



EVALUATION DE LA COMPLEXITE D'IMPLANTATION EN VLSI PAR LA SYNTHÈSE ARCHITECTURALE : UNE EXPERIENCE EN FILTRAGE ADAPTATIF

P. MARTINEAU*, **A. BAGANNE***,
O. SENTIEYS**, **J.L. PHILIPPE****,
E. MARTIN*

* LESTER Université de Bretagne Sud, 10 rue jean ZAY 56100 Lorient

** LASTI Université de Rennes 1, 6 rue de Kerampont 22305 Lannion

Le problème de l'estimation de la complexité d'implantation des algorithmes de traitement du signal se pose dès leur phase de conception. Afin de disposer d'un moyen fiable pour cette estimation, nous proposons d'utiliser la synthèse d'architecture de circuits VLSI. Cette démarche a été appliquée à un jeu représentatif d'algorithmes de filtrage adaptatifs. Les résultats obtenus permettent de conclure à l'efficacité de la démarche, qui devrait intéresser d'autres domaines du traitement du signal, de l'image ou des télécommunications.

1. INTRODUCTION

Les applications de traitement du signal, comme c'est le cas en télécommunication, utilisent intensément le filtrage adaptatif. De nombreuses études ont porté ces dernières années sur l'amélioration du filtrage opéré et la diminution de la complexité de calcul [1]. Pour ce dernier critère, des techniques d'adaptation par bloc, de convolution circulaire, de sous échantillonnage ou autres ont permis de réduire la complexité arithmétique (C.A.) évaluée en nombre d'additions, de multiplications ou encore de multiplication-accumulations (MAC) [2]. Cette évaluation de la complexité permet une évaluation assez correcte du temps de traitement pour une mise en oeuvre sur processeur séquentiel (CISC, RISC ou DSP). Cependant dès que l'on utilisera le parallélisme pour les applications fortement contraintes, des paramètres difficiles à évaluer comme le parallélisme intrinsèque, la régularité [3], interviennent dans l'évaluation de la complexité d'implantation. Nous présentons dans cet article la synthèse d'architecture pour les circuits VLSI dédiés (ASICs) au §2, puis la technologie ciblée au §3. Nous montrons au §4 l'utilisation de la synthèse architecturale pour l'évaluation de la complexité d'implantation de filtres adaptatifs.

The problem of estimating the implementation complexity of a signal processing algorithm arises at the first step of the algorithm development. Architectural synthesis of VLSI circuits is reliable for this estimation. We applied this technique on a representative set of adaptive filters. The obtained results show the efficiency of the implementation complexity evaluation thanks to architectural synthesis. Various applications in signal or image processing, as well as telecommunication may be interesting in a such approach.

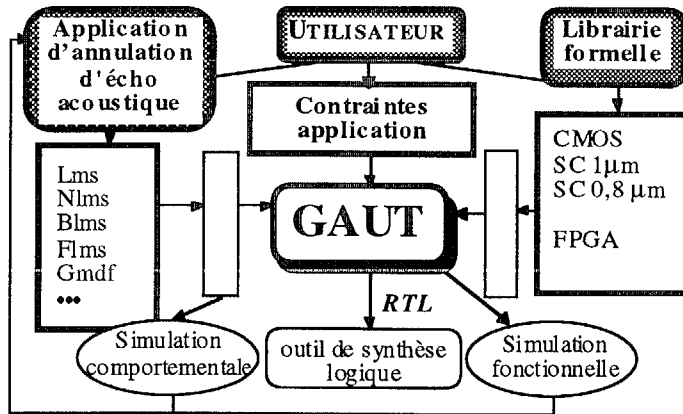
2. SYNTHÈSE D'ARCHITECTURE

Il existe actuellement de nombreux outils de synthèse logique commercialisés. Ceux-ci permettent d'obtenir à partir de la description d'une architecture (niveau de description RTL, ou transfert de registres), le plan de masse d'un circuit intégré dédié (ASIC). La synthèse d'architecture est une étape de conception amont à la synthèse logique [4]. A partir de la description comportementale d'une application qui fait abstraction de toute solution matérielle, l'outil de synthèse d'architecture génère la description d'une architecture de circuit qui peut être parallèle et qui permet la mise en oeuvre de l'application à moindre coût, tout en respectant les contraintes de temps d'exécution.

La première étape de la synthèse d'architecture consiste à transformer et paralléliser le code du programme qui décrit une application. Cette étape aboutit à une représentation interne, de type graphe flot de données (DFG), qui permet d'exprimer le parallélisme maximal des traitements. Dans une seconde étape, la synthèse d'architecture sélectionne le jeu approprié de composants à partir d'une bibliothèque fonctionnelle considérée, et ordonne les traitements, de manière à satisfaire la contrainte de temps d'exécution et de minimisation du coût de l'architecture. La synthèse d'architecture s'achève par l'optimisation des chemins de données (bus, registres,



etc....) et par la description à un niveau RTL de l'architecture et de son contrôleur. Nous avons développé au Laboratoire un outil de synthèse d'architecture dédié aux applications temps réel de traitement du signal. Il permet de concevoir des architectures de traitement de type coeur de processeur de traitement du signal, spécifiquement optimisées pour les applications considérées [5].



Environnement de la synthèse d'architectures.

La synthèse d'architecture permet aussi bien l'obtention d'un circuit dédié à une application, que la comparaison de la complexité d'implantation de différents algorithmes. Nous pouvons de plus évaluer, pour une application donnée la faisabilité de mise en oeuvre sur différentes technologies (ASICs en Standard Cells pour différentes technologies CMOS, circuits programmables de différentes familles (FPGA), etc.).

3. TECHNOLOGIE VLSI

Comme nous l'avons souligné précédemment, la synthèse d'architecture est une étape de la CAO de circuits VLSI s'interposant entre la spécification comportementale et la synthèse logique. Cette dernière étape de la conception permet de réaliser autant d'opérateurs optimisés que peut l'imaginer un concepteur. L'outil de synthèse d'architecture n'utilise qu'une vue fonctionnelle des différents opérateurs conçus au niveau de la synthèse logique. Il permet la réalisation de circuits d'autant plus optimisés qu'il dispose d'une importante bibliothèque d'opérateurs. L'outil GAUT utilise une bibliothèque complète d'opérateurs traitant des nombres sur 16 bits en virgule fixe cadrée à gauche. Chaque composant de la bibliothèque est modélisé en VHDL par différents paramètres génériques, comme le temps de fonctionnement, la surface, le format des entrées et

sorties, ou encore le nombre de tranches de pipeline ou de fonctions réalisées.

Nous donnons ci-dessous un résumé des principaux opérateurs modélisés en technologie ES2 CMOS 1µm à l'aide de l'outil de synthèse logique COMPASS.

Composants (16 bits)	Type	Temps (ns)	Coût (mm ²)
Multiplieur	Datapath VHDL	56 / 40	1,26 / 3,16
		53 / 74	2,5 / 2,02
Mult. pipeline	Datapath	32	1,65
		57	1,49
Additionneur	Datapath VHDL	12	0,35
		19 / 11	0,11 / 0,21
Soustracteur	Datapath VHDL	12	0,356
		20 / 11	0,11 / 0,217
Additionneur/ Soustracteur	Datapath VHDL	23 / 16	0,19 / 0,39
		20 / 12	0,32 / 0,58
Comparateur	VHDL	16	0,285
Add/Sous/Comp	VHDL	21	0,65
Mult. / Add. Mult/Add pipeline	Datapath Datapath	67	1,83
		41	1,96
Registre / Multiplexeur	VHDL	2,5 / 1,2	0,144 / 0,04
Sortie trois-états	VHDL	5,1 / 1,1	0,062/0,029

Librairie ES2 1µ CMOS SC:

Exemples d'opérateurs conçus par synthèse logique

L'étape de la conception chargé de la sélection de module a pour objet de retenir parmi les nombreuses possibilité le jeu minimum d'opérateurs satisfaisant les contraintes de la synthèse: temps d'exécution et minimisation du coût de l'architecture.

4. COMPLEXITE D'IMPLANTATION

Nous avons utilisé l'outil de synthèse d'architecture GAUT [5], développé au Laboratoire, pour analyser la complexité d'implantation d'algorithmes de filtrage adaptatif. Ce travail a été mené dans le cadre d'une convention CNET-GDR TDSI. Les algorithmes étudiés ont été sélectionnés pour représenter les différentes tendances en terme de développement algorithmique dans les familles des algorithmes de moindre carré et des algorithmes stochastiques du gradient.

Deux applications ont été retenues, traitant de l'annulation d'écho acoustique. La première concerne les télécommunications en téléconférence, la seconde concerne les télécommunications en radio mobile. Ces deux application reposent sur les mêmes algorithmes de filtrage adaptatif, mais elles supportent des contraintes différentes (longueur de la réponse impulsionnelle et fréquence d'échantillonnage bien

plus importantes pour la première que la seconde application).

4.1. PRESENTATION DES ALGORITHMES ADAPTATIFS

L'un des algorithmes le plus utilisé est l'algorithme LMS, dit du gradient stochastique (ou sa version normalisée NLMS). Ses différentes variantes, par bloc dans le temps (BLMS) ou en fréquence (FLMS, GMDF) forment la première famille d'algorithmes adaptatifs étudiés. L'algorithme RLS, ou des moindres carrés résout le problème du filtrage optimal sans faire de supposition sur les statistiques des signaux d'entrée. Il présente de meilleures qualités de convergence que le LMS, mais est très complexe en nombre de calculs. Ainsi ses versions rapides, ou moindres carrés rapides (FRLS, FTF) forment la seconde famille d'algorithmes étudiés. Différents auteurs, dont [3], ont indiqué les complexités, en nombre d'opérations de ces filtres. Celles-ci sont reprises dans le tableau suivant:

Filtre	(N)LMS	BLMS	FLMS	RLS
Complexité \times	2N	2N	$20\log_2(2N)+16$	$3N^2+4N$
Complexité $+$	2N	2N	$30\log_2(2N)+8$	$2N^2+2N$
Filtre	FTF81	GAL	GMDF α	
Complexité \times	$8N + 13$	$3N + 5C^\dagger$	$\alpha[10K+3\log_2(N)-3]$	
Complexité $+$	$8N+5$	$2N+4C^\dagger$	$\alpha[8K+9\log_2(N)-10]$	

Complexité théorique des algorithmes (N est l'ordre du filtre).
 \dagger : le nombre C de cellules est compris entre 1 et N.

L'évaluation de la complexité des algorithmes est mesurée par le nombre de multiplications, ou par une somme pondérée des multiplications plus des additions - soustractions. La pondération dépend directement pour les processeurs CISCs du nombre de cycles nécessaires à traiter les différents types d'opérations.

Afin de comparer les différents algorithmes par cette mesure de complexité d'implantation, nous avons considéré comme référence l'algorithme LMS, puis nous déterminons l'augmentation ou la réduction de complexité d'implantation des algorithmes, relativement à cet algorithme de base.

4.2. APPLICATION TELECONFERENCE

Les contraintes d'une application d'annulation d'écho acoustique en téléconférence sont la taille du filtre (approximativement 1000 points) et la fréquence d'échantillonnage (16 KHz). Les circuits sont conçus

en technologie ES2 CMOS 1 μ m à l'aide d'opérateurs présentés au §3.

Les résultats de synthèse sont comparés comme suit: Les filtre LMS a la complexité unitaire (en nombre de multiplication d'une part et en surface des circuits d'autre part). La complexité d'implantation des autres filtres est indiquée relativement au filtre LMS. Nous précisons dans le tableau suivant les principaux opérateurs mis en oeuvre.

Filtre 1024 pts 16KHz	C.A.	C.I. mm ²	Opérateurs					
			*	+	-	+/-	RegMux	
Blms bloc 4	1	1,81 6,140	2	2	1		13	11
FTF 8L	4	5,72 19,394	5			8	37	12
Gal 30 cellules	1,58	2,28 7,738	2			2	16	8
Gal N cellules	4	7,48 25,348	10			8	42	33
Gmdf K=128,N=8, $\alpha=4$	5,02	5,36 32,560	5	3	2		36	29
Lms	1	1 3,388	1				1	6
Nlms	1	1,02 3,462	1	1	1		6	4

Complexités relatives des filtres adaptatifs: en surface de silicium et en nombre d'opérations. (Reg: registre; Mux: multiplexeur)

C.A. Complexité arithmétique; C.I. Complexité d'implantation. (les divisions sont traitées par soustraction - décalage; Multi pipe: multiplieur pipeline)

On pourra constater dans cet exemple que la complexité compté en nombre d'opération peut être inférieure jusqu'à 80% à la complexité d'implantation en mm². On peut constater que le rôle des registres n'est pas négligeable dans le coût final de l'architecture, et que leur nombre diminue lorsque les algorithmes sont de nature très séquentielle (un seul opérateur de chaque type dans l'architecture).

4.3. APPLICATION RADIO MOBILE

Pour une application d'annulation d'écho acoustique en radio mobile, les contraintes sont la taille du filtre (approximativement 128 points) et la fréquence d'échantillonnage (8 KHz). La bibliothèque utilisée est la même que précédemment



Filtre 128 pts 8KHz	C.A.	C.I. <i>mm</i> ²	Opérateurs					4	2
			*	+	-	+/-Reg	Mux		
Blms bloc16	1	1 2,456	1	1	1				
Blms bloc32	1	1 2,456	1	1	1				
Blms bloc8	1	1 2,456	1	1	1				
Flms	0,69	0,97 2,382	1				1	4 2	
Ftf 8l	4	2,22 5,470	1	1	1	1	15	8	
Gal	4	1 2,468	1	1	1			4 2	
Gmdf K=8 N=16, α=4	2,82	1,31 3,220	1				1	7 5	
Lms	1	1 2,458	1	1	1			4 3	
Nlms	1	1,04 2,550	1	1	1			5 2	

Complexité des filtres adaptatifs pour l'application radio mobile, en surface de l'unité de traitement.

C.A. Complexité arithmétique; C.I. Complexité d'implantation

On constatera dans ce second exemple que la limite technologique n'est pas atteinte, c'est à dire que tous les filtres sont implantés sur une unité de traitement de complexité minimale (un seul opérateur de chaque type).

Nous sommes en présence d'applications pour lesquelles la complexité d'implantation de ces filtres sera largement déterminée par la taille et la gestion des mémoires à mettre en oeuvre. Ainsi, nous avons relevé le nombre de transferts entre la mémoire et l'unité de traitement constaté lors de la synthèse de ces différents filtres. La taille mémoire sera évidemment inférieure au nombre de transferts.

Filtre	LMS	NLMS	BLMS	
Transferts mémoire	257	262	536 (bloc 8) 613 (bloc 16) 613 (bloc 32)	
Filtre	GMDF	FLMS	FTF8L	GAL
Transferts mémoire	4770	2561	1305	1024

Nombre de transferts mémoire pour les différents filtres.
Ces transferts majorent la taille de la mémoire à mettre en oeuvre.

5. PERFORMANCES ET LIMITATIONS DE LA DEMARCHE

L'apport de la synthèse d'architecture, pour l'évaluation de la complexité d'implantation des applications de traitement du signal, doit se traduire

par un gain de temps substantiel. On peut ainsi concevoir que l'évaluation comparée de filtre permette d'affiner les directions dans lesquelles doit travailler le spécialiste en algorithme. Notons par exemple que le traitement par bloc dans le domaine fréquentiel des algorithmes, pose un problème important au niveau de la gestion mémoire. Celle-ci est particulièrement critique pour les applications peu contraintes.

Les limitations actuelles de la démarche sont liées principalement à l'interaction qu'il existe entre le style d'écriture au niveau comportemental d'un algorithme et l'optimisation de l'architecture. Nous avons noté dans [6] que la modification de la description pouvait faire varier l'optimisation de l'architecture dans un rapport pouvant aller jusqu'à 30% de la surface de silicium.

Cette limitation devrait disparaître avec la mise en oeuvre des transformations utilisées au niveau des compilateurs logiciels, grâce notamment à l'interfaçage de GAUT avec des langages et environnements de spécification comme SIGNAL, PTOLEMY, etc.

BIBLIOGRAPHIE

- [1] O. Macchi & all « Comparaison d'algorithmes adaptatifs en milieu non stationnaire » Revue Traitement du signal - Vol 6 n°5, 1990.
- [2] P. Duhamel, J. Sanchez & H. BARAL "Expériences en implantation" Colloque Adéquation Algorithme Architecture GDR 134, Grenoble 20-21 janvier 1994.
- [3] J. Sanchez « Prédiction et évaluation de performance des algorithmes adaptatifs implantés sur machines parallèles », *Thèse Télécom Paris 93 E 021*, (Novembre 1993).
- [4] D. Gajski, N. Dutt, A. Wu & S. Lin « High level synthesis » *Kluwer academic publisher* 1988.
- [5] E. Martin, J.L. Philippe, H. Dubois & O. Sentieys « GAUT: An architectural synthesis tool for dedicated signal processor » Conférence EURODAC 93, 20-24 septembre 1993, Hambourg.
- [6] E. Martin, J.L. Philippe « Parallélisme à grain fin pour la synthèse d'architectures en traitement du signal » Conférence RENPAR 6, Juin 1994, Lyon.