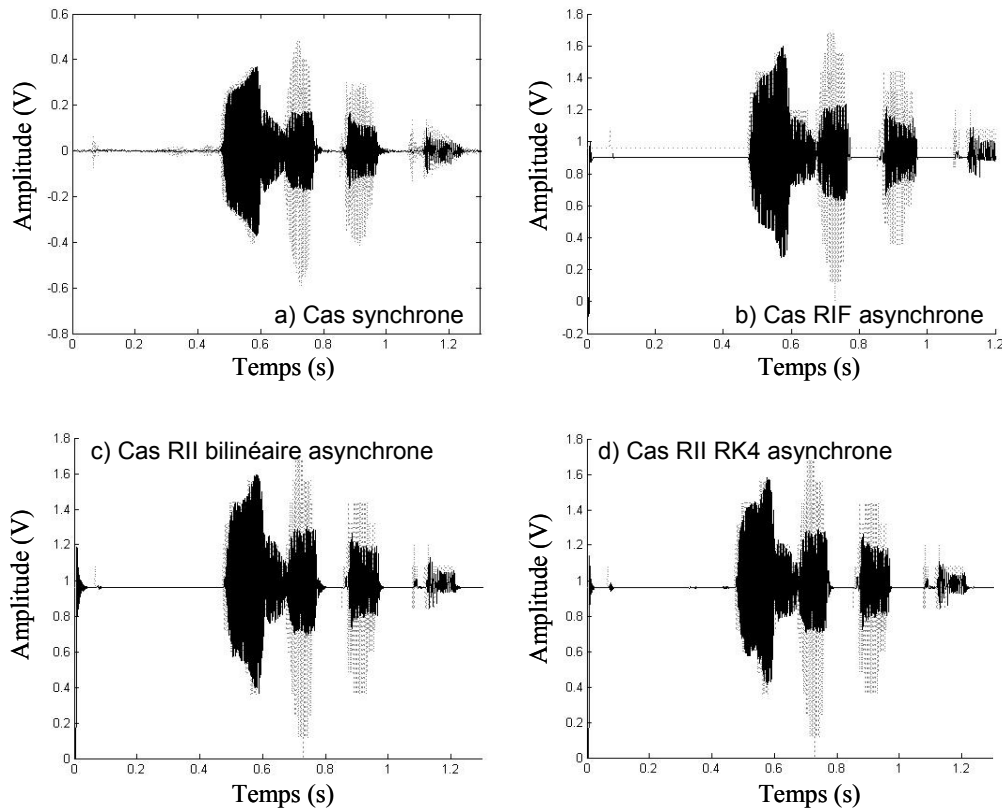


Figure 53 : Filtrage avec différents schémas numériques d'un signal de parole échantillonné non uniformément par traversée de niveaux

IV.3.4.2 Critère d'évaluation

L'exemple que nous venons de présenter nous a permis d'introduire le traitement d'un signal à activité variable par une chaîne asynchrone. Jusqu'à présent, le traitement numérique du signal est essentiellement basé sur un échantillonnage régulier et une conception synchrone des éléments de la chaîne. Nous avons vu précédemment que la prise en compte des caractéristiques du signal est primordiale pour diminuer l'activité de la chaîne et donc sa consommation (ce que l'on ne fait pas dans le cas synchrone). Nous avons vu également que le traitement d'un signal échantillonné non uniformément était un peu plus complexe que dans le cas synchrone. Il est donc important de définir un critère de comparaison des deux approches.

Les deux principales différences entre une chaîne de traitement synchrone et asynchrone sont d'une part la complexité du calcul et d'autre part le principe d'échantillonnage. On peut donc proposer le critère suivant :

Équation 29

$$G_{T,\varphi} = \frac{f_{sync}(\varphi) \cdot N_{sync}}{f_{async}(\varphi) \cdot N_{async}}$$

Soit un traitement φ donné, f_{sync} et f_{async} sont des fonctions modélisant la complexité de calcul du traitement φ dans les cas synchrone et asynchrone. Ces fonctions peuvent être à des niveaux d'abstraction variés. Dans le cas présent, nous nous limitons au calcul de nombre total d'additionneurs et multiplieurs nécessaires au filtrage. Ce modèle peut être affiné lors des phases de conception du système en y intégrant les caractéristiques matérielles des opérateurs comme la consommation, la rapidité, la surface, ... Pour une fenêtre temporelle T , N_{sync} et N_{async} représentent le nombre de points échantillonnés régulièrement et non uniformément sur cette durée. $G_{T,\varphi}$ est donc le rapport entre le nombre total de calculs sur la

durée T dans le cas synchrone sur le nombre total de calculs sur la même durée T dans le cas asynchrone. Si $G_{T,\varphi} < 1$, la chaîne synchrone sera plus performante pour réaliser le traitement φ , et si $G_{T,\varphi} > 1$, la chaîne asynchrone sera plus performante.

Le gain $G_{T,\varphi}$ a été évalué pour les trois filtres présentés dans ce manuscrit (cf. Équation 30, Équation 31, Équation 32).

$$\text{Équation 30} \quad G_{T,RIF} = \frac{2N \cdot N_{sync}}{4N \left(1 + \frac{N_{async}}{N_{sync}}\right) \cdot N_{async}} = \frac{1}{2 \left(1 + \frac{N_{async}}{N_{sync}}\right)} \frac{N_{sync}}{N_{async}}$$

$$\text{Équation 31} \quad G_{T,RII \text{ bilinéaire}} = \frac{(2N + 1 + 2N) \cdot N_{sync}}{(5N + 9N) \cdot N_{async}} = \frac{4N + 3}{14N} \frac{N_{sync}}{N_{async}}$$

$$\text{Équation 32} \quad G_{T,RII \text{ RK4}} = \frac{(2N + 1 + 2N + 2) \cdot N_{sync}}{(12N + 1 + 9N + 1) \cdot N_{async}} = \frac{4N + 3}{21N + 1} \frac{N_{sync}}{N_{async}}$$

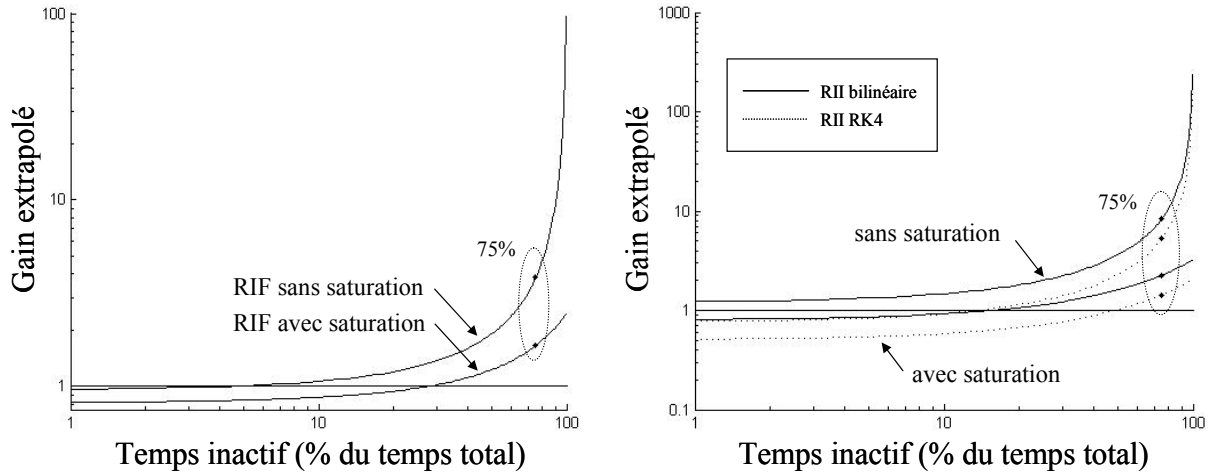
(Le paramètre N correspond à l'ordre du filtre)

L'évolution du gain en fonction du rapport entre le nombre d'échantillons synchrones et asynchrones a été analysée et on a déterminé le ratio entre échantillons synchrones et asynchrones qui donnait un gain unitaire. Les résultats correspondant à notre signal de parole sont consignés dans le tableau ci-dessous :

Traitement	RIF	RII Bilinéaire	RII RK4
Report	2,7	3,5	5,25

Le choix de la technologie dépend donc du rapport entre le nombre de points échantillonnés dans chaque cas, donc du processus d'échantillonnage. Cependant, il n'est pas possible de relier directement ce rapport au signal et plus particulièrement à son activité. En effet, dans le cas synchrone, le nombre d'échantillons est soit fixé par la fréquence maximale du signal, soit par des contraintes matérielles qui imposent la fréquence d'échantillonnage. Dans le cas asynchrone, le nombre d'échantillons dépend à la fois des variations du signal et du convertisseur (nombre de niveaux, profondeur du timer). Or, l'échantillonnage par traversée de niveaux ne génère pas forcément le même nombre de points pour deux signaux actifs différents; seules les zones inactives sont échantillonnées de la même manière. Il est donc nécessaire d'affiner l'analyse des résultats et prendre en considération d'autres paramètres. En effet, si l'on considère l'inactivité du signal comme un paramètre, il est possible de faire varier cette durée tandis que la zone active reste identique, c'est-à-dire échantillonnée de la même manière. Dans le cas synchrone, le silence est échantillonné régulièrement à la fréquence d'échantillonnage du système. Dans le cas asynchrone, le silence est également échantillonné régulièrement, mais à faible cadence, du fait de la saturation du timer. En conservant la durée active de la parole constante, nous avons fait varier la plage de silence entre 0% et 99% du temps total afin d'obtenir une extrapolation du critère sur l'activité du signal. Nous tenons bien à préciser que les résultats présentés sur la Figure 54 ne sont valables que dans le cas particulier du signal considéré. Ils permettent simplement d'étudier le gain de la technologie asynchrone sur la technologie synchrone si ce même signal avait eu une activité différente.

Par ailleurs, la saturation du timer a pour conséquence que les intervalles de temps sont toujours inférieurs ou égal à une valeur limite. Compte tenu de cette valeur, tous les filtres considérés sont stables même celui utilisant le schéma RK4. Cependant, lorsque le signal est constant, une série de points identiques est prélevée; le traitement devient redondant ce qui induit une activité accrue de la chaîne.



Traitement		G_T min (1%)	G_T à 75%	G_T max (99%)	Seuil $G_T = 1$
<i>RIF</i>	Avec saturation	0,81	1,65	2,45	28%
	Sans saturation	0,96	3,83	95,8	5%
<i>RII bilinéaire</i>	Avec saturation	0,81	2,26	3,22	16%
	Sans saturation	1,22	8,41	239	/
<i>RII RK4</i>	Avec saturation	0,53	1,49	2,11	46%
	Sans saturation	0,81	5,54	157	12%

Figure 54 : Gains extrapolés en fonction de l'activité du signal de parole pour les différents filtres

Pour le signal de parole considéré précédemment, nous pouvons noter que lorsqu'il présente une inactivité égale à 75% du temps total, l'utilisation d'une chaîne de traitement asynchrone du signal est préférable. Il est également intéressant de remarquer que pour tous les traitements, excepté le filtre RII bilinéaire sans saturation du timer, il existe un seuil en dessous duquel l'approche synchrone est à privilégier. Sans le prouver formellement, cette extrapolation permet de montrer que plus un signal possède de longue période d'inactivité, plus le rapport entre le nombre de points synchrones et asynchrones augmente et plus

l'approche asynchrone doit être préconisée pour le traitement numérique du signal. Par ailleurs, ce seuil se réduit considérablement si le traitement des points issus de la saturation du timer n'est pas considéré. Cela démontre que, quand cela est possible, l'usage de l'algorithme de sélection d'activité est bénéfique car il offre un fonctionnement correspondant au cas sans saturation. Dans ces conditions, on a en effet des ratios entre périodes d'activité et d'inactivité qui deviennent petits.

IV.4 Conclusion

Ce chapitre démontre tout l'intérêt que l'on peut avoir à reconsidérer complètement une chaîne de traitement du signal, y compris en exploitant des techniques d'échantillonnage non conventionnelles. Ainsi, il apparaît clairement qu'il est :

- possible de réaliser un dispositif de conversion analogique-numérique asynchrone à échantillonnage par traversée de niveaux,
- d'exploiter directement ces signaux pour y effectuer un traitement numérique,
- de concevoir des architectures spécifiques et adaptées à cette approche,
- et d'obtenir des gains substantiels sur l'activité et la consommation d'un système.

Par ailleurs, des travaux ont été réalisés sur des aspects concernant l'analyse spectrale et d'autres sont en cours pour développer des techniques d'analyse temps-fréquence exploitant l'échantillonnage par traversée de niveaux. L'ensemble de cette activité, orientée traitement du signal, m'a amené à reconsidérer beaucoup de choses et je suis aujourd'hui convaincu que la conception d'une chaîne de traitement du signal devrait commencer par l'analyse des caractéristiques du signal et pas uniquement l'analyse de l'application. En effet, il paraît opportun de choisir sa technique d'échantillonnage en fonction de la connaissance *a priori* que l'on a du signal. De nouveaux schémas d'échantillonnage seront étudiés à l'avenir pour répondre à des problèmes spécifiques utilisant des signaux dont la nature est bien définie comme c'est le cas dans les systèmes de communications, en vidéo, en médecine ou en sismique.

V. Conclusion et perspectives scientifiques

L'ensemble des travaux que j'ai menés avec mes collègues du TIMA, plus particulièrement Marc Renaudin et Gilles Sicard, et les nombreux étudiants qui sont passés dans le groupe CIS m'ont permis de dégager trois axes principaux de recherche. Le dénominateur commun de ces activités de recherche est l'absence de signal de synchronisation globale et de cadencement dans les dispositifs que nous étudions. C'est ainsi que mes activités se sont articulées autour de trois thèmes : la conception de systèmes intégrés asynchrones, les flots de conception des circuits asynchrones et le traitement numérique de signaux non uniformément échantillonnés.

V.1 *Systemes intégrés asynchrones*

L'étude de systèmes intégrés asynchrones nous a permis de mettre en évidence les bienfaits de ce type d'approche dans de nombreux secteurs tels que la faible consommation, la modularité de leur conception, la robustesse aux variations PVT, la synchronisation dans les systèmes GALS et la sécurité des données. On pourrait ajouter aux travaux présentés dans ce manuscrit les possibilités de maîtriser le rayonnement électromagnétique d'une puce dès la phase de conception [PAN 04]. La conception asynchrone présente des avantages indéniables par rapport à son homologue synchrone malgré un coût en surface généralement supérieur. Le ratio de surface entre un circuit synchrone et un circuit asynchrone est facile à obtenir mais son interprétation demeure délicate dans la mesure où le circuit asynchrone ne rend généralement pas exactement le même service que son « équivalent » synchrone. Par exemple, un circuit de chiffrement DES asynchrone exécute le même algorithme qu'un DES synchrone mais leurs consommations, vitesses et robustesses aux attaques par canaux cachés ne sont pas identiques. C'est pour ces raisons que la comparaison brute des surfaces est à mon sens peu pertinente.

Les résultats obtenus au sein du groupe ont démontré formellement des avancées significatives en démontrant que la consommation dynamique d'un système était mieux gérée avec un circuit asynchrone. En effet, seule la partie traitant les données consomme de l'énergie. Le reste est placé naturellement dans un état de « veille active ». La simple apparition de données (requête) à l'entrée d'un bloc suffit à le réactiver. Enfin, il apparaît qu'en matière de gestion d'énergie, il est aisé de tirer parti de la grande robustesse aux variations de tension des circuits asynchrones quasi-insensibles aux délais. En effet, comme les circuits QDI restent fonctionnels sur de larges plages de tensions, il est possible d'utiliser leur tension d'alimentation pour maîtriser leur vitesse de fonctionnement et du même coup leur consommation énergétique. Ce dispositif de gestion peut être très facilement contrôlé par un système d'exploitation approprié dans le cas d'un microprocesseur. L'avantage par rapport au processeur synchrone est que le dispositif est bien plus simple car il n'est pas nécessaire de gérer la fréquence de l'horloge en sus de la tension. Par ailleurs, un processeur asynchrone bénéficie par essence de la « veille active ».

La logique asynchrone offre aussi de grandes possibilités dans le domaine de la synchronisation et de l'échantillonnage de signaux. En effet, il est possible de concevoir des synchroniseurs sûrs grâce à l'utilisation de logique insensible aux délais. Même s'il est impossible d'empêcher l'apparition d'états métastables lors de l'échantillonnage d'un signal, la logique asynchrone (DI et QDI) permet néanmoins de s'en accommoder en autorisant l'attente (aussi longue soit-elle) de la résolution de la métastabilité. Cette possibilité offre de nombreuses perspectives que l'on peut exploiter pour concevoir des réseaux de communication sur puce (NoC) que l'on rencontre de plus en plus fréquemment dans les systèmes dits « Globalement Asynchrones, Localement Synchrones (GALS) ». En outre, il a

été montré que l'usage de logique asynchrone dans un réseau de communication sur puce (NoC) permettait de réduire les latences induites par les dispositifs de synchronisation traditionnellement utilisés.

L'ensemble de ces travaux nous a incité à étudier l'usage de la logique asynchrone sur des circuits programmables du commerce tout d'abord, puis à concevoir nos propres structures programmables. Il s'agit ici d'un vaste chantier car il existe de nombreuses façons de concevoir les circuits asynchrones et potentiellement autant de réaliser des circuits logiques configurables et asynchrones. Principalement pour des raisons académiques, nous nous sommes pas focalisés sur un style de logique asynchrone mais avons étudié des circuits programmables capables de supporter la plupart des styles de logique asynchrone. Ces travaux et ceux menés dans le domaine de la sécurité des données électroniques, qui constitue indéniablement une niche pour la logique asynchrone, nous a amené à considérer les aspects sécuritaires sur des structures programmables. En effet, le bénéfice d'une structure programmable pour les industriels est qu'elle offre de la flexibilité pour leurs produits qui peuvent aisément être mis à jour, intégrer des correctifs sécuritaires ou supporter de nouveaux algorithmes de chiffrement. Ce travail mené en collaboration avec l'ENST de Paris a montré qu'il était important de concevoir un matériel robuste aux attaques par canaux cachés mais qu'il était tout aussi essentiel de disposer d'une technique de configuration empêchant la fuite d'informations.

V.2 Outils de synthèse de circuits asynchrones

Si le premier chapitre a démontré la pertinence de l'approche asynchrone dans bon nombre de situations, il n'en demeure pas moins que l'on peut rester dubitatif quant aux moyens et aux outils qu'il faut déployer pour réaliser des circuits asynchrones complexes. Sans aborder le problème de la formation des ingénieurs aux techniques asynchrones, il nous est apparu très rapidement indispensable de proposer un flot de conception cohérent et capable de réaliser la synthèse de circuits asynchrones complexes. Ce travail a été initié dès 1999 et a donné lieu à plusieurs thèses.

Le premier travail a été de définir un ensemble de modèles capables de décrire efficacement les circuits asynchrones. Les modèles de plus haut niveau ont été tirés des travaux d'Alain Martin (Caltech). Nous sommes partis des descriptions CHP telles qu'elles sont pratiquées à Caltech et nous les avons enrichies. Nous avons intégré la possibilité de gérer la hiérarchie et la notion de composants mais aussi nous avons ajouté des notions permettant de différencier les choix déterministes et non déterministes par exemple. Cette étude nous a aussi conduit à définir des règles d'écriture garantissant que nos outils seraient à même de synthétiser nos circuits. Ces règles, appelées DTL, sont le pendant asynchrone des règles RTL du synchrone. Enfin, il a été nécessaire de définir des formes intermédiaires supportant la conversion depuis un langage de haut niveau tel que CHP et facilitant les optimisations et la projection technologique sur des circuits asynchrones. Les travaux effectués au sein du groupe ont montré la pertinence de l'utilisation de formats intermédiaires tels que les réseaux de Petri et les diagrammes de décision multi-valués.

Sur la base de ces modèles formels, deux techniques de synthèse ont été développées. La première exploite directement le réseau de Petri et extrait les équations de dépendance utiles à la construction d'un circuit asynchrone QDI. La seconde utilise les diagrammes de décision multi-valués (MDDs). Cette dernière possède l'avantage de garantir que le circuit généré sera réellement QDI. En effet, il a été formellement prouvé que l'algorithme de synthèse n'effectue que des transformations qui préservent les propriétés d'insensibilité aux délais des circuits. Ainsi les propriétés de robustesse des circuits QDI, essentielles pour être tolérant aux variations PVT (Process, Voltage, Temperature) ou réaliser une synchronisation

sûre, sont garanties par construction. Cela pourrait bien s'avérer à terme comme un élément important de choix de cette technologie.

Enfin, toujours dans le domaine des preuves formelles, des travaux ont été menés pour réaliser des moniteurs d'observation asynchrones capables de surveiller des systèmes intégrés complexes synchrones ou asynchrones. L'avantage de ce type de technologie est sa capacité à être très robuste (comparativement à un circuit synchrone) et à s'accommoder des états métastables. Ainsi, il a été possible de réaliser des observateurs matériels à partir d'une description PSL (Property Specification Language) prouvés corrects par construction et d'une complexité proportionnelle à l'expression à vérifier. Ces petits circuits présentent un grand intérêt pour la surveillance et la sécurité de systèmes complexes tels qu'un avion ou un système intégré multiprocesseur (MPSoC).

V.3 Systèmes à échantillonnage non uniforme

Dans ce document, je me suis efforcé de démontrer que l'usage d'une horloge d'échantillonnage n'était pas toujours pertinent. En effet, il apparaît qu'un échantillonnage uniforme respectant le théorème de Shannon est souvent très contraignant pour le dispositif matériel de traitement du signal. Ce constat nous a amené à étudier de nombreuses solutions alternatives et reconsidérer dans son ensemble la chaîne de traitement numérique du signal ainsi que la théorie associée.

Cette aventure, débutée en 2000, a commencé par l'étude et la conception d'un dispositif d'échantillonnage dit « par traversée de niveaux ». C'est ainsi que nous avons réalisé un prototype de convertisseur analogique numérique asynchrone (CANA ou A-ADC en anglais) en technologie CMOS 130 nm. On parle ici de CAN asynchrone car le circuit dispose d'un mécanisme exploitant de la logique asynchrone mais aussi parce que les instants d'échantillonnage ne sont pas connus à l'avance. Même si cette technique d'échantillonnage n'était pas la seule envisageable, nous l'avons étudiée en profondeur car elle nous a mis en perspectives beaucoup de possibilités dans le domaine de la faible consommation. Il est à noter que cet échantillonnage que l'on peut qualifier d'événementiel se marie très bien avec la logique asynchrone qui elle aussi fonctionne par événements.

La réalisation du dispositif de conversion n'a été en fait que le premier pas car il a fallu ensuite aborder des problèmes ardu de mathématique afin d'exploiter les échantillons produit par notre CANA. Les études ont été menées sur plusieurs fronts et ont permis d'écrire des algorithmes de filtrage RIF et RII ainsi que de concevoir les architectures matérielles associées. Les travaux ont également montré que l'on pouvait significativement réduire l'activité et donc la consommation de ces dispositifs de traitement du signal. Il reste néanmoins un certain nombre de points délicats. Même si nous savons calculer le spectre d'un signal non uniformément échantillonné, nous nous sommes heurtés à l'absence d'une « transformée de Fourier » pour l'échantillonnage non uniforme. C'est notamment une des raisons qui nous a incitée à étudier les techniques de sélections d'activité couplées à un rééchantillonnage (uniforme cette fois !) afin de réaliser certaines opérations pour lequel l'arsenal mathématique n'est pas encore été développé.

V.4 Perspectives scientifiques

L'ensemble de ces travaux de recherche mené au sein du laboratoire TIMA depuis 8 ans me conduit à imaginer un certain nombre de perspectives que l'on pourrait suivre dans les années à venir. Avant de définir des orientations, je pense qu'il est très important de dresser un constat. En effet, après avoir connu entre 1990 et 2000 une phase d'accélération, la microélectronique est probablement entrer dans une phase de refondation. Les méthodes et techniques subissent aujourd'hui une profonde mutation pour au moins deux raisons. La première est liée à l'évolution de la technologie vers les dimensions décanométriques et les

faibles tensions et la seconde est liée à la complexité des systèmes intégrés (le matériel et le logiciel) qui a crû exponentiellement ces dernières années. Une mutation est en train de s'opérer à plusieurs niveaux au point que les méthodes de conception traditionnelles sont en passe d'être remplacées.

V.4.1 Conception dans les technologies décanométriques

Avec l'avènement des technologies décanométriques, il apparaît de nouvelles contraintes de conception : une grande sensibilité aux variations des procédés de fabrication, des tensions d'alimentation très basses (inférieures au Volt). A cela s'ajoute les contraintes applicatives qui imposent de plus en plus souvent de maîtriser la consommation, voire le rayonnement électromagnétique. L'ensemble de ces difficultés semble être très favorable à la conception de circuits asynchrones. En effet, ces derniers, dans leur version insensible aux délais, offrent une très grande robustesse aux variations des procédés de fabrication, des tensions ou de la température. C'est aspect risque de devenir prédominant à l'avenir malgré les efforts consentis par les technologues pour amoindrir l'impact des variations de « process ». La logique asynchrone est probablement une réponse possible. Par ailleurs, vis-à-vis du rayonnement électromagnétique, il est possible d'utiliser la logique asynchrone pour en réduire ces effets. Mais les potentiels de la logique asynchrone ne s'arrêtent pas là. Son insensibilité aux délais lui offre des perspectives dans le domaine de la synchronisation et de l'arbitrage qui sont des dispositifs essentiels dans les communications radiofréquences ou les réseaux sur puce inscrits dans un paradigme GALS. La logique asynchrone ne peut probablement pas répondre à tous ces défis mais elle sera probablement une réponse appropriée dans beaucoup de situations.

Une autre façon de lutter contre les variations de « process » est de concevoir de façon répétitive. Ainsi, comme c'est déjà le cas depuis fort longtemps avec les mémoires, il est possible de rajouter des blocs redondants susceptibles de remplacer les blocs défectueux. Cette approche est sans aucun doute bénéfique à une augmentation du rendement lors de la fabrication. C'est ainsi qu'une évolution naturelle devrait être la conception de systèmes constitués de blocs identiques. Nous observons déjà une amorce de ce phénomène, plutôt pour un gain de puissance de calcul il est vrai, avec les architectures « multi-cœurs » et multiprocesseurs. Si l'on ajoute à cela les contraintes applicatives qui demandent toujours plus de flexibilité, je n'ai pas de doute que nous allons voir apparaître des systèmes constitués de blocs identiques et configurables. Ainsi, il sera toujours possible d'ajuster un design même après fabrication de la puce et de réduire les pertes de fabrication. Ce concept sera probablement étendu à une caractérisation des performances de chacun des blocs lors d'une phase d'analyse (à l'initialisation par exemple). Cette mesure servira à programmer et à répartir intelligemment la charge de traitement en fonction des performances intrinsèques de chacun des blocs. Si l'on y ajoute la possibilité de déconnecter, de réguler la tension (Dynamic Voltage Scaling) et la vitesse de chacun de ces blocs (Dynamic Speed Scaling), on obtient une puce constituée de blocs programmables capables de gérer les contraintes de faibles consommations de nombreuses applications. Enfin, j'ajouterai que l'usage d'un bloc asynchrone peut aider à exploiter localement le contrôle de la tension et des polarisations de substrat par un usage adéquat des nombreux signaux de synchronisation existants dans ce type de circuit.

Les circuits analogiques sont également très fortement impactés par ces évolutions. Leur conception va devenir plus délicates, d'une part du fait des variations de « process » et, d'autre part des très faibles tensions d'alimentation. En effet, la simple construction d'une paire différentielle ne pourra plus se faire selon les schémas traditionnels, les transistors n'étant plus correctement polarisés du fait de l'usage de tension inférieure au volt. Il ne sera plus possible d'empiler plus de deux transistors entre masse et alimentation. Il faudra donc

recourir à de nouvelles pratiques et à des « astuces » pour parvenir à construire les fonctions analogiques élémentaires. De plus, l'abaissement des tensions rendra la tâche d'autant plus délicate aux designers que la sensibilité au bruit sera accrue. En outre, les variations de « process » vont imposer aux designers analogiciens de nouvelles règles de dessin pour les masques. Il faudra peut-être envisager un design très régulier afin de limiter ces effets. Les solutions viendront peut-être de solutions de plus haut niveau. Par exemple, la construction d'oscillateurs impose des contraintes drastiques sur le bruit de phase pour les systèmes de radiocommunication. Là aussi, il sera bon de chercher des solutions alternatives au design traditionnel comme des oscillateurs asynchrones (oscillateurs avec un dispositif de requêtes et d'acquiescements). Des études préliminaires ont montré que le bruit de phase était plus limité avec ce type d'approche.

V.4.2 Synthèse de systèmes complexes et sûrs

L'apparition des technologies décanométriques va de pair avec l'apparition de nouvelles approches de conception à haut niveau car la complexité des circuits a crû très rapidement ces dernières années. Cette croissance a été tellement rapide qu'il est apparu un manque de réactivité entre les attentes d'un marché très demandeur de nouveautés et les capacités de l'industrie à y répondre dans les temps. Aujourd'hui, nous pouvons observer un grand nombre de méthodes permettant de réaliser des synthèses de haut niveau. Ces approches, décrivant souvent le circuit dans un style « C-like », offre la possibilité de concevoir très efficacement la partie opérative d'un système. Elles s'avèrent souvent moins appropriées (ou pas appropriées) à la conception complète de systèmes complexes.

Ce constat et l'étude des techniques de synthèse des circuits asynchrones me laisse penser que l'évolution naturelle des outils devrait s'orienter vers une synthèse agissant sur des modèles de type TLM (Transaction Level Modeling). En effet, ce niveau est relativement adéquat pour concevoir des systèmes complexes avec une lisibilité du modèle acceptable pour un concepteur. Par ailleurs, ce niveau offre la possibilité de maîtriser de nombreux détails tout en cachant des signaux tels que les signaux de synchronisation et d'initialisation. La synthèse d'un système complexe et de ces « transacteurs » pose encore des difficultés et notamment la possibilité de gérer le non déterminisme. Il est donc essentiel de développer de nouvelles approches de synthèse pour concevoir des systèmes GALS afin de les architecturer autour d'un système de communication adapté et spécifique et non autour d'un réseau générique produit par un générateur. A plus long terme, une synthèse d'encore plus haut niveau sera probablement envisageable mais pour cela il faudra encore franchir de nombreuses étapes.

En marge de ces systèmes complexes, il apparaît aussi des besoins pour des systèmes plus simples pour lesquels il est important de garantir le bon fonctionnement du dispositif, y compris dans un environnement hostile. C'est typiquement le cas des systèmes électroniques pour l'aéronautique, l'espace ou l'automobile. Par exemple, l'aviation a défini des normes (regroupées sous le label DO254) qui impose à l'électronique de résister à diverses agressions. Aujourd'hui, il s'agit plus d'un catalogue de vérifications à effectuer que de règles de conception. Les choses sont en train d'évoluer et le secteur aéronautique est prêt à accepter des techniques à base de preuves formelles pour garantir le bon fonctionnement du système. Cette évolution devrait engendrer à terme des techniques de synthèse prouvées mathématiquement et de nouveaux outils qui seront certifiés par des organismes indépendants. Dans ce contexte où la fiabilité et la robustesse des systèmes sont primordiales, les circuits asynchrones peuvent apporter des solutions à ces besoins. Néanmoins leur acceptation ne sera possible que si des preuves formelles sont données. Il est vrai que le mot « asynchrone » suscite plus de méfiance que de confiance (à tort, hélas !).

V.4.3 Optimiser en sortant du cadre restrictif de la microélectronique

Nous avons vu que de nombreuses solutions existent pour optimiser les circuits en consommation, vitesse, rayonnement. Toutefois, les gains cumulés de ces techniques (synthèse, bibliothèque de cellules, gestion de l'énergie, ...) ne permettent pas de gagner plusieurs ordres de grandeur par rapport à une solution classique. Il faut donc rechercher des solutions plus en amont de la conception afin d'obtenir ces gains supplémentaires. Avec les études menées sur les techniques de gestion de l'énergie par les systèmes d'exploitation, les algorithmes asynchrones et de nouvelles techniques d'échantillonnage des signaux analogiques, il devient possible d'imaginer de nombreuses possibilités d'optimisation. Par exemple, les travaux de Bruno Galilée ont montré qu'un algorithme de partage des eaux asynchrones [GAL 07] permettait d'augmenter d'un facteur 1000 la vitesse tout en réduisant la consommation par rapport aux solutions habituelles.

De même, les techniques d'échantillonnage et de traitement du signal associées mises au point au laboratoire permettent d'envisager des gains substantiels inatteignables par les simples progrès des techniques de conception. Ainsi, le choix de la technique d'échantillonnage n'est pas anodin et peut influencer grandement les performances d'un système. Aujourd'hui, tout le monde utilise le même schéma, mais je suis convaincu qu'à l'avenir nous aborderons le problème de l'échantillonnage différemment. En effet, il n'y a aucune raison de choisir le même schéma d'échantillonnage pour une modulation en fréquence et la capture d'un signal naturel provenant d'un sismographe ! Je pense que la bonne démarche à adopter est d'étudier les caractéristiques du signal avant de décider du procédé d'échantillonnage le plus adapté. Dans nos travaux, nous avons mis en évidence les bienfaits d'un échantillonnage par traversée de niveaux, mais il existe de nombreuses autres façons pour échantillonner.

Dans une chaîne de traitement du signal, l'échantillonnage n'est qu'une étape du système et il est opportun d'étudier la façon de traiter l'information. Bien évidemment, l'approche suggérée montre qu'il est possible de réaliser des fonctions avec des réductions d'activité allant jusqu'à deux ordres de grandeur, mais de nombreux obstacles mathématiques subsistent. Pour une part, je suis convaincu que les solutions mathématiques existent déjà et pour le reste, elles seront à trouver ! Ainsi, les systèmes de communication numériques, l'instrumentation et les systèmes embarqués devraient tirer parti de ces approches non conventionnelles. Par exemple, l'activité inutile d'une horloge d'échantillonnage et l'incapacité des systèmes traditionnels à déterminer les zones d'inactivité montrent combien la marge de manœuvre est grande pour optimiser la consommation, la vitesse ou le rayonnement. Cette remise en cause profonde d'une chaîne de traitement du signal ouvre d'immenses perspectives de recherche et montre qu'il est important d'optimiser les systèmes à haut niveau avant d'en entreprendre la conception.

Bibliographie

- [ABA 00] Y. Abarbanel et. al., FoCs: Automatic Generation of Simulation Checkers from Formal Specifications, in Computer Aided Verification, vol. 1855, LNCS, London, UK, 2000, pp. 538-542.
- [ABR 01] Abrial (A.), Bouvier (J.), Renaudin (M.), Senn (P.) and Vivet (P.), "A New Contactless Smart Card IC using On-Chip Antenna and Asynchronous Microcontroller". IEEE Journal of *Solid-State Circuits*, Vol. 36 (2001) 1101-1107.
- [ACC 07] Accelera, Aug. 07, <http://www.eda.org/ovl>.
- [AES 04] Aeschlimann F., Allier E., Fesquet L. et Renaudin M. "Asynchronous FIR Filter: Towards a New Digital Processing Chain". Tenth IEEE International Symposium of Asynchronous and Systems, ASYNC'04. 198-206. Crète, Grèce. 19-23 Avril 2004.
- [AES 06] Fabien Aeschlimann, Traitement du signal échantillonné non uniformément : algorithme et architecture, Thèse de doctorat INPG, soutenue le 6 février 2006.
- [ALL 02] E. Allier, L. Fesquet, M. Renaudin, G. Sicard, "Low-Power Asynchronous A/D Conversion", Twelfth International Workshop on Power And Timing Modeling, Optimization and Simulation (PATMOS), September 11-13, 2002, Sevilla, Spain.
- [ALL 03] Emmanuel Allier, Interface analogique numérique asynchrone : Une nouvelle famille de convertisseurs basés sur la quantification du temps, Thèse de doctorat INPG, soutenue le 27 novembre 2003.
- [ALL 03b] E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "A New Class of Asynchronous A/D Converters Based on Time Quantization", IEEE Async'03, Vancouver, Canada, May 12-16, 2003, pp. 196-205.
- [ALL 03c] E. Allier, L. Fesquet, M. Renaudin, G. Sicard, "Procédé et dispositif de conversion analogique-numérique", brevet n° FR2835365 (déposé le 31 janvier 2002), publié le 2003-08-01.
- [ALL 05] E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "Asynchronous Level Crossing Analog to Digital Converters", Special Issue on ADC Modelling and Testing of Measurement, Vol. 37, Issue 4 , June 2005, Pages 296-309.
- [BAR97] Bardsley A., Edwards D., Compiling the Language Balsa to Delay-Insensitive Hardware, in Kloos C. D. and Cerny E., Ed., Hardware Description Languages and their Applications (CHDL), (1997), pp. 89-91.
- [BAR 00] A. Bardsley and D. A. Edwards. The Balsa asynchronous circuit synthesis system. In Forum on Design Languages, September 2000.
- [BEI 74] J. Beister. A unified approach to combinational hazards. IEEE Transactions on Computers, C-23(6), pp. 566-575, June 1974.
- [BER 91] Berkel K. V., Kessels J., Roncken M., Saeijs R. and Schalijs F., The VLSI-programming language Tangram and its translation into handshake circuits, in Proc. European Conference on Design Automation (EDAC) (1991), pp 384-389.
- [BER 93] Berkel K. V., Handshake circuits: an asynchronous architecture for VLSI Programming, vol. 5 of International Series on Parallel Computation, Cambridge University Press, 1993.
- [BLU 00] Ivan Blunno and Luciano Lavagno. Automated synthesis of micro-pipelines from behavioral Verilog HDL. In Proc. International Symposium on Advanced

- Research in Asynchronous Circuits and Systems (ASYNC), pages 84-92. IEEE Computer Society Press, April 2000.
- [BOR 05] D. Borrione, M. Liu, K. Morin-Allory, P. Ostier, L. Fesquet, "On-Line Assertion-Based Verification with Proven Correct Monitors", IEEE Intl. Conf. on Information & Communications Technology, Dec. 2005, Cairo, Egypt, pp 125—143.
- [BOR 06] Dominique Borrione, Miao Liu, Pierre Ostier, Laurent Fesquet, "PSL-based Online Monitoring of Digital Systems" in *Advances in Design and Specification Languages for SoCs* (best selected contributions from FDL'05), Springer 2006.
- [BOU 06] M. Boulé, Z. Zilic, Efficient automata-based assertion-checker synthesis of PSL properties, in 24th IEEE International Conference on Computer Design (ICCD'06), nov., 2006.
- [BRE 04] V. Brégier, B. Folco, L. Fesquet, M. Renaudin, "Modeling and Synthesis of Multi-rail Multi-protocol QDI Circuits", IWLS'04, San Diego, USA, June, 2004.
- [BRE 07] Vivian Brégier, Synthèse automatisée de circuits optimisés prouvés quasi-insensibles aux délais, thèse de doctorat INPG soutenue le 14 septembre 2007
- [BRU 89] Brunvand E., Sproull R. F., Translating concurrent programs into delay-insensitive circuits, in *International Conference on Computer-Aided Design (ICCAD)*, (1989), IEEE Computer Society Press.
- [BRU 95] E. Brunvand, "Low latency self-timed flow-through FIFOs", *Proceedings of the 16th Conference on Advanced Research in VLSI*, Chapel Hill, NC, March 1995.
- [BUR 95] Burd (T.D.), Brodersen (R.W.), "Energy-Efficient CMOS Microprocessor Design". IEEE Proc. 28th Hawaiï Int. Conf. on System Sciences, Vol. 1 (Jan 1995) 288–297.
- [BUR 96] S. M. Burns. General condition for the decomposition of state holding elements. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*. IEEE Computer Society Press, March 1996
- [CHA 92] Chandrakasan (A.P.), Sheng (S.), Brodersen (R.W.), "Low Power CMOS Digital Design". IEEE Journal of Solid-State Circuits, Vol. 27 (April 1992) 473–484.
- [CHE 00] T. Chelcea et S. M. Nowick, "Low-Latency Asynchronous FIFO's Using token Rings", *Proceedings of the Sixth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC'00*, Eilat, Israel, 4-6 April 2000.
- [CHE 01] T. Chelcea et S. M. Nowick, "Robust Interfaces for Mixed-Timing Systems with Application to Latency-Insensitive Protocols", *Proceedings of the 38th Conference on Design Automation (DAC'01)*, Las Vegas, Nevada, United States, 13-16 March 2001.
- [CHE 04] T. Chelcea et S. M. Nowick, "Robust Interfaces for Mixed-Timing Systems", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 8, pp. August 2004.
- [COR 97] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev. Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers. *IEICE Transactions on Information and Systems*, E80-D(3):315-325, March 1997.

- [COR 02] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev. Logic Synthesis of Asynchronous Controllers and Interfaces. Springer-Verlag, 2002.
- [CHU 87] Chu T. A., Synthesis of self-timed VLSI Circuits from Graph-theoretic Specifications, PhD Thesis, Massachusetts Institute of Technology, 1987.
- [DAL 98] W. J. Dally et J. W. Poulton, "Digital Systems Engineering", Cambridge University Press, 1998.
- [DAV 93] Davis A., Coates B. and Stevens K., The PostOffice experience: Designing a large asynchronous chip, in Proc. of the 26th Annual Hawaii International Conference on System Science (1993), IEEE Computer Science Press, pp. 409-418.
- [DAV 97] A. Davis and S.M. Nowick, "An Introduction to Asynchronous Circuit Design", University of Utah technical report, Department of Computer Science, UUCS-97-013, Sept. 19, 1997.
- [DIJ 76] E. W. Dijkstra, A Discipline of Programming, Prentice Hall, Englewood Cliffs, N.J. 1976
- [DIK 99] C. Dike et E. Burton, "Miller and Noise Effects in a Synchronizing Flip-flop", IEEE Journal of Solid-State Circuits, vol. 34, no. 6, pp. 849-855.
- [DIN 02] Anh Vu Dinh Duc, Laurent Fesquet, Marc Renaudin, "Synthesis of QDI Asynchronous Circuits from DTL-style Petri-Net" IWLS-02, 11th IEEE/ACM International Workshop on Logic & Synthesis, New Orleans, Louisiana, June 4-7, 2002.
- [DIN 02b] A.V. Dinh Duc, J.-B. Rigaud, A. Rezzag, A. Sirianni, J. Fragoso, L. Fesquet, M. Renaudin, "TAST CAD Tools: Tutorial", tutorial given at the International Symposium on Advanced Research in Asynchronous Circuits and Systems ASYNC'02, Manchester, UK, April 8-11, 2002, TIMA internal report ISRN:TIMA-RR-02/04/01—FR, <http://tima.imag.fr/cis>.
- [DIN 03] Anh Vu Dinh Duc, "Synthèse automatique de circuits asynchrones QDI", Thèse de doctorat INPG, soutenue le 14 mars 2003.
- [EBE 91] Jo C. Ebergen. A formal approach to designing delay-insensitive circuits. Distributed Computing, 5(3):107-119, 1991.
- [ESS 02] Es Salhiene (M.), Fesquet (L.), Renaudin (M.), "Dynamic Voltage Scheduling for Real Time Asynchronous Systems". Twelfth International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS). Sevilla, Spain (11-13 September 2002).
- [FAN 96] Karl M. Fant and Scott A. Brandt. NULL conventional logic: A complete and consistent logic for asynchronous digital circuit synthesis. In International Conference on Application-specific Systems, Architectures, and Processors, pages 261-273, 1996.
- [FES 03] Laurent Fesquet, Mohammed Es Salhiene, Marc Renaudin, "La technologie asynchrone au service de la réduction d'énergie dans les systèmes embarqués", Annales des Télécommunications, Tome 59, n°7-8, juillet-août 2004.
- [FES 05] L. Fesquet, M. Renaudin, "A Programmable Logic Architecture for prototyping clockless circuits", FPL'05, Tampere, Finland, August 24-26, 2005, pp 293-298.
- [FES 06] Laurent Fesquet, Bertrand Folco, Mathieu Steiner and Marc Renaudin, "State-holding in Look-Up Tables: application to asynchronous logic", VLSI-SoC 2006, October 16-18, 2006, Nice, France, pp 12-17.
- [FLE 01] M. Fleischmann, "Crusoe Longrun Power Management". Transmeta Corporation internal report, 17 January 2001.

- [FOL 07] Bertrand Folco, Contribution à la synthèse de circuits asynchrones quasi insensibles aux délais, application aux systèmes sécurisés, thèse de doctorat INPG, soutenue le 4 octobre 2007.
- [FON 01] Fontaine L. et Ragot J. "Filtrage de signaux à échantillonnage irrégulier". *Revue du Traitement du signal* 18(2): 89-101. 2001.
- [FUH 99] R. M. Fuhrer, S. M. Nowick, M. Theobald, N. K. Jha, B. Lin, and L. Plana. Minimalist: An environment for the synthesis, verification and testability of burst-mode asynchronous machines. Technical Report TR CUUCS-020-99, Columbia University, NY, July 1999.
- [GAO 96] B. Gao. "A globally asynchronous locally Synchronous configurable array architecture for algorithm embeddings". PhD thesis, University of Edinburgh, December 1996.
- [GAL 07] Galilee B., Mamalet F., Renaudin M., Coulon P.Y., "Parallel asynchronous watershed algorithm-architecture", *IEEE Transactions on Parallel and Distributed Systems*, Volume: 18, Issue 1, 2007, pp 44-56.
- [GAU 03] Bruno Gaujal, Nicolas Navet, and Cormac Walsh, "A linear algorithm for real-time scheduling with optimal energy use", Technical report 4886, INRIA, 2003.
- [GIN 02] R. Ginosar, "Synchronization and Arbitration", Proceedings of the ACiD Summer School on Asynchronous Circuit Design, Grenoble, France, July 15-19 2002.
- [GOW 98] Gowan (M.K.), Biro (L.L.), Jackson (D.B.), "Power Considerations in the Design of the Alpha 21264 Microprocessor". *IEEE proc. 35th Design Automation Conference*, San Francisco, CA USA (June 1998) 726-731.
- [HO 02] Quoc Thai Ho, J.-B. Rigaud, L. Fesquet, M. Renaudin, R. Rolland, "Implementing asynchronous circuits on LUT based FPGAs", *The 12th International Conference on Field Programmable Logic and Applications (FPL)*, September 2-4, 2002, Montpellier (La Grande-Motte), France, pp 36-46.
- [HOA 78] Hoare C. A. R., *Communicating Sequential Processes*, *Communications of the ACM* 21, vol. 8, (April, 1978), pp. 666-677
- [IEE 05] IEEE Computer Society, IEEE standard 1850 for Property Specification Language, Reference manual, oct., 2005
- [IEE 05b] IEEE Computer Society, SystemVerilog IEEE standard 1800, Reference Manual, 2005.
- [ITR 07] <http://www.itrs.net/>
- [KAR 01] F. Karim, A. Nguyen et S. Dey, "An Interconnect Architecture for Networking Systems on Chips", *IEEE Micro*, vol. 22, no. 5, pp. 36-45, September/October 2002.
- [KUM 00] Kumar (P.), Srivastava (M.), "Predictive Strategies for Low-Power RTOS Scheduling". *IEEE Proc. Int. Conf. On Computer design: VLSI in computers and processors*, Austin TX (Sept. 2000) 343-348.
- [LI 03] Li (Y.W.), Patounakis (G.), Jose (A.), Shepard (K.L.), Nowick (S.M.), "Asynchronous datapath with software controlled on-chip adaptative voltage scaling for multirate signal processing application". *IEEE Proc. Int. Symp. On Asynchronous Circuits and Systems*, Vancouver, BC, Canada (May 2003) 216-225.
- [LIN 96] D. H. Linder and J. C. Harden., "Phased logic supporting the synchronous design paradigm with delay-insensitive circuitry", *IEEE Transactions on Computers*, 45(9):1031-1044, 1996.

- [MAN 96] Rajit Manohar and Alain J. Martin. Quasi-delay-insensitive circuits are Turing-complete. Second International Symposium on Advanced Research in Asynchronous Circuits and Systems (invited), March 1996.
- [MAR 81] Mark J. W. et Todd T. D. "A Nonuniform Sampling Approach to Data Compression". IEEE Transactions on Communications COM-29(1): 24-32. 1981.
- [MAR 90] Alain J. Martin. Programming in VLSI: From communicating processes to delay-insensitive circuits. In C. A. R. Hoare, editor, Developments in Concurrency and Communication, UT Year of Programming Series, pages 1-64. Addison-Wesley, 1990.
- [MAR 93] A.J. Martin, "Synthesis of Asynchronous VLSI Circuits", Internal Report, Caltech-CS-TR-93-28, California Institute of Technology, Pasadena, 1993.
- [MAR 01] Farokh Marvasti, Nonuniform Sampling : Theory and Practice, Springer, Nov., 2001, 912 pages
- [MAR 05] Marvasti F. A. "Recovery of Missing Samples using a Novel Iterative Method". International Workshop on Sampling Theory and Applications, SAMPTA. Samsun, Turkey. July 2005
- [MAY 90] D. May. Compiling OCCAM into Silicon. Developments in Concurrency and Communication. Edited by C.A.R. Hoare, Addison Wesley, pages 87-106, 1990.
- [MEN 07] Mentor Graphics, Aug. 2007
<http://www.mentor.com/products/fv/cdv/checkerware/index.cfm>
- [MOL 85] Molnar C. E., Fang T. P. and Rosenberger F. U., Synthesis of delay-insensitive modules, in 1985 Chapel Hill Conference on VLSI, (1985), H. Fuchs, Ed., Computer Science Press, pp. 67-86.
- [MOR 06] Katell Morin-Allory, Laurent Fesquet, Dominique Borrione, "Asynchronous on-line monitoring of PSL assertions", Rapid System Prototyping (RSP 2006), June 14-16, 2006, Chania, Crete.
- [MOR 07] Katell Morin-Allory, Laurent Fesquet, Dominique Borrione, "Asynchronous online monitoring of logical and temporal assertions", FDL, Barcelona, Spain, 18-20 septembre, 2007.
- [MUL 59] Muller D. E. and Bartky W. S., A Theory of Asynchronous Circuits, in Proc. of an International Symposium of the Theory of Switching, 1959, pp. 204-243
- [MYE 92] Chris Myers and Teresa H.-Y. Meng. Synthesis of timed asynchronous circuits. In Proc. International Conf. Computer Design (ICCD), pages 279-282. IEEE Computer Society Press, October 1992
- [NIE 94] Nielsen (L.S.), Niessen (C.), Sparso (J.), Van Berkel (J.), "Low Power Operation Using Self-Timed Circuits and Adaptative Scaling of the Supply Voltage". IEEE Transaction on Very large Scale Integration (VLSI) Systems, Vol. 2 (December 1994) 391-397.
- [NOW 93] Steven M. Nowick. Automatic Synthesis of Burst-Mode Asynchronous Controllers. PhD thesis, Stanford University, Department of Computer Science, 1993.
- [PAN 04] Dhanistha Panyasak, "Electromagnetic Emission Reduction in Integrated Circuits: The Asynchronous Alternative", Thèse de doctorat INPG, soutenue le 14 juin 2004.
- [PAY 97] Robert Payne. "Self Timed Field Programmable Gate Array Architectures". PhD thesis, University of Edinburgh, 1997.

- [PED 97] Pedram (M.), "Design Technologies for Low Power VLSI". Encyclopedia of Computer Science and Technology, Vol. 36. Inc. Marcel Dekker (1997) 73–96.
- [PER 00] Pering (T.), Burd (T.), Brodersen (R.), "Voltage Scheduling in the lpARM Microprocessor System". Proc. Int. Symposium on Low Power Electronic Design, Rapallo, Italy (July 2000) 96–101.
- [PEE 04] Ad Peeters, "Bringing handshake technology to the openmarket", Asynchronous Circuits and Systems, Proceedings of the 10th International Symposium, 2004.
- [PET 62] C.A Petri. Kommunikation mit automaten. PhD Thesis, Bonn, Institut für Instrumentelle Mathematik, 1962.
- [POU 01] Poulton D. et Oksman J. "Filtrage de signaux à échantillonnage non uniforme". Revue du Traitement du signal 12(2): 81-88. 2001.
- [QAI 06] Saeed Mian Qaisar, Laurent Fesquet, Fabien Aeschlimann, Marc Renaudin, "Spectral Analysis of a Signal Driven Sampling Scheme", EUSIPCO 2006, September 4-8, 2006, Florence Italy.
- [QAI 07a] Saeed Mian Qaisar, Laurent Fesquet, Marc Renaudin, "Adaptive Rate Filtering for a Signal Driven Sampling Scheme", ICASSP 2007, Honolulu, Hawaiï, USA, 15-20 April 2007.
- [QAI 07b] Saeed Mian Qaisar, Laurent Fesquet, Marc Renaudin, "Adaptive Rate Sampling and Filtering for Low Power Embedded Systems", Sampling Theory and Applications (SampTA'07), Thessaloniki, Greece, 1 - 5 June 2007.
- [QAI 07c] Saeed Mian Qaisar, Laurent Fesquet, Marc Renaudin, "Computationally Efficient Adaptive Rate Sampling and Filtering", EUSIPCO, Poznan, Poland, 3-7 September 2007.
- [QUA 04] Jérôme Quartana, "Conception de réseaux de communication sur puce asynchrones : application aux architectures GALS", Thèse INPG soutenue le 20 décembre 2004 (Direction : Marc Renaudin, Co-direction : Laurent Fesquet).
- [QUA 05] Jérôme Quartana, Salim Renane, Arnaud Baixas, Laurent Fesquet, Marc Renaudin, "GALS systems prototyping using multiclock FPGAs and asynchronous network-on-chips", FPL'05, Tampere, Finland, August 24-26, 2005, pp 299-304.
- [REN 98] Renaudin (M.), Vivet (P.), and Robin (F.), "ASPRO-216: A standard-cell QDI 16-bit RISC asynchronous microprocessor". Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, San Diego, CA, USA (1998) 22-31.
- [REN 99] Renaudin (M.), Vivet (P.), Robin (F.), "ASPRO: an Asynchronous 16-Bit RISC Microprocessor with DSP Capabilities". Proc. ESSCIRC 99, Duisburg, Germany (21-23 Sept. 1999) 28–31.
- [REN 99b] M. Renaudin, P. Vivet, and F. Robin. A design framework for asynchronous/synchronous circuits based on CHP to HDL translation. In Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC), pages 135-144, April 1999.
- [REN 00] Marc Renaudin, "Asynchronous circuits and systems: a promising design alternative", Microelectronic Engineering, Elsevier, vol. 54, p. 133-149, December 2000.
- [REN 04] M. Renaudin et A. Yakovlev, "Arbiters", tiré de l'ouvrage "Synchronization (à paraître)", R. Ginosar (Ed.), KLUWER academic publishers, chapitre 5, 2004.
- [REN 07] Marc Renaudin, Laurent Fesquet, Taha Beyrouthy, Alin Razafindraibe, Philippe Hoogvorst, Sylvain Guilley, Sumanta Chaudhuri, Jean-Luc Danger "A

- Reconfigurable Cell for a Multi-Style Asynchronous FPGA”, RecoSoC, Montpellier, 18-20 juin, 2007.
- [REZ 04] Amine Rezzag, Synthèse logique de circuits asynchrones micropipeline, thèse de doctorat INPG, soutenue le 13 décembre 2004.
- [RIG 01] J. B. Rigaud, J. Quartana, L. Fesquet et M. Renaudin, "Modeling and design of asynchronous priority arbiters for on-chip communication systems", Proceedings of the VLSI-SOC'01, 11th IFIP International Conference on Very Large Scale Integration, Montpellier, France, 3-5 Dec. 2001, pp. 424-429.
- [RIG 02] J. B. Rigaud, "Spécification de Bibliothèques pour la Synthèse de Circuits Asynchrones", Thèse de Doctorat, Institut National Polytechnique de Grenoble, 2002. Directeur de Thèse: M. Renaudin.
- [RIG 02b] J. B. Rigaud, J. Quartana, L. Fesquet et M. Renaudin, "Modeling and design of asynchronous priority arbiters for on-chip communication systems", tiré de l'ouvrage "SOC Design Methodologies", Hardbound M.-L. Flottes (Ed.), KLUWER academic publishers, chapitre 5, 2002.
- [RIG 02c] J. B. Rigaud, J. Quartana, L. Fesquet et M. Renaudin, "High-level modeling and design for on-chip communication systems", Proceedings of the Design Automation and Test in Europe.
- [ROS 85] L. Y. Rosenblum and A. V. Yakovlev. Signal graphs: from self-timed to timed ones. In Proceedings of International Workshop on Timed Petri Nets, pages 199-207, Torino, Italy, July 1985. IEEE Computer Society Press.
- [SAY 96] Sayiner N., Sorensen H. V. et Viswanathan T. R. "A Level Crossing Scheme for A/D Conversion". IEEE Transactions on Circuits and Systems II 43(4): 335-339. 1996.
- [SEM 03] Y. Semiat et R. Ginosar, "Timing Measurements of Synchronization Circuits", Proceedings of the Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC'03, Vancouver, Canada.
- [SEN 92] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli. SIS: A system for sequential circuit synthesis. Technical report, U.C. Berkeley, May 1992.
- [SIA 07] <http://www.sia-online.org>
- [SUT 01] I. E. Sutherland et S. Fairbanks, "GasP: A Minimal FIFO Control", Proceedings of the Seventh International Symposium on Advanced Research on Asynchronous Circuits and Systems, Asynch 2001, Salt Lake City, Utah, USA, March 2001.
- [TAR 05] Tarczynski, Andrzej and Allay, Najib (2005) Evaluation of several random sampling schemes for DASP applications. In: SampTA'05: Sampling Theory and Applications, 10-15 Jul 2005, Samsun, Turkey.
- [TEI 03] John Teifel and Rajit Manohar. "Programmable Asynchronous Pipeline Arrays". In FPL 2003, Lisbon, Portugal, September 2003.
- [TEI 04] John Teifel and Rajit Manohar. "Highly Pipelined Asynchronous FPGAs". In 2th ACM International Symposium on Field-Programmable Gate Arrays, Monterey, CA, February 2004
- [TRA 07] Transmeta Corporation, "Crusoe Processor System Design Guide". <http://www.transmeta.com>.
- [TRI 01] C. Tristam, "It's Time for Clockless Chips", Technology Review, no. 36-41, October 2001.

- [UNG 71] S.H. Unger, "Asynchronous sequential switching circuits with unrestricted input changes", IEEE Transactions on Computers, C-20(12), pp. 1437-1444, December 1971.
- [VAR 90] Varshavsky V. I., Ed. "Self-timed Control of Concurrent Processes: The Design of Aperiodic Logical Circuit's in Computers and Discrete Systems, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.
- [WAK 02] J. Wakerly: Digital Design: Principles & Practices, Prentice Hall; 3rd edition, July 2002.
- [WEI 94] Weiser (M.), Welch (B.), Demers (A.), Shenker (S.), "Scheduling for reduced CPU energy". Proc. USENIX Symposium on Operating Systems Design and Implementation, Monterey, CA, USA (Nov. 1994)13–25.
- [WON 03] Catherine G.Wong, Alain J.Martin, Peter Thomas, "An Architecture for Asynchronous FPGAs". In FPT 2003, Tokyo, Japan, 15-17 December, 2003
- [XAN 01] X. Xanthopoulos and al., The Design and Analysis of the Clock Distribution Network for a 1.2 GHz Alpha Microprocessor, in IEEE International Solid-State Circuits Conference, 2001, pp. 402-403.
- [YAO 95] F. Yao, A. Demers, S. Shenker, "A scheduling model for reduced CPU energy" Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95), Milwaukee, Wisconsin, October 23-25, 1995, p 374.
- [YAK 00] A. Yakovlev, L. Gomes and L. Lavagno. Hardware Design and Petri Nets. Kluwers Academic Publishers, 2000.
- [YUN 92] Yun K. Y., Dill D. L. and Nowick S. M., Synthesis of 3D asynchronous state machines, in Proc. International Conference on Computer Design (ICCD), (1992), IEEE Computer Society Press, pp. 346-350.
- [ZHE 98] Zheng H., Specification and compilation of timed systems, Master thesis, University of Utah, 1998.

Partie II : Curriculum Vitæ

I. Cursus universitaire et professionnel

1993	Ingénieur de l'Ecole Nationale Supérieure de Physique de Strasbourg, option traitement d'images.
1993	DEA Photonique et Image de l'Université Louis Pasteur de Strasbourg.
1994	Elève fonctionnaire de l'Ecole Normale Supérieure de Cachan.
1994	Agrégation de Sciences Physiques, option physique appliquée.
1997	Doctorat de l'Université Paul Sabatier - Toulouse III. Titre : « Intégration de sous-systèmes photoniques dans les architectures de communication multiprocesseurs », thèse préparée au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), soutenue le 9 décembre 1997 à Toulouse.
1998	Professeur agrégé en classe de BTS électrotechnique à Brive
1999	Maître de conférences à l'INPG-ENSERG, chercheur au laboratoire TIMA

II. Thèmes de recherche

J'ai intégré le laboratoire TIMA au sein de l'équipe CIS en septembre 1999. Mes travaux de recherche dans le groupe CIS porte sur l'étude des circuits sans horloge (aussi dénommés "asynchrones"). Ces travaux visent à explorer, identifier et quantifier les bénéfices qu'apporte le concept d'asynchronisme dans la problématique de l'intégration VLSI de fonctions complexes et évoluées. Les études que je mène dans ce domaine couvrent plusieurs aspects :

- Le traitement des signaux non uniformément échantillonnés,
- les algorithmes et les architectures des systèmes,
- les outils et les méthodes de synthèse pour les circuits asynchrones,
- la réalisation de circuits et de fonctions logiques et analogiques.

II.1 Traitement des signaux non uniformément échantillonnés

Ce travail de recherche porte sur le traitement de signaux à l'aide de fonctions réalisées en logique asynchrone. L'objectif est de se libérer de l'échantillonnage régulier communément utilisé, au profit d'un échantillonnage irrégulier contrôlé par le signal lui-même. La chaîne classique de traitement, "conditionnement du signal", "échantillonnage" et "traitement numérique synchrone" est remise en cause car elle ne prend pas en compte la nature de l'information contenue dans le signal.

Ce travail s'articule autour de deux points :

- Une nouvelle structure pour la conversion analogique-numérique a été mise au point. Ainsi, on ne représente plus le signal par des échantillons prélevés à des instants prédéfinis mais on détecte les variations du signal et on leur associe une information temporelle. Cet échantillonnage irrégulier est parfaitement adapté au traitement numérique asynchrone car, comme lui, il est contrôlé par des événements dont les instants d'occurrence sont inconnus a priori. On sort ici du cadre restrictif d'une algorithmique prévue pour le traitement des signaux numériques à échantillonnage régulier destiné à s'exécuter sur des machines synchrones. Pour expérimenter et valider ce nouveau concept, un prototype de convertisseur analogique/numérique asynchrone (A-ADC) a été fabriqué et validé. Les résultats de ce travail ont également fait l'objet d'un dépôt de brevet.

- Des travaux théoriques sur le traitement des signaux irrégulièrement échantillonnés ont été effectués. Les résultats de ces études ont été appliqués à l'analyse spectrale et au filtrage numérique. La maîtrise de la conception de machines asynchrones et de l'architecture des systèmes nous a conduit à considérer une algorithmique basée sur une représentation différente des signaux et sur des concepts d'analyse numérique. Il apparaît clairement que les bénéfices d'un tel système de traitement sont exploitables pour la conception des systèmes intégrés en termes de consommation, de bruit et de modularité. Cette approche ouvre des perspectives nouvelles pour la conception de "capteurs intelligents" (Signal/Image) et "objets communicants".

II.2 Algorithmes et architectures des systèmes

Une approche globale de la conception des systèmes conduit à l'élaboration de solutions innovantes par la maîtrise des couches basses, architecture matérielle et implémentation logique et analogique, et des couches hautes, logiciels applicatifs et systèmes (machine virtuelle, système d'exploitation, « middleware »). Les travaux, qui sont menés dans ce domaine, incluent les systèmes d'exploitation. En effet, il est stratégique de disposer pour les systèmes à microprocesseurs asynchrones d'un noyau multitâches, voire temps réel, tirant parti des caractéristiques spécifiques des processeurs asynchrones, notamment en terme de consommation. Cette étude a donné lieu à des résultats théoriques sur les stratégies d'ordonnancement des tâches sur les microprocesseurs asynchrones mais aussi sur la gestion de leurs tensions d'alimentation. Ce travail a également des incidences sur les couches matérielles car elle va nous permettre de concevoir des systèmes matériels adaptés au système d'exploitation. Ainsi, cette stratégie permettra à terme de réduire de manière significative la consommation des systèmes intégrés.

En outre, l'absence d'horloge offre également une grande flexibilité et permet d'explorer des solutions algorithmiques et architecturales nouvelles. Les deux principales raisons à cela sont : le contrôle des circuits asynchrones est distribué et l'absence de synchronisation globale permet une implémentation fiable de modèles indéterministes, notamment d'arbitres pour les systèmes de communication intra-puce. Une illustration de cette approche est l'adoption du paradigme GALS (Globally Asynchronous Locally Synchronous) pour la conception de systèmes sur puce.

Enfin, le développement de techniques de prototypage de nos circuits asynchrones sur circuits programmable nous a permis d'évaluer de nombreux circuits, y compris un réseau de communication intra-puce. Cette expérience, nous permet de développer des structures logiques reconfigurables dédiées aux circuits asynchrones. Ces structures reconfigurables pourront être à terme insérées dans nos designs.

II.3 Outils et Méthodes de synthèse pour les circuits asynchrones

Le développement d'outils de conception automatiques pour les circuits asynchrones constitue aujourd'hui un verrou très important et difficile à lever. Ces outils doivent être du niveau de performance et de maturité des outils communément utilisés pour la conception de circuits synchrones. La maîtrise de méthodes et le développement d'outils automatiques pour la conception de circuits asynchrones sont essentiels pour, d'une part, explorer leur potentiel et, d'autre part, favoriser leur diffusion et leur acceptation dans l'industrie. Je participe donc à l'effort de recherche et de développement d'un environnement de conception pour des circuits asynchrones dont l'acronyme est TAST (pour "Tima Asynchronous Synthesis Tools"). C'est un environnement de conception principalement composé d'un compilateur-synthétiseur capable de cibler différents types de logiques asynchrones à partir d'une description de haut

niveau exprimée dans un langage proche de CSP (Communicating Sequential Processes). Après analyse, les programmes sont représentés en interne dans l'outil par des réseaux de Petri associés à des graphes de type flot de données. Ce type de représentation est particulièrement adéquat dans le domaine des circuits asynchrones. Une forme synthétisable du langage, dénommée DTL (data transfer level), a été définie pour permettre la génération automatique de différents styles de circuits asynchrones : les circuits insensibles aux délais (QDI) et les circuits Micropipeline. Actuellement, nous travaillons sur une nouvelle génération de l'outil TAST basée sur les diagrammes de décision multivalués. Cette évolution permettra d'obtenir automatiquement des circuits logiques asynchrones optimisés.

III. Responsabilités administratives et scientifiques

III.1 Responsabilités scientifiques

Président du chapitre français IEEE de la Solid-State Circuits Society

J'assume la présidence du chapitre français IEEE de la Solid-State Circuits Society depuis le 7 novembre 2000. J'ai développé les activités de l'IEEE dans le domaine de la micro-électronique (Solid-State Circuits) en France en favorisant les parrainages de conférences par IEEE, en initiant (voire en organisant) des réunions techniques, des séminaires et des conférences. Avec mes collègues des chapitres CAS (Circuits And Systems) et CPMT (Components, Packaging and Manufacturing Technology), nous nous sommes organisés en consortium pour fédérer nos activités afin de couvrir l'ensemble des aspects de la micro-électronique (du boîtier au système). En outre, je participe aux activités du bureau de la section française de IEEE qui regroupe l'ensemble des sociétés de IEEE.

Direction du groupe CIS du laboratoire TIMA

Depuis le 1^{er} octobre 2007, j'assure la co-direction du groupe « Concurrent Integrated Systems » avec mon collègue Gilles Sicard. Le groupe est constitué d'une vingtaine de personnes dont une douzaine de doctorants.

Comités de programme

- Membre du comité de programme de l'« IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC) » en 2008 et 2006
- Membre du comité de programme de la conférence « Design Automation and Test in Europe (DATE) » en 2008
- Membre du comité de programme de l'« International Conference on Micro and Nanotechnologies (ICMNT) » en 2006

Revue internationale

- Rapporteur pour « Microelectronics journal », Elsevier
- Rapporteur pour « IEEE Transactions on Signal Processing »

III.2 Organisation de manifestations

Sampling Theory and Applications (SAMPTA) 2009, avril ou mai 2009 à Marseille

- Co-general chair de SAMPTA 2009

Cette conférence traite de la théorie de l'échantillonnage et de ses applications. La théorie de l'échantillonnage a connu un net regain d'intérêt depuis les années 1990, grâce notamment à ses liens avec la théorie des bases d'ondelettes et plus généralement avec l'analyse temps-fréquence. L'idée forte est maintenant de rendre l'échantillonnage adaptatif, c'est à dire

caractériser une fonction donnée, avec une précision fixée, par le plus petit nombre possible d'échantillons (valeurs ponctuelles de la fonction, de sa transformée de Fourier, ou autres extensions). Cette problématique, assez générale, recouvre de difficiles problèmes de mathématiques pures aussi bien qu'appliquées. Il faut également signaler l'intérêt porté à l'échantillonnage par le secteur industriel, notamment le secteur microélectronique, où se posent de difficiles problèmes d'échantillonnage sur lesquels portent des contraintes pratiques (échantillonnage irrégulier, contraintes de temps réel,...)

12th IEEE International Symposium on Asynchronous Circuits and Systems 2006 (ASYNC'06), 13-15 mars 2006, Grenoble

- Industrial exhibition chair
- Local arrangement chair

The International Symposium on Asynchronous Circuits and Systems provides a high quality forum for scientists and engineers to present their latest research findings. Topics of interest include, but are not limited to:

- Mixed synchronous/asynchronous architectures, interfaces, and circuits.
- High-speed/low-power asynchronous logic, memories, and interconnects.
- High-level design and synthesis of self-timed circuits.
- Physical design of unlocked logic and pipelines.
- Formal methods for correctness and performance analysis of asynchronous designs.
- Test, reliability, security, and radiation tolerance.
- CAD for asynchronous design and validation.
- Asynchronous System-on-a-chip (SoC) and Network on Chip (NoC).

European Solid-State Device Research Conference 2005

European Solid-State Circuit Conference 2005

ESSDERC-ESSCIRC'05, 12-16 septembre 2005, Grenoble

- Editeur des actes de la conférence ESSCIRC. (Le livre comporte environ 550 pages!)

The aim of the **ESSDERC** conference is to provide an annual European forum for the presentation and discussion of recent advances in solid-state devices and technologies. **ESSDERC** and its sister conference **ESSCIRC**, which deals with solid-state circuits, are jointly organised.

The increasing level of integration for system-on-chip design made available by advances in silicon technology requires a deeper interaction amongst technologists, device physicists and circuit designers. While keeping separate Technical Programme Committees, **ESSDERC** and **ESSCIRC** will share plenary keynote presentations and tutorials bridging both communities. Delegates registered for either conference are encouraged to attend any of the scheduled parallel sessions.

Journée System on Chip: Design for Low Power, 14 octobre 2003, Aix-en-Provence

- Co-organisateur de la manifestation

Cette journée organisée avec IEEE France et ATMEL Rousset est dédiée aux nouvelles techniques de conception pour la faible consommation dans les systèmes sur puce. L'atelier s'est focalisé sur les problèmes liés à la réduction des tensions d'alimentation et aux dimensions fortement submicroniques des transistors.

Smart Object Conference, 15-17 mai 2003, Grenoble

- J'ai été membre du comité de pilotage de cette manifestation et j'ai participé à la mise en place de cette manifestation. Cette conférence a donné naissance à SoC-Eusai.

Cette conférence avait pour objet le nouveau champ de recherches et d'applications défini par l'association non exclusive d'expressions-clés comme «informatique ubiquitaire», «informatique diffuse», «intelligence ambiante», «ordinateurs évanescents», «capteurs et actionneurs en réseau». Les «objets communicants» visés par cette conférence sont observés de deux points de vue complémentaires : un point de vue centré sur les objets eux-mêmes, poussé par la technologie et un point de vue centré sur les utilisateurs finaux.

Summer School on asynchronous Circuits Design, 15-19 juillet 2002, Grenoble

- Co-organisateur de la manifestation

Cette manifestation a pour but de former les industriels et les universitaires aux techniques de la conception de circuits logiques asynchrones et notamment à l'usage des outils de synthèse spécialisés dans le domaine de l'asynchrone. Cette école d'été s'inscrit dans le cadre du groupe de travail ACiD (Asynchronous Circuit Design) du projet européen Esprit 21949.

Asynchronous Circuit Design Workshop, 30-31 janvier, 1^{er} février 2000, Grenoble

- Co-organisateur de la manifestation

Cette manifestation a pour but de réunir les spécialistes européens universitaires ou industriels qui sont impliqués dans des recherches sur la conception de circuits asynchrones complexes. Cette conférence s'inscrit dans le cadre du groupe de travail ACiD du projet européen Esprit 21949.

Technologies nouvelles, Interconnexions et Communications dans les Systèmes Distribués et Parallèles, 16-17 juin 1997, Toulouse

- Co-organisateur de la manifestation

Cet atelier est organisé dans le cadre d'une coopération entre les trois principaux laboratoires toulousains en informatique (Centre d'Etude et de Recherche de Toulouse de l'ONERA, l'Institut de Recherche en Informatique de Toulouse, le Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS). L'intention des organisateurs est de:

- Cerner les besoins présents et futurs en communication des systèmes multiprocesseurs ou des systèmes distribués et examiner les capacités des technologies électroniques et photoniques pour les résoudre.
- Réunir à la fois des concepteurs de machines parallèles, des concepteurs de systèmes de communication, des utilisateurs et des technologues susceptibles d'exposer les possibilités offertes par les technologies optiques et électroniques.

III.3 Participation à des projets

III.3.1 Projets européens

Projet Européen Medea+ TRUST-eS 2004-2005

Le projet Trust-eS aborde cinq grands thèmes techniques que sont, l'architecture de système intégrant une carte et un lecteur de carte, les applications client/serveur pour les cartes à puce, les techniques d'authentification, les techniques de reconfiguration et l'intégration des lecteurs de cartes dans l'environnement système. Le groupe CIS du TIMA était impliqué dans l'étude des techniques de durcissement des cartes à puces vis-à-vis d'attaques dites par injection de fautes. J'étais par ailleurs responsable de cette tâche pour le projet. Le projet a été arrêté avant son terme en 2005.

Partenaires : Gemplus, STMicroelectronics, Atmel, Infineon, Schlumberger Sema, Thales, iRoC Technologies, TIMA, SystemPlus, Telephonica, Universidad Carlos III de Madrid, CNM, CEA-LETI, Soisic, M2000, GET, Opsis, Halmstad University, University of Surrey, University of Fribourg, Tarragona University, Innova Card

Projet européen Medea+ EsP@ss-IS 2001-2004

The growth of value-added electronic and mobile commerce devices can be greatly enhanced by the availability of both hardware- and software-based open smart-card platforms. At a secondary level, the development of such platforms can provide a set of reusable innovative technological bricks from which future generations of high security smart-card products may be created. The EsP@ss-IS project aims to deliver all of the necessary hardware and software for open smart-card platforms destined to support the availability of value-added electronic and mobile commerce services. These are intended to meet the needs of operators in the mobile telecommunications, banking and pay-TV sectors.

Partenaires : CEA-LETI, TIMA, Interpay, Philips Consumer Electronics, Schlumberger, ST Microelectronics, Thomson Multimedia, TIM, Trusted Logic

III.3.2 Projets nationaux

Projet ARAVIS du pôle de compétitivité Minalogic 2007-2010

La « scalabilité » des technologies CMOS n'est plus applicable à partir des nœuds technologiques 32nm compte tenu de la très forte variabilité des paramètres technologiques sur une même puce, avec pour incidence une nécessité de prise en compte au niveau système de ces effets. De même, les architectures de traitement doivent être repensées et adaptées pour permettre de pallier à ces contraintes de dispersion technologique. Trois technologies clés y sont combinées pour concevoir et fabriquer un système sur puce SoC-ARAVIS en technologie CMOS 32nm : La structure reconfigurable à grain grossier DSPfabric de ST, les techniques asynchrones, et les techniques d'automatisme avancées pour la gestion dynamique de puissance et d'activité.

Partenaires : Orange-FT, ST, Thomson, CEA-LETI, INRIA, TIMA

Projet ANR SAFE 2005-2008

Ce projet vise l'étude et la réalisation d'un FPGA embarqué dédié à la technologie asynchrone et aux applications sécurisées comme celles que l'on rencontre dans les cartes à puce.

Partenaires : ENST Paris, TIMA

Projet RNRT Aresa 2006-2009

ARESA is focused on large scale wireless sensor networks distributed in a monitoring area. Because the sensor nodes are autonomous, they should preserve their energy during

processing, communication, etc. Finally, the energy is the key problem and this is considered in the different working packages of the project. Sensor networks should consider the energy consumption in all the topics possible: hardware, operating systems, networking, applications, etc. In this project and based on a specific scenario, we address the network protocols design, the performance evaluation, the protocols validation, etc.

Partenaires : CITI, INSA Lyon, Coronis Systems, France Telecom R&D, LIG, TIMA, Verimag

Projet ANR SFINCS 2008-2011

Le projet SFINCS (Semi-Formal INstrumentation for Circuits and Systems) recherche, étudie et développe de nouvelles techniques nécessaires à la validation des systèmes sur puce. SFINCS se situe dans le cadre de l'*Assertion-Based Verification* (ABV). Le but de ce projet est de développer et d'intégrer des méthodologies permettant l'application de l'ABV à une palette de composants matériels, au moyen d'une approche uniforme basée sur une technologie conçue au Laboratoire TIMA. Ce projet permettra de développer des outils de génération de moniteurs de surveillance synchrones ou asynchrones et des générateurs de séquences de test que l'on pourra embarquer dans les SoCs.

Partenaires : Dolphin Integration, Thalès Communications, TIMA

III.3.3 Projets collaboratifs

Projet SPASS avec le Laboratoire Jean Kuntzmann (LJK) 2000-

Ce projet vise l'étude mathématique des propriétés liées aux techniques d'échantillonnage par traversée de niveaux. Ce travail d'analyse numérique constitue la pierre fondatrice pour la théorie que nous développons dans le domaine du traitement des signaux non uniformément échantillonnés.

Projet ADAC avec l'Université Sharif de Téhéran 2007-

Le schéma d'échantillonnage non uniforme par traversée de niveaux est étudié à la fois à TIMA (CNRS-INPG-UJF) et à ACRI (Université Sharif de Téhéran). A TIMA, les études sont principalement focalisées sur les aspects systèmes et sur la conception de circuits intégrés. A ACRI, la recherche étant orientée sur le développement d'algorithmes pour la reconstruction de signaux non uniformément échantillonnés par méthodes itératives, une coopération a été mise en place. Ce travail collaboratif a donné lieu à une demande de projet Hubert Curien (Gundishapur).

Projet UWB-IR avec FT R&D 2005

Ce projet, en collaboration avec France Télécoms R&D, vise l'exploration de solution en technologie asynchrone pour réaliser un récepteur pour de la radio impulsionnelle à ultra-large bande spectrale. Cette collaboration a été initié fin 2004 et à donner lieu à deux soutenances d'ingénieurs. Elle est actuellement poursuivie dans le cadre d'une thèse CIFRE.

ERC et ERC2 avec ST-Microelectronics 2000-2008

Il ne s'agit pas à proprement parler d'un projet, ERC signifiant Equipe de Recherche Commune. ST Microelectronics finance le laboratoire TIMA pour des activités de recherche amont. Le groupe CIS du TIMA a contribué à l'étude des techniques de synthèse des circuits asynchrones, à l'architecture d'un réseau de communication sur puce asynchrone et à la conception d'un système de recouvrement d'horloge asynchrone utilisé dans un système de communication.

III.4 Responsabilités administratives

- Responsable scientifique de la plateforme « Conception, Prototypage et Test (CPT) » du CIME-Nanotech depuis 2006

Je suis en charge du développement et des évolutions techniques et scientifiques de la plateforme CPT. La plateforme offre l'ensemble des moyens nécessaires à la conception de circuits et systèmes intégrés, c'est-à-dire logiciels de conception, moyens de prototypage et de test. Cette activité est également supportée par trois ingénieurs à temps plein.

- Président du Comité de Coordination des Services Techniques (CCST) du CIME-Nanotech depuis 2007

Ce comité de coordination est à la charge de définir les informations à faire remonter au conseil d'administration du CIME-Nanotech

- Membre du conseil d'administration du CIME-Nanotech depuis 2007
- Membre nommé du conseil du laboratoire TIMA
- Membre élu du conseil d'administration de l'ENSERG de 2002 à 2006
- Membre élu de la commission de spécialistes de l'INPG en 61^{ème} section depuis 2003

IV. Publications

IV.1 Thèse

1. Laurent Fesquet, "Intégration de sous-systèmes photoniques dans les architectures de communication multiprocesseurs", thèse de l'Université Paul Sabatier de Toulouse soutenue le 9 décembre 1997.

IV.2 Brevet

1. E. Allier, L. Fesquet, M. Renaudin, G. Sicard, "Procédé et dispositif de conversion analogique-numérique", brevet n° FR2835365 (déposé le 31 janvier 2002), publié le 2003-08-01

IV.3 Edition d'ouvrages

1. Laurent Fesquet, Andreas Kaiser, Sorin Cristoloveanu, Michel Brillouët, "IEEE European Solid-State Circuits Conference (ESSCIRC) ", 12-16 septembre 2005, Grenoble, France, 538 pages, ISBN 0-7803-9205-1, IEEE catalog number 05EX1088.

IV.4 Contributions à des ouvrages

3. Kamel Slimani, Joao Fragoso, Mohammed Es Sahliene, Laurent Fesquet, Marc Renaudin, "Low Power Asynchronous Processors", in Low Power Electronic design, edited by Christian Piguet, Chap. 22, CRC Press, July 2004, ISBN 0849319412, pp 22.1-22.23.
2. J.B. Rigaud, J. Quartana, L. Fesquet, M. Renaudin, "Modeling and Design of Asynchronous Priority Arbiters for On-Chip Communication Systems", in "Communication Systems SoC Design Methodologies", Michel Robert, Bruno

Rouzeyre, Christian Piguet, Marie-Lise Flottes Eds., Kluwer Academic Publishers, July 2002, ISBN 1-4020-7148-5.

1. Laurent Fesquet, Jacques Collet, Rainer Buhleier, "Optical Interconnections and Parallel Processing: Trends at the Interface", Ed. Pascal Berthomé, Alfonso Ferreira, Kluwer, January 1998, Chapitre 4 : "Low Latency Asynchronous Optical Bus for Distributed Multiprocessor Systems", 18 pages

IV.5 Journaux

3. E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "Asynchronous Level Crossing Analog to Digital Converters", Special Issue on ADC Modelling and Testing of *Measurement*, Vol. 37, Issue 4 , June 2005, Pages 296-309.
2. A new type of Asynchronous Analog to Digital Interface Author(s): Allier E., Sicard G., Fesquet L., Renaudin M. Journal: Journal of International Measurement Confederation, Vol.35, Issue 2, March 2004.
1. Laurent Fesquet, Mohammed Es Salhiene, Marc Renaudin, "La technologie asynchrone au service de la réduction d'énergie dans les systèmes embarqués", Annales des Télécommunications, Tome 59, n°7-8, juillet-août 2004.

IV.6 Papiers de conférences sélectionnés

5. Katell Morin-Allory, Laurent Fesquet, Dominique Borrione, "Asynchronous online monitoring of logical and temporal assertions", " in Embedded Systems Specification and Design Languages, to be published by Springer, Selected papers from FDL 2007
4. Dominique Borrione, Miao Liu, Pierre Ostier, Laurent Fesquet , "PSL-based Online Monitoring of Digital Systems" in Advances in Design and Specification Languages for SoCs (best selected contributions from FDL'05), Springer 2006
3. Jérôme Quartana, Laurent Fesquet, Marc Renaudin, "Modular Asynchronous Network-on-Chip: Application to GALS Systems Rapid Prototyping" in VLSI-SOC: From Systems to Chips, (best selected contributions from VLSI-SoC 2005), Series: IFIP International Federation for Information Processing, Vol. 200, Glesner, M.; Reis, R.; Indrusiak, L.; Mooney, V.; Ekeking, H. (Eds.), Springer , 2006, pp 195-208
2. Bertrand Folco, Vivian Brégier, Laurent Fesquet, Marc Renaudin, "Technology Mapping for Area Optimized Quasi Delay Insensitive Circuits", in VLSI-SOC: From Systems to Chips, (best selected contributions from VLSI-SoC 2005), Series: IFIP International Federation for Information Processing, Vol. 200, Glesner, M.; Reis, R.; Indrusiak, L.; Mooney, V.; Ekeking, H. (Eds.), Springer, 2006, pp 55-70
1. Laurent Fesquet, Jacques Collet , "Applications of photonic technology 2", Ed. G.A. Lampropoulos, R.A. Lessard, Plenum Press, 1997, Chapitre : "Low Latency Optical Bus for Multiprocessor Architecture", (best selected contributions from the International Conference on Applications of Photonics Technology 1996), pages 189-194

IV.7 Conférences invitées

4. Laurent Fesquet, Jérôme Quartana, Marc renaudin, "Asynchronous Systems on Programmable Logic", Reconfigurable Communication-centric SoCs, ReCoSoC'05, Montpellier, France, June 27-29, 2005, pp 105-112, ISBN 2 – 9517 – 4611 – 3.
3. Laurent Fesquet, Fraidy Bouesse, Yannick Monnet, Marc Renaudin, "Implementing Asynchronous Circuits on LUT Based FPGAs", 3rd International Workshop on Cryptographic Architectures Embedded in Reconfigurable Devices - CryptArchi 2005, Saint-Etienne, France, June 8 - 11th 2005.

2. Marc Renaudin, Fraïdy Bouesse, Yannick Monnet, Laurent Fesquet, "Secure asynchronous circuits design and prototyping", 3rd International Workshop on Cryptographic Architectures Embedded in Reconfigurable Devices - CryptArchi 2005, Saint-Etienne, France, June 8 - 11th 2005.
1. Laurent Fesquet, Stéphane Monfray, Jean-Pierre Carrère, "Les technologies CMOS avancées au service des designers", Ecole ARCHI'05, Autrans, France, March 21-25, 2005.

IV.8 Conférences internationales avec actes

36. Jérémie Hamon, Laurent Fesquet, Benoit Miscopain, Marc Renaudin, "High-Level Time-Accurate Model for the Design of Self-Timed Ring Oscillators", 14th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC 2008), Newcastle upon Tyne, UK, 7th - 11th April 2008.
35. Sumantha chaudhuri, Philippe Hoogvorst, Sylvain Guilley, Jean-Luc Danger, Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, Marc Renaudin, "Physical Design of FPGA Interconnect to Prevent Information Leakage", International conference on Applied Reconfigurable Computing ARC'08, Imperial College, London, UK, March 26-28, 2008
34. Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, Marc Renaudin, Philippe Hoogvorst, Sylvain Guilley, Sumanta Chaudhuri, Jean-Luc Danger, "A novel asynchronous e-FPGA architecture for security applications", ICFPT, Kokurakita, Kitakyushu, Japan, December 12th - 14th, 2007, pp 369-372, ISBN 1-4244-1471-7.
33. Katell Morin-Allory, Laurent Fesquet, Dominique Borrione, "Asynchronous online monitoring of logical and temporal assertions", FDL, Barcelona, Spain, 18-20 septembre, 2007.
32. Saeed Mian Qaisar, Laurent Fesquet, Marc Renaudin, "Computationally Efficient Adaptive Rate Sampling and Filtering", EUSIPCO, Poznan, Poland, 3-7 September 2007.
31. Marc Renaudin, Laurent Fesquet, Taha Beyrouthy, Alin Razafindraibe, Philippe Hoogvorst, Sylvain Guilley, Sumanta Chaudhuri, Jean-Luc Danger "A Reconfigurable Cell for a Multi-Style Asynchronous FPGA", RecoSoC, Montpellier, 18-20 juin, 2007, pp 15-22, ISBN: 2-9517461-3-X.
30. Saeed Mian Qaisar, Laurent Fesquet, Marc Renaudin, "Adaptive Rate Sampling and Filtering for Low Power Embedded Systems", Sampling Theory and Applications (SampTA'07), Thessaloniki, Greece, 1 - 5 June 2007.
29. Saeed Mian Qaisar, Laurent Fesquet, Marc Renaudin, "Adaptive Rate Filtering for a Signal Driven Sampling Scheme", IEEE ICASSP 2007, Honolulu, Hawaiï, USA, 15-20 April 2007, vol. 3, pp 1465-1468.
28. Laurent Fesquet, Bertrand Folco, Mathieu Steiner and Marc Renaudin, "State-holding in Look-Up Tables: application to asynchronous logic", VLSI-SoC 2006, October 16-18, 2006, Nice, France, pp 12-17.
27. Saeed Mian Qaisar, Laurent Fesquet, Fabien Aeschlimann, Marc Renaudin, "Spectral Analysis of a Signal Driven Sampling Scheme", EUSIPCO 2006, September 4-8, 2006, Florence Italy.
26. Katell Morin-Allory, Laurent Fesquet, Dominique Borrione, "Asynchronous on-line monitoring of PSL assertions", Rapid System Prototyping (RSP 2006), June 14-16, 2006, Chania, Crete.
25. D. Borrione, M. Liu, K. Morin-Allory, P. Ostier, L. Fesquet, "On-Line Assertion-Based Verification with Proven Correct Monitors", IEEE Intl. Conf. on Information & Communications Technology, Dec. 2005, Cairo, Egypt, pp 125—143.

24. Bertrand Folco, Vivian Brégier, Laurent Fesquet, Marc Renaudin, "Quasi Delay Insensitive circuits synthesis using Multi-valued Decision Diagrams", IFIP VLSI-SoC 2005, Perth, Australia, October 17-19, 2005.
23. Jérôme Quartana, Laurent Fesquet, Marc Renaudin, "Modular Asynchronous Network-on-Chip: Application to GALS Systems Rapid Prototyping ", IFIP VLSI-SoC 2005, Perth, Australia, October 17-19, 2005.
22. Dominique Borrione, Miao Liu, Pierre Ostier, Laurent Fesquet, "PSL-based online monitoring of digital systems", FDL'05, Lausanne, Switzerland, September 27-30, 2005.
21. Jérôme Quartana, Salim Renane, Arnaud Baixas, Laurent Fesquet, Marc Renaudin, "GALS systems prototyping using multiclock FPGAs and asynchronous network-on-chips", FPL'05, Tampere, Finland, August 24-26, 2005, pp 299-304.
20. L. Fesquet, M. Renaudin, "A Programmable Logic Architecture for prototyping clockless circuits", FPL'05, Tampere, Finland, August 24-26, 2005, pp 293-298.
19. Fabien Aeschlimann, Emmanuel Allier, Laurent Fesquet, Marc Renaudin, "Sampling scheme for the design of an asynchronous non-uniform signal processing chain", Sampling Theory and Applications (SampTA'05), Samsun, Turkey, July 10-15, 2005.
18. N. Huot, H. Dubreuil, L. Fesquet, M. Renaudin, "FPGA Architecture for Multi-style asynchronous logic", DATE 2005, München, Germany, March 7-11, 2005, pp 32-33.
17. Anh Vu Dinh Duc, Laurent Fesquet, Marc Renaudin, "A new language-based Approach for Specification of Asynchronous Systems", Research Innovation and Vision for the Future (RIFV 2005), Can Tho, Vietnam February 21-24, 2005, Eds Suger.
16. V. Brégier, B. Folco, L. Fesquet, M. Renaudin, "Modeling and Synthesis of Multi-rail Multi-protocol QDI Circuits", IWLS'04, San Diego, USA, June, 2004.
15. F. Aeschlimann, E. Allier, L. Fesquet, M. Renaudin, "Asynchronous Finite Impulse Response Digital Filter: A New Digital Signal Processing Chain", IEEE Async'2004, Hersonisos, Crete, April 19-23, 2004, pp. 198-206.
14. E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "Asynchronous ADCs: Design methodology and case study", IEEE IWADC'03, Perugia, Italy, September 8-10, 2003, pp. 29-32.
13. E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "A New Class of Asynchronous A/D Converters Based on Time Quantization", IEEE Async'03, Vancouver, Canada, May 12-16, 2003, pp. 196-205.
12. E. Allier, L. Fesquet, M. Renaudin, G. Sicard, "Low-Power Asynchronous A/D Conversion", Twelfth International Workshop on Power And Timing Modeling, Optimization and Simulation (PATMOS), September 11-13, 2002, Sevilla, Spain.
11. M. Es Salhiene, L. Fesquet, M. Renaudin, "Dynamic Voltage Scheduling for Real Time Asynchronous Systems", Twelfth International Workshop on Power And Timing Modeling, Optimization and Simulation (PATMOS), September 11-13, 2002, Sevilla, Spain.
10. Quoc Thai Ho, J.-B. Rigaud, L. Fesquet, M. Renaudin, R. Rolland, "Implementing asynchronous circuits on LUT based FPGAs", The 12th International Conference on Field Programmable Logic and Applications (FPL), September 2-4, 2002, Montpellier (La Grande-Motte), France.
9. Anh Vu Dinh Duc, Laurent Fesquet, Marc Renaudin, "Synthesis of QDI Asynchronous Circuits from DTL-style Petri-Net" IWLS-02, 11th IEEE/ACM International Workshop on Logic & Synthesis, New Orleans, Louisiana, June 4-7, 2002.
8. J.-B. Rigaud, J. Quartana, L. Fesquet, M. Renaudin, "High-Level Modeling and Design of Asynchronous Arbiters for On-Chip Communication Systems", Design Automation and Test Conference (DATE), 4-7 March, 2002, Paris, France.

7. JB. Rigaud, J. Quartana, L. Fesquet, M. Renaudin, " Modeling and design of asynchronous priority arbiters for on-chip communication systems", in the proceedings of VLSI'SoC, 3-5 December, Montpellier, France, 2001.
6. W. Hlayhel, J. Collet, L. Fesquet, "Implementing snoop-coherence protocol for future SMP architectures", Conference Europar'99, Juin 1999, Toulouse, pp.745-752
5. W. Hlayhel, D. Litaize, L. Fesquet, J. Collet, "Optical bus versus electronic bus for address transactions for future SMP architecture", International Conference on Parallel Architectures and Compilation Techniques, october 14th-17th 1998, Paris, France, pages 22-29
4. Yannick Saouter, Daniel Litaize, Jacques H. Collet, Laurent Fesquet, "ATM Systolic Switch Architecture for Gigabit Networks", Extended version of Conference IFIP ATM, 20th-22nd july 1998, Ilkley, West Yorkshire, U.K., 11 pages
3. W. Hlayhel, D. Litaize, L. Fesquet, J. Collet, "Optical bus versus electronic bus for address transactions for future SMP architecture", International Conference on Parallel Architectures and Compilation Techniques, october 14th-17th 1998, Paris, France, pages 22-29
2. Jacques Collet, Laurent Fesquet, "Comparison of the communication latency for an optical bus and for several 2D electronic network topologies", 11th International Parallel Processing Symposium, Workshop on Optics and Computer Science (WOCS), April 1st-5th 1997 Geneva Switzerland, 12 pages
1. Laurent Fesquet, Jacques Collet, "Ultrafast Optical Bus for Multiprocessor Systems", International Conference on Applications of Photonics Technology 1996, July 29th - August 1st 1996 Montreal Canada, pages 13-12, 13-13

IV.9 Conférences internationales sans actes

4. E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "A New Class of Asynchronous Analog-to-Digital Converters", in the ACiD-WG workshop, Heraklion, Crete, January 27-28, 2003.
3. A.V. Dinh Duc, J.-B. Rigaud, A. Rezzag, A. Sirianni, J. Fragoso, L. Fesquet, M. Renaudin, "TAST CAD Tools", in the ACiD-WG workshop, Munich, Germany, January 28-29, 2002.
2. F. Bouesse, L. Fesquet, M. Renaudin, "QDI Circuits to Improve Smartcard Security", in the ACiD-WG workshop, Munich, Germany, January 28-29, 2002.
1. Laurent Fesquet, Jacques Collet, "Low Latency Optical Bus for Distributed Multiprocessor Systems", International Workshop on New Technologies, Interconnects and Communications in Distributed and Parallel Systems, June 16th-17th 1997, Toulouse, France

IV.10 Conférences nationales

9. Taha Beyrouthy, Alin Razafindraibe, Laurent Fesquet, Marc Renaudin, Secure Asynchronous FPGA for Embedded Systems (SAFE), Journées du GDR SoC-SIP, Paris, 13-15 juin, 2007.
8. Taha Beyrouthy, Laurent Fesquet, Marc Renaudin, Sumanta Chaudhuri, Sylvain Guilley, Philippe Hoogvorst, Jean-Luc Danger, "FPGA Asynchrone Sécurisé", présenté aux journées PARISTIC sur le programme sécurité et informatique, 22-24 Novembre 2006, Nancy.
7. Fabien Aeschlimann, Emmanuel Allier, Laurent Fesquet, Marc Renaudin, "Etude spectrale de l'échantillonnage par traversée de niveaux", GRETSI'05, Louvain-la-Neuve, Belgique, 6-9 septembre 2005, pp 827-830.

6. Alain Guyot, Laurent Fesquet, Laurent Martin et Matthieu Arnaud, "Circuit d'addition virgule flottante avec arrondi et dénormalisé", journées Arineews printemps 2004, 25-26 mai 2004.
5. Emmanuel Allier, Gilles Sicard, Laurent Fesquet, Marc Renaudin, "Conversion analogique-numérique faible consommation : conception asynchrone et échantillonnage irrégulier", TAISA'03, Louvain-la-Neuve, Belgique, 25-26 septembre 2003, pp. 53-56.
4. K. Slimani, Y. Remond, A. Sirianni, G. Sicard, L. Fesquet, M. Renaudin, "Estimation et optimisation de la consommation d'énergie des circuits asynchrones", FTFC'2003, Paris, May 15-16, 2003, pp. 59-64.
3. Mohammed Es Salhiene, Laurent Fesquet, Marc Renaudin, "Adaptation dynamique de la puissance des systèmes embarqués : les systèmes asynchrones surclassent les systèmes synchrones", FTFC'2003, Paris, May 15-16, 2003, pp. 51-58.
2. Mohammed Es Salhiene, Laurent Fesquet, Quoc Thai Ho, Frédéric Lhuillery, Marc Renaudin, "L'asynchronisme de la puce au système : état de l'art et exemples", Séminaire Objets Communicants, Grenoble, 17-18 octobre, 2001.
1. Mohammed Es Salhiene, Laurent Fesquet, Quoc Thai Ho, Frédéric Lhuillery, Marc Renaudin, "Démonstration de prototypes d'objets communicants en technologie asynchrone", Séminaire Objets Communicants, Grenoble, 17-18 octobre, 2001.

IV.11 Séminaires

2. L.Fesquet, "L'avenir des circuits numériques est-il asynchrone ? Principes, conception et intérêts des circuits asynchrones ", Séminaire du LAAS, Toulouse, 10 mai 2005.
1. L. Fesquet, M. Renaudin, "Asynchronous Circuit Design", Séminaire Institut Supérieur d'Electronique de Paris, November 17, 2003.

IV.12 Autres publications

Tutorial

1. A.V. Dinh Duc, J.-B. Rigaud, A. Rezzag, A. Sirianni, J. Fragoso, L. Fesquet, M. Renaudin, "TAST CAD Tools: Tutorial", tutorial given at the International Symposium on Advanced Research in Asynchronous Circuits and Systems ASYNC'02, Manchester, UK, April 8-11, 2002, TIMA internal report ISRN:TIMA-RR-02/04/01—FR, <http://tima.imag.fr/cis>.

Enseignement

2. Emmanuel Allier, Laurent Fesquet, "Conception de circuits intégrés analogiques", cours pour la formation à distance ELAN (niveau Bac+4), 130 pages, 2003.
1. Sylvain Boulmé, Laurent Fesquet, Florence Maraninchi, "Logiciel de base en C pour le processeur Pentium et le format ELF", Ensimag/Télécom, 54 pages, avril 2003

Vulgarisation

2. Mohammed Es Salhiene, Laurent Fesquet, Marc Renaudin, "Gérer dynamiquement la consommation des systèmes embarqués", Electronique, Vol. 140, p 45-48, Décembre 2003.
1. Mohammed Es Salhiène, Laurent Fesquet, Marc Renaudin, "L'adaptation dynamique de la puissance va mieux à l'asynchrone", Tribune, Electronique, Septembre 2003.

Journées de doctorants

4. Vivian Brégier, Laurent Fesquet, Marc Renaudin, "synthèse et optimisation de circuits quasi-insensibles aux délais", Journée des Doctorants, Grenoble, France, 3 février 2005.

3. Amine Rezzag, L. Fesquet, M. Renaudin, "Synthèse logique de circuits asynchrones micropipeline : traitement de différents protocoles de communication", JNRDM 2003, 15-16 mai 2003, Toulouse, France.
2. Fabien Aeschlimann, L. Fesquet, M. Renaudin, "Echantillonnage irrégulier et circuits intégrés asynchrones", Prix du meilleur poster, Journée des doctorants de l'ED EEATS 2003, 11 décembre 2003, Grenoble.
1. Laurent Fesquet, "Architecture multiprocesseur à latence réduite par bus optique multiplexé", 1er Séminaire de l'Ecole Doctorale d'Electronique de Toulouse, 9-11 septembre 1997, Toulouse

Rapports

4. Laurent Fesquet, Caractérisation d'une caméra 200 images/s, Rapport de stage de D.E.A. Photonique et Image, 1993
3. Laurent Fesquet, Spécification, conception et réalisation d'un exécutif temps réel, Rapport de stage ingénieur chez Julabo Labortechnik, Seelbach (Allemagne), 1992
2. Laurent Fesquet, Funktionalitätbeschreibung (« Spécification » d'un système temps réel), Documentation Julabo Labortechnik, Seelbach (Allemagne), 1992
1. Laurent Fesquet, RTKPX Real Time Kernel, Documentation Julabo Labortechnik, Seelbach (Allemagne), 1992

Logiciels

1. Laurent Fesquet, "RTKPX : Développement d'un noyau temps réel pour un biprocesseur", stage ingénieur chez Julabo Labortechnik à Seelbach en Allemagne, 1992

V. Encadrement de thèses

Dérogação pour l'encadrement de la thèse d'Oussama Elissati. Cette thèse Cifre avec ST sur les oscillateurs asynchrones à faible bruit de phase devrait démarrer début 2008.

V.1 Thèses en cours

**Oussama Elissati - Oscillateurs en anneau et lignes à retard asynchrones
Applications aux PLLs, et aux systèmes de « Clock recovery »**

Débutée le 18 février 2008

Soutenance prévue en 2011

Direction : Laurent Fesquet

Co-direction : Sébastien Dedieu (ST) et Sébastien Rieubon (ST)

**Hatem Zakaria – Régulation asynchrone intégrée pour technologies décanométriques:
application à un système parallèle reconfigurable pour l'embarqué**

Débutée le 1^{er} octobre 2007

Soutenance prévue en 2010

Direction : Marc Renaudin

Co-direction : Laurent Fesquet

**Taha Beyrouthy – Logique programmable asynchrone pour systèmes embarqués
sécurisés**

Débutée le 1^{er} octobre 2006

Soutenance prévue en 2009

Direction : Marc Renaudin

Co-direction : Laurent Fesquet

Eslam Yayah – Réseaux de communication sur puce en logique asynchrone
Débutée le 1^{er} octobre 2005
Soutenance prévue en 2008
Direction : Marc Renaudin
Co-direction : Laurent Fesquet

Thèse Jérémie Hamon – Etude d’implémentation en logique asynchrone de traitement de signaux impulsionnels à ultra-large bande spectrale
Débutée le 1^{er} octobre 2005
Soutenance prévue en 2008
Direction : Marc Renaudin
Co-direction : Laurent Fesquet

Saeed Mian Quaisar – Systèmes de traitement de signaux non uniformément sur-échantillonnés localement en temps – application aux objets communicants
Débutée le 1^{er} octobre 2005
Soutenance prévue en 2008
Direction : Marc Renaudin
Co-direction : Laurent Fesquet

Julien Goulier – Etude et conception d’un système de conversion analogique-numérique reconfigurable intégré en technologie CMOS avancée. Application à une chaîne de réception multi-standard
Débutée le 1^{er} janvier 2005
Soutenance prévue le 26 mai 2008
Direction : Marc Renaudin
Co-direction : Laurent Fesquet et Eric André (ST)

Aurélien Bührig – Optimisation de la consommation des nœuds d’un réseau de capteurs
Débutée le 1^{er} octobre 2004
Soutenance prévue le 29 avril 2008
Direction : Marc Renaudin
Co-direction : Laurent Fesquet
Situation actuelle du doctorant : Ingénieur

V.2 Thèses soutenues

Vivian Brégier – Synthèse automatisée de circuits optimisés prouvés quasi-insensibles aux délais
Débutée le 1^{er} octobre 2003
Soutenu le 14 septembre 2007
Direction : Laurent Fesquet
Co-direction : Marc Renaudin
Situation actuelle du doctorant : Ingénieur

Ce travail de thèse porte sur la mise au point d'une technique de synthèse automatique de circuits asynchrones quasi insensibles aux délais (QDI). La méthode de synthèse permet, à partir d'une spécification du circuit dans un langage de haut niveau, CHP, de synthétiser un circuit en portes logiques élémentaires (comprenant les portes de Muller utiles pour la réalisation de circuits QDI). Le circuit est totalement décomposé : il n'est constitué que de

portes à deux entrées, ce qui permet d'effectuer une projection technologique en effectuant uniquement des fusions de portes (transformation qui préserve la quasi insensibilité aux délais d'un circuit). De plus, une étude formelle réalisée durant la thèse démontre que les circuits synthétisés respectent la contrainte de quasi insensibilité aux délais. Cette technique de synthèse a été développée au sein du projet TAST. Elle a été validée sur un ensemble de circuits de tests, qui sont des opérateurs arithmétiques générés automatiquement, de taille variable.

**Fabien Aeschlimann – Traitement du signal échantillonné non uniformément :
algorithme et architecture**

Débutée le 1^{er} octobre 2003

Soutenue le 6 février 2006

Direction : Marc Renaudin

Co-direction : Laurent Fesquet

Situation actuelle du doctorant : Ingénieur

Le travail mené dans le cadre de cette thèse est focalisé sur le traitement du signal en vue de la réalisation de fonctions en logique asynchrone. L'objectif est de se libérer de l'échantillonnage régulier au profit d'un échantillonnage non uniforme contrôlé par le signal lui-même dit « par traversée de niveaux ». Nous remettons en cause la chaîne classique de traitement, « conditionnement du signal », « échantillonnage » et « traitement numérique synchrone », car elle ne prend pas en compte la nature de l'information contenue dans le signal. C'est pourquoi en combinant la conception en logique asynchrone et l'échantillonnage par traversée de niveaux, une nouvelle classe de chaîne de traitement numérique du signal est définie sur la base d'un convertisseur Analogique/Numérique Asynchrone (CANA). La chaîne traite ainsi des échantillons prélevés en fonction des variations du signal d'entrée, associant une information sur l'amplitude et une information temporelle caractérisant la non uniformité, donc le signal. Cet échantillonnage est parfaitement adapté au traitement numérique asynchrone car, comme lui, il est contrôlé par des événements dont les instants d'occurrence sont a priori inconnus. On sort d'ici du cadre restrictif de la conception d'une algorithmique prévue pour le traitement des signaux échantillonnés régulièrement destinés à s'exécuter sur des circuits synchrones. La maîtrise de la conception de machines asynchrones nous amène donc à considérer une algorithmique basée sur une autre représentation des signaux numériques. L'étude théorique de l'échantillonnage par traversée de niveaux pour différentes classes de signaux montre que l'information temporelle est nécessaire lors d'un traitement afin de pondérer l'amplitude des échantillons. Des outils mathématiques ont ainsi été développés et mis en œuvre afin réaliser des opérations de base spécifiques à l'échantillonnage non uniforme : transformée de Fourier, filtrage à réponse impulsionnelle finie et infinie. Les traitements sont formalisés, analysés en terme d'erreur et de complexité combinatoire et comparés au cas régulier ; des architectures matérielles en sont alors déduites en vue de leur intégration. Les simulations fonctionnelles montrent qu'en dépit d'une complexité accrue, le traitement asynchrone de signaux peu actifs permet une forte réduction de la consommation électrique du circuit par rapport au cas synchrone du fait de l'absence de redondance dans l'information traitée. Il apparaît clairement que les bénéfices de cette approche seront exploitables pour la conception de circuits intégrés complexes ouvrant donc des perspectives nouvelles pour la réalisation de « capteurs intelligents » et de systèmes de communication.

Jérôme Quartana - Conception de réseaux de communication sur puce asynchrones : applications aux architectures GALS

Débutée le 1^{er} octobre 2000

Soutenue le 20 décembre 2004

Direction : Marc Renaudin

Co-direction : Laurent Fesquet

Situation actuelle du doctorant : Enseignant-Chercheur

Cette thèse porte sur l'étude d'architectures de communication sans horloge pour la conception de réseaux de communication asynchrones appliqués aux systèmes globalement asynchrones et localement synchrones. Elle s'intègre également dans le cadre du développement de l'outil de conception automatique de circuits asynchrones TAST (TIMA Asynchronous Synthesis Tool). L'importance des besoins de communication au sein des systèmes intégrés modernes fait du réseau d'interconnexion un acteur majeur de la complexité et des performances de ces systèmes. Parmi les nombreuses méthodologies de synchronisation de systèmes existantes, nous montrons l'intérêt de choisir un réseau d'interconnexion sans horloge pour la communication des systèmes globalement asynchrones et localement synchrones. La thèse étudie une méthodologie de conception d'un réseau d'interconnexion qui utilise les propriétés de modularité des circuits sans horloge. Nous découpons la construction de nos réseaux asynchrones en quatre modules majeurs : arbitrage, transport, routage et synchronisation. L'objectif de ce découpage simple est de permettre à terme la synthèse automatique d'arbitres et de réseaux de communication sans horloge dans l'outil de conception TAST. Les modules du réseau sont spécifiés en CHP, un langage de modélisation de haut niveau adapté à la synthèse de circuits asynchrones. A travers ces modélisations, nous mettons en relief l'importance des problèmes d'arbitrage et de synchronisation entre les blocs du système. Nous présentons un système de communication qui illustre cette méthodologie de construction de réseau par assemblage de modules et son degré d'automatisation actuel.

Antoine Sirianni - Modélisation, simulation et vérification de circuits numériques asynchrones dans le standard SystemC

Débutée le 1^{er} octobre 2001

Soutenue le 18 juin 2004

Direction : Marc Renaudin

Co-direction : Laurent Fesquet

Situation actuelle du doctorant : Ingénieur

Le sujet de la thèse se situe dans le contexte de la modélisation, de la simulation et de la vérification des circuits asynchrones. Le but de ces travaux est de définir un modèle approprié à la simulation et à la vérification des circuits asynchrones dans un environnement SystemC. Ce modèle permet notamment de valider les propriétés d'insensibilité aux délais des circuits asynchrones par le biais d'une instrumentation développée également en SystemC. Le fonctionnement du noyau de simulation est analysé en détail et une technique de vérification par simulation est mise en œuvre afin de garantir les propriétés d'insensibilité aux délais des circuits asynchrones modélisés. Ce travail apporte une contribution intéressante pour la modélisation et la vérification des circuits asynchrones avec SystemC.

Mohammed Es Salhiene – Spécification et conception d'un système d'exploitation faible consommation pour processeurs asynchrones

Débutée le 1^{er} octobre 2000 (à mi-temps jusqu'en 2002)

Non soutenue

Direction : Marc Renaudin
Co-direction : Laurent Fesquet
Situation actuelle du doctorant : Professeur certifié de génie électrique

Le contexte de cette thèse se situe dans le cadre la conception matérielle et logicielle de systèmes embarqués complexes. Le sujet de la thèse porte sur la spécification et la conception d'un système d'exploitation dédié aux microprocesseurs asynchrones (sans horloge). On s'intéresse en particulier à des systèmes faible consommation utilisant un OS associé à des processeurs asynchrones particulièrement adéquats pour être couplés à un système de régulation de la tension d'alimentation par un algorithme de DVS intégré à l'OS. La thèse a été menée à son terme mais pas soutenue, le candidat ayant renoncé pour des raisons personnelles. La soutenance était prévue en 2004.

Amine Rezzag - Synthèse logique de circuits asynchrones micropipeline

Débutée le 1^{er} octobre 2000

Soutenue le 13 décembre 2004

Direction : Marc Renaudin
Co-direction : Laurent Fesquet
Situation actuelle du doctorant : Ingénieur

Ce travail de thèse porte sur la définition d'une méthodologie de conception de circuits intégrés asynchrones micropipeline. La synthèse micropipeline est une approche qui exploite à la fois les outils commerciaux de synthèse pour le chemin de données, et la synthèse de contrôleurs asynchrones pour le contrôle (« STG » avec Petrify , « BURST MODE » avec Minimalist). La méthodologie générale pour la modélisation et la synthèse de circuits asynchrones est basée sur la spécification dite DTL (Data Transfer Level) qui définit une façon d'écrire les codes sources garantissant une synthèse rapide et systématique pouvant cibler plusieurs styles de circuits asynchrones. Cette méthode de conception part d'une spécification basée sur un langage de haut niveau (CHP ou Concurrent Hardware Processes). Elle permet en sortie de générer des circuits en portes logiques élémentaires et en portes de Muller. Il a été procédé à un prototypage de cette méthode de synthèse. Ce prototype est conçu pour être intégré dans l'outil de conception automatique de circuits asynchrones TAST (Tima Asynchronous Synthesis Tool) dont le synthétiseur génère des circuits asynchrones QDI, pour l'étendre à la génération de circuit micropipelines. Par ailleurs, la méthodologie de synthèse a été étendue à l'utilisation de différents types de contrôleurs asynchrones susceptibles d'en améliorer les performances en termes de vitesse et de consommation.

Anh Vu Dinh Duc – Synthèse automatique de circuits asynchrones QDI

Débutée le 1^{er} octobre 1999

Soutenue le 14 mars 2003

Direction : Marc Renaudin
Co-direction : Laurent Fesquet
Situation actuelle du doctorant : Enseignant-Chercheur

Cette thèse s'inscrit dans le domaine de la conception de circuits intégrés asynchrones. Son but était la création d'un outil de synthèse pour les circuits asynchrones les plus robustes, dits quasi-insensibles aux délais. A partir d'une description comportementale en langage CHP de la fonction à intégrer, on génère automatiquement un réseau de portes qui respecte le modèle quasi-insensible aux délais. La partie théorique de ce travail a mis en évidence une méthodologie de synthèse formalisée. En effet, un sous-ensemble synthétisable du langage

CHP a été défini et une forme intermédiaire indépendante des langages et des technologies, les équations de dépendances, a été mise en œuvre dans l'outil. L'optimisation logique des circuits asynchrones et la réalisation du matériel en utilisant exclusivement des cellules standard ont également été intégrées dans cet outil. Cette thèse a donné lieu à une réalisation logicielle conséquente.

VI. Activités d'enseignement

VI.1 Conception microélectronique

VI.1.1 INPG - ENSERG

- *Cours de conception avancée des circuits intégrés analogiques, 3^{ème} année (cours 7h30)*

Ce cours traite de la conception de circuits analogiques et plus spécifiquement de la synthèse des filtres à temps continu et à temps discrets, des dispositifs de conversion analogiques-numériques et des boucles à verrouillage de phase. Cet enseignement s'intéresse surtout au dimensionnement de ces dispositifs et aux contraintes de réalisation.

- *Cours de synthèse matérielle des algorithmes de traitement du signal (Cours 18h, TP 24h)*

Ce cours aborde les techniques formelles de synthèse pour traduire un algorithme de traitement du signal en matériel. L'ensemble du cours s'appuie sur des techniques de transformation de graphes et de projection sur les axes de temps et d'espace.

- *TP d'architecture des systèmes intégrés (TP 24h)*

L'objectif de ce TP est de confronter les étudiants à la réalisation et à l'optimisation d'un système intégré (sur une plateforme SoPC) par le biais d'une accélération matérielle. Les étudiants conçoivent tout d'abord un filtre RIF par logiciel (grâce au processeur embarqué sur la plateforme), puis implémentent l'algorithme matériellement, ajoutent un DMA et finalement gèrent l'ordonnancement des tâches du système par un petit système d'exploitation (en utilisant des sémaphores).

- *Cours de modélisation et de synthèse des systèmes matériels, 2^{ème} année (cours 12h30, TD 5h, TP 16h)*

Ce cours adresse les problèmes de la modélisation des circuits intégrés numériques et les techniques de synthèse à appliquer à un modèle de haut niveau pour obtenir un réseau de portes. Les techniques de prototypage et de validation sont abordées dans le cadre d'un TP.

- *Projet de microélectronique, 2^{ème} année (projet 72h)*

Ce projet vise la réalisation d'un circuit intégré numérique à partir d'une spécification de haut niveau jusqu'au layout. Ce travail implique la définition d'une architecture, l'écriture d'un modèle RTL, la synthèse du modèle et le placement-routage du circuit. Chaque étape est vérifiée et simulée. Dans la mesure du possible, une stratégie de test (DFT) est définie et implémentée. Les outils mis en œuvre sont les outils des suites de Mentor, Cadence et Synopsys. Exemples de sujet : microprocesseur MIPS R2000, microprocesseur ColdFire, hub à la norme USB, liaison infrarouge à la norme IRDA, cryptoprocresseurs DES et RSA, Cordic, coprocresseur virgule flottante à la norme IEEE, ...

VI.1.2 Formation continue INPG

J'ai contribué à la mise en place de deux filières de formation destinées à des ingénieurs.

- *Formation à la conception de circuits intégrés analogiques (1 semaine, cours et TPs)(2004)*

Cet enseignement comporte l'étude de tous les blocs et techniques de base de l'électronique analogiques : sources de courant, amplification, filtrage, compensation en fréquence, bandgap, techniques pour le dessin des masques, ...

- *Formation à la conception de circuits intégrés numériques (1 semaine, cours et TPs)(2004)*

Cet enseignement intègre l'étude de l'architecture et du flot de conception des circuits numériques. Notamment, il a été présenté les techniques de simulations, de vérification, de synthèse et de placement - routage. Les cours ont été prolongés par des séances de TPs qui avaient pour but l'étude la vérification, la synthèse et le routage hiérarchique d'un cœur de processeur MIPS R2000 (RISC 32 bits, pipeline à 5 étages).

- *Formatech : Cours de HDL et de synthèse (20h cours et TPs)(2005)*

Ce cours adresse les problèmes de la modélisation des circuits intégrés numériques et les techniques de synthèse. L'ensemble des connaissances est mis en œuvre par des TPs.

VI.1.3 Ecole Nationale Supérieure de Physique de Grenoble

- *Cours de conception des circuits intégrés analogiques, 2ème année (cours 16h)*

Ce cours traite de l'acquisition des bases nécessaires à la conception de circuits analogiques. L'enseignement s'intéresse aux cellules élémentaires des circuits analogiques et à leurs dimensionnements

VI.1.4 MASTER CSINA ET MNE

- *Projet de microélectronique*

Ce projet vise la réalisation d'un circuit intégré numérique à partir d'une spécification de haut niveau jusqu'au layout. Ce travail implique la définition d'une architecture, l'écriture d'un modèle RTL, la synthèse du modèle, sa vérification formelle et le placement-routage du circuit.

VI.2 Informatique

VI.2.1 INPG – ENSIMAG et Département télécoms

- *Projet informatique et langage C, cours de première année (cours 3h, projet 24h)*

Ce projet propose l'écriture d'un simulateur ou d'un assembleur compatible avec un processeur Intel Pentium capable de lire ou d'écrire le format ELF.

VI.2.2 INPG-ENSERG

- *Projet informatique et langage C, cours de deuxième année (projet 24h et tutorat 15h)*

Ce projet propose l'écriture d'un simulateur ou d'un assembleur compatible avec un processeur. Les processeurs étudiés sont : le 68000 de Motorola, le processeur Aspro, l'Intel 80386, le microcontrôleur Mica, le Pentium, le MIPS.

Annexes

Les publications sélectionnées et jointes au manuscrit sont listées ci-dessous.

I. Modélisation et conception d'arbitres pour NoC

J.-B. Rigaud, J. Quartana, L. Fesquet, M. Renaudin, "High-Level Modeling and Design of Asynchronous Arbiters for On-Chip Communication Systems", Design Automation and Test Conference (DATE), 4-7 March, 2002, Paris, France.

II. Réseau de communication sur puce asynchrone

Jérôme Quartana, Laurent Fesquet, Marc Renaudin, "Modular Asynchronous Network-on-Chip: Application to GALS Systems Rapid Prototyping" in VLSI-SOC: From Systems to Chips, (best selected contributions from VLSI-SoC 2005), Series: IFIP International Federation for Information Processing, Vol. 200, Glesner, M.; Reis, R.; Indrusiak, L.; Mooney, V.; Ekeking, H. (Eds.), Springer, 2006, pp 195-208

III. Synthèse logique de circuits QDI

Bertrand Folco, Vivian Brégier, Laurent Fesquet, Marc Renaudin, "Quasi Delay Insensitive circuits synthesis using Multi-valued Decision Diagrams", IFIP VLSI-SoC 2005, Perth, Australia, October 17-19, 2005.

IV. Convertisseur analogique numérique asynchrone

E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "Asynchronous Level Crossing Analog to Digital Converters", Special Issue on ADC Modelling and Testing of *Measurement*, Vol. 37, Issue 4, June 2005, Pages 296-309.

V. Filtrage RIF de signaux non uniformément échantillonnés

F. Aeschlimann, E. Allier, L. Fesquet, M. Renaudin, "Asynchronous Finite Impulse Response Digital Filter: A New Digital Signal Processing Chain", IEEE International Symposium on Asynchronous Circuits and Systems (Async'04), Hersonisos, Crete, April 19-23, 2004, pp. 198-206.

High-Level Modeling and Design of Asynchronous Arbiters for On-Chip Communication Systems

J.-B. Rigaud, J. Quartana, L. Fesquet, M. Renaudin Authors' affiliation 1
TIMA Laboratory, 46 avenue Félix Viallet, 38031 Grenoble, France
{Jean-Baptiste.Rigaud, Jerome.Quartana, Laurent.Fesquet, Marc.Renaudin}@imag.fr

Abstract

This paper addresses the design of complex arbitration modules, like those required in SoC communication systems. Clock less, delay-insensitive arbiters are studied in the perspective of making easier and practical the design of future GALS or GALA SoCs. The paper first focuses on high-level modeling of fixed and dynamic priority arbiters using CHP (Communicating Hardware Processes). It then describes delay-insensitive implementations derived from the CHP descriptions. Pre-layout simulations show that arbiters able to process several hundreds mega requests per second can be designed using the .18 CMOS process of STMicroelectronics.

1. Introduction

One of the critical components of a SoC is the communication system, commonly named on-chip bus. Such an on-chip communication system has to be very flexible to interface in-house and external virtual components, providing high bandwidth, low latency, low power, arbitration mechanisms and routing capabilities.

In a SoC the on-chip bus connects the components to each other and dynamically allocates a path from one block to another. Several blocks running concurrently may require accessing the same resource leading to contentions. In this case, an arbiter is needed to solve the conflicts and to ensure that only one block is accessing the resource. The choice is done using priorities affected to each request.

Several arbitration algorithms were proposed in the past to solve the problem of accessing a unique resource from an arbitrary number of blocks. These algorithms can be classified according to the characteristics of their corresponding hardware implementation. To mention a few, arbitration structures can be distributed or centralized, can be linear like daisy-chain arbiters, or ring-based like token-ring and round-robin arbiters.

Most of on-chip communication systems and the arbitration modules they include are today designed with synchronous circuits. In this paper, delay-insensitive asynchronous arbiters are considered, to be part of future

on-chip busses of GALA (globally asynchronous locally asynchronous) or GALS (globally asynchronous locally synchronous) SoCs.

In the SoC design perspectives, delay-insensitive arbiters have this main advantage of being hundred-percent reliable (enough time is given to resolve metastability). Today, reliability of on-chip communication systems is becoming a major issue because transaction rates' increase is dangerously reducing the so-called Mean Time Before Failure characterizing clocked synchronizers.

As far as power consumption is concerned, such event driven communication/arbitration structures have a minimal electrical activity. Indeed, unless clocked circuits, power consumption of delay-insensitive asynchronous arbiters is proportional to access rates [6].

Furthermore, delay insensitivity enables the design of fast "long distance" communication busses [5].

Finally yet importantly, such delay-insensitive communication systems are fully autonomous blocks, which can easily be used/reused in complex SoC architectures as soft, firm or hardware virtual components, hence decreasing design time and complexity.

Based on these motivations, the paper contributes to two fundamental issues. Section 2 deals with modeling arbitration algorithms using a high-level description language called CHP (Communicating Hardware Processes). Section 3 describes the delay-insensitive arbiters derived from the CHP specifications. Section 4 reports pre-layout simulation results. Priority arbiters are considered as representative case studies throughout the paper. The last section concludes this work and mentions the main prospects.

2. Arbiter Modeling

This section first considers some traditional arbitration algorithms (daisy chain and tree structures) to introduce the main CHP language constructs. The section then focuses on recent priority-arbiter architectures proposed in [1] and shows how they are modeled in CHP. The CHP language, proposed by Martin [2], is naturally adopted in this work because i) it includes non deterministic choice structures required to model arbitration, and ii) it is very

well suited to model and synthesize delay-insensitive circuits [2][3].

2.1. Fixed-priority arbiter modeling

Fixed-priority arbiters have to choose among input requests with predefined hardware-coded priority values.

The CHP program of figure 1 describes a 4-input daisy-chain arbiter implementing a sequential priority scheme. The process is by default in sleep mode, and wakes up each time an activity is detected on at least one of the input requests. The probe CHP operator (denoted #) is used to watch activity on inputs and trigger processing (Figure 1, line 1). Priority is simply modeled by sequentially testing the requests. Highest priority input R3 is analyzed first. It is tested active, #R3 statement (Figure 1, line 2). When active, the shared resource S is attributed to R3 by writing to S using the CHP write statement S!. Concurrently (“,” CHP notation), request R3 is granted using the CHP read statement denoted R3?. R3 is also tested not active, statement !#R3 (Figure 1, line 3). In this case, next channel R2 is analyzed the same way, and so on for R1 and R0.

Note that the stability of guards #R3 and !#R3 is not guaranteed [2]. In fact, the request signal level R3 may change while it is evaluated. In this case, a non-deterministic choice denoted “@@” is used in CHP. At the hardware level, it corresponds to using a synchronizer in charge of solving metastability that may occur while deciding whether R3 is zero or one.

```
* [#( R0 ∨ R1 ∨ R2 ∨ R3 )
→ [ #R3 → S!, R3?
  @@ !#R3 → [ #R2 → S!, R2?
             @@ #R2 → [ #R1 → S!, R1?
                    @@ !#R1 → S!, R0?
  ] ] ] ]
```

Figure 1. Daisy-chain arbiter CHP model

Another very common arbitration structure, a binary-tree arbiter, is modeled in figure 2. The program clearly shows the tree structure of the selection process.

```
* [#( R0 ∨ R1 ∨ R2 ∨ R3 )
→ [ [ [ #R3 → s1 := 2 (1)
      @@ #R3 → [ #R2 → s1 := 1
                @@ !#R2 → s1 := 0
      ] ] ],
  [ [ #R1 → s0 := 2 (2)
    @@ !#R1 → [ #R0 → s0 := 1
              @@ !#R0 → s0 := 0
    ] ] ] ; (3)
[ s1 = 2 → R3?, S!
@ s1 = 1 → R2?, S!
@ s1 = 0 → [ s0 = 2 → R1?, S!
           @ s0 = 1 → R0?, S!
  ] ] ]
```

Figure 2. Tree arbiter CHP model.

Here again, line one is used to sense input activities. Part one and part two of the program (respectively labeled 1 and 2 in figure 2), concurrently solve R3/R2 and R1/R0 contentions in the first stage. Priority of R3 over R2

(respectively R1 over R0) is modeled by a two-stage linear structure first checking R3 (respectively R1) and then R2 (respectively R0). Then, the CHP sequential operator “;” is used to specify that the concurrent parts of the first stage have to complete before stage 2 can process (label 3 in figure 2). In this last stage, deterministic choices are used since variables s0 and s1 are stable (“@” notation). According to s0 and s1 values, the winning input is granted and the shared resource is concurrently accessed.

Recently, Bystrov, Kinniment and Yakovlev introduced in [1] an enhanced version of priority arbiters that decouples request signals sampling (synchronizer module) and contention solving (priority module). This structure outperforms previously proposed structures in terms of complexity (one synchronizer per input) and latency. Moreover, unlike the daisy chain and the tree arbiters, this new arbiter structure is strong modular, and priorities can be modified by only redesigning the priority module. However, in [1] the design of this priority arbiter’s family was very intuitive and performed by hand.

Figure 3 gives a formal CHP specification of a 4-way fixed-priority arbiter like those proposed in [1]. Line 1 is used to sense request activities. Then, four identical subparts are modeling the concurrent sampling of the request signals (labeled synchronizer 3 to 0). Variables s0 through s3 are used to store the samples, which are then exploited by the fixed-priority module to figure out which input has to elect (labeled fixed-priority module).

```
* [#( R0 ∨ R1 ∨ R2 ∨ R3 ) -- loop control
→ [ [ [ #R3 → s3 := 1 -- synchronizer 3
      @@ !#R3 → s3 := 0
    ],
  [ [ #R2 → s2 := 1 -- synchronizer 2
    @@ !#R2 → s2 := 0
  ],
  [ [ #R1 → s1 := 1 -- synchronizer 1
    @@ !#R1 → s1 := 0
  ],
  [ [ #R0 → s0 := 1 -- synchronizer 0
    @@ !#R0 → s0 := 0
  ] ] ] ;
-- fixed-priority module
[ s3 = 1 → R3?, S!
@ s3 = 0 → [ s2 = 1 → R2?, S!
           @ s2 = 0 → [ s1 = 1 → R1?, S!
                     @ s1 = 0 → R0?, S!
           ] ] ] ] ] ] ]
```

Figure 3. Fixed-priority arbiter CHP model of [1].

2.2. Dynamic-priority arbiter modeling

In a dynamic-priority arbiter, each input channel is carrying request signaling as well as priority values. The dynamic-priority arbiter performs priority comparisons of active requests and grants the one with the highest value. The input having the highest index is by convention selected in case of identical priority levels. Figure 4 presents the program of a 4-way dynamic-priority arbiter. Each input R_i is a dual rail channel, encoding the request signal together with a priority value ranging from 0 (lower

to 1 (higher). Only two priority-levels are considered for the sake of simplicity. This can easily be extended and adapted to the requirements of real applications.

```

* [ #( R0 ∨ R1 ∨ R2 ∨ R3 )                -- loop control
→ [ [   #R3 → s3 := 1, [ #R3 = 0 → r3 := 0   -- synchronizer 3
    @ #R3 = 1 → r3 := 1 ]
  @@ #R3 → s3 := 0
  ],
  [   #R2 → s2 := 1, [ #R2 = 0 → r2 := 0   -- synchronizer 2
    @ #R2 = 1 → r2 := 1 ]
  @@ #R2 → s2 := 0
  ],
  [   #R1 → s1 := 1, [ #R1 = 0 → r1 := 0   -- synchronizer 1
    @ #R1 = 1 → r1 := 1 ]
  @@ #R1 → s1 := 0
  ],
  [   #R0 → s0 := 1, [ #R0 = 0 → r0 := 0   -- synchronizer 0
    @ #R0 = 1 → r0 := 1 ]
  @@ #R0 → s0 := 0
  ] ];
-- dynamic-priority module
-- 1st stage: 2-way request analysis and priority comparators
[ [ (s0, s1) = (0, 0) → v1_0 := 0
  @ (s0, s1) = (1, 0) → v1_0 := 1
  @ (s0, s1) = (0, 1) → v1_0 := 2
  @ (s0, s1) = (1, 1) → [ r1 ≥ r0 → v1_0 := 2
                        @ r1 < r0 → v1_0 := 1 ]
  ],
  [ (s2, s3) = (0, 0) → v3_2 := 0
  @ (s2, s3) = (1, 0) → v3_2 := 1
  @ (s2, s3) = (0, 1) → v3_2 := 2
  @ (s2, s3) = (1, 1) → [ r3 ≥ r2 → v3_2 := 2
                        @ r3 < r2 → v3_2 := 1 ]
  ] ];
-- 1st stage acknowledgment
[ (v1_0, v3_2) = (1, 0) → comp := 0, R0?, S!
  @ (v1_0, v3_2) = (2, 0) → comp := 0, R1?, S!
  @ (v1_0, v3_2) = (0, 1) → comp := 0, R2?, S!
  @ (v1_0, v3_2) = (0, 2) → comp := 0, R3?, S!
  -- multiplexer
  @ (v1_0, v3_2) = (1, 1) → comp := 1, C0 := r0, C1 := r2
  @ (v1_0, v3_2) = (1, 2) → comp := 1, C0 := r0, C1 := r3
  @ (v1_0, v3_2) = (2, 1) → comp := 1, C0 := r1, C1 := r2
  @ (v1_0, v3_2) = (2, 2) → comp := 1, C0 := r1, C1 := r3
  ];
-- 2nd stage: 2-way priority comparator
[ comp = 1 → [ C1 ≥ C0 → out := 1
              @ C1 < C0 → out := 0 ]
  @ comp = 0 → skip
  ];
-- 2nd stage acknowledgment
[ comp = 1 → [ out = 1 → [ (v1_0, v3_2) = (1, 1) → R2?, S!
                          @ (v1_0, v3_2) = (1, 2) → R3?, S!
                          @ (v1_0, v3_2) = (2, 1) → R2?, S!
                          @ (v1_0, v3_2) = (2, 2) → R3?, S! ]
                @ out = 0 → [ (v1_0, v3_2) = (1, 1) → R0?, S!
                              @ (v1_0, v3_2) = (1, 2) → R0?, S!
                              @ (v1_0, v3_2) = (2, 1) → R1?, S!
                              @ (v1_0, v3_2) = (2, 2) → R1?, S! ]
  ];
  @ comp = 0 → skip
  ] ];

```

Figure 4. Dynamic-priority arbiter CHP model.

The first stage is very similar to that of the fixed-priority arbiter (subparts labeled synchronizer 3 to 0). The priority module is much more complicated because it has to perform priority-level comparisons of competing inputs. The program proposed in figure 4 describes a two-stage comparator structure, but any other structure could be modeled leading to different tradeoffs in complexity, speed

and power. The first stage concurrently analyses pairs of inputs and if necessary concurrently performs priority comparisons. Variable $v1_0$ indicates whether no request occurs (value 0), request $R0$ (value 1) or request $R1$ (value 2) passed through stage one. Variable $v3_2$ plays the same role for inputs $R3$ and $R2$. Variables $v1_0$ and $v3_2$ cannot be both at zero since priority comparators are triggered on requests' activity. If only one request comes out stage one, the corresponding input is immediately granted (label 1st stage acknowledgment in figure 4). If two requests come out stage one, then another comparison has to take place. A multiplexer identifies the two requests in competition and sends their priority values to the second stage's comparator (subpart labeled 2nd stage: 2-way priority comparator in figure 4). Finally, the issue of the comparison together with the couple of the inputs in contention is used to figure out which input is the winner (subpart labeled 2nd stage acknowledgment in figure 4).

3. Arbiter design

In this section, delay-insensitive gate-level implementations are derived from the CHP specifications of the fixed and dynamic priority arbiters. All implementations use a four-phase handshaking protocol. Single rail input channels are used for the fixed-priority arbiter, whereas dual-rail input channels are used for the dynamic-priority arbiter. R_i denotes the request signal associated to the single-rail channel R . R_i_0 and R_i_1 denote the two rails of the dual-rail input channel R . R_o denotes the acknowledge signal of input channel R . Finally, the shared resource is assumed to be accessed through a single-rail channel S .

In the schematics, gates marked with a C are Muller C elements. An R (respectively S) is added when the gate output needs to be set to zero (respectively one) at reset. A "+" is added to specify asymmetric behavior. Input marked with a + only contributes to driving the output to one.

3.1. Fixed-priority arbiter design

The structure of the 4-way fixed-priority arbiter is described in figure 5. It is composed of three blocks. The loop control block is in charge of reactivating the arbiter after an input request has been served and the shared resource accessed. This block includes the trigger block, which senses input activities to keep the arbiter quiet as long as no request occurs. The second block consists in four synchronizers [2] that sample the input requests. The third block is the fixed-priority module that determines the input request to grant according to hardware-coded priorities. Figure 6 presents a delay-insensitive implementation of the fixed-priority logic block.

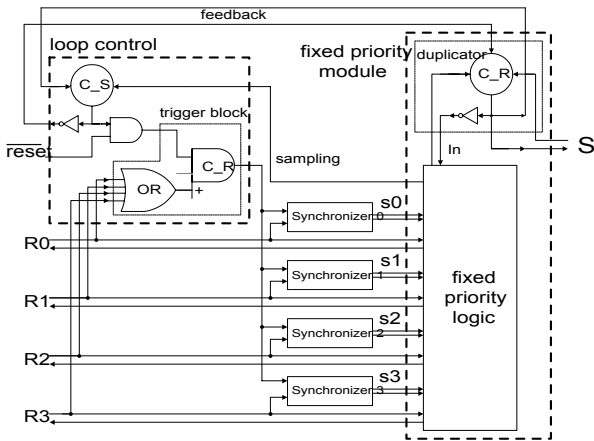


Figure 5. Fixed-priority arbiter architecture.

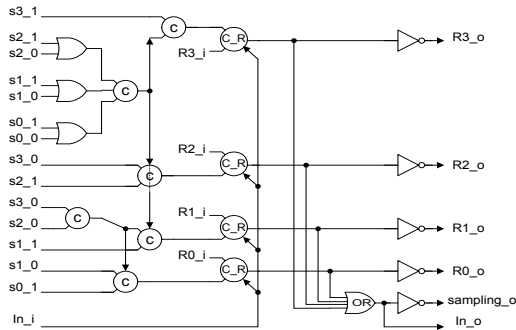


Figure 6. Fixed-priority logic block.

3.2. Dynamic-priority arbiter design

Figure 7 presents the architecture of the 4-way dynamic-priority arbiter specified by the CHP program of figure 4.

The two 2-way request analyzer & priority comparator blocks concurrently compare two pairs of requests. Each of them delivers the priority-holder request. Details of the gate-level structures are given in figure 8. It can be seen that the 2-way priority comparator is only activated when both input requests are active. The first stage of acknowledgment detects if a unique request is active at the output of the two 2-way request analyzers and priority comparators. In that case, the corresponding input is immediately granted and the shared resource accessed, preventing from useless power consumption and latency.

The logic implementation is described in figure 9. The upper part implements the favorable cases when a unique request passed through the first stage of the priority module. When more than one request passed through, the two multiplexers described at the bottom of figure 9, propagate the priority levels to the final two-way priority comparator included in the second stage. The issue of the comparison is used by the second acknowledgment stage to grant the selected input request (Figure 10 gives the second acknowledgment stage gate-level implementation).

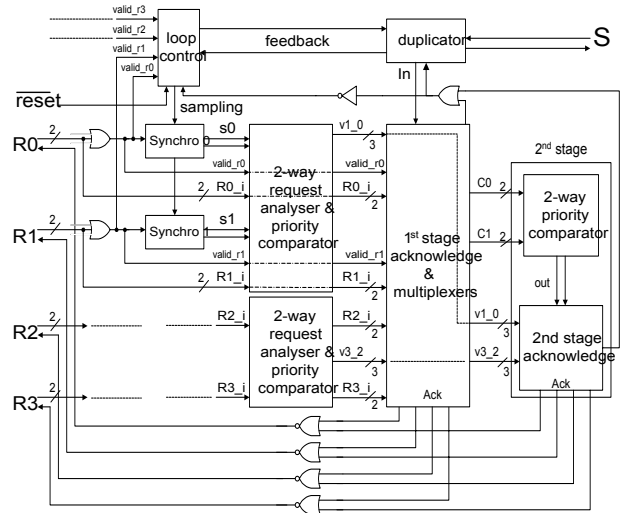


Figure 7. Dynamic-priority arbiter architecture.

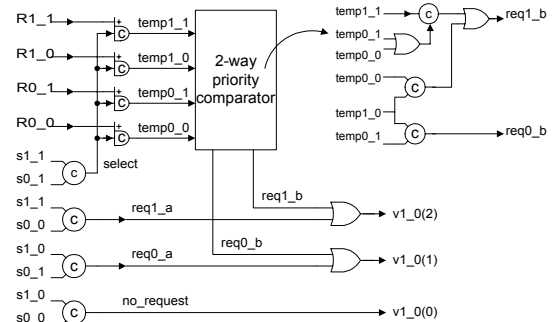


Figure 8. 2-way request analyzer and priority comparator.

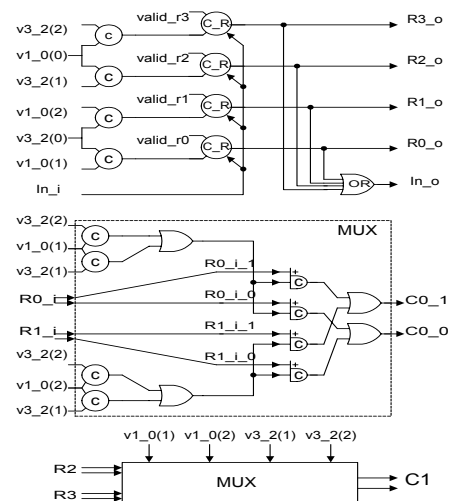


Figure 9. 1st stage acknowledgment & multiplexers.

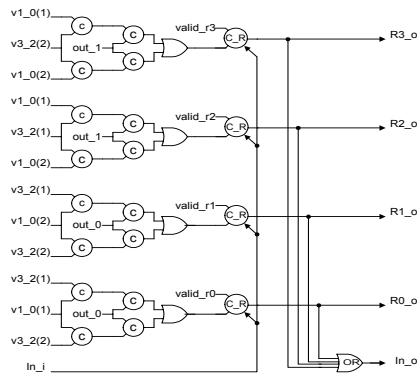


Figure 10. 2nd acknowledgment stage.

4. Design methodology & simulation results

Deriving the gate-level circuits from the CHP program is not fully automated so far. However, the schematics are obtained following a formal procedure that is beyond the scope of this paper and will be detailed in future communications. It is based on the formal methodology presented in [2].

The design flow used is the one presented in [4]. Gate-level implementations of the CHP programs are all described using VHDL gate netlists that are verified by back-annotable logic simulations with timing. Then, circuit netlists are imported into the Cadence™ framework for electrical simulations, placement and routing. The technology used for this study is a CMOS .18 μm process from STMicroelectronics.

A pre-layout electrical simulation is shown in figure 11 for the fixed-priority arbiter. Table 1 summarizes latency and cycle time figures obtained simulating the gate-level netlists. The fixed-priority arbiter has a forward latency of about 1.5 ns and a cycle-time of 2.2 ns corresponding to an access rate around 450 Mhz. The dynamic-priority arbiter was simulated with two input sets to measure best and worst case latencies (as explained in section 3). It can sustain a request rate of about 350 MHz and 180 MHz.

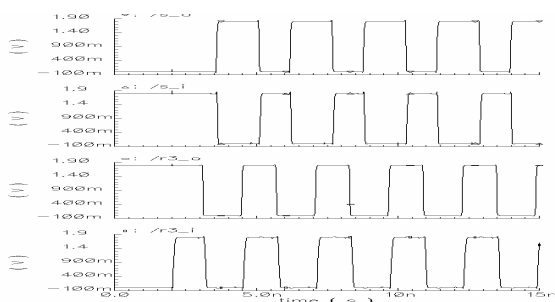


Figure 11. fixed-priority arbiter simulation.

Using more aggressive optimizations based on fast handshaking components [7] would increase the speed.

	Fixed-priority arbiter	Dynamic-priority arbiter	
		Best case	Worst case
Fwd latency	1.49 ns	2.22 ns	3.51 ns
Cycle time	2.22 ns	2.85 ns	5.59 ns
Throughput	450 MHz	351 MHz	179 MHz

Table 1: latency and cycle-time of simulated arbiters.

5. Conclusion

In this paper, it is shown that arbiters can be modeled using the high-level language CHP and their corresponding delay-insensitive implementations derived. Pre-layout electrical simulations report that delay-insensitive priority arbiters, processing several million requests per second, can be designed using an up-to-date CMOS technology.

This work actually proves that it is today possible to cleanly and formally model and design delay-insensitive arbiter modules that are reliable, modular and fast. It definitely constitutes an enabling factor for the asynchronous technology to be more and more seriously considered in the design of SoCs.

Prospective works will be focused on the automation of the synthesis process and the improvement of arbiter architecture and circuit performances. “n to p” fixed or dynamic priority arbiters will be considered to target the design of complex, multi-way on-chip routing systems.

References

- [1] A. Bystrov, D. J. Kinniment, A. Yakovlev, “Priority Arbiters”, in “International Symposium on Advanced Research in Asynchronous Circuits and Systems” (ASYNC), Eilat, Israel, April 2000, pp. 128-137.
- [2] A.J. Martin, “Synthesis of Asynchronous VLSI Circuits”, Internal Report, Caltech-CS_TR-93-28, California Institute of Technology, Pasadena, 1993.
- [3] J-B. Rigaud, M. Renaudin, “Modeling and design/synthesis of arbitration problems”, AINT’2000, Proceedings of the Asynchronous Interfaces: Tools, Techniques and Implementations Workshop, TU Delft, The Netherlands, July 19-20th 2000, pp. 121-128.
- [4] M. Renaudin, P. Vivet, F. Robin, “A Design Framework for Asynchronous/Synchronous Circuits Based on CHP to HDL Translation”, in “International Symposium on Advanced Research in Asynchronous Circuits and Systems” (ASYNC), Barcelona, Spain, April 19-21, pp 135-144, 1999.
- [5] W.J. Bainbridge, S.B. Furber, “Delay Insensitive System-On-Chip Interconnect using 1-of-4 Data Encoding”, in “International Symposium on Advanced Research in Asynchronous Circuits and Systems”, Salt Lake City, USA, March 11-14, pp 118-126, 2001.
- [6] C. Piguet, M. Renaudin, T. Omnes, “Special Session on Low-power Systems on Chips, Design Automation and Test in Europe (DATE), Munich, Germany, March 13-16, 2001, pp. 488-494.
- [7] Andrew Lines, “Pipelined Asynchronous Circuits”, Master Thesis, Caltech-CS-TR-95-21, 1995, revised June 1998.

Modular Asynchronous Network-on-Chip: Application to GALS Systems Rapid Prototyping

Jérôme Quartana, Laurent Fesquet, Marc Renaudin
TIMA Laboratory, 46 av. Felix Viallet 38031 Grenoble Cedex, France
{Jerome.Quartana, Laurent.Fesquet, Marc.Renaudin}@imag.fr

Abstract

This paper presents an innovating methodology for fast and easy design of Asynchronous Network-on-Chips (ANoCs) dedicated to GALS systems. A topology-independent building-block approach permits to design modular, scalable and reliable ANoCs with low-power and low-complexity requirements. A crossbar generator is added to the existing design flow for fast system architecture exploration. A multi-clock FPGA allows a fast prototyping of a complex ANoC-centric GALS system. It includes synchronous standard IP cores and asynchronous modules connected through an asynchronous 5x5 crossbar. First results about communication costs/performances across the Asynchronous NoC are reported.

1. Introduction

GALS paradigm is to partition a system design in decoupled clock-independent modules [1]. Design parameters of each block can be adjusted independently (performance, power consumption or clock-tree management to name but a few). Another benefit of GALS paradigm is to separate the design of communication from functionality by using handshake protocol synchronization (amongst other techniques).

Asynchronous NoCs (ANoCs) strongly benefit to such a globally asynchronous design methodology. Clockless interconnect networks improve reliability by removing clock-domain crossing synchronizations and by using delay-insensitive arbiters for solving routing conflicts [2] [3]. Global design constraints are released. They also offer robust communications thanks to an automatic data transfer regulation (elastic pipeline): no data item can be lost or duplicated. Moreover, regular distributed network topologies (any topology based on point-to-point links, such as meshes, torus or crossbars), built of independent routing nodes, fully exploit modularity and locality design properties of asynchronous circuits. To illustrate these benefits of using ANoCs for GALS systems, several publications bringing major research contributions can be cited.

In [4] a stoppable clock methodology, based on asynchronous wrappers around synchronous blocks, is used to compare topology performances by using ad-hoc

synchronous peripherals adapted to the asynchronous networks. Such techniques need training sessions and suffer from PVT sensitivity [21] and from penalties in restarting the clocks.

Beigne et al. present in [3] an asynchronous mesh topology providing a high Quality-of-Service (QoS), using a multi-level design flow. This very efficient ad-hoc architecture is dedicated to a specific application and has a high complexity cost. Bolotin and al. use in [5] a generic architecture to evaluate four classes of packet services. After a training session on every class, the most appropriate service is implemented onto a point-to-point link between two components, according to the communication requirements. This NoC architecture is more modular than [3] but for a higher complexity cost.

In [6] and [7], Bainbridge and Lovett develop a modular and low-complexity ANoC design methodology, using simple one-to-two and two-to-one switches to build regular topology networks. In such structures, arbiters are very simple and so efficient for packet routers with few channels to drive. However, assembling these switches will heavily increase latency and area costs for large multi-inputs/outputs routers.

Compared to these works, our purpose is to provide a simple and flexible generic structure which allows fast design of a large spectrum of ANoC topologies for GALS systems requiring efficient communications at a low complexity cost. According to this motivation, this paper presents in section 2 a topology-independent structure which is strongly modular, scalable and robust and which permits by using accurate-function building blocks to design ANoCs for high-reliability, low-power and low-complexity requirements. Section 3 gives some details to the self-timed FIFO structure of the synchronous/asynchronous domains interfaces. The section 04 details the design flow methodology. A crossbar generator has been developed for fast system architecture exploration. As such a flexible ANoC structure is well-suited for rapid GALS system prototyping [9], we remind a special methodology [8] to synthesize asynchronous modules onto FPGA, with an extension for non-deterministic arbiter circuits [9]. In section 5, this methodology is applied to implement an ANoC-centric GALS system onto a multi-clock FPGA. First results about communication costs/performances across the Asynchronous NoC are reported.

2. Asynchronous NoC design

Our methodology fully exploits the modularity of asynchronous circuits. We provide a basic layered structure of ANoC with no predefined topology, by using a building-block approach. Each block or layer has been accurately defined to efficiently deliver one of the major functions of an interconnect network (these functions are detailed in section 02.2 with the description of each block):

- service-level communication protocols,
- synchronization interfaces at mixed-timing domains,
- signal-level information transport,
- packet arbitration and routing in interconnect nodes.

Moreover, the basic blocks have been designed with an objective of reliability improvement (section 2.1) and with respect to low-complexity, “easy-plug” and scalability features. The result is a simple and flexible structure having efficient latency and throughput and a wide variety of high-level services at low-complexity and low-power costs. Such structure allows fast design of any ANoC regular distributed topology.

2.1 Focus on synchronization bolts

Our methodology for designing ANoCs is focused in part on solving synchronization problems. The two major synchronization bolts for a GALS system are: synchronization at clock domain boundaries and arbitration between concurrent requests [14]. Such circuits have a non-deterministic behavior. We put special invest to improve reliability/performance tradeoffs of these synchronizer circuits.

Clocked synchronization. As discussed in the introduction, using an ANoC is in itself a reliability improvement by removing clock-domain crossing synchronizations through the interconnect network. However *clocked* synchronizers are still required between Synchronous peripheral Blocks (SB) and the ANoC. Discussion on this synchronous/asynchronous interface is developed in section 02.2 and structure details are given in section 3.

Delay-insensitive arbiters. Arbitration circuits, or simply arbiters, are required where a restricted number of resources are allocated to different user or client processes. Packet routers are such cases. In the case of an ANoC, delay-insensitive arbiters have this main advantage of being hundred-percent reliable (enough time is given to resolve metastability). Reliability of on-chip communication systems is becoming a major issue since the increase transaction rates are drastically reducing the so-called Mean Time Between Failure characterizing clocked synchronizers. In [2] we present a class of delay-insensitive arbiters which decouple the sampling of incoming requests from the arbitration process in a strong modular and reliable structure. Such arbiters use a Parallel-Request-Sampling structure and are used in [3], [9] and in the following ANoC structure.

2.2 Modular ANoC structure

We cut out the construction of ANoCs in five basic components or layers, as illustrated in Figure 1.

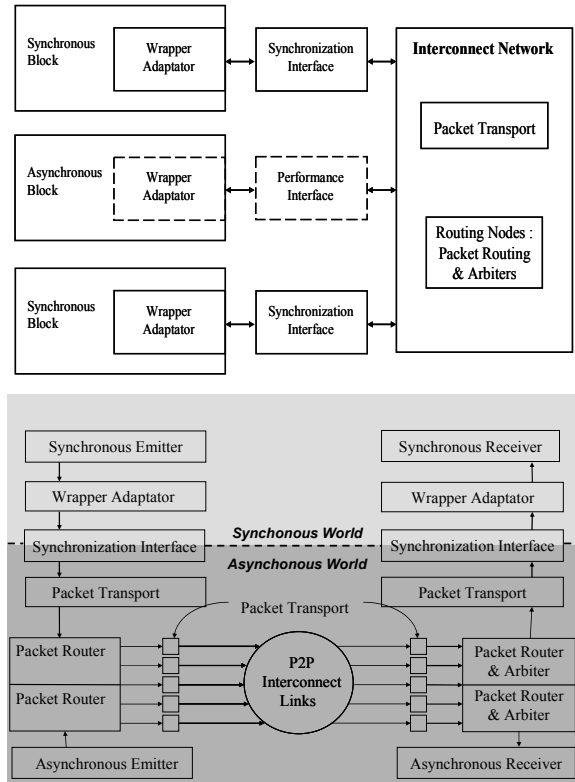


Figure 1. ANoC-centric GALS architecture:
a) abstract structure b) layered structure

1. Wrapper Adaptor (WA). This resource is required to translate between the communication protocols used by a synchronous or asynchronous peripheral and the interconnect network. The WA component adapts both flit and packet levels of the communication protocols. The details of these protocols are beyond the scope of this paper [3].

2. Synchronization & Performance Interface (SPI). This component binds the SB clock domain with the ANoC using a FIFO decoupling method. The SPI consists of a standard double flip-flop (DFF) synchronizer and of an asynchronous FIFO. Such simple synchronization interface facilitates plugging of standard synchronous IP cores.

The DFF resynchronizes asynchronous signals with the SB clock. The DFF offers actually a very sufficient reliability/latency tradeoff (two clock cycles per input signal sampling) [15], compared to numerous clocked synchronizer’s improvements [16].

The asynchronous FIFO transforms the synchronous protocol in the corresponding asynchronous protocol, adapting relative speeds between the SB and the ANoC. For AB, such a FIFO is optional and can be used for pipeline performance optimization. In this case we call it Performance Interface (PI) (Figure 1). Details of the

asynchronous FIFO structure are presented in section 3. This architecture is based on an existing asynchronous FIFO [17]. The level of parallelism between data and control flows is improved and two versions are delivered: a low-latency version or a low-power consumption version, according to design requirements.

3. Packet Transport (PT). This resource adapts the physical level (or signal-level) of the communication protocol. The PT component provides successive protocol conversions from SPI component to delay-insensitive NoC core for best power consumption and robustness. Between SPI and PR layers, bundle data protocols are converted in delay-insensitive protocols for better robustness. Between the packet routers (PR layer), the four-phase protocols can be converted in 2-phase protocols for long interconnect links for lower power consumption and higher speed [18].

4. Parallel-Request-Sampling Priority-Arbitrer (PRS-PA). This resource provides a self-timed arbiter with a decoupled arbitration process and a 100% reliable request sampling structure based on delay-insensitive parallel synchronizers [2][19] (section 2.1).

5. Packet Routing (PR). This resource offers a modular routing of data items for transaction services (packet level services such as burst mode or split transactions). PRS-PA and PR resources are parts of ANoC routing nodes, as detailed in section 2.3.

2.3 Switches architecture for ANoC routing nodes

Packet router is the core component of an interconnect network. The packet routers are assembled with modular elementary blocks, as shown in Figure 2, with the same objectives of low-complexity, easy “plug-and-play” and scalability as for the complete ANoC.

Figure 2 illustrates two switch instances. The n-to-1 switch, or Emitter, is built around the PR and PRS-PA components, as previously presented in section 2.2. The PR resource is decomposed in three modules: Packet Analyzer (PA), Data Path Controller (DPC) and MUX module. The Emitter component delivers two major classes of packet level services: arbitration service and transaction service. The PA block decodes *Channel_ctrl* message in order to extract arbitration and transaction information parts and to drive it respectively to the PRS-PA and DPC modules. Arbitration information is a request signal incoming with priority level (optional), used by the PRS-PA module to arbiter incoming requests and to elect a *Channel_data*. Transaction information delivers packet status, such as single flit packet or for burst mode: start-packet flit, body flit, end-of-packet flit. DPC exploits it to control data flow (burst mode or split transaction) on the elected *Channel_data* and to drive the switch output (MUX module).

The 1-to-m Switch, or Receiver, is a PR component which realizes the dual operation by driving the unique input to the selected *Target Address*. No arbitration is needed here. By composing these switches we can build

in short design time fast and efficient routing nodes (sections 4.2 & 5.1).

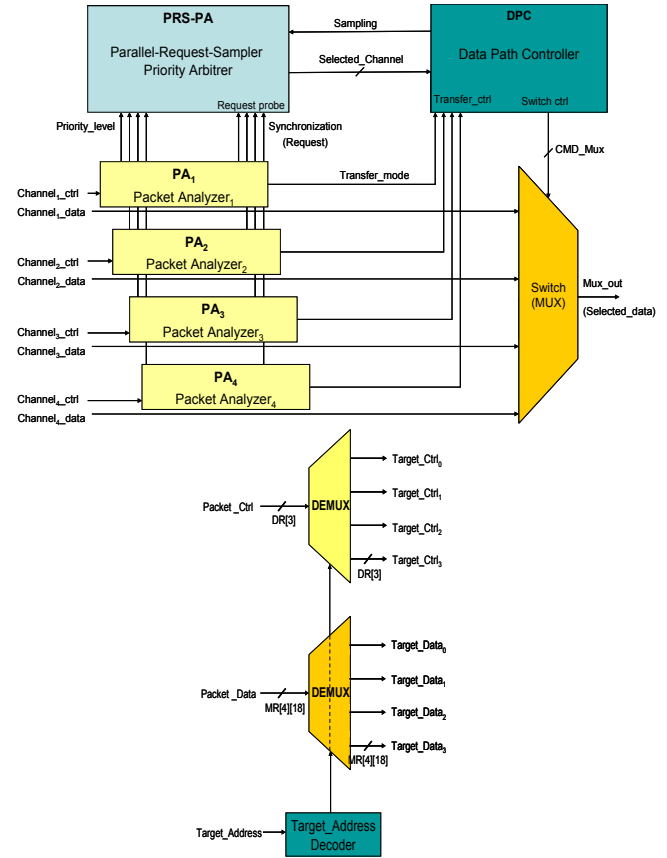


Figure 2. Switch components: a) Emitter (n-to-1 switch) b) Receiver (1-to-m switch)

3. Asynchronous FIFO for mixed domain interfaces

3.1 Reference work

Asynchronous FIFO. Chelcea and Nowick present in [17] several mixed-timing FIFO designs. The designs are implemented as a core of micropipeline-style circular arrays of identical cells connected to common data buses. Data items are not moved around the array once they are enqueued, preserving power consumption. Control is made with two tokens: one that allows enqueueing data and the other that allows dequeuing data. This asynchronous array is scalable and modular and offers very low latency.

Mixed-timing FIFO. The core of these asynchronous FIFO cells are used to design instances of double-clock FIFOs and in your concern mixed synchronous/asynchronous FIFOs. In this last case, either a Put or a Get complex interface (respectively for data item enqueueing or dequeuing) is added to the synchronous part to synchronize data transfers and to control the full or empty state of the FIFO. Full and empty state detectors in the Put and Get interfaces guarantee a write or read operation per cycle on the

synchronous part of the FIFO in normal working (FIFO neither full nor empty), i.e. for closed working frequencies on each part of the FIFO.

But such a situation can not be guaranteed for a mixed-timing FIFO interfacing a SB with an ANoC: these two domains will present very different working speeds. Preliminary results analysis on the FPGA platform confirms such different relative speeds (see section 5.2). Therefore the FIFO will often be empty or full and speed performances will be degraded by a global three clock cycles latency cost: two clock cycles for a DFF synchronizer plus one more flip-flop to prevent a greedy path [16] on the empty detector's DFF synchronizer.

We decide not to use this mixed version of the FIFO to avoid the use of such complex and latency-penalizing interfaces. With the help of a standard DFF synchronizer which reduces the latency to two cycles, an improved version (section 3.2) of the fully asynchronous FIFO is provided to interface synchronous and asynchronous working domains. This solution is robust (see section 2.1) and efficient to adapt domains with large difference in working speeds. Details of the asynchronous FIFO architecture improvements will be presented in a future paper, but we can say a few about it.

3.2 Improved asynchronous FIFO

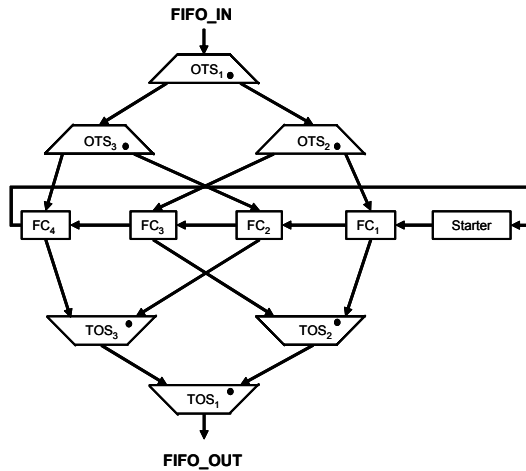


Figure 3. FIFO structure with mux/demux trees.

The architecture of the fully asynchronous FIFO is transformed in two ways to improve its performances.

1. Improved level of parallelism. This architecture as a limited degree of parallelism between control and data paths (token passing and data enqueueing/enqueueing operations). We use the TAST tool suite features to improve it, and consequently the speed of the FIFO. A FSM modeling of the FIFO in CHP language allows a decoupling of token passing and data enqueueing/dequeueing operations. TAST synthesizer options allow to parameter the synchronization point between these operations and therefore ensure the correctness of the FIFO. Both delay-insensitive and micropipeline versions of a FIFO can be synthesized.

2. Low-power and fast architecture exploration.

The common data buses give increasing power consumption penalties for deep FIFOs. Moreover, the bus buffers have to be re-designed for each new FIFO size. We replace these high-loaded tri-state buses with two components called *One-to-Two Sequential* switch (OTS) and *Two-to-One Sequential* switch (TOS). These components are bonded in a vertical binary tree of switches as shown in Figure 3. A

Figure 3 shows the horizontal array of FIFO Cells (FC) with the distributed right-to-left token passing control path [17]. Data items move vertically across a path of OTS, FC and TOS components. Each OTS component is a 1-to-2 demultiplexer with automatic switch token. Each data item is alternatively driven to one of the two output path, starting on right path. The TOS components are the reciprocal 2-to-1 multiplexers, receiving the first data item on the right input path and then automatically switching from one input to the other.

This architecture ensures the correctness deadlock-free operations of the FIFO. OTS and TOS components work as supplementary data memorization cells. Moreover, the cells' structure for data paths is identical for FC, OTS and TOS components, i.e. a data latch plus a Muller gate (to control channel's request signal). Consequently, the input and output loads of each cell are balanced. Compared to the common bus solution, the mux/demux binary trees solution provide the following features: design of the FIFO is simplified, scalability and power consumption are improved and latency is degraded.

3. Conclusion. A circular asynchronous (either QDI or μP style) FIFO with a high degree of parallelism is delivered to robustly interface SB and ANoC modules in an ANoC-centric GALS system. Two versions are available: a mux/demux binary-tree version for fast system architecture exploration (especially for optimal FIFO size) and low-power; a common-bus version for low-latency requirement.

4. Design flow

4.1 Design methodology

We specify and model asynchronous circuits in CHP (for Communicating Hardware Processes), a high-level description language based on communication processes [10] [11]. The processes are synthesized using TAST, a suite tool [12] dedicated to asynchronous circuit synthesis. The TAST tool enables to map the CHP specification onto a standard-cell library and/or a specific cell library [13] when targeting ASICs, or to map onto FPGA for rapid system prototyping [9] [8].

4.2 Automatic crossbar generation

We use an automatic crossbar topology generation tool to implement the 5x5 crossbar ANoC. The tool controls adjustable design parameters for some of the

five ANoC modular blocks/layers. It supports fully-interconnect or Octagon [20] topology generation and modular routing node cores generation, which can be hand-adapted and assembled in more complex regular interconnect topologies, such as meshes. The choice of crossbar or fully-interconnect topologies ensures a fast, flexible and low-complexity system architecture exploration. It allows implementing efficient Emitter and Receiver components in term of routing complexity, latency and throughput and in term of control cost. The Receiver component supports high packet service extensions thank to its strong modularity.

So far, the adjustable parameters are:

1. *Crossbar size*. It depends of the number of the system's components.
2. *Point-to-point (p2p) interconnects width*. This parameter defines the width of each interconnect path according to the required bandwidth of each p2p linked SB or AB.
3. *Priority algorithm*. The priority solving function can be programmed. Available policies are round-robin, FIFO and non-interruptible two-level priority policies. The FIFO policy can be programmed independently for each routing node.
4. *Transaction services*. DPC module can be programmed to support data transaction services. For the time being, only the burst mode is available. All routing nodes must support the same transaction services.

4.3 Synthesis of QDI circuits onto FPGAs

This section presents an ANoC-centric GALS architecture implemented onto a multiclock Stratix Altera FPGA. We give in [8] a generic synthesis methodology to properly place and route asynchronous elements or mixed synchronous/asynchronous circuits onto a FPGA, respecting the specific timing assumptions of either QDI or micropipeline (μP) asynchronous design techniques. This methodology is extended in [9] to synthesize arbiter circuits with non-deterministic behavior, due to their synchronizer elements. A special circuit mapping is presented for delay-insensitive synchronizers devoted to asynchronous arbiters.

This FPGA-prototyping methodology is applied to the clock-less modules of the following architecture (ANoC and DES). The ANoC is designed according to the modular building method of sections 2 and 3.

5. Validation platform

5.1 PACMAN platform

We demonstrate our network-centric GALS building methodology with a case-study implemented on a Stratix Altera FPGA. This system is a first prototype version of a generic GALS platform called PACMAN, for Programmable And Configurable Multiprocessor Asynchronous Network.

The PACMAN first-version architecture is shown in Figure 4. It includes an ANoC interconnecting four processing elements.

The *asynchronous NoC* is a 5x5 crossbar with a direct output parallel communication link. The ANoC delivers both arbitration and transaction services (section 2.4). The arbitration policy is a non-interruptible two-level priority policy. When concurrent incoming requests need arbitration, a request with the high-priority level is selected and low-priority level requests are suspended. For equal priority-level concurrent requests, a First-In First-Granted (FIFG) policy is used. A former selected channel can not be interrupted by an incoming higher priority-level request during a burst mode data transfer. The high-priority level is assigned to the MIPS processors. The transaction service delivers burst mode or simple on-flit packet transfer modes, plus a special service called Indirect-Response (IR). In IR mode, a peripheral A, initiator of a communication, notify the receiver B not to answer to A, but to a third peripheral C.

The four processing elements are:

- *Two independently clocked MIPS* with local RAM banks and serial communication links. One MIPS is running at 10MHz for interfacing purposes whereas the other MIPS is running at 50MHz for number crunching applications.
- *A self-timed DES module* (Data Encryption Standard).
- *A shared RAM bank*.

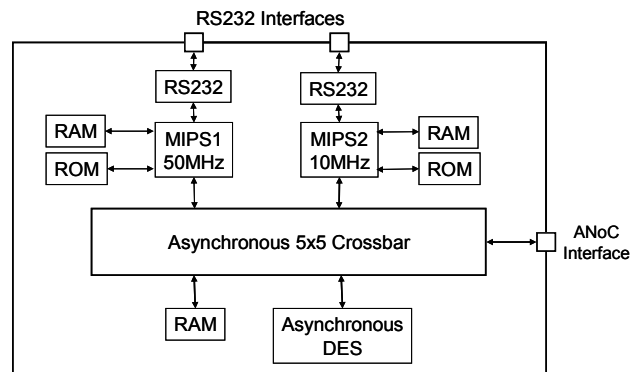


Figure 4. Structure of PACMAN case-study version for FPGA implementation

5.2 Preliminary results

The Stratix Altera FPGA platform we have been using successfully supports the PACMAN architecture implementation. Post place-and-route simulations show that the ANoC has a flit transport latency of 5ns from packet router to packet router and a cycle time of 18ns. There is no pipelining in this version of the ANoC, so the corresponding throughput is 55.6Mflit/s. This data transmission rate can easily be improved with pipelining.

For communication between MIPS1 (50MHz) and MIPS2 (10MHz), two WA modules are involved. The latency is 78ns and the cycle time is 148ns due to the clocked synchronizers penalty. Therefore, the throughput of the MIPS to MIPS communication is 6,8Mflit/s.

6. Conclusion

In this paper we provide a simple and flexible structure of Asynchronous NoC for GALS systems requiring efficient communications at low-complexity, high-reliability and low-power costs. Such a structure is modular, robust and scalable. The interconnect topology generator delivers several configurable interconnect topologies which facilitate the system architecture exploration, helped by a scalable and easy-to-plug (flexible?) self-timed FIFO. Then a low-latency FIFO version can be instanced in the final architecture. Using a multi-clock FPGA allows a fast prototyping of a complex ANoC-centric GALS system with mixed synchronous and asynchronous components. First result analysis gives promising ANoC abilities to deliver fast and robust communications. Closely analyses of the FPGA platform are currently performed to extract complete results, in order to improve both ANoC and entire GALS system design. Especially for the Wrapper Adaptors which are the bottleneck of the communication performances.

Prospective works will be to extend the topology generator to the other regular distributed topologies, with a large variety of arbitration policies and transaction services. Another work will be to integrate formal verification methods into the design flow. The aim is to deliver a dedicated synthesis tool for asynchronous interconnect networks generation.

References

- [1] F. K. Gürkaynak, S. Oetiker, N. Felber, H. Kaeslin et W. Fichtner, "Is there hope for GALS in the future ?" Proceedings of the 4th Asynchronous Circuit Design Workshop (ACiD 2004), Turku, Finland, June 28-29, 2004.
- [2] J. B. Rigaud, J. Quartana, L. Fesquet et M. Renaudin, "Modeling and design of asynchronous priority arbiters for on-chip communication systems", Proceedings of the VLSI-SOC'01 Conference on Very Large Scale.
- [3] E. Beigné, F. Clermidy, P. Vivet, A. Clouard, M. Renaudin, "An Asynchronous NOC Architecture Providing Low Latency Service and its Multi-Level Design Flow", 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), March 14-16, New York, USA, 2005.
- [4] T. Villiger, H. Kaeslin, F. Gurkaynak, S. Oetiker et W. Fichtner, "Self-timed Ring for Globally-Asynchronous Locally-Synchronous Systems", Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC'03, Vancouver, Canada, May 12-16, 2003.
- [5] E. Bolotin, E. Cidon, R. Ginosar et A. Kolodny, "QNoC : QoS architecture and design process for network on chip", Journal of Systems Architecture, no. June 2003.
- [6] W. J. Bainbridge et S. Furber, "CHAIN: A Delay Insensitive CHip Area INterconnect", IEEE Micro, vol. 22, no. 5, pp. 16-23, September/October 2002.
- [7] W. O. Lovett, "CHip Area Network Simulation", Master of Science, University of Manchester, 2002.
- [8] T. Q. Ho, J. B. Rigaud, M. Renaudin, L. Fesquet et R. Rolland, "Implementing Asynchronous Circuits on LUT Based FPGAs", Proceedings of the Field-Programmable Logic and Applications, Reconfigurable Computing Is Going Mainstream, 12th International Conference, FPL 2002, Montpellier, France, September 2-4, 2002.
- [9] J. Quartana, S. Renane, A. Baixas, L. Fesquet, M. Renaudin, "GALS Systems Prototyping using Multiclock FPGAs and Asynchronous Network-on-Chips", 15th Field-Programmable Logic and Applications Conference (FPL'05), August 24-26, Tampere, Finland. Submitted.
- [10] A.J. Martin, "Programming in VLSI: from communicating processes to delay-insensitive circuits", in C.A.R. Hoare, editor, Developments in Concurrency and Communication, UT Year of Programming Series, 1990, Addison-Wesley, p. 1-64.
- [11] Anh Vu Dinh Duc, Laurent Fesquet, Marc Renaudin, "Synthesis of QDI Asynchronous Circuits from DTL-style Petri-Net" IWLS-02, 11th IEEE/ACM International Workshop on Logic & Synthesis, New Orleans, Louisiana, 2002.
- [12] A.V. Dinh Duc, J.B. Rigaud, A. Rezzag, A. Sirianni, J. Fragoso, L. Fesquet, M. Renaudin, "TAST CAD Tools: Tutorial", tutorial given at the International Symposium on Advanced Research in Asynchronous Circuits and Systems ASYNC'02, Manchester, UK, April 8-11, 2002, and at the ACiD Summer School on "Asynchronous circuits design", Grenoble, France, July 15-19, 2002. TIMA internal report ISRN:TIMA-RR-02/07/01—FR, <http://tima.imag.fr/cis>.
- [13] Ph. Maurine, J.B. Rigaud, F. Bouesse, G. Sicard, M. Renaudin, "Static Implementation of QDI asynchronous primitives", PATMOS'03 - 13th International Workshop on Power and Timing Modeling, Optimization and Simulation. Torino, Italy, September 10-12, 2003.
- [14] R. Ginosar, "Synchronization and Arbitration", Proceedings of the ACiD Summer School on Asynchronous Circuit Design, Grenoble, France, July 15-19 2002.
- [15] Y. Semiat et R. Ginosar, "Timing Measurements of Synchronization Circuits", Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC'03, Vancouver, Canada, May 12-16, 2003.
- [16] R. Ginosar, "Fourteen ways to fool your Synchronizer", Proceedings of the Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC'03, Vancouver, Canada, May 12-16, 2003.
- [17] T. Chelcea et S. M. Nowick, "Robust Interfaces for Mixed-Timing Systems", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 8, 2004.
- [18] R. Ho, J. Gainsley et R. Drost, "Long wires and asynchronous control", Proceedings of the Asynch'04, 2004.
- [19] A. Bystrov, D. J. Kinniment, A. Yakovlev, "Priority Arbiters", in "International Symposium on Advanced Research in Asynchronous Circuits and Systems" (ASYNC), Eilat, Israel, April 2000, pp. 128-137.
- [20] F. Karim, A. Nguyen et S. Dey, "An Interconnect Architecture for Networking Systems on Chips", IEEE Micro, vol. 22, no. 5, pp. 36-45, September/October 2002. Integration, Montpellier, France, 3-5 Dec. 2001.
- [21] C. Piguet, M. Renaudin, T. Omnés "Low-power systems on chips (SOCs)", Proceedings of the DATE Conference, Munich, Germany, 2001.

Technology Mapping for Area Optimized Quasi Delay Insensitive Circuits

Bertrand Folco, Vivian Brégier, Laurent Fesquet, Marc Renaudin

TIMA Laboratory, 46 av. Félix Viallet

38031 Grenoble – FRANCE

{Bertrand.Folco, Vivian.Bregier, Laurent.Fesquet, Marc.Renaudin}@imag.fr

Abstract

Quasi delay insensitive circuits are functionally independent of delays in gates and wires (except for some particular wires). Such asynchronous circuits offer high robustness but do not perform well to automatically synthesize and optimize. This paper presents a new methodology to model and synthesize data path QDI circuits. The model used to represent circuits is based on Multi-valued Decision Diagrams and allows obtaining QDI circuits with two-input gates. Optimization is achieved by applying a technology mapping algorithm with a library of asynchronous standard cells called TAL. This work is a part of the back-end of our synthesis flow from high level language. Throughout the paper, a digit-slice radix 4 ALU is used as an example to illustrate the methodology and show the results.

1. Introduction

Asynchronous circuits do not have a global signal to synchronize them. Synchronization between blocks is locally done. Those circuits show very interesting properties such as low power consumption, noise emission, security, robustness, reusability, etc [1].

Today, to adopt the asynchronous technology the industry needs powerful asynchronous tools similar to synchronous ones.

This work is part of the TAST [2, 3] (Tima Asynchronous Synthesis Tool) project, aimed at developing and prototyping such tools. The synthesized circuits in TAST are quasi-delay insensitive (or QDI [4]). QDI circuits are functionally correct independently of delays in gates and wires, apart from the assumption that some forks are isochronic. This kind of asynchronous circuit is particularly robust. But robustness has a cost; these circuits usually have more transistors than the others, especially when standard cells are targeted. Many efforts are directed towards circuit optimization and transistor reduction; one of the main difficulties is to preserve the property of quasi-delay insensitivity [5-9].

2. Contributions

This paper presents a complete standard cells based design flow we have developed as illustrated in Figure 1. Our method uses Multi-valued Decision Diagrams as a model of the circuit that can be optimized while preserving the QDI property. Firstly, the model is generated from a CHP description. Secondly, the model is optimized. A two-input gates circuit is synthesized from the model. Thirdly, a technology mapping algorithm produces the final circuit, using gates from a library of standard asynchronous cells called TAL (TIMA Asynchronous Library).

This design flow includes a general technology mapping algorithm dedicated to QDI circuits. It enables to target any standard cells library, including or not asynchronous cells. The main objective of this work is to reduce the area of the asynchronous circuits. In fact, this is one of the main challenge for the asynchronous circuits to be adopted. Accordingly, the last part of the paper compares results obtained for our asynchronous circuits to its synchronous equivalent.

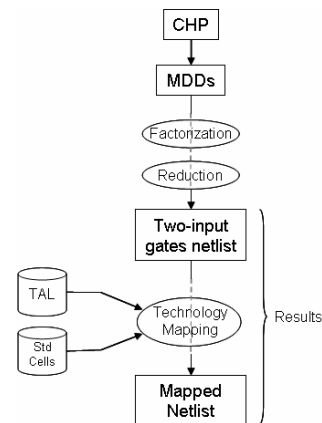


Figure 1 : Asynchronous Design Flow

3. Asynchronous Circuits

3.1. Communication channels and handshake protocol

In asynchronous circuits, a local mechanism is used to perform the synchronization called handshake

protocol. It relies on two signals: request and acknowledgment. When a block needs to transmit data to another, it sends a request signal along with the data, and holds them until it receives the acknowledgment. The request and acknowledgment signals may not be reset before the next communication, making two possible handshake protocols, well-known as two-phase and four-phase protocols. Asynchronous circuits considered in TAST implement the latter. Request, acknowledgment and data are linked together; therefore we consider them as a single entity called communication channel.

3.2. Quasi Delay Insensitivity

A circuit is said QDI (Quasi Delay Insensitive) when its correct operation does not depend on the delays of gates or wires, except for certain wires that form isochronic forks [10]. If a circuit is QDI, a transition on its input must cause a transition on its output. It is said that the transition on the output acknowledges the transition on the input. Mutual exclusion plays a very important role to prove this causality relationship [11].

3.3. Delay Insensitive Code

In QDI circuits, a mechanism must guarantee that when a channel emits a request, its data are available. To achieve this, the request is encoded with the data using a 1-of-n code: n rails are used to implement n possible values, numbered 0 to n-1. When all the rails are '0', there is no data and the request is '0'. The channel is said invalid. When one of the rails is '1', its number is the value of the data, and the request is '1'. The channel is said valid. Other codes, when several rails are '1', are out of the code, and therefore forbidden. The code is said Delay Insensitive since it guarantees that the request signal is always synchronized with the data.

3.4. The Muller gate

Asynchronous circuits need a gate that synchronizes several signals. This gate is called Muller gate (or C-element): when all inputs are equal, the output takes their value; when inputs are different, the output holds its value. Its symbol is a circle.

3.5. An example

Throughout this article, we illustrate our method with the example presented in Figure 2. This example is a digit-slice radix 4 ALU: it computes the function Op between its operands A and B, using the carry Cin and Cout when needed (addition and subtraction). Radix 4 was chosen to demonstrate that the method is not limited to dual rail. The ALU can compute seven different operations (add, sub, and, or, xor, neg, not); therefore Op is encoded with a 1-of-7 code. The CHP code is given in Figure 3.

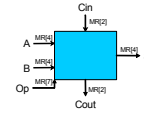


Figure 2: A digit-slice radix 4 ALU.

```

process alu_digit_slice
port (op: in di MR[7], a: in di MR[4],
      b: in di MR[4], cin: in di MR[2],
      s: out di MR[4], cout: out di MR[2]);
begin
variable op: MR[7], a: MR[4], b: MR[4], c: MR[2];
*[
Op?op;
@[
op = '0' => A?a, B?b;
@[ a+b<3 => Cout!0, [Cin?c, S!(a+b+c)]; --K
  a+b=3 => Cin?c, [Cout!c, S!(c=0?3:0)]; --P
  a+b>3 => Cout!1, [Cin?c, S!(a+b+c-4)]; --G
op = '1' => A?a, B?b; --sub
@[ b-a<3 => Cout!0, [Cin?c, S!b-a+c]; --K
  b-a=3 => Cin?c, [Cout!c, S!(c=0?3:0)]; --P
  b-a>3 => Cout!1, [Cin?c, S!(b-a+c-4)]; --G
op = '2' => A?a, B?b; S!a and b; --and
op = '3' => A?a, B?b; S!a or b; --or
op = '4' => A?a, B?b; S!a xor b; --xor
op = '5' => A?a; S!(not a+1); --neg
op = '6' => A?a; S!(not a); --not
]]
end

```

Figure 3: CHP code of the example

4. Circuit modeling using MDDs

The first step of our method is to model the circuit with Multi-valued Decision Diagrams (MDDs). It is presented in this section.

A MDD [12] is a generalized BDD (Binary Decision Diagram, [13]) structure. This structure is very interesting for QDI circuits synthesis because it exhibits the notion of mutual exclusion, which plays a valuable role in quasi delay insensitivity.

4.1. Presentation of the Multi-valued Decision Diagrams

A MDD is a rooted directed acyclic graph. Each non-terminal vertex is labeled by a multi-valued variable and has one out-going arc for each possible value of the variable. Each terminal vertex is labeled by a value. Figure 4 presents an example of MDD.

Each path of the MDD from its root to a terminal vertex maps to an input vector (a state of the input variables). The value of the terminal vertex specifies the value that the MDD has to take under this input vector.

The above definition of MDDs does not specify what the label of a vertex can be. Obviously, it can be input ports of the circuit: the logical function that specifies the outputs depends on the inputs.

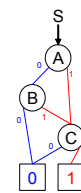


Figure 4: A simple example of MDD

We also want to be able to use internal variables in the circuit. To achieve this goal, we consider an internal variable as a MDD. Therefore, the label of a vertex can also be another MDD, which specifies an internal variable.

4.2. Direct and acknowledgment MDDs

A communication channel holds not only data, but also request and acknowledgment signals. The request signal is computed with the data, thanks to the 1-of-n DI code.

However the acknowledgment signal of the input channels needs to be computed separately. Moreover,

not all input channels are read at each computation level; the circuit must not acknowledge an input channel that has not been read.

For each output channel, our model contains a MDD that specifies the logic function computed and is called a direct MDD. For each input channel, it contains one MDD, called an acknowledgment MDD. Acknowledgment signals are considered as 1-of-n DI code with $n=1$: an acknowledgment MDD has only one terminal, and specifies the conditions under which the channel must be acknowledged. Figure 5 illustrates the MDDs of the example 3.4.

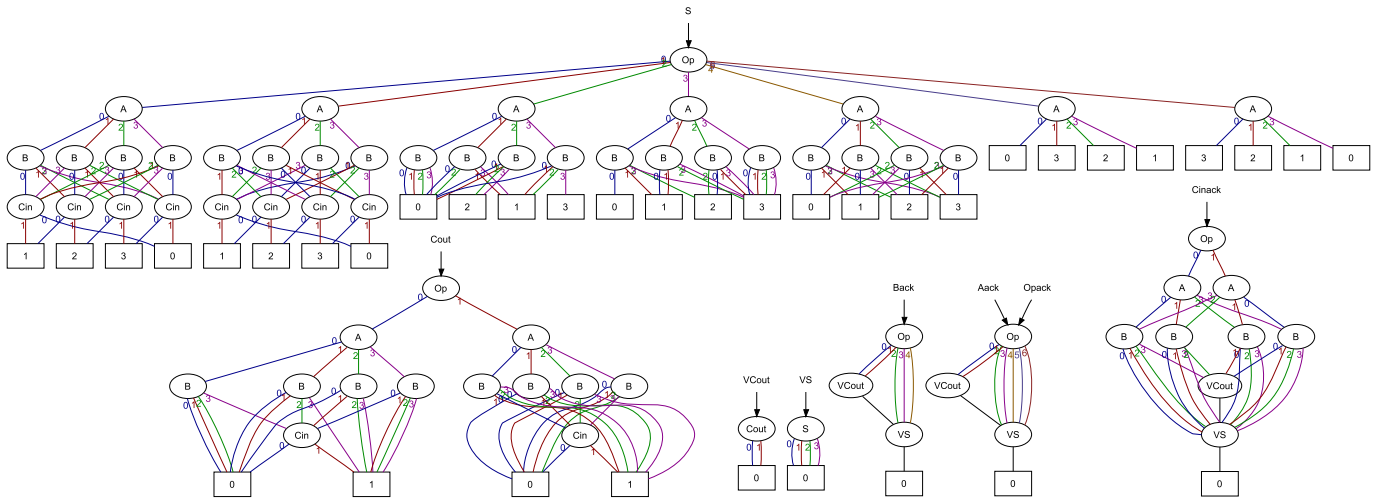


Figure 5: MDDs modeling the circuit specified in 2.4.

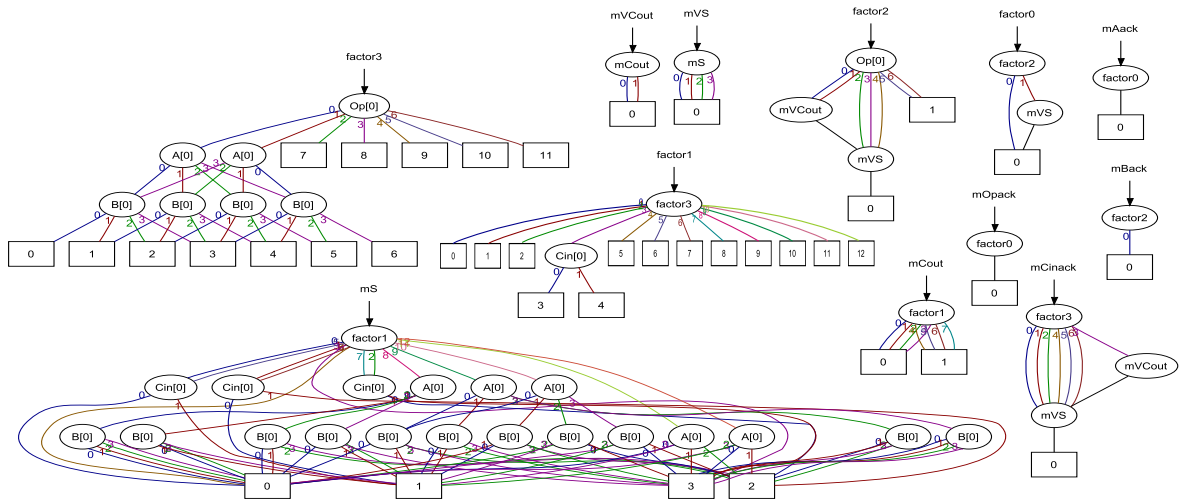


Figure 6: Result of the factorization over Figure 5

5. Basic gates synthesis from the MDDs

There are several steps to synthesize a circuit using basic two-input gates. First, a factorization is done between the different MDDs to share the common parts. Then, a reduction is applied to decrease the number of vertices in each MDD. Finally, each node of each MDD is synthesized using two-input gates.

5.1. Factorization

The factorization algorithm extracts the common part of a set of MDDs as an internal MDD, as illustrated in Figure 7.

To preserve the QDI property, the factorization algorithm must ensure that it extracts at least one node in each path of the MDD: otherwise, the extracted

MDD could become valid but be ignored in the calculation of the circuit's outputs, remaining unacknowledged and therefore violating the QDI property. To ensure this, the algorithm only extracts common parts that include the root vertex. Since we try all possible ordering of the variables, this restriction does not limit the efficiency of the algorithm. Figure 6 shows the result of this algorithm when applied to the MDDs of Figure 5.

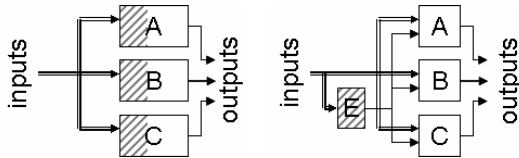


Figure 7: Before and after the factorization of a set of MDDs. E is the common part extracted from A, B and C.

5.2. Reduction

This step is similar to the reduction of BDDs: it merges the identical vertices of the MDD, which decreases their number and thus the size of the circuit. Note that this is different from factorization: the reduction acts on the structure of one MDD, whereas the factorization acts on the logical functions represented by a set of MDDs, independently of their structure.

5.3. Synthesis using basic two-input gates

To synthesize the circuit modeled by composed MDDs, each MDD is synthesized as a block of the circuit.

The algorithm is specified by the following rules:

- Each arc in a MDD corresponds to a rail in the circuit.
- Multiple arcs directed to the same vertex are grouped by an OR gate.
- A non-terminal vertex is implemented as set of two-input Muller gates that synchronize each rail of its variable with the in-going arc. The Muller gates outputs are the out-going arcs of the vertex.
- A terminal vertex with value i represents rail number i of the MDD.

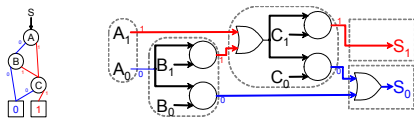


Figure 8: Example of basic two-input gates synthesis of a MDD.

Figure 9 presents the synthesized circuit from the MDDs of Figure 5.

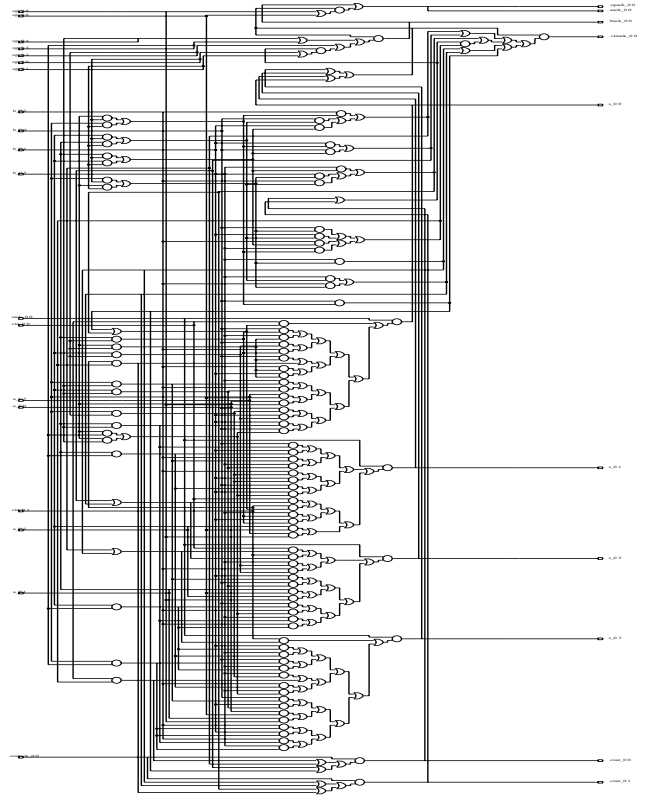


Figure 9: Basic two-input gates circuit synthesized from the MDDs of Figure 5.

6. Technology mapping

We first present a library of asynchronous standard cells we have developed and called TAL. Then, we give different results obtained by using this library in the design of the digit-slice radix 4 ALU, instead of the ST standard library. Finally we compare our asynchronous circuit to a synchronous equivalent circuit.

6.1. TAL library

The TAL library has been developed to design asynchronous circuits with the aim to reduce their area, consumption and increase their speed [14]. This library contains about 160 cells (representing 42 functionalities), and has been designed with the 130nm technology of STMicroelectronics. The main functionalities of the library are useful asynchronous functions as Muller gate, Half-Buffers, Mutex and complex gates as Muller-Or, Muller-And, ...

To clarify what gains should be attributed to a dedicated asynchronous library, we can view in Table 1 the comparison, between basic cells of the TAL library and their standard cells equivalent, in terms of number of transistors and area. For example, the Muller gate presented in 3.4 is build with 9 transistors in the TAL library (for a Muller gate with 2 inputs). With standard cells we have to use an optimized AO222 gate with a loop as described in Figure 10, made of 14 transistors, to find the functionality of a Muller gate.

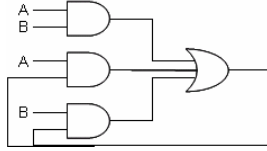


Figure 10 : Muller Gate in standard cells

Function	TAL Lib Nb of transistors/ Area (μm^2)	Std cells Nb of transistors/ Area (μm^2)	Gain (area)
Muller 2	9 tr. / 14,12	14 tr. / 20,17	30 %
Muller 4	13 tr. / 18,15	42 tr. / 60,51	70 %
Half-Buffer	28 tr. / 40,34	44 tr. / 62,53	35 %

Table 1 : Differences between TAL and Std cells implementations of basic functions.

The average gain in term of area for all the TAL library compared to the standard ST library is around 35%.

6.2. Technology Mapping algorithms

The main difficulty before mapping a library on asynchronous circuits is to decompose them and ensure to keep their property of quasi delay insensitivity.

For example, it's difficult to decompose a Muller gate with 3 inputs in 2 Muller gates with 2 inputs without introducing a hazard. This decomposition is automatic for an OR gate. This is described in Figure 11.

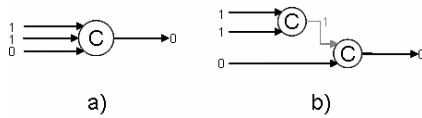


Figure 11 : Naïve Muller decomposition introduces hazard

In case a), the three inputs of the Muller gate are different and the output keeps its value 0. After the decomposition (b), the first Muller gate output switches while the output of the second one doesn't change. Thus the output of the first Muller gate is not acknowledged causing a possible glitch in the circuit with the next set of inputs.

The synthesis method presented in 0 ensures that the circuits obtained are QDI and formed of two-input gates. Thus the decomposition phase is done and the technology mapping consists in merging gates to obtain an optimized circuit following a selected criteria (area, speed, ...). Merging gates do preserve delay insensitivity.

We decide to implement known synchronous algorithms of technology mapping [15-17] and adapt them to asynchronous circuits. Some algorithms of technology mapping exist for asynchronous circuits [18-20], but the aim of these algorithms is mainly to decompose circuits without hazards, and as we have seen before, the decomposition is solved.

Moreover, technology mapping has been an important domain of research in the synchronous world and the resulting algorithms are very powerful. Thus we extend the method presented in [16] because the technology mapping algorithm presented in this paper has really great performances. Thereby we represent the input library cells as tree of OR, AND and MULLER gate and we keep the structural relationship between the library cells using lookup table. These trees are then mapped on the netlist representing the circuit with the same algorithms as for synchronous circuits.

6.3. Results

In the following section, we intend to evaluate in terms of area the gain due to the TAL library and the gain due to the technology mapping algorithms.

The circuit netlist of Figure 9 comprises 95 OR gates and 107 MULLER gates. The Table 2 compares the number of transistors and the area of the circuit, before place and route, using the TAL library or the ST standard library.

Table 2 : Circuits with TAL or ST standard cells

	TAL library	Standard ST cells
Nb of transistors	1533	2068
Area (μm^2) (before placement and routage)	2469	3116,36

We can conclude out of this figure that without any optimization of the netlist, if we only use TAL cells instead of the standard cells to build Muller gates, the number of transistors decreases by 35% and the area of the circuit decreases by 21%.

Now we want to evaluate the gain brought by the technology mapping algorithms on the netlist of the digit-slice radix 4 ALU. We can view results of algorithms in the Table 3. During the mapping phase, only complex gates of the TAL library are used as Muller-Or22, Muller-Or21. OR2 gates are also merged in OR3 and OR4 gates.

Table 3 : Results of technology mapping algorithms

	Native TAL netlist	Optimized TAL netlist
Nb of transistors	1533	1034
Area (μm^2) (before placement and routage)	2469	1401,95

We can notice a decrease of 32% of the number of transistors, and a decrease of 43% of the area of the circuit compared to the same circuit netlist using the TAL library without technology mapping algorithm applied. We thus note a decrease of around 50% of the

number of transistors and area compared to the initial netlist using the ST standard cells library.

Another interesting point is to compare these circuit characteristics with an equivalent synchronous digit-slice radix 4 ALU. The asynchronous circuits remain bigger than their synchronous equivalent because of the delay insensitive code and the local controls of the circuit. However our goal is to reduce this difference as much as possible by applying aggressive technology mapping algorithms on the circuit and by using cells library specially designed for asynchronous circuit.

We describe the digit-slice radix 4 ALU using the VHDL language. As we want to compare our version to a synchronous circuit, we add a clock in the description. In fact, the outputs are memorized in the asynchronous circuit with the Muller gate. In the synchronous version, we have to add registers on each output, to achieve this memorization.

To synthesize this circuit, we used Design Analyser from Synopsys and the ST standard cells library. Table 4 shows the results.

Table 4 : Comparison with the equivalent synchronous circuit

	Optimized TAL netlist	Synchronous netlist
Nb of transistors	1034	386
Area (μm^2) (before placement and routage)	1401,95	476, 06

We can conclude that the synchronous circuit is less than 2,9 times smaller, and contains 2.7 times less transistors than the asynchronous one.

7. Conclusion

This paper presents a general method to model and synthesize asynchronous optimized QDI circuits. The method allows synthesizing circuits using multi-rail logic and maps them on to single output standard cells. Direct and reverse (acknowledge) paths are automatically and jointly synthesized. A first netlist of the circuit, containing only two-input gates is generated. Technology mapping is then applied targeting a dedicated asynchronous library to optimize the circuit area. Others criteria of optimization could be selected as well but the paper focuses on area which is one of the most important challenge.

The method based on Multi-valued Decision Diagrams, is illustrated on a digit-slice radix 4 ALU. We present different versions of the same circuit to evaluate the gain introduced by the asynchronous library and by the technology mapping algorithm. The last results show that our circuit is still 2.9 times larger than the synchronous one.

Future work will be focused on improving the methodology by working in two directions: logic synthesis and complex cells specification.

8. References

1. Renaudin, M., *Asynchronous circuits and systems: a promising design alternative*. Microelectronic Engineering, 2000. **54**(1-2): p. 133 - 149.
2. Dinh Duc, A.V., L. Fesquet, and M. Renaudin. *Synthesis of QDI Asynchronous Circuits from DTL-style Petri Net*. in *11th IEEE/ACM International Workshop on Logic & Synthesis*. 2002. New Orleans, Louisiana.
3. Dinh Duc, A.V., et al. *TAST CAD Tools*. in *ACiD-WG workshop*. 2002. Munich, Germany.
4. Martin, A.J., *The Limitations to Delay-Insensitivity in Asynchronous Circuits*, in *Advanced Research in VLSI*, W.J. Dally, Editor. 1990, MIT Press. p. 263-278.
5. Manohar, R., T.K. Lee, and A.J. Martin. *Projection: A Synthesis Technique for Concurrent Systems*. in *The 5th IEEE International Symposium on Asynchronous Circuits and Systems*. 1999.
6. Toms, W.B. *QDI Implementation of Boolean Graphs*. in *14th UK Asynchronous Forum*. 2003.
7. Burns, S.M., *General Condition for the Decomposition of State Holding Elements*, in *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*. 1996, IEEE Computer Society Press.
8. Lemberski, I. and M.B. Josephs. *Optimal Two-Level Delay-Insensitive Implementation of Logic Functions*. in *PATMOS*. 2002. Spain.
9. Nielsen, C.D. *Evaluation of Function Blocks for Asynchronous Design*. in *eurodac*. 1994: icsp.
10. Martin, A.J., *The Limitations to Delay-Insensitivity in Asynchronous Circuits*, in *Advanced Research in VLSI*, W.J. Dally, Editor. 1990, MIT Press. p. 263--278.
11. Bregier, V., et al. *Modeling and Synthesis of multi-rail multi-protocol QDI circuits*. in *International Workshop on Logic Synthesis*. 2004.
12. Kam, T., et al., *Multi-valued decision diagrams: Theory and applications*. International Journal on Multiple-Valued Logic, 1998. **4**(1-2): p. 9-24.
13. Dreschler, R. and B. Becker, *Binary Decision Diagrams, Theory and Implementation*. Kluwer Academic Publishers ed. 1998: Kluwer Academic Publishers.
14. Maurine, P., et al. *Static Implementation of QDI Asynchronous Primitives*. in *PATMOS: 13th International Workshop on Power and Timing Modeling, Optimization and Simulation*. 2003.
15. Keutzer, K. *DAGON: technology binding and local optimization by DAG matching*. in *Proceedings of the 24th ACM/IEEE conference on Design automation*. 1987. Miami Beach, Florida, United States.
16. Zhao, M. and S.S. Sapatnekar. *A new structural pattern matching algorithm for technology mapping*. in *The 38th Conference on Design Automation*. 2001. Las Vegas, Nevada, United States.
17. Matsunaga, Y. *On Accelerating Pattern Matching for Technology Mapping*. in *International Conference on Computer Aided Design*. 1998. San Jose, California, United States.
18. Cortadella, J., et al. *Decomposition and technology mapping of speed-independent circuits using Boolean relations*. in *Proc. International Conf. Computer-Aided Design (ICCAD)*. 1997.
19. Myers, C.J., P.A. Beerel, and T.H.-Y. Meng, *Technology Mapping of Timed Circuits*, in *Asynchronous Design Methodologies*. 1995, Elsevier Science Publishers. p. 138-147.
20. Siegel, P.S.K., *Automatic Technology Mapping for Asynchronous Designs*. 1995, Stanford University.

Available online at www.sciencedirect.com**Measurement**

Measurement xxx (2005) xxx–xxx

www.elsevier.com/locate/measurement

Asynchronous level crossing analog to digital converters

E. Allier *, G. Sicard, L. Fesquet, M. Renaudin

TIMA Laboratory/Concurrent Integrated Systems Group, 46 Avenue Félix Viallet, 38031 Grenoble Cedex, France

Abstract

We present a new class of asynchronous analog to digital converters (A-ADCs), based on an level-crossing sampling scheme of the analog signal, and an asynchronous design. Because these ADCs are not conventional, a design methodology is also presented, it takes place at a system level, then a transistor level. Its purpose is to determine the characteristics of an A-ADC given the required effective number of bits and the properties of the analog signal to convert, such as to minimize the complexity, the activity, and the power consumption. A prototype has been designed for speech applications, using the 0.18 μm CMOS technology from STMicroelectronics, and a voltage mode approach for the analog parts of the converter. Electrical simulations prove that the Figure of Merit (FoM) of this converter is increased by more than one order of magnitude compared to synchronous Nyquist ADCs. Moreover, behavioural simulations prove that the activity of the A-ADC is reduced by two orders of magnitude compared to its synchronous counterpart.

© 2005 Published by Elsevier Ltd.

Keywords: Analog to digital converters (CANs); Level-crossing sampling; Asynchronous design; Figure of merit (FoM); Analog voltage mode design

1. Introduction

It is well known that asynchronous designs exhibit interesting properties [1]. This kind of designs has been used in a few publications to improve the performances of Nyquist ADCs such as: immunity to metastable behaviour [2], reduction of the elec-

tromagnetic interferences [3], speed [4], power consumption savings [5], etc.

Moreover, most of the systems using ADCs bring signals with interesting statistical properties into operation, but Nyquist signal processing architectures do not take advantage of them. Actually, these signals (such as temperature sensors, pressure sensors, electro-cardiograms, speech signals, etc.) are almost always constant and may vary significantly only during brief moments. Thus, classical regular sampling and converting systems are highly constrained, due to the Shannon theory, which is to ensure for the sampling

* Corresponding author. Tel.: +334 76 57 45 00; fax: +334 76 47 38 14.

E-mail addresses: emmanuel.allier@imag.fr (E. Allier), gilles.sicard@imag.fr (G. Sicard), laurent.fesquet@imag.fr (L. Fesquet), marc.renaudin@imag.fr (M. Renaudin).

39 frequency to be at least twice the input signal fre-
 40 quency bandwidth. Therefore, in the time domain,
 41 this condition can be translated as a large number
 42 of samples without any relevant information. This
 43 effect implies a useless increase of activity of the
 44 circuit compared to the supplied output digital
 45 information relevance, and so a useless increase
 46 of the power dissipation. It has been proved in
 47 [6] and [7] that ADCs using a non equi-repartition
 48 of the samples in time lead to interesting power
 49 savings compared to Nyquist ADCs.

50 The new class of ADCs proposed in this paper
 51 consists in using both the “level-crossing” sam-
 52 pling scheme proposed in [6] and an asynchronous
 53 implementation of the circuit (no global clock).
 54 The theory and the design of such converters is
 55 completely different from classical Nyquist ADCs.
 56 Hence a complete design methodology is pre-
 57 sented, in order to minimize activity, power con-
 58 sumption, and hardware of the circuit, according
 59 to the statistical properties of the analog signal
 60 to convert, and the signal-to-noise ratio (SNR)
 61 of the targeted application.

62 After an introduction on the irregular sampling
 63 scheme and the SNR theory, the new contribu-
 64 tions of this work are presented in Section 2. The
 65 design methodology is presented in Section 3 so
 66 as to design such converters. Section 4 describes
 67 the design and electrical simulations of an asyn-
 68 chronous ADC targeted for speech signals. Figure
 69 of Merit, and comparisons of signal processing
 70 chains are discussed in Section 5. Lastly, Section
 71 6 concludes the paper.

72 2. Principles and architecture

73 2.1. Irregular sampling

74 For irregular sampling (cf. Fig. 1), and a M -bits
 75 resolution, $2^M - 1$ quantization levels are regu-
 76 larly disposed along the amplitude range of the sig-
 77 nal. A sample is taken only when the analog input
 78 signal V_{in} crosses one of them. Contrary to classi-
 79 cal Nyquist sampling, samples are not regularly
 80 spaced out in time, because it depends on the sig-
 81 nal variations. Thus, together with the value of
 82 the sample b_i , the time Dt_i elapsed since the previ-

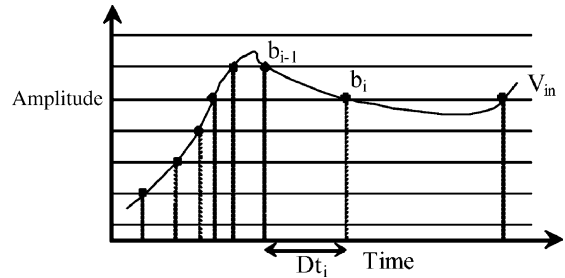


Fig. 1. Irregular sampling.

ous sample b_{i-1} must be recorded. A local timer of
 period T_C is dedicated to this task. The amplitude
 of the sample is precise, but the time elapsed since
 the previous sample is quantized according to the
 precision T_C of the timer [8].

After computation, the signal-to-noise ratio
 (SNR) is defined by:

$$\text{SNR}_{\text{dB}} = 10 \cdot \log \left(\frac{3 \cdot P(V_{in})}{P\left(\frac{dV_{in}}{dt}\right)} \right) + 20 \cdot \log \left(\frac{1}{T_C} \right). \quad (1)$$

The first term of Eq. (1) is only determined by
 the statistical properties of the input signal V_{in} .
 The SNR depends on the timer period T_C , and
 not on the number of quantization levels. Thus,
 for a given implementation of the irregular sam-
 pling A/D converter (a fixed number of quantiza-
 tion levels: $2^M - 1$), the SNR can be externally
 tuned by changing the period T_C of the timer.
 For example, if the SNR has to be improved by
 6.02 dB i.e. must have one more bit of effective res-
 olution (or ENOB—Equivalent Number Of Bit),
 T_C has to be divided by 2. Eq. (1) can be used
 to determine the theoretical SNR of a pure sine wave,
 a speech signal, an audio signal... In theory, for
 irregular sampling, the SNR can be improved as
 far as it is needed, by reducing T_C . The limit is
 the accuracy of the analog blocks: they determine
 the precision of the quantization levels position
 in Fig. 1. This point is discussed in Section 3.2
 for the design methodology.

2.2. Asynchronous ADC

The block diagram of the architecture bringing
 this sampling scheme into operation is shown in

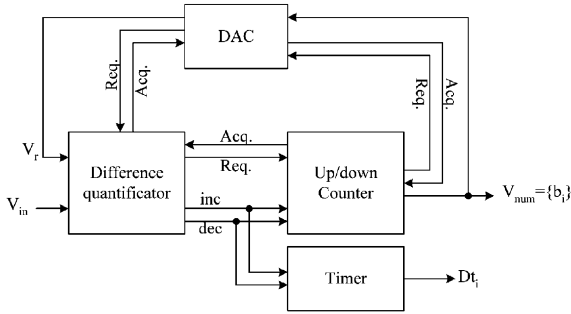


Fig. 2. Block diagram of the A-ADC.

116 Fig. 2. The converter resolution M and its dynamic
117 range ΔV_{in} set the quantification step

$$q = \frac{\Delta V_{in}}{2^M - 1}. \quad (2)$$

120 The output digital value V_{num} is converted to V_r
121 by the DAC, and compared to V_{in} . If the difference
122 is greater than $\frac{1}{2} \cdot q$, the state variable is incre-
123 mented ($inc = '1'$), if it is lower than $-\frac{1}{2} \cdot q$, it is
124 decremented ($dec = '1'$). In all other cases, nothing
125 is done ($inc = dec = '0'$): the converter output sig-
126 nal V_{num} remains constant, there is no activity.
127 The output signal is composed of couples (b_i, Dt_i)
128 where b_i , is the digital value of the sample (with
129 $V_{num} = \{b_i\}$, $i \in N$), and Dt_i the time elapsed since
130 the previous converted sample b_{i-1} , given by the
131 timer. The term A-ADC for “Asynchronous

ADC” is now used. Contrary to other works [2–
4], the term “asynchronous” defines not only the
sampling principle but also the design mode. Infor-
mation transfer is locally managed with a bi-direc-
tional control signalling. Each “data” signal is
associated with two “control” signals: a *request*
and an *acknowledgement* [1].

Let δ be the total delay of the loop, the slope of
 V_{in} must verify the “tracking condition”

$$\left| \frac{dV_{in}}{dt} \right| \leq \frac{q}{\delta}. \quad (3)$$

2.3. Implementation of the A-ADC

The A-ADC presented in Fig. 2 has been imple-
mented with a three stage micro-pipelined archi-
tecture [9], and a 4-phase protocol [1] (cf. Fig. 3).
It is composed of two parts: a quasi delay-insensi-
tive control part composed of delays, Muller gates
(or C-elements [10]) and standard gates, and a data
path part composed in our case of digital and ana-
log blocks. The delay element *delay2* (respectively,
delay3) is matching the counter critical path
(respectively, DAC critical path). The timer is
not represented on the circuit. The bits *Setb* and
Resetb on the circuit are controlled by the same
initialisation signal *RESET*. The analog latch is
here to link the control part and the data path
part, and to insure the correct running of the con-
verter [8].

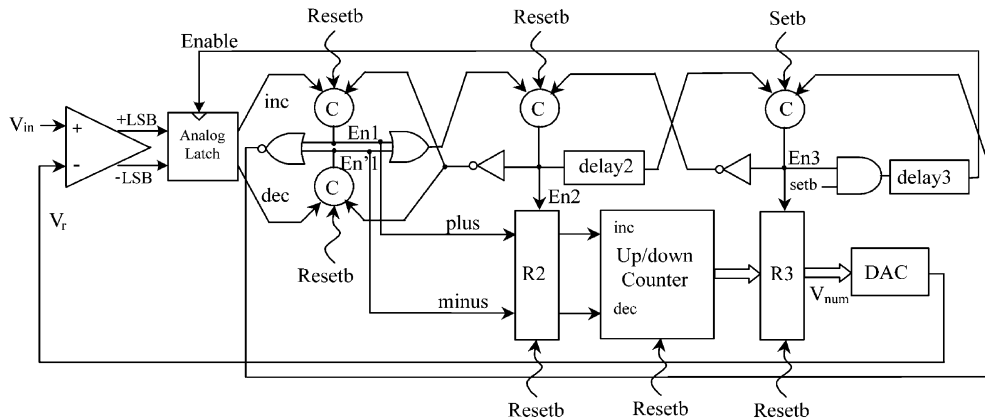


Fig. 3. Implementation of the A-ADC.

161 3. Design methodology

162 As it is explained in Section 2.1, because of the
163 irregular sampling scheme, the theory associated
164 with the A-ADC is completely different from clas-
165 sical Nyquist ADCs. Thus, the purpose of the pro-
166 posed methodology is to enable the designers to
167 calculate the design parameters of an A-ADC, gi-
168 ven an application.

169 First, this methodology takes place at a system
170 level. The input parameters are the analog signal
171 characteristics and the desired ENOB, from which
172 the four parameters characterizing an A-ADC
173 must be computed. There is the resolution M (that
174 defines the number of quantization levels), the
175 loop delay δ , the period T_C of the timer, and the
176 resolution of the timer M_{timer} , that means the
177 number of bits of the time intervals Dt_i . The input
178 signal must be perfectly known: its power spectral
179 density (PSD), its bandwidth f_{max} , its maximum
180 amplitude ΔV_{in} , and its probability density $p(x)$.
181 It is assured that the amplitude range ΔV_{in} of the
182 signal fits the entire input range of the converter,
183 so it is a known parameter.

184 Second, the methodology takes place at a tran-
185 sistor level, where all the characteristics of each
186 analog block of the A-ADC is analysed such as
187 to reach the *ENOB-bits* of effective resolution for
188 the A-ADC. In this methodology, only a voltage
189 mode approach is studied for the analog part of
190 the A-ADC. Actually, it has been shown that this
191 kind of design leads to very low-power circuits
192 [11,12], that is the goal of our new approach for
193 asynchronous analog to digital conversion.

194 3.1. System level methodology

195 3.1.1. Computing M

196 In the synchronous case, the Shannon theorem,
197 insures the reconstruction of the original signal
198 with a *sinc* basis. For irregular sampling, a recon-
199 struction condition is deduced from the number of
200 quantization levels (i.e. the resolution M), which
201 directly influences the number of samples. It de-
202 pends also on the signal variations. In an irregular
203 sampling scheme, Beutler [13] proved that recon-
204 struction of the original continuous signal is possi-
205 ble if the average sampling rate is greater than

twice the signal bandwidth f_{max} . For a given type 206
of signals: the density probability of its amplitude 207
 $p(x)$ and its maximum slope $|\frac{dV_{\text{in}}}{dt}|$ are known. By 208
statistical considerations, the average sampling 209
rate \bar{f}_{sample} can be computed theoretically. It de- 210
pends on M , which minimum value can be deter- 211
mined so as to respect the Beutler condition. The 212
same method like the one given in [6] for a band- 213
limited gaussian signal can be applied to determine 214
the average sampling rate \bar{f}_{sample} . To conclude, the 215
average number of samples per second will be min- 216
imum to insure a reconstruction, and so the activ- 217
ity and the power consumption will be minimized. 218
Moreover, a low value for M insures a low hard- 219
ware complexity and a reduction of the silicon 220
area. 221

3.1.2. Computing δ

222 The knowledge of the amplitude range and the 223
bandwidth of V_{in} enables us to find the boundaries 224
of the slope using the Bernstein theorem: 225

$$\left| \frac{dV_{\text{in}}}{dt} \right| \leq 2 \cdot \pi \cdot f_{\text{max}} \cdot \Delta V_{\text{in}}. \quad (4)$$

229 The number of quantization levels is now 229
known: $2^M - 1$. According to the tracking condi- 230
tion (3), and the Bernstein inequality (4), the loop 231
delay δ must verify: $\delta \leq \delta_{\text{max}}$, with 232

$$\delta_{\text{max}} = \frac{1}{2 \cdot \pi \cdot f_{\text{max}} \cdot (2^M - 1)}. \quad (5)$$

3.1.3. Computing T_C

237 For a given application, the effective resolution 237
ENOB is known. The classical formula of the SNR 238
for the Nyquist sampling scheme is perfectly 239
known: 240

$$\text{SNR}_{\text{dB}} = 10 \cdot \log \left(\frac{12 \cdot \sigma_{\text{in}}^2}{q^2} \right), \quad (6)$$

244 where σ_{in} is the standard deviation of the input sig- 244
nal V_{in} and q is the quantum. Thus the theoretical 245
SNR we want to reach is perfectly known. In the 246
asynchronous sampling scheme, the theoretical 247
SNR is given by Eq. (1), the first term can be com- 248
puted from the PSD of the signal V_{in} . Thus, with 249
Eqs. (1) and (6), the timer period T_C can be 250
computed. 251

252 3.1.4. Computing M_{timer}

253 When the input signal V_{in} is quiet, no sample is
 254 converted by the A-ADC according to the level-
 255 crossing sampling scheme. Nevertheless, the A-
 256 ADC must provide a sample when the timer
 257 reaches its overflow. The maximum time interval
 258 Dt_{max} that can be provided by the timer is

$$Dt_{max} = (2^{M_{timer}} - 1) \cdot T_C. \quad (7)$$

261 If a voltage mode implementation is considered
 262 for the analog part, the voltage reference V_r is gi-
 263 ven by the DAC and comes from charged capaci-
 264 tances (cf. Section 3.2). When V_{in} is quiet, the
 265 reference V_r should be constant. Because of the
 266 leakage currents, this voltage reference drifts and
 267 may corrupt the targeted SNR of the application.
 268 The leakage current by surface unit $I_{leak/C}$ is con-
 269 stant for a given CMOS process. Let V_C be the
 270 voltage at the output of a capacitance, the varia-
 271 tion in time of V_C is given by

$$\frac{dV_C}{dt} = \frac{I_{leak}}{C} = I_{leak/C}. \quad (8)$$

274 In the worst case, the input signal V_{in} is constant
 275 during Dt_{max} , the variation of V_r becomes:

$$\Delta V_{C-max} = I_{leak/C} \cdot (2^{M_{Timer}} - 1) \cdot T_C. \quad (9)$$

278 To insure an effective resolution of ENOB-bits for
 279 the A-ADC, this variation must not be greater
 280 than half a LSB of effective resolution (cf. Eq.
 281 (12)). Thus, according to Eqs. (10) and (12), the
 282 maximum value of M_{timer} is given by:

$$M_{timer} \approx \ln_2 \cdot \left(\frac{\Delta V_{in}}{2 \cdot I_{leak/C} \cdot T_C \cdot (2^{ENOB} - 1)} \right). \quad (10)$$

3.1.5. Summary

286 The system level design methodology is
 287 summed up in Fig. 4. The desired ENOB gives
 288 T_C , and the signal properties give M , δ_{max} and
 289 M_{timer} . In conclusion, exploiting the statistical
 290 properties of the signal leads to a low value for
 291 M , while keeping a high ENOB. The hardware
 292 complexity is thus much lower than in synchro-
 293 nous Nyquist ADCs. Hence, together with the sil-
 294 icon area reduction, such A-ADC has a reduced
 295 electrical activity.
 296

3.2. Transistor level

297 All the system level parameters being com-
 298 puted, the analog part of the A-ADC must be de-
 299 signed very carefully because its characteristics
 300 must not deteriorate the performances of the con-
 301

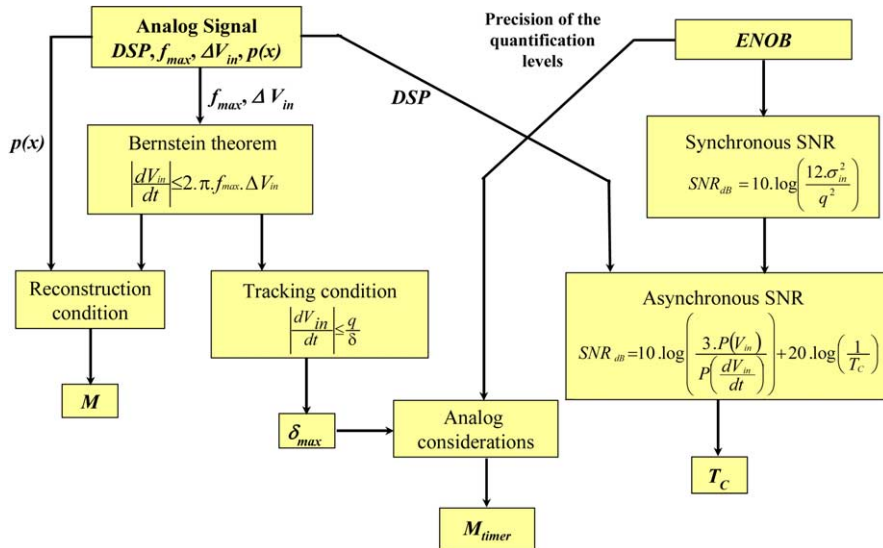


Fig. 4. System level design flow of an A-ADC given a targeted application.

302 verter. In this section, each parameter of each ana-
 303 log block (the DAC and the comparator) is clearly
 304 analysed such as for the A-ADC to reach the tar-
 305 geted resolution of ENOB-*bits*. In fact, the analog
 306 imperfections are the source of imprecisions in the
 307 position of the reference levels given in Fig. 1, that
 308 may reduce the value of the theoretical SNR given
 309 by Eq. (1).

310 Let q_{eff} be the effective quantum corresponding
 311 to the effective resolution of the converter. It is gi-
 312 ven by the relation:

$$q_{\text{eff}} = \frac{\Delta V_{\text{in}}}{2^{\text{ENOB}} - 1}. \quad (11)$$

315 Supposing that all the contributions are decor-
 316 related, let δV be the maximum error in amplitude
 317 for a reference level introduced by a given param-
 318 eter of an analog block. The ENOB-*bits* of the A-
 319 ADC are verified if the following relation is
 320 observed:

$$\delta V \leq \frac{q_{\text{eff}}}{2}. \quad (12)$$

324 The analog part of the A-ADC is given in Fig.
 325 5. There are two comparators, the analog latch
 326 and a DAC that must provide two reference volt-
 327 ages: $V_{\text{DAC}} + \frac{q}{2}$ and $V_{\text{DAC}} - \frac{q}{2}$. The analog latch is
 328 not studied in this section because it has only a
 329 digital running, and this kind of analog constraints
 330 does not exist for its design.

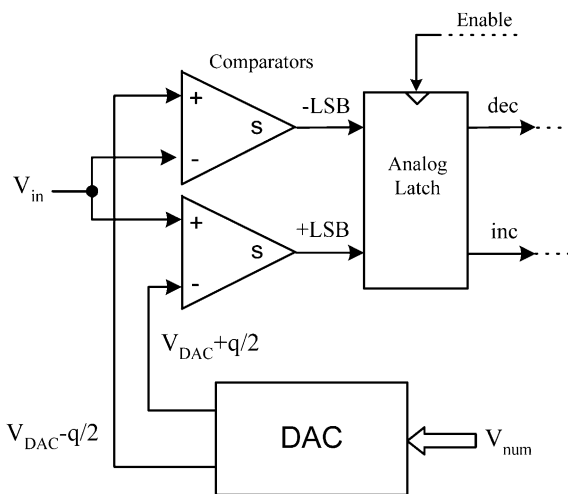


Fig. 5. Analog part of the A-ADC given in Fig. 3.

3.2.1. The DAC

331 Most of the time, in a voltage mode approach,
 332 digital to analog converters are designed using
 333 switched capacitor circuits. The problem is that
 334 many imperfections can appear, concerning the
 335 Operationnal Amplifier, the switches, and the
 336 capacitances. The reference circuit that has been
 337 used in our methodology is presented in [14] and
 338 given in Fig. 6.

339 Φ_1 and Φ_2 are non-overlapping clocks, V_{ref} is a
 340 reference voltage, $x_i, i \in \{0, \dots, 2^M - 1\}$ is the digi-
 341 tal input of the DAC using a thermometer code.
 342 The analog output of the DAC V_{DAC} is given by:
 343

$$V_{\text{DAC}} = \left(\frac{C \cdot \sum_{i=0}^{2^M-2} x_i}{C_{\text{ref}}} \right) \cdot V_{\text{ref}}. \quad (13)$$

346 The circuit given in Fig. 6 has been modified
 347 such as to provide the two reference voltages and
 348 so to be inserted in the analog part of the A-
 349 ADC given in Fig. 5. The schematic is given in
 350 Fig. 7, and the clocks in Fig. 8. These different
 351 clocks are generated using combinations of the *re-*
 352 *quest* and *acknowledgement* signals of the control
 353 part of the A-ADC (cf. Fig. 3). All the capaci-
 354 tances are made by connecting unit capacitors
 355 C_u . If m is the number of capacitances that are

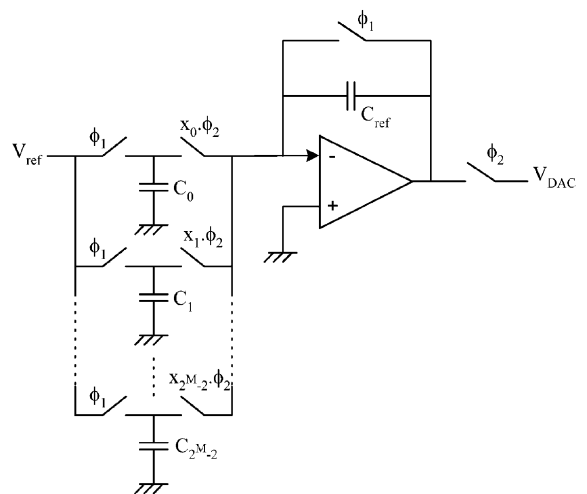


Fig. 6. Reference switched capacitors DAC used for the transistor level methodology.

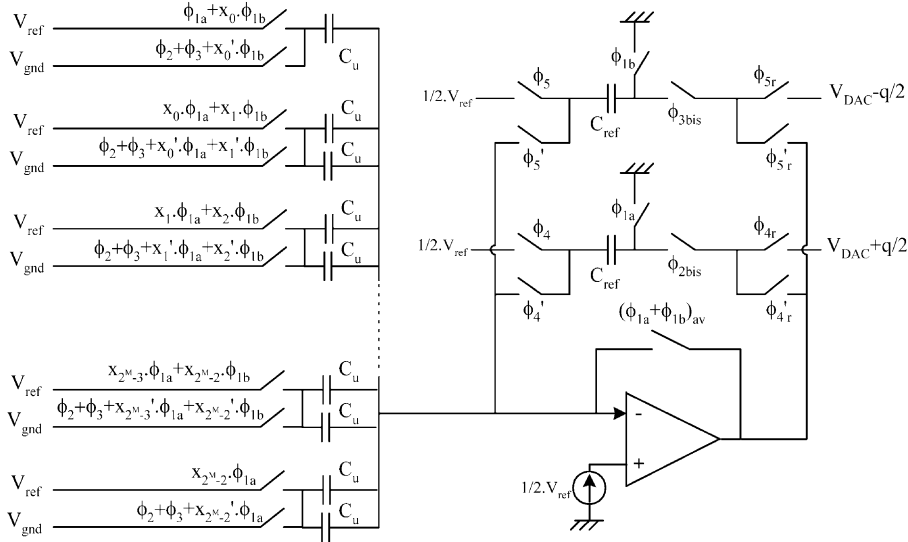


Fig. 7. Schematic of the DAC used for the A-ADC.

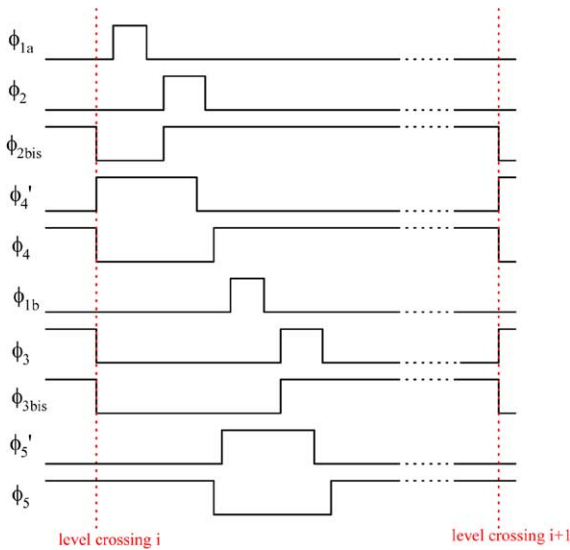


Fig. 8. Clocks used in the DAC given in Fig. 7.

356 switched on, and if $C_{ref} = (2^M - 1) \cdot C_u$, the ratio r
 357 between the voltage output V_{DAC} and V_{ref} is given
 358 by:

$$r = \frac{V_{DAC}}{V_{ref}} = \frac{2 \cdot m \cdot C_u}{2 \cdot (2^M - 1) \cdot C_u} \quad (14)$$

362 (without the offset voltages $+\frac{q}{2}$ and $-\frac{q}{2}$).

363
 364
 365
 366
 367
 3.2.1.1. Capacitances. For capacitances, considerations concerning the CMOS process used for the design must be taken into account. Actually, a capacitance can only be designed with a relative error given by the Pelgrom formula [15]

$$\sigma\left(\frac{\Delta C}{C}\right)\% = \frac{A}{\sqrt{W \cdot L}} + B. \quad (15)$$

371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500

The standard deviation of r is given by

$$\left(\frac{\sigma_r}{r}\right)\% = \left(\frac{\sigma_{C_u}}{C_u}\right)\% \cdot \sqrt{\frac{1}{2 \cdot m} + \frac{1}{2 \cdot (2^M - 1)}}. \quad (16)$$

If we want that 99.6% of the DACs verify the ENOB-bits of effective resolution, and supposing that $\Delta V_{in} = V_{ref}$, the following condition must be verified:

$$3 \cdot \sigma_{V_{dac}} \leq q_{eff} = \frac{V_{ref}}{2^{ENOB} - 1}. \quad (17)$$

389 The worst case appears when all the capaci-
390 tances are switched on: $m = 2^M - 1$. Supposing
391 that the capacitances are square shaped, according
392 to Eqs. (15)–(17), the minimum side L_{\min} of the
393 capacitances is given by

$$L_{\min} (\mu\text{m}) = \frac{3 \cdot A \cdot (2^{\text{ENOB}} - 1)}{100 \cdot \sqrt{2^M - 1}}. \quad (18)$$

396 Lastly, another kind of problems concerns the
397 non-linearity of a capacitance when its input volt-
398 age varies. This phenomenon is given by

$$C(V) = C_0 \cdot (1 + \text{DCDV1} \cdot V + \text{DCDV2} \cdot V^2). \quad (19)$$

401 where DCDV1 and DCDV2 are constants for a gi-
402 ven technology. With the orders of magnitudes of
403 actual CMOS processes, this aspect can be ne-
404 glected in our methodology.

405 *3.2.1.2. Switches.* Non-ideal switches have a finite
406 resistance R_{on} when they are on, it is given by
407 the formula:

$$R_{\text{on}} = \frac{1}{\mu_0 \cdot C_{\text{ox}} \cdot \frac{W}{L} \cdot (V_{\text{gs}} - V_{\text{r}} - V_{\text{ds}})}. \quad (20)$$

411 Then, the charges and discharges of the capaci-
412 tances through the switches in Fig. 7 can be mod-
413 elled as RC circuits. The whole loop delay δ of the
414 A-ADC can be attributed to the DAC because it is
415 the slowest block of the data path part of the A-
416 ADC. With the implementation of Fig. 7, four cy-
417 cles (Φ_{1a} , Φ_2 , Φ_{1b} and Φ_2) are needed to provide the
418 two output voltage references. The settling time
419 t_{DAC} of the DAC is given by

$$t_{\text{DAC}} = 2^{M+2} \cdot \ln(2^{\text{ENOB}} - 1) \cdot R_{\text{on}} \cdot C_u. \quad (21)$$

422 It must verify

$$t_{\text{DAC}} \leq \delta. \quad (22)$$

426 Thus, the maximum value of R_{on} can be computed
427 such as to verify relation (22). Then the minimum
428 size of the switches can be computed using Eq.
429 (20). Nevertheless, a tradeoff must then be used:
430 the size of the switches must not be as high as
431 we want because the clock feed through effect
432 might also reduce the effective resolution of the

A-ADC. To minimize this effect, certain clocks
presented in Fig. 7 are delayed. 433 434

3.2.1.3. Operationnal amplifier. Considering a finite
static gain A_{v_0} for the Operationnal Amplifier of
the DAC, the theoretical output voltages of the
DAC without the generation of the two voltage
offsets $\frac{q}{2}$ and $-\frac{q}{2}$ becomes 435 436 437 438 439

$$V_{\text{DAC}} = (V_{\text{ref}} - V_{\text{gnd}}) \cdot \frac{m}{2^M - 1} \cdot \left[\frac{1}{1 + \frac{2^M - 1 + m}{A_{v_0} \cdot (2^M - 1)}} \right]. \quad (23)$$

Supposing that A_{v_0} is very high, Eq. (24) becomes 442

$$V_{\text{DAC}} \approx \frac{m \cdot V_{\text{ref}}}{(2^M - 1)} \cdot \left[1 - \frac{2^M - 1 + m}{A_{v_0} \cdot (2^M - 1)} \right]. \quad (24)$$

Thanks to Eqs. (14) and (24), the error in ampli-
tude δV bring by the finite gain A_{v_0} is then 446 447

$$\delta V = \frac{(2^M - 1 + m) \cdot m \cdot V_{\text{ref}}}{A_{v_0} \cdot (2^M - 1)^2}. \quad (25)$$

For the worst case, all the capacitances are
switched on, and Eq. (12) must be verified. Using
Eq. (25), the following condition is obtained: 451 452 453

$$A_{v_0} \geq 4 \cdot (2^{\text{ENOB}} - 1). \quad (26)$$

Moreover, the loop delay δ of the A-ADC can be
attributed to the DAC. With the schematic of the
DAC given in Fig. 7 implemented in the converter,
four cycles are needed to provide the two output
voltage references. Thus, the cutoff frequency of
the Operationnal Amplifier must verify: 456 457 458 459 460 461

$$f_c \geq \frac{4}{\delta}. \quad (27)$$

3.2.2. The comparator 464

The level-crossing principle of the sampling
mode of the A-ADC leads to strong constraints
that do not exist for comparators of ADCs imple-
menting the Shannon sampling theory. Actually, a
comparator detects the instants where the analog
input signal crosses a reference level, and triggers
the conversion of a sample. This instant must be
detected very precisely because it is directly corre-
lated with the SNR of the A-ADC. Because of the 465 466 467 468 469 470 471 472 473

474 delay of the comparator that depends on the slope
475 of the analog signal V_{in} that must be converted, it
476 exists an error in time t_{comp} that may be translated
477 in an error in amplitude δV of the reference levels.

478 3.2.2.1. *Input capacitance.* Let $C_{in-comp}$ be the input
479 capacitance of the comparator. Let us find the
480 minimum value of $C_{in-comp}$ such as to verify the
481 effective resolution of the A-ADC. The value of
482 the analog output of the DAC without the gener-
483 ation of the voltage offsets $\frac{q}{2}$ and $-\frac{q}{2}$ becomes

$$V_{DAC-real} = \left(\frac{2 \cdot m \cdot C_u - 0.5 \cdot C_{in-comp}}{C_{ref} - C_{in-comp}} \right) \cdot V_{ref}. \quad (28)$$

487 The corresponding error δV in amplitude is given
488 by

$$\delta V = |V_{DAC} - V_{DAC-real}|, \quad (29)$$

492 it must always verify the condition given by Eq.
493 (12). Thus, thanks to Eqs. (28), (29) and (12), the
494 input capacitance of the comparator must verify:

$$C_{in-comp} \leq \frac{2 \cdot (2^M - 1) \cdot C_u}{2^{ENOB} + 1}. \quad (30)$$

497 3.2.2.2. *Gain and bandwidth.* A voltage comparator
498 can be modelled as a first-order system with a sta-
499 tic gain Av_0 , and a cutoff frequency f_c .

500 The corresponding transfer function is given by

$$\frac{V_{out}}{V_{in}} = \frac{Av_0}{1 + \tau_{comp} \cdot p} \quad (31)$$

503 with

$$f_c = \frac{1}{2 \cdot \pi \cdot \tau_{comp}} \quad (32)$$

506 and

$$f_T = Av_0 \cdot f_c. \quad (33)$$

509 When the analog signal V_{in} to convert crosses a
510 reference level, it can be approximated as a
511 straight line with the slope:

$$\left(\frac{\partial V_{in}}{\partial t} \right)_{V_{in} \approx \text{level } i}. \quad (34)$$

514 The corresponding output V_{out} of the comparator
515 is given by

$$V_{out}(p) = \left(\frac{Av_0}{1 + \tau_{comp} p} \right) \frac{1}{p^2} \cdot \left(\frac{\partial V_{in}}{\partial t} \right)_{V_{in} \approx \text{level } i} \quad (35)$$

In the time domain, Eq. (35) becomes 519

$$V_{out}(t) = Av_0 \cdot \alpha \cdot \left[\tau_{comp} \cdot \exp\left(\frac{-t}{\tau_{comp}}\right) - \tau_{comp} + t \right]. \quad (36)$$

The resolution time t_{comp} is given by the formula: 522

$$V_{out}(t_{comp}) = \frac{V_{dd}}{2}, \quad (37)$$

where V_{dd} is the power supply. It is the time 526
527 elapsed between a level crossing of V_{in} , and the in-
528 stant when the output of the comparator goes
529 high.

Solving Eq. (37), using Eq. (35), the resolution 530
531 time t_{comp} of the comparator is

$$t_{comp} \approx \sqrt{\frac{\tau_{comp} \cdot V_{dd}}{Av_0 \cdot \alpha}}. \quad (38)$$

The error in time t_{comp} can be translated in an 535
536 error in amplitude δV according to

$$\delta V = \left(\frac{\partial V_{in}}{\partial t} \right)_{V_{in} \approx \text{level } i} \cdot t_{comp}. \quad (39)$$

The precision of ENOB-bits given by Eq. (12) 540
541 must also be verified, according to Eqs. (38),
542 (39), and the Bernstein inequality (4), we have:

$$f_T = Av_0 \cdot f_c \geq 4 \cdot f_{max} \cdot (2^{ENOB} - 1)^2. \quad (40)$$

The loop delay is δ , thus the cutoff frequency of 545
546 the comparator must verify

$$f_c \geq \frac{1}{\delta}. \quad (41)$$

3.2.2.3. *Slew rate.* The slew-rate SR is defined as 549
550 the maximum slope of the output signal of the
551 comparator when the input varies with a high
552 dynamic

$$SR = \left(\frac{\partial V_{out}}{\partial t} \right)_{max}. \quad (42)$$

Let t_{SR} be the rising time of the output of the 555
556 comparator, it is given by

$$t_{SR} = \frac{1}{2} \cdot \frac{V_{dd}}{SR}. \quad (43)$$

560 The voltage error to this finite value of t_{SR} is gi-
561 ven by

$$\delta V = \frac{\partial V_{in}}{\partial t} \cdot t_{SR}. \quad (44)$$

565 The condition of precision (12) must be verified.
566 Using the Bernstein inequality (4), Eqs. (43) and
567 (44), the slew rate of the comparator must verify

$$SR \geq 2 \cdot \pi \cdot f_{max} \cdot V_{dd} \cdot (2^{ENOB} - 1). \quad (45)$$

570 3.2.3. Summary

571 This transistor level methodology is summed up
572 in Table 1. For a given application, the purpose is
573 to verify all these constraints for the comparator
574 and the DAC. We must also respect all these con-
575 ditions by insuring the lowest polarisation currents
576 for the analog blocks, such as to preserve a very
577 low power design for the A-ADC.

578 4. Case study

579 In this section, the design methodology pre-
580 sented in Section 3 is used in order to design an
581 A-ADC for a speech application.

582 4.1. Computation of the A-ADC parameters

583 The case study considered to illustrate our ap-
584 proach is the processing of speech signals. The
585 parameters of such signals are perfectly known.
586 The bandwidth of the signal is set to: $f_{max} =$
587 4 kHz, its amplitude ΔV_{in} is set to match 5–95%
588 of the A-ADC input range. The probability den-

sity function p of the amplitude of the signal is a
Laplacian distribution [16]:

$$p(x) = \frac{1}{\sqrt{2} \cdot \sigma_x} \cdot \exp\left(\frac{-\sqrt{2} \cdot |x|}{\sigma_x}\right). \quad (46)$$

It is easy to show that 99.65% of the speech
samples belong to the range

$$-4 \cdot \sigma_x \leq x \leq 4 \cdot \sigma_x. \quad (47)$$

It is deduced that

$$\sigma_x = \frac{0.9 \cdot \Delta V_{in}}{8}. \quad (48)$$

When $M \geq 4$, the Beutler condition is verified: the
average sampling frequency is: $\bar{f}_{sample} \approx 8.2 \text{ kHz} >$
 $2 \cdot f_{max}$. The resolution of the A-ADC is then:
 $M = 4$ -bits.

The tracking condition (3) gives the maximum
loop delay: $\delta_{max} = 2.65 \mu\text{s}$.

The PSD is given in [7], and thus Eq. (1)
becomes

$$SNR_{dB} = -66.3 + 20 \cdot \log\left(\frac{1}{T_C}\right). \quad (49)$$

Using Eqs. (6) and (48), the synchronous SNR of a
speech application is

$$SNR_{dB} = -7.27 + 6.02 \cdot ENOB. \quad (50)$$

Then, from Eqs. (49) and (50), we obtain the va-
lue of T_C according to the ENOB

$$T_C = 10^{\frac{59.03 + 6.02 \cdot ENOB}{20}}. \quad (51)$$

For example, to reach a 8-bits of ENOB, only a lo-
cal timer of $T_C^{-1} = 228.8 \text{ kHz}$ will have to be
considered.

With the considered technology, and according
to Eq. (10), whatever the ENOB is, the timer will
be implemented with $M_{timer} = 12$ -bits.

All the parameters being computed, the A-ADC
can now be designed. An ENOB of 8-bits is also
targeted.

4.2. Design of the A-ADC

We used the $0.18 \mu\text{m}$ CMOS technology from
STMicroelectronics. A voltage mode design has
been chosen to implement the analog part, and
all the characteristics of the analog part have been

Table 1

Transistor level design flow of an A-ADC given a targeted application

Comparator	$C_{in-comp} \leq \frac{2 \cdot (2^M - 1) \cdot C_u}{2^{ENOB} + 1}$ $f_T = A_{v0} \cdot f_C \geq 4 \cdot f_{max} \cdot (2^{ENOB} - 1)^2$ $f_C \geq \frac{1}{8}$ $SR \geq 2 \cdot \pi \cdot f_{max} \cdot V_{dd} \cdot (2^{ENOB} - 1)$
DAC	$L_{min} (\mu\text{m}) = \frac{3 \cdot A \cdot (2^{ENOB} - 1)}{100 \cdot \sqrt{2^M - 1}}$ $A_{v0} \geq 4 \cdot (2^{ENOB} - 1)$ $f_C \geq \frac{4}{8}$

Table 2

Electrical characteristics of the A-ADC targeted for a speech application

Hardware resolution	$M = 4$ -bits
Timer	12-bits, T_C^{-1} up to 1.37 MHz
ENOB	Up to 8-bits
Technology	0.18 μm CMOS
Power supply	$V_{\text{dd}} = 1.8$ V
Voltage dynamic	$\Delta V = 900$ mV
Loop delay	$\delta = 106$ ns
Input signal bandwidth	$f_{\text{max}} = 100.1$ kHz
Total static power consumption of the A-ADC	$P_{\text{avg}} = 95$ μW
Total dynamic power consumption of the A-ADC	$P_{\text{avg}} = 210$ μW at A-ADC max. speed
Timer consumption	$P_{\text{timer}} = 0.014$ mW (for ENOB = 8-bits)
Analog area	$S_{\text{analog}} = 120$ $\mu\text{m} \times 51$ μm
Digital area	$S_{\text{digital}} = 160$ $\mu\text{m} \times 80$ μm

635 determined according to the transistor level meth-
 636 odology described in Section 3. The A-ADC has
 637 been designed using classical architectures for the
 638 analog and digital blocks. So, in the future, opti-
 639 mizations will be done by studying specific low
 640 power designs for each block of the converter.
 641 The prototype (including the conversion loop,
 642 the timer and the synchronization interface) en-
 643 ables us to perform electrical simulations (cf. Table
 644 1). The total measured delay δ of the conversion
 645 loop, needed to convert a sample, is $\delta = 106$ ns.
 646 This value is very far from the calculated limit of
 647 speech signals ($\delta_{\text{max}} = 2.65$ μs). Thus, with Eq.
 648 (5), a maximum input frequency of $f_{\text{max}} =$
 649 100.1 kHz could be accepted. The fact that the
 650 loop is faster is not important because activity is
 651 only determined by the input signal and the num-
 652 ber of quantization levels (Table 2).

653 5. Discussion

654 5.1. Figure of merit

655 The Figure of Merit (FoM) we have chosen is
 656 defined by

$$\text{FoM} = \frac{2^{\text{ENOB}} \cdot f_{\text{samp}}}{P_m \cdot S}, \quad (52)$$

where P_m is the average dissipated power in Watts
 (W), f_{samp} is the sampling frequency of the con-
 659 verter in Hertz (Hz), and S the area of the core
 660 of the circuit in m^2 . More than forty recent ADCs
 661 published in IEEE journals and IEEE conferences
 662 proceedings, from 2000 to 2002 have been used for
 663 the comparison. The FoMs have been computed
 664 and are summed up in Fig. 9. Each dot is below
 665 10^{18} and it can be seen that the FoM of the A-
 666 ADC is one order of magnitude higher beyond
 667 ENOB = 8-bits: it reaches $\text{FoM} = 1.28 \times 10^{19}$.
 668 These results are pessimistic because the FoM of
 669 the A-ADC is computed with the total average
 670 power consumption of the converter. The FoM does
 671 not take into account the reduction of activity due
 672 to the level crossing sampling scheme. That is why
 673 behavioural simulations of the converter must be
 674 made in order to evaluate the gain in activity of
 675 the converter compared to its synchronous
 676 counterpart.

677 5.2. Conversion and signal processing

Let us now consider the use of ADCs in a sig-
 680 nal-processing chain to estimate the global perfor-
 681 mances of a whole system.

The purpose of A-ADCs is to have an asyn-
 683 chronous system, processing the irregular sampled
 684 data. The digital output of the A-ADC is com-
 685 posed of couples (b_i, Dt_i) (cf. Section 2.1). Usual
 686 Nyquist signal theory cannot be used, because
 687 samples are not time equi-spaced. Hence, new dig-
 688 ital asynchronous circuits, performing Fourier
 689 Transforms, filtering, detection... directly on the
 690 irregular stream of data, are now under investiga-
 691 tion in our research team, to implement such fully
 692 “asynchronous SoCs”.

Nevertheless, the A-ADC can firstly be used as
 694 a low-power front-end in a classical Nyquist signal
 695 processing system. In this way, an interpolation
 696 stage must be added to re-sample the samples in
 697 a regular way. This aspect has been studied in
 698 [7]. It is proved that the stream of data could be
 699 re-sampled to the Shannon frequency, respecting
 700 the initial A-ADC SNR, using a simple second-order
 701 polynomial interpolation.

Let us now consider a similar approach for the
 703 synchronous Nyquist ADC. It is perfectly known
 704

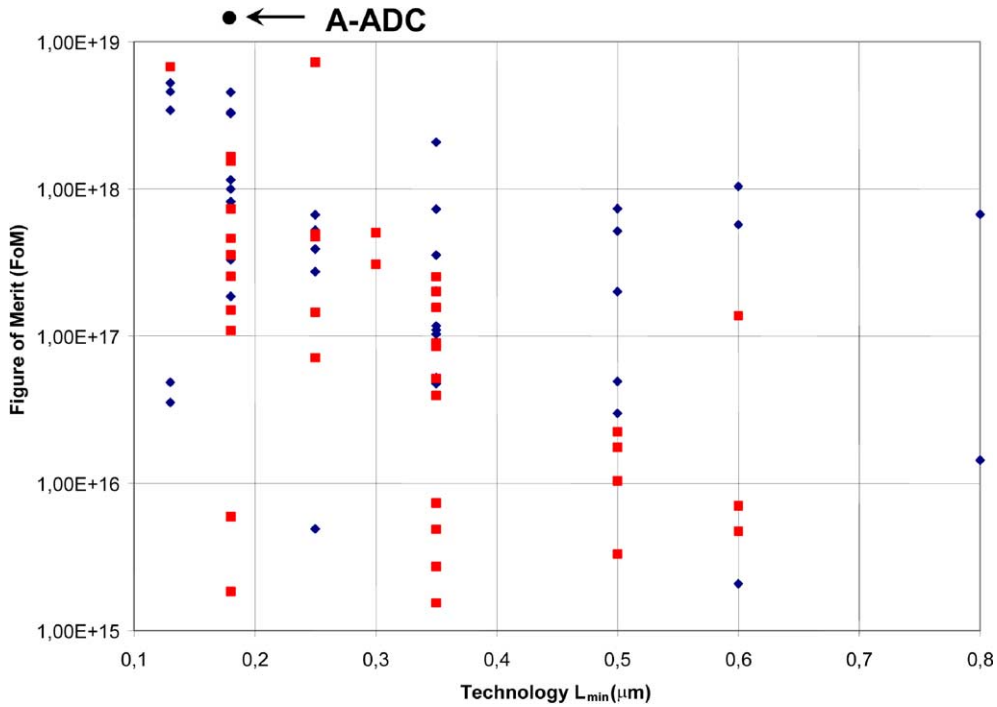


Fig. 9. FoM of state-of-the-art ADCs and A-ADC.

705 that the ENOB of any Nyquist ADC can be im- 726
 706 proved by over-sampling at f_{sample} the input signal 727
 707 V_{in} and filtering the quantization noise outside the 728
 708 signal bandwidth. In this case:

$$\text{SNR}_{\text{dB}} = 1.76 + 6.02 \cdot N + 10 \times \log \left(\frac{f_{\text{sample}}}{f_{\text{Nyq}}} \right). \quad (53)$$

711 The ENOB is improved by 1-bit, when the sam- 734
 712 pling frequency is multiplied by four. Given an M - 735
 713 bits resolution ADC, an effective resolution of 736
 714 ENOB-bits can be achieved if the sampling fre- 737
 715 quency is:

$$f_{\text{sample}} = 4 \cdot (\text{ENOB} - M) \cdot f_{\text{Nyq}}. \quad (54)$$

719 To preserve a similar hardware complexity for 740
 720 both types of conversions, we consider a Succes- 741
 721 sive Approximation ADC (SA-ADC) for the syn- 742
 722 chronous reference. The SA-ADC output must 743
 723 be processed by a decimation filter to obtain the 744
 724 ENOB-bits of resolution and a sampling frequency 745
 725 of f_{Nyq} . It limits the bandwidth of the over-sam-

726 pled input signal and down-samples the output. 727
 728 To simplify our comparison, we can consider the 729
 729 output digital stages of the two types of A/D con- 730
 730 verters (second-order polynomial interpolation 731
 731 and decimation filter) as equivalent in terms of 732
 732 complexity and power consumption, although this 733
 733 approximation is certainly unfair to the A-ADC. 734

735 Let us now analyze the two front-end conver- 736
 736 sion stages. For a speech signal, we have 737
 737 $f_{\text{Nyq}} = 2 \cdot f_{\text{max}} = 8$ kHz. For an SA-ADC, M cycles 738
 738 are needed to convert a sample. Using Eq. (54), the 739
 739 number of cycles per second is 740

$$f_{\text{cycle}} = 4 \cdot M \cdot (\text{ENOB} - M) \cdot f_{\text{Nyq}}. \quad (55)$$

741 For the A-ADC, we always have $\bar{f}_{\text{cycle}} = 742$
 742 $\bar{f}_{\text{sample}} \approx 8.2$ kHz. The gain brought by the 743
 743 A-ADC in terms of the number of cycles per sec- 744
 744 ond is 745

$$G_{\%} (\text{cycles/s}) = 100 \cdot \left[\frac{2 \cdot N \cdot f_{\text{max}} - \bar{f}_{\text{sample}}}{2 \cdot N \cdot f_{\text{max}}} \right]. \quad (56)$$

Table 3
A-ADC vs. ov-samp. SA-ADC

ENOB	SA-ADC nb cyc/s	T_C (Hz)	A-ADC avg nb cyc/s	G (%)
8-bit	512 k	754 k	≈ 8.2 k	98.39
10-bit	768 k	3.01 M	≈ 8.2 k	98.93
12-bit	1.024 M	12.06 M	≈ 8.2 k	99.19

746 The results are given in Table 3. Above
747 ENOB = 10-bits, the average number of cycles is
748 reduced by two orders of magnitude for the A-
749 ADC. Hence, in first approximation, a similar gain
750 in power savings is achieved. Thus, an A-ADC
751 could easily be used as a low-power front-end in
752 a classical Nyquist rate signal processing chain.
753 In this case, a second-order polynomial interpola-
754 tion stage must be used.

755 6. Conclusion

756 In this paper, a new class of ADCs based on an
757 asynchronous design mode and an irregular sam-
758 pling scheme is proposed. These two characteris-
759 tics are the source of a significant reduction in
760 power consumption, complexity and area. A meth-
761 odology is also presented in order to design such
762 converters, respecting a required ENOB, and
763 exploiting the statistical properties of the analog
764 input signal. An A-ADC has been designed for
765 speech applications in the 0.18 μm STMicroelec-
766 tronics CMOS process. Electrical simulations
767 demonstrate that the Figure of Merit (FoM) is in-
768 creased by one order of magnitude compared to
769 state-of-the-art synchronous Nyquist ADCs. This
770 result remains pessimistic because the FoM do
771 not take into account the reduction of activity of
772 the A-ADC due to the level crossing sampling
773 scheme. Nevertheless, behavioural simulations
774 prove that the activity of the circuit is decreased
775 by two orders of magnitude compared to a fully
776 synchronous counterpart. This aspect exists
777 thanks to the principle of the sampling scheme.
778 In terms of applications, A-ADCs can be used in
779 classical signal processing systems as a low-power
780 conversion front-end. Nevertheless, our goal is to
781 use these converters in fully asynchronous systems
782 combining irregular sampling and asynchronous

processing means. This solution is under investiga- 783
tion in our research team [17], and we believe it is 784
one of the most promising approaches to reduce 785
the power consumption of integrated mixed signal 786
circuits by more than one order of magnitude. 787

References 788

- [1] M. Renaudin, Asynchronous circuits and systems: a 789
promising design alternative, *Journal of Microelectronic* 790
Engineering 54 (2000) 133–149. 791
- [2] D. Kinniment, A. Yakovlev, B. Gao, Synchronous and 792
asynchronous A-D conversion, *IEEE Transactions on* 793
VLSI Systems 8 (2) (2000) 217–220, April. 794
- [3] D.J. Kinniment, A.V. Yakovlev, Low power, low noise 795
micropipelined flash A-D converter, *IEEE Proceedings on* 796
Circuits Devices Systems 146 (5) (1999) 263–267, October. 797
- [4] M. Conti, S. Orcioni, C. Turchetti, G. Biagetti, A current 798
mode multistable memory using asynchronous successive 799
approximation A/D converter, in: *Proceedings of the IEEE* 800
International Conference on Electronics, Circuits and 801
Systems, Cyprus, September 1999. 802
- [5] L. Alacoque, M. Renaudin, S. Nicolle, An irregular 803
sampling and local quantification scheme A–D converter, 804
IEE Electronics Letters 39 (n° 3) (2003) 263–264, February. 805
- [6] J.W. Mark, T.D. Todd, A nonuniform sampling approach 806
to data compression, *IEEE Transactions on Communica-* 807
tions COM-29 (n° 4) (1981) 24–32, January. 808
- [7] N. Sayiner, H.V. Sorensen, T.R. Viswanathan, A level- 809
crossing sampling scheme for A/D conversion, *IEEE* 810
Transactions on Circuits and Systems II 43 (n° 4) (1996) 811
335–339, April. 812
- [8] E. Allier, G. Sicard, L. Fesquet, M. Renaudin, A new class 813
of asynchronous A/D converters based on time quantiza- 814
tion, in: *Proceedings of the 9th IEEE International* 815
Symposium on the Design of Asynchronous Circuits and 816
Systems (ASYNC'03), 12–16 May 2003, Vancouver, BC, 817
Canada. 818
- [9] I.E. Sutherland, Micropipelines, *Communication of the* 819
ACM 32 (June) (1989) 720–738. 820
- [10] M. Shams, J.C. Ebergen, M.I. Elmasry, Optimizing CMOS 821
implementations of the C-Element, in: *Proceedings of the* 822
International Conference on Computer Design, pp. 700– 823
705, 1997. 824
- [11] J. Sauerbrey, D. Schmitt-Landsiedel, R. Thewes, A 0.5 V, 825
1 μV successive approximation ADC, in: *Proceedings of* 826
the 28th European Solid State Circuits Conference, 24–26 827
September 2002, Firenze, Italy. 828
- [12] M.D. Scott, B.E. Boser, K.S.J. Pister, An ultra-low power 829
ADC for distributed sensor networks, in: *Proceedings of* 830
the 28th European Solid State Circuits Conference, 24–26 831
September 2002, Firenze, Italy. 832
- [13] F.J. Beutler, Error-free recovery from irregularly spaced 833
samples, *SIAM Review* 8 (3) (1966) 328–335, July. 834

- 835 [14] U. Moon, J. Silva, J. Steensgaard, G.C. Temes, A switched
836 capacitor DAC with analog mismatch correction, IEEE
837 International Symposium on Circuits and Systems, May
838 2000.
- 839 [15] M. Pelgrom, A. Duinmaijer, A. Welbers, Matching prop-
840 erties of MOS transistors, IEEE Journal Solid State
841 Circuits 24 (December) (1989) 1433–1440.
- [16] L.R. Rabiner, R.W. Schafer, Digital Processing of Speech
842 Signals, Prentice Hall Inc., Englewood Cliffs, NJ, 1978. 843
- [17] F. Aeschlimann, E. Allier, L. Fesquet, M. Renaudin,
844 Asynchronous FIR filter: towards a new digital processing
845 chain, the 10th, in: IEEE International Symposium on the
846 Design of Asynchronous Circuits and Systems, 19–23 April
847 2004, Hersonissos, Crete, Greece. 848
849

Asynchronous FIR Filters: Towards a New Digital Processing Chain

F. Aeschlimann, E. Allier, L. Fesquet, M. Renaudin
TIMA Laboratory / Concurrent Integrated Systems Group
46 Avenue Félix Viallet, 38031 Grenoble Cedex, FRANCE.
Tel: +334 76 57 45 00 - Fax : +334 76 47 38 14

{Fabien.Aeschlimann, Emmanuel.Allier, Laurent.Fesquet, Marc.Renaudin}@imag.fr

Abstract

This paper is a contribution to the definition of a new kind of digital signal processing chain. It is focused on Finite-Impulse-Response filtering (FIR) applied to irregularly sampled signals obtained from an asynchronous analog to digital converter. The paper first formalizes the convolution operator in the irregular sampling context. The computational complexity is deduced and compared to the one of standard synchronous FIR filters. It shows that a significant reduction of the computational complexity is achievable, hence a reduction in terms of energy. The paper then describes the architecture of the asynchronous filter. It finally reports the simulations performed on a speech application, resulting in a reduction of the processing power of about one order of magnitude.

Keywords - Level-crossing sampling, Asynchronous design, FIR filter, Irregular sampling, Speech processing.

1. Introduction

This work is a contribution to a drastic change in classical digital signal processing chains: A/D converters, digital processing circuits and D/A converters. The main objective is to reduce their power consumption by one or two orders of magnitude by completely rethinking their architectures and the associated signal processing theory.

It is well known that asynchronous designs exhibit interesting properties [1]. This kind of design has been used in a few publications to improve the performances of Nyquist ADCs such as: immunity to metastable behavior [2], reduction of the electromagnetic interference [3], speed [4], and power consumption savings [5]. Moreover, most of the systems using ADCs imply signals with interesting statistical properties, but Nyquist signal processing architectures do not take advantage of them. Actually, these signals (such as temperature sensors, pressure sensors, electro-cardiograms, speech signals...) are almost always constant and may vary significantly only during short moments. Thus, classical regular

sampling and converting systems are highly constrained, due to the Shannon theory, which is to ensure for the sampling frequency to be at least twice the input signal frequency bandwidth. It has been proved in [6] and [7] that ADCs using a non equi-repartition of the samples in time leads to interesting power savings compared to Nyquist ADCs. A new class of ADCs called A-ADCs, based on irregular sampling, has been developed by the CIS group of the TIMA laboratory [8] [9]. This new class of ADCs consists of using both the “level-crossing” sampling scheme proposed in [6] and an asynchronous implementation of the circuit (no global clock).

This paper follows this previous contribution, joining up the A-ADC to an asynchronous implementation of a circuit processing the irregular samples.

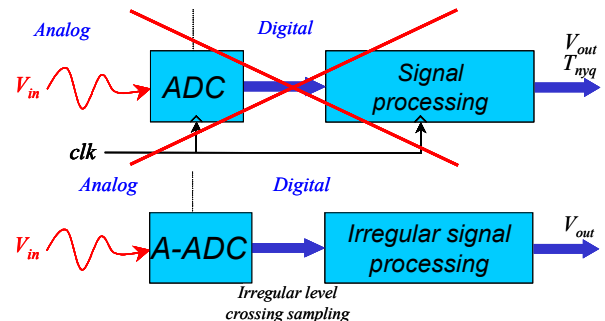


Figure 1: Dual approaches of signal processing chains.

Our new approach of signal processing is to combine asynchronous designs with signal event triggered processes in order to reduce dynamic activity.

Many studies deal with irregular signal theory but are limited to mathematical aspects like recovery of additive-random or jittered sampling process [10] [11]. The signal processing approach of this work is dedicated to low power applications especially in the area of Smart Devices and Communicating Objects. The Finite Impulse Response (FIR) filters have been chosen for their stability and convergence properties to illustrate this work [12]. Although FIR filtering is widely used in regular sampling

processing, no study has been referenced in the profuse literature about the case of irregular sampling.

This paper shows that the combination of asynchronous design and irregular sampling reduces the power consumption by one order of magnitude. This gain is due to the reduction of the number of samples processed.

After introducing the irregular sampling scheme, the A-ADC is presented in section 2. A new computation of the FIR processing for irregularly sampled signals is developed in section 3 in order to design a new kind of FIR architecture in section 4. Next, section 5 describes a complete new digital processing chain devoted to speech signals. Lastly, section 6 concludes the paper.

2. Asynchronous Analog to Digital

The asynchronous analog-to-digital converter (A-ADC) presented in [8][9] is based on an irregular sampling scheme called level-crossing sampling [6] and an asynchronous implementation. The system is only driven by the information present in the input signal. It only reacts to the analog input signal variations.

2.1. Irregular sampling

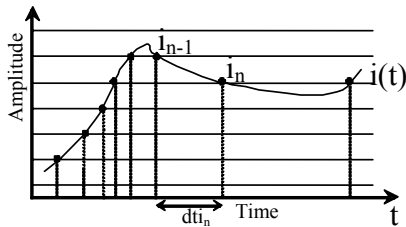


Figure 2: Irregular sampling.

For irregular sampling (cf. Figure 1) and a M-bit resolution, $2M-1$ quantization levels are regularly disposed along the amplitude range of the signal. A sample is taken only when the analog input signal $i(t)$ crosses one of them. Contrary to classical Nyquist sampling, the samples are not regularly spaced out in time, because they depend on the signal variations. Thus, together with the value of the sample i_n , the time dti_n elapsed since the previous sample i_{n-1} must be recorded. A local timer of period T_C is dedicated to this task. The amplitude of the sample is precise, but the time elapsed since the previous sample is quantized according to the precision T_C of the timer. The SNR depends on the timer

period T_C , and not on the number of quantization levels. Thus, for a given implementation of the irregular sampling A/D converter (a fixed number of quantization levels: $2M-1$), the SNR can be externally tuned by changing the period T_C of the timer. In theory, for irregular sampling, the SNR can be improved as far as it is needed, by reducing T_C . The limit is the accuracy of the analog blocks: they determine the precision of the quantization levels position in Figure 2.

2.2. Asynchronous ADC

The block diagram of the architecture performing this sampling scheme is shown in Figure 3. The converter resolution M and its dynamic range Δ set the quantification step:

$$q = \frac{\Delta}{2^M - 1} \quad \text{Eq. 1}$$

The output digital value V_{num} is converted to V_{ref} by the DAC, and compared to the input signal $i(t)$.

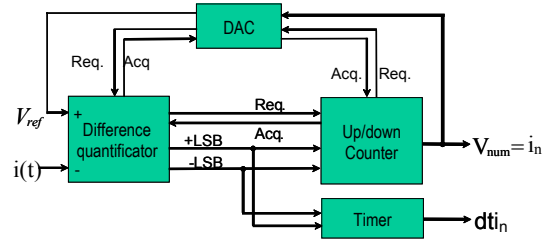


Figure 3: Block diagram of the A-ADC.

If the difference is greater than $\frac{1}{2}q$, the counter is incremented ($inc = '1'$), if it is lower than $-\frac{1}{2}q$, it is decremented ($dec = '1'$). In all other cases, nothing is done ($inc = dec = '0'$): the converter output signal V_{num} remains constant, there is no activity. The output signal is composed of couples (i_n, dti_n) where i_n is the digital value of the sample, and dti_n the time elapsed since the previous converted sample i_{n-1} , given by the timer. Information transfer is locally managed with a bi-directional control signalling. Each “data” signal is associated with two “control” signals: a request and an acknowledgement [1]. Let δ be the total delay of the loop, the slope of $i(t)$ must satisfy the “tracking condition”:

$$\left| \frac{di(t)}{dt} \right| \leq \frac{q}{\delta} \quad \text{Eq. 2}$$

In digital signal filtering, an input signal $i(t)$ represented by its samples (i_n, dti_n) is processed to obtain an output signal $o(t)$ represented by its samples (o_n, dto_n) . From an initial time t_0 , the time ti_n of the n^{th} input sample is reconstructed by the relation:

$$ti_n = ti_{n-1} + dti_n. \quad \text{Eq. 3}$$

This value is used to describe our approach but not in the algorithms since it is not available in practice.

3. Asynchronous Finite Impulse Response filtering

This section formally defines the FIR filtering computation when processing irregularly sampled digital signals.

3.1. FIR filtering computation of irregularly sampled signals

3.1.1. Principle

In a regular scheme, an N^{th} order FIR filter computes a digital convolution product, where T_{sample} is the sampling period, as described in Eq. 4.

$$o(t_n) = T_{\text{sample}} \sum_{k=0}^N h(t_k) i(t_{n-k}). \quad \text{Eq. 4}$$

Contrary to the regular scheme, the sampling time of the k^{th} sample of the impulse response h does not necessarily correspond to the sampling time of the $(n-k)^{\text{th}}$ sample of the input signal i (see Eq. 5)

$$th_k \neq ti_{n-k}. \quad \text{Eq. 5}$$

The product of the two samples is thus meaningless. The difficulty is bypassed by processing the convolution product between the resampled interpolated impulse response \hat{h} and the resampled interpolated input \hat{i} . Indeed, the output of a filtering process is defined as the numerical result of an analog convolution product of two analog signals obtained from two irregularly sampled signals by an interpolation. Actually, the convolution product is an area computation. It is well known that the easier way to compute this area is the rectangle method i.e. a zero-order method. Moreover in the classical digital computation, the output is also an area computation using the rectangle method (see Eq. 4), but all the intervals are constant, which it is not the case in irregular sampling. This means that it is necessary:

- to interpolate both input and impulse response signals at order 0,
- to resample the impulse response signal at the input sampling times,
- to resample the input signal at the impulse response sampling times.

At each new input sampling step, two new sampled signals \hat{h} and \hat{i} are calculated. Let \hat{h}_n and \hat{i}_n be the new irregular series of samples used to process the n^{th} output sample. The time of this sample corresponds to the delay introduced in the computation. The output is computed at the times of the input samples, which could be at any time. Then, the output signal is given in Eq. 5.

$$\begin{cases} o_n = o(ti_n) = \int \hat{h}_n(\tau) \hat{i}_n(ti_n - \tau) d\tau, \\ dto_n = dti_n. \end{cases} \quad \text{Eq. 6}$$

Moreover, the simple zero-order interpolation allows to split the area corresponding to the convolution product in a sum of rectangle areas with different widths. Figure 4 shows the general principle of resampling for the input signal and the impulse response of the filter. The continuous lines represent the original samples whereas the dashed lines correspond to the new calculated samples. After this step, each sample of \hat{i}_n corresponds to a sample of \hat{h}_n . It is now possible to process the convolution product.

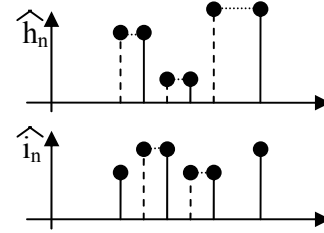


Figure 4: Principle of the resampling scheme used in the irregular FIR computation.

3.1.2. Optimization of the algorithm

For each new input sample, the algorithm is a priori

- to calculate both series at the sampling times,
- to split and sort the series,
- to resample both signals,
- to compute the sum of all the rectangle areas.

It is clear that such a procedure is not efficient. We should therefore optimize the convolution product procedure.

The solution is to compute each area without resampling but using an iterative loop. Figure 5 presents an example of the loop: at step 1, the first sub-area is the product of the two sample magnitudes i_n and h_0 by the minimum of the sample intervals: $i_n h_0 dt_{\text{min}}$. Then the input interval is subtracted from the impulse response interval: $dti_n = dti_n - dt_{\text{min}}$. The second h sample magnitude h_1 is introduced for the next computation. At step 2, the second

sub-area is the product of the two sample magnitudes i_n and h_j by the minimum of the sample intervals: $i_n h_j dti_n$. The impulse response interval is subtracted from the input interval: $dth_j = dth_j - dti_n$. The third computation introduces the second i sample magnitude i_{n-1} . Finally an algorithmic description of the loop is proposed in Eq. 7:

$$\begin{cases} o_n = \sum \min(dt_{i_{n-k}}, dth_j) i_{n-k} h_j, \\ dt_{i_n} = dt_{i_n}, \end{cases}$$

- if $\min = dt_{i_{n-k}}$ then $dth_j = dth_j - dt_{i_{n-k}}$;
 $k = k + 1$, Eq. 7
- if $\min = dth_j$ then $dt_{i_{n-k}} = dt_{i_{n-k}} - dth_j$;
 $j = j + 1$,
- if $\min = dth_j = dt_{i_{n-k}}$ then $k = k + 1$; $j = j + 1$.

The initial values of the indices k and j are zero as in the regular case. The break condition only depends on the impulse response since it is the only signal which is bounded in time (the impulse response consists in a finite number N of samples). Thus the loop is broken when j corresponds to the first null sample h_N .

This implies that the number of input samples used is variable and a priori unknown. Indeed, the length of the impulse response is th_N . During that time, the number of input samples used varies from one input to another. Therefore, a characteristic of the new convolution product defined here is that the number of samples of the input signal involved in the computation depends on both the input signal and the impulse response.

3.2. Computational complexity

3.2.1. General case

The performance of the loop is estimated in terms of the number of operations to compare the efficiency of the synchronous and asynchronous FIR filters. The output sample of a FIR filter is computed by the sum of S terms. In a synchronous $N-1$ order FIR filter, N multiplications/accumulations are computed using the N samples of the input signal and the impulse response, i.e. $S = N$. For the asynchronous FIR computation, S depends on the indices k and j but no analytical expression can be found. We thus suggest the following formulation:

$$S = N + M - \text{card}(t_{i_{n-k}} = th_j) + 1 \quad \text{Eq. 8}$$

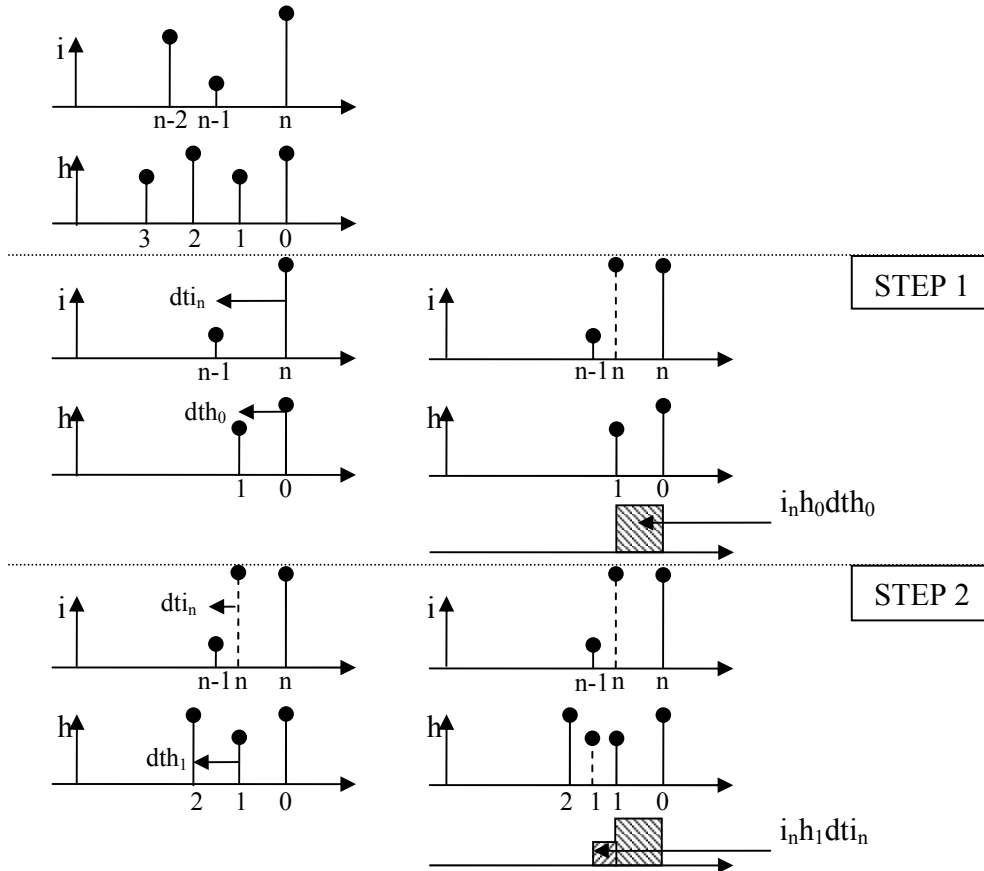


Figure 5: Optimization of the irregular FIR computation.

where M is the number of input samples taken in th_N (the impulse response duration). The value of M is computed as the smallest integer such that $\sum_{k=0}^M dti_{n-k} \geq th_N$.

The number of samples such that $ti_{n-k} = th_j$ is $card(ti_{n-k} = th_j)$ and has to be subtracted since no resampling is needed.

Except for the first iteration using i_n and h_0 , the probability that two samples are taken at the same time is almost zero. As a consequence $card(ti_{n-k} = th_j) = 1$. A way to estimate M is to consider its average value. Its expression is:

$$\begin{cases} M \approx \frac{th_N}{mean(dti_{n-k})} \\ th_N = N \cdot mean(dth_j) \end{cases} \quad \text{Eq. 9}$$

Therefore, $S = N+M$ and Eq. 8 becomes:

$$S \approx N \left(1 + \frac{mean(dth_j)}{mean(dti_{n-k})} \right). \quad \text{Eq. 10}$$

3.2.2. Regularly sampled impulse responses

The asynchronous FIR algorithm can process the convolution product of two irregularly sampled signals. However, a regularly sampled signal is a particular case with constant sample intervals. Thus in order to simplify the algorithm, it is possible to filter an irregularly sampled signal by a regularly sampled impulse response. This allows us to use all classical techniques available to compute the impulse response of the FIR filter.

Now, considering an $N-1$ order FIR filter with $th_N = NT_{sample}$, where T_{sample} is the sampling period used for the filter synthesis, equations Eq. 9 and Eq. 10 become:

$$\begin{cases} M \approx \frac{NT_{sample}}{mean(dti_{n-k})} \\ S \approx N \left(1 + \frac{T_{sample}}{mean(dti_{n-k})} \right) \end{cases} \quad \text{Eq. 11}$$

Eq. 11 leads to consider three cases for S according to the relation between T_{sample} and $mean(dti_{n-k})$:

- $mean(dti_{n-k}) \gg T_{sample}$: S tends to the classical cost N of a synchronous filter. In fact the computation of an output only requires one input sample which corresponds to the filtering of a constant signal. In terms of signal processing, this case is of limited interest.

- $mean(dti_{n-k}) \ll T_{sample}$: S tends theoretically to infinity. In fact, the number of input samples to consider within the T_{sample} time frame is very high. Nevertheless, S is bounded because $mean(dti_{n-k})$ is bounded by $\min(dti_{n-k}) = \delta$. Considering an input signal $i(t)$, the A-ADC parameter δ is designed according to the Bernstein inequality [9]:

$$\left| \frac{di(t)}{dt} \right| \leq \pi F_{MAX} \Delta = \frac{q}{\delta}, \quad \text{Eq. 12}$$

where F_{MAX} is the maximum frequency of the input signal and Δ the full dynamic range of the input signal. As $F_{MAX} = \frac{F_{sample}}{k} \leq \frac{F_{sample}}{2}$ with $k \geq 2$, the worst ratio becomes:

$$\frac{T_{sample}}{\delta} = \frac{\pi}{k} \frac{\Delta}{q} \leq \frac{\pi}{2} (2^M - 1). \quad \text{Eq. 13}$$

Thus S is bounded by S_{MAX} :

$$S_{MAX} = N \left(1 + \frac{T_{sample}}{\delta} \right) = N \left(1 + \frac{\pi}{2} (2^M - 1) \right). \quad \text{Eq. 14}$$

In this case, the upper bound of S is much larger than N (synchronous filtering). However, note that the statistical properties of the input signal are in this case not adequate with the proposed asynchronous filtering processing. A synchronous FIR filter should be preferred to the asynchronous FIR filter in this case.

- $mean(dti_{n-k}) \approx T_{sample}$: S tends to $2N$. In this case, the computational complexity is twice larger than for a synchronous FIR filter.

The choice of T_{sample} has of course to do with the filter characteristics. However, from the above discussion, it is clear that the choice of T_{sample} to synthesize the FIR filter should also take into account these estimations of the computational complexity.

Finally, note that, at the scale of a single output sample, the computational complexity can be higher than the synchronous case: $3S$ operations (2 multiplications and 1 addition) per output sample for the asynchronous convolution product versus $2N$ (1 multiplication and 1 addition) for the regular process. But on average, over the entire signal, a significant reduction of the computational complexity occurs since the number of irregular output samples computed is reduced compared to the regular case. Section 5 will illustrate this result.

4. Architecture of the asynchronous FIR filter

Many synchronous architectures exist for FIR filters like transversal direct form or iterative structures. Asynchronous implementations of the standard digital convolution scheme have also been proposed as a pipelined lattice structure [15] or a low-power FIR filter bank [16]. The architecture described in this section implements the new convolution algorithm proposed in section 3. Implementation details are beyond the scope of this study. The asynchronous FIR computation is implemented using four phase asynchronous circuits. Design techniques for low-power can of course be applied.

At a starting point, we consider an iterative structure using one multiplier and one accumulator, both controlled by a Finite State Machine (FSM). Five blocks compose such a FIR filter architecture as shown in Figure 6. A delay line stores the input samples and the impulse response coefficients. A multiplier and an accumulator compute the convolution and the result is stored in the output buffer. This architecture requires N clock cycles to compute an output sample.

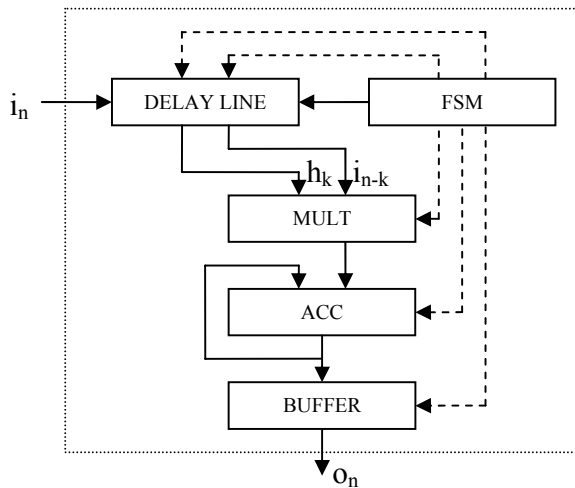


Figure 6: Classical FIR iterative architecture.

This iterative structure is well adapted to implement the asynchronous FIR filtering of Eq. 7. In fact, it easily supports the variable number of iterations required by the asynchronous filter to compute an output sample. Therefore, the algorithm described in section 3.1.2 is implemented with the modified architecture described in Figure 7.

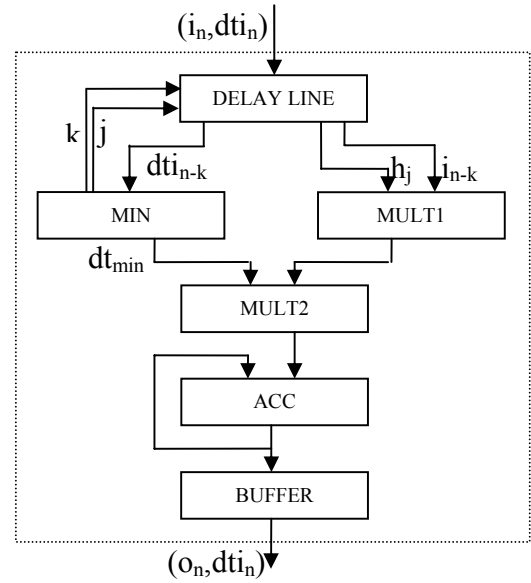


Figure 7: The new iterative architecture for the A-FIR filter.

The DELAY LINE block is the memory of the FIR filter. It stores the input samples (magnitudes and time intervals) in a RAM. Write and read pointers are used to access the input samples in the RAM. The impulse response coefficients are stored in a ROM accessed using a simple address counter.

On an A-ADC request, the new sample is stored in the RAM, and the read and write pointers are updated. As explained before, the number of input samples used during the convolution computation is not constant. Therefore, the number of input samples to be stored in the RAM is variable. As explained in section 3.2.2, this number is bounded by M_{MAX} :

$$M_{MAX} = N \frac{T_{sample}}{\min(dt_{n-k})} = N \frac{T_{sample}}{\delta}, \quad \text{Eq. 15}$$

where δ is the A-ADC loop time. In practice, this bound is never reached as shown in section 5. The RAM has to be sized using the input signal statistical properties. A software implementation can of course be used exploiting asynchronous microcontrollers [13] or microprocessors [14] features.

To compute the convolution according to Eq. 7, the RAM receives the index k from the MIN block, sends dt_{n-k} back to the MIN block and sends i_{n-k} to the MULT1 block. In addition, the ROM receives the index j from the MIN block and sends h_j to the MULT1 block. The

DELAY LINE block inputs/outputs are described in Figure 8.

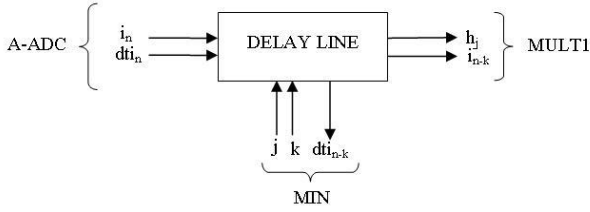


Figure 8: Description of the DELAY LINE block.

The MIN block is in charge of computing indices k and j as well as the current minimum time interval ($dtmin$) as specified in Eq. 7. To this aim, the MIN block handles two local variables for k and j , a local variable for the current time interval (dt_{n-k} or dth_j) and the T_{sample} constant value. The MIN block inputs/outputs are described in Figure 9.

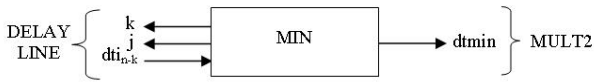


Figure 9: Description of the MIN block.

The MULT1, MULT2, ACC, and BUFFER are standard blocks.

5. Case study: a speech processing application

As an illustrative application of the proposed method, and to evaluate its efficiency, the filtering of a speech signal is considered. The application is to measure the fundamental frequency of a voice in order to detect the speaker's gender. It is well known that a man has a fundamental voice frequency in the interval [100; 150] Hz. The woman voice fundamental frequency lies in the interval [200; 300] Hz [17]. The audio signal bandwidth is limited to 5 kHz. It is hence sampled at $F_{sample} = 11.2$ kHz. This sampling frequency is taken as a reference to compare the synchronous and asynchronous approaches.

To extract the fundamental frequency, an asynchronous low pass FIR filter, with a cut off frequency of 300Hz is synthesized. To ensure a proper detection, T_{sample} is chosen as $1/F_{sample}$, and the obtained FIR filter has 321 coefficients.

The speech signal processed comes from a dialog. It is well known that in this situation, a speaker is only speaking 25% of the time [18]. Thus, the A-ADC remains inactive 75% of the time whereas a regular converter would work all the time. Moreover, it is proven that for speech signals, the A-ADC is reducing the number of samples [9].

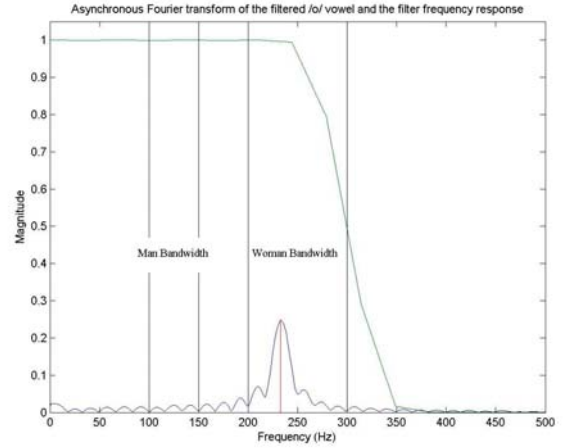


Figure 10: The asynchronous Fourier transform of the filtered 'o' vowel.

Figure 10 shows the functional result of the filtering using an asynchronous Fourier transform [19] performed on a window centered on the 'o' vowel (in Figure 11 between the instants 0.7s and 0.8s). It clearly shows that the fundamental frequency is 233Hz and that the speaker is a woman.

In Figure 11, the computational load of the asynchronous filtering is given. The analysis is focused on a window of width $T_i = 1.22$ s (active speaker, no silence). Within this window, the A-ADC converts $P_{ADC} = 2287$ samples, whereas $P_{REG} = 13692$ samples would be required by a synchronous ADC running at 11.2 kHz.

The number of samples is here reduced by a factor 6. At the scale of the entire dialog (25% of speech and 75% of silence) the reduction is increased by a factor 4. Therefore the number of asynchronous convolution product performed on the input signal is reduced by a factor 24.

As highlighted on Figure 11, the processing load of the considered audio signal is variable according to the $T_{sample}/mean(dti)$ ratio (see Table 1). The consonants are sampled with large time intervals since they behave like low amplitude noise: low computational power. The vowels, like any harmonic signals, are over sampled. Then, the computational power is higher.

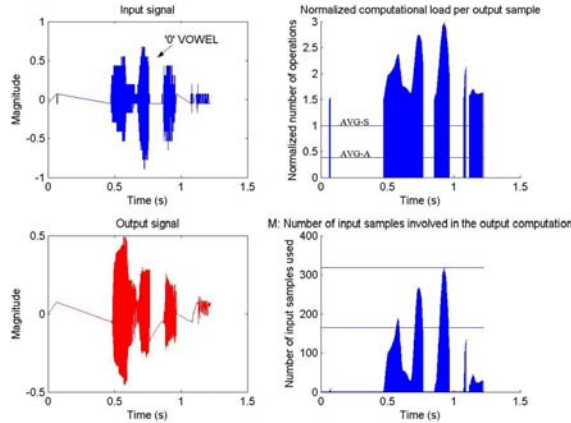


Figure 11: Results of the A-FIR filtering of a 1.2s speech signal. The ‘o’ vowel used to detect the speaker’s gender is centered on 0.75s. The computational load is normalized by the number of operations for a synchronous FIR filter. In this case, the average number of operations for the 1.2s period is $AVG-S = 1$. The number of operations for the asynchronous filter on the 1.2s period exhibits a significant reduction of the computational load: $AVG-A = 0.38$.

Table 1: Statistics on the asynchronous filtering computation. It confirms that $S = M+N$ and so $card(t_{i_{n-k}} = t_{i_j})$ tends to one.

	min	mean	max
M_n: number of input samples involved in the n^{th} output sample computation	1	165	318
S_n: number of taps required for the computation of the n^{th} output sample	321	485	638
$3S_n$: number of operations required for the computation of the n^{th} output sample	963	1455	1914
$2N$: number of operations required for the computation of any output sample in the regular case	642	642	642
$3S_n/2N$: normalized number of operations performed for the n^{th} output sample computation	1.5	2.26	2.98

As reported on Table 1, on average over a speech period of time (Figure 11), the asynchronous FIR filtering processes fewer input samples M_n but requires more operations ($3S_n$) per sample than a synchronous convolution ($2N$) per sample. However, if we now include the silence time periods, by considering the entire speech signal, the total number of output samples computed is much fewer than in the synchronous case. Therefore the overall computational power is decreased. This is formalized by expressing the gain G as the ratio of

the computational load required by a synchronous FIR filter over the computational load required by the asynchronous FIR filter, and this in a given amount of time:

$$G = \frac{2NP_{REG}}{3 \sum_{n=1}^{P_{A-ADC}} S_n} \text{ over a given time interval. Eq. 16}$$

In our example, $G \approx 10.5$ which means that the proposed asynchronous FIR filter leads to a significant reduction of the total number of operations.

6. Conclusion

In this paper a new class of digital signal processing chain is studied. The basic idea and main innovation lies in the processing of irregularly sampled signals, implemented using asynchronous hardware parts. In this context, the convolution operator is considered since it is a basic tool in the signal processing domain. It is first formally specified and a hardware-architecture derived. Then, it is formally proven that the computational complexity of the asynchronous FIR filter can be far lower than the computational complexity of the synchronous FIR filter, provided that the signal statistics are well exploited. A speech application is considered to illustrate this new approach. It demonstrates that the computational requirements, and hence the energy, can be reduced by more than one order of magnitude when compared to the standard regular sampling scheme. It should be noted that the gain could even be higher for other applications (medical, monitoring...).

Following the study on asynchronous ADCs [9], this study on asynchronous FIR filters confirms that the combination of irregular sampling and asynchronous technology is one of the most promising approaches to drastically reduce the power consumption of integrated mixed-signal circuits. Other signal processing tools treating irregularly sampled signals such as Infinite Impulse Response (IIR) filters, digital spectral analysis and digital to analog converters are currently being studied. We also plan to design and fabricate a hardware platform, based on A-ADCs and asynchronous digital processing parts, in order to measure and quantify the benefits of the approach (in terms of energy, speed, electromagnetic emissions...).

Acknowledgment

We would like to thank all the reviewers for discussion and comprehension efforts.

References

- [1] M. Renaudin, "Asynchronous Circuits and Systems: a Promising Design Alternative", *Journal of Microelectronic Engineering*, Vol. 54, pp. 133-149, 2000.
- [2] D. Kinniment *et al.*, "Synchronous and Asynchronous A-D Conversion", *IEEE Trans. on VLSI Syst.*, Vol. 8, n° 2, pp. 217-220, April 2000.
- [3] D.J. Kinniment *et al.*, "Low Power, Low Noise Micropipelined Flash A-D Converter", *IEE Proc. on Circ. Dev. Syst.*, Vol. 146, n° 5, pp. 263-267, Oct. 1999.
- [4] M. Conti *et al.*, "A Current Mode Multistable Memory Using Asynchronous Successive Approximation A/D Converter", *IEEE Int. Conf. on Elec. Circ. and Sys.*, Sept. 1999, Cyprus.
- [5] L. Alacoque *et al.*, "An Irregular Sampling and Local Quantification Scheme A-D Converter", *IEE Electronics Letters*, Vol. 39, n° 3, pp. 263-264, Feb. 2003.
- [6] J.W. Mark *et al.*, "A Nonuniform Sampling Approach to Data Compression", *IEEE Trans. on Communication.*, Vol. COM-29, n° 4, pp. 24-32, Jan. 1981.
- [7] N. Sayiner *et al.*, "A Level-Crossing Sampling Scheme for A/D Conversion", *IEEE Trans. on Circ. and Syst. II*, Vol. 43, n° 4, pp. 335-339, April 1996.
- [8] E. Allier, L. Fesquet, G. Sicard, M. Renaudin, "Low Power Asynchronous A/D Conversion", *Proceedings of the 12th International Workshop on Power and Timing, Modeling, Optimization and Simulation (PATMOS'02)*, September 11-13 2002, Sevilla, Spain.
- [9] E. Allier, G. Sicard, L. Fesquet, M. Renaudin, "A New Class of Asynchronous A/D Converters Based on Time Quantization", *ASYNC Proceedings*, pp. 197-205, May 12-16 2003, Vancouver, Canada.
- [10] R.J. Martin, "Irregular sampled signals theories and techniques for analysis", PhD thesis, *University College London*, England, January 1998.
- [11] Publications of NUHAG (The Numerical Harmonic Analysis Group), *University of Vienna*, Austria: <http://tyche.mat.univie.ac.at/papers/index.html>
- [12] G.V. Oppenheim, A.S. Willsky, I.T. Young, "Signals and systems", *Prentice-Hall*, 1995.
- [13] A. Abrial, J. Bouvier, M. Renaudin, P. Senn, P. Vivet, "A New Contactless Smart Card IC using On-Chip Antenna and Asynchronous Microcontroller", *IEEE Journal of Solid-State Circuits*, Vol. 36, 2001, pp. 1101-1107.
- [14] M. Renaudin, P. Vivet, F. Robin, "ASPRO: an Asynchronous 16-Bit RISC Microprocessor with DSP Capabilities", *ESSCIRC proceedings*, September 21-23, 1999, Duisburg, Germany.
- [15] U.V. Cummings, A.M. Lines, A.J. Martin, "An Asynchronous Pipelined Lattice Structure Filter", *ASYNC proceedings*, pp. 126-133, Nov. 3-5 1994, Salt Lake City, Utah.
- [16] L.S. Nielsen, J. Sparso, "A Low-power Asynchronous Data-path for a FIR filter bank", *ASYNC proceedings*, pp. 197-207, March 18-21 1996, Aizu-Wakamatsu, Fukushima, Japan.
- [17] L.R. Rabiner *et al.*, "Digital Processing of Speech Signals", *Prentice Hall Inc.*, 1978.
- [18] P.G. Fontollet, "Systèmes de Télécommunications", *Dunod*, 1983.
- [19] F. Aeschlimann, "Traitement Numérique du Signal sur des Signaux Echantillonnés Irrégulièrement", Master thesis, *Institut National Polytechnique de Grenoble*, Grenoble, France, June 2002.

Habilitation à Diriger des Recherches

Systèmes intégrés asynchrones et de traitement des signaux non uniformément échantillonnés

Laurent Fesquet

Résumé

Les travaux présentés dans cette habilitation sont le fruit d'une partie des recherches effectuées au sein du groupe CIS du laboratoire TIMA. Ils se sont focalisés sur des techniques « alternatives » de conception des systèmes intégrés et de traitement de l'information. Ces recherches ont mis en évidence la pertinence de l'approche asynchrone dans bien des domaines. Les techniques asynchrones permettent, par exemple, de concevoir des dispositifs de synchronisation sûrs, de sécuriser les circuits de chiffrement contre les attaques par canaux cachés mais aussi de concevoir plus aisément dans les technologies décanométriques où les problèmes liés aux variations de procédés de fabrication, les faibles tensions d'alimentation et la consommation statique sont devenus des enjeux délicats à traiter. La formalisation des méthodes de conception asynchrone a également permis de concevoir des outils de synthèse pour des circuits quasi-insensibles au délais et micropipelines. Enfin, une nouvelle approche pour le traitement du signal, se mariant bien avec la logique asynchrone qui est par essence événementielle, est proposée. Les recherches démontrent notamment les bénéfices que l'on peut tirer d'un échantillonnage non uniforme pour réduire d'un à deux ordres de grandeur la consommation d'un système intégré en traitement du signal.

Mots clés : systèmes intégrés asynchrones, synthèse de circuits asynchrones, traitement de signaux non uniformément échantillonnés.

Asynchronous integrated systems and non-uniformly sampled signal processing

Laurent Fesquet

Abstract

The studies presented in this HDR thesis result from a part of the CIS group activities at the TIMA laboratory. These researches have been focused on alternative techniques for designing signal processing and integrated systems (System-on-Chip). These works have demonstrated many times in many situations the asynchronous approach relevance. For instance, asynchronous techniques are suitable for designing reliable synchronization mechanisms, securing crypto-processors against side-channel attacks and easing designs in decanometric technologies where process variations, low-voltages and static leakage are challenging issues to address. The formalism of asynchronous design methods has also been studied in order to develop CAD tool for synthesizing Quasi-Delay Insensitive and micropipeline circuits. Finally a novel signal processing approach, fully compliant with the event-driven asynchronous logic principles, has been proposed. In particular, the researches show the benefits of non-uniformly sampled signals which are able to help reducing the integrated system power consumption by one to two orders of magnitude in the field of signal processing.

Keywords: Asynchronous integrated systems, Asynchronous logic synthesis, non-uniformly sampled signal processing.

ISBN : 978-2-84813-118-4