

IMPLANTATION DE FILTRES NUMERIQUES SUR COMPOSANTS LOGIQUES PROGRAMMABLES

Prof. LEICH H. Dr.Ir. HANCQ J.

Faculté Polytechnique de Mons
Service de Théorie des Circuits et de Traitement du Signal
31, Boulevard DOLEZ, B-7000 MONS (Belgique)
Tel/Fax : +32[65]374128 / 374300, E-Mail : hancq@tcts.fpms.ac.be

RÉSUMÉ

La disponibilité de composants programmables et reconfigurables comme les FPGA permet d'envisager de nouvelles implantations en "traitement du signal". En particulier, la réalisation de filtres numériques dans de telles structures est ici envisagée. Après avoir comparé diverses réalisations "pipeline", le principe des outils nécessaires pour la résolution d'un tel problème est présenté ainsi qu'un bilan des performances qui peuvent être atteintes avec les structures actuelles. Dans le cadre des outils de développement, des algorithmes sous-optimaux de minimisation du coût de réalisation des filtres FIR et IIR ont été développés.

ABSTRACT

The availability of reconfigurable field-programmable gate arrays has opened up new possibilities for digital signal processing implementation. In this paper, we discuss the realization of programmable digital filters. Different pipeline realizations are compared and the principle of design tools required to resolve this problem is described as also the performances attainable with the existing components. Within the framework of FIR and IIR filters design tools, suboptimal cost minimization algorithms for their implementation have been conceived.

1. INTRODUCTION

La disponibilité de composants programmables tels que les CPLD et les FPGA, permet d'envisager de nouvelles solutions programmées pour la réalisation de structures de traitement des signaux. Les applications développées ici sont des réalisations nécessitant des filtres numériques dans des domaines de fréquence d'échantillonnage qui ne peuvent être couverts par les microprocesseurs DSP conventionnels, sans toutefois recourir à des circuits spécialisés de filtrage.

On prendra principalement en compte les composants à cellules de base et connexions définies par des zones de mémoire statique SRAM tels que ceux présentés par les firmes Xilinx, Altera, Plessey, Algotronix et Concurrent Logic qui offrent en plus une possibilité de reconfiguration en cours de fonctionnement. Toutefois, la méthodologie reste applicable à tout type de tels composants.

La recherche d'une structure optimale de filtrage, liée aux caractéristiques des cellules de base disponibles dans ces composants et ce, de façon à réduire au minimum le coût en cellules consommées pour l'implantation de filtres permet d'offrir des degrés de liberté

- soit en sélectionnant un composant de coût réduit
- soit en autorisant l'adjonction dans un même composant, de fonctions additionnelles, solution des plus intéressantes vis-à-vis de l'emploi de circuits spécialisés de filtrage.

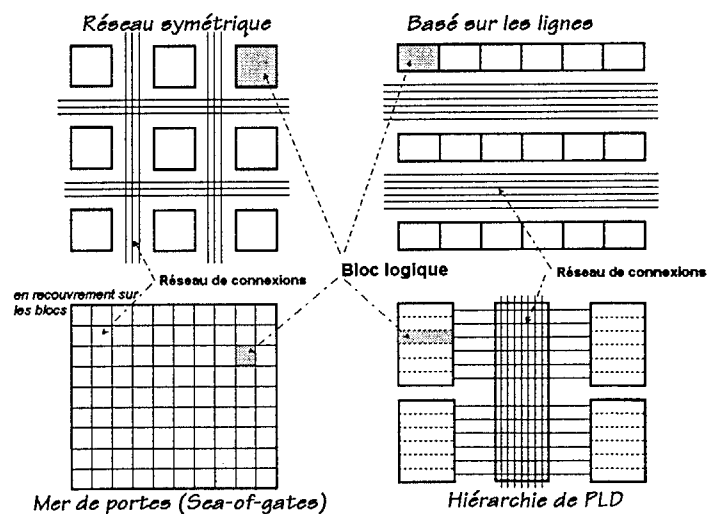


Figure 1 - Principales structures rencontrées au niveau des FPGA

La figure 1 représente les formes structurelles généralement rencontrées au niveau des CPLD et FPGA [4,6]. Parmi celles-ci, les composants reconfigurables rencontrés sont du type:

- "réseau symétrique" comme c'est le cas chez Xilinx et Altera. Les cellules élémentaires sont alors structurées autour de "blocs universels" permettant la réalisation de n'importe quelle "fonction de N-entrées" décrite sous la forme d'une table comme le montre la figure 2 pour un composant de la famille XC2000 chez Xilinx.
- "mer de portes" comme c'est le cas chez Concurrent Logic, Plessey et Algotronix. Les cellules élémentaires possèdent diverses structures utilisant soit des multiplexeurs, soit des blocs universels.

2. STRUCTURES DES COMPOSANTS CIBLES

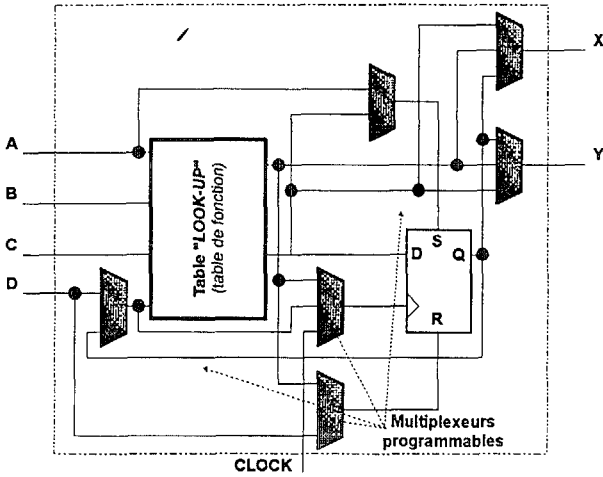


Figure 2 - Exemple d'une cellule élémentaire chez Xilinx

3. CHOIX D'UNE ARITHMETIQUE

Le choix se porte sur les représentations du type "complément à deux" pour lesquelles on note les avantages suivants:

- L'addition de 2 nombres peut être réalisée avec un additionneur en binaire naturel dit "additionneur complet à 1 bit".
- La soustraction revient à ajouter la forme complémentée.
- La tolérance aux dépassements locaux en tout point d'accumulation; si le résultat est borné et que cette borne est connue avec certitude, on ne doit pas disposer de bits de garde vis-à-vis de cette borne.

Une implantation "pipeline" de la multiplication-accumulation, opération de base du filtrage numérique [5], est donnée à la figure 3. Elle présente l'avantage de limiter les propagations à un étage d'additionneur complet à 1 bit.

Les éléments "délais" présents dans la branche "horizontale" inférieure mémorisent les accumulations partielles. Si (p+1) désigne le nombre de bits du multiplicande (signe compris) et (q+1) celui du multiplicateur (signe compris), ces éléments délais sont de longueur (p+q+1) bits. Chaque multiplicateur est implanté sous la forme de (q+1) cellules d'addition mises en cascade et une cellule d'addition notée "accumulateur" est nécessaire en sortie des cellules de multiplication.

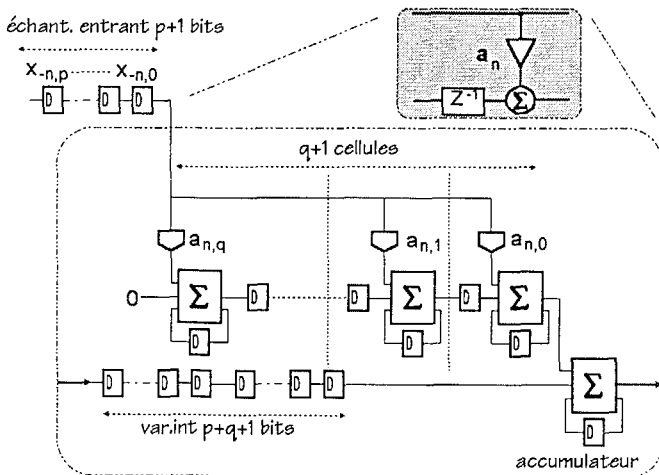


Figure 3 - Structure pipeline d'un multiplieur-accumulateur.

La multiplication est ainsi réalisée sur base de sommes décalées en se déplaçant parallèlement à une diagonale dans la matrice de multiplication [5]. En complément à 2, on montre que le résultat correct est obtenu en :

- considérant les représentations dans les sommes partielles étendues à (p+q+1) bits
- soustrayant le multiplicande décalé de 2^q au lieu de l'ajouter si le multiplicateur est négatif (dit "traitement du bit de signe"), ce qui ne coûte rien au niveau des cellules.

On note qu'une extension de signe doit être réalisée lors du décalage dans le registre contenant le multiplicande.

Une forme particulière de codage du multiplicateur est déduite directement de l'algorithme de **BOOTH**. Elle consiste en décomposer celui-ci sur base des facteurs binaires [-1,0,1] associés aux puissances binaires ce qui se traduit par des additions/soustractions du multiplicande décalé.

4. IMPLANTATION DES FILTRES

La limitation des propagations dans le cadre de l'implantation pipeline conduit à utiliser les structures transposées. Ainsi, la structure d'un filtre **FIR** est donnée à la figure 4 où l'accroissement de performances en fréquence d'échantillonnage se traduit par un coût accru en mémorisation car il faut maintenir les accumulations partielles d'où la nécessité d'opérateurs délais à (p+q+1) bits.

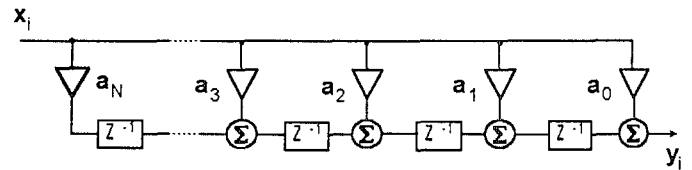


Figure 4 - Structure transposée ou à délai de propagation réduit pour filtre FIR

Pour les filtres **IIR**, la structure est donnée à la figure 5. Toutefois la nécessité de disposer d'une sortie déjà calculée pour faciliter l'implantation pipeline conduit à transformer la cellule transposée et à proposer la cellule modifiée de la figure 6 dont la figure 7 donne un exemple d'utilisation en implantation cascade.

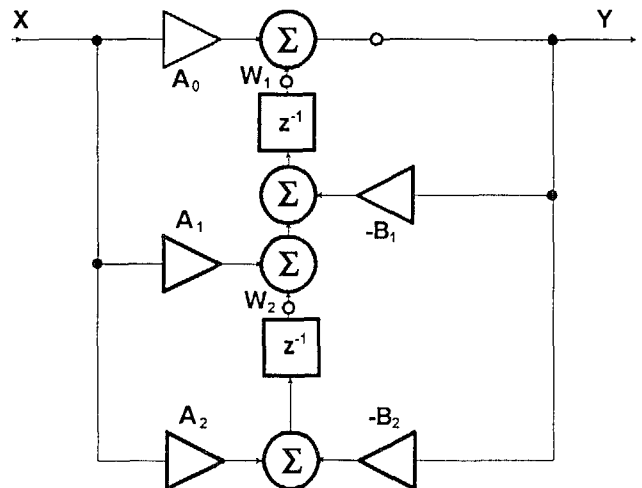


Figure 5 - Structure transposée pour filtre IIR

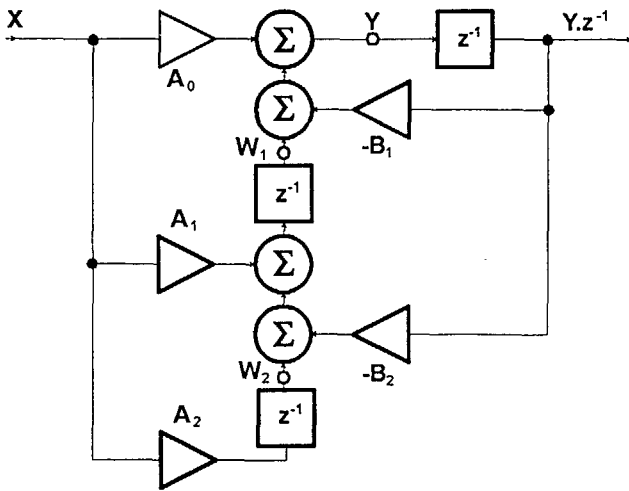


Figure 6 - Structure transposée modifiée pour filtre IIR

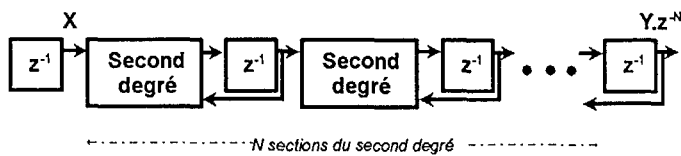


Figure 7 - Implantation cascade avec la structure transposée modifiée

L'optimisation prend en compte que chaque coefficient $a_{j,m}$ nul conduit à un additionneur recevant constamment "0" sur l'une de ses entrées et se comportant dès lors comme un élément transfert comme le montre la figure 8. Les coefficients $a_{j,m}$ égaux à -1 ne nécessitent que la complémentation du multiplicande avec introduction d'un report initialement à "1" (préchargement du bistable de report).

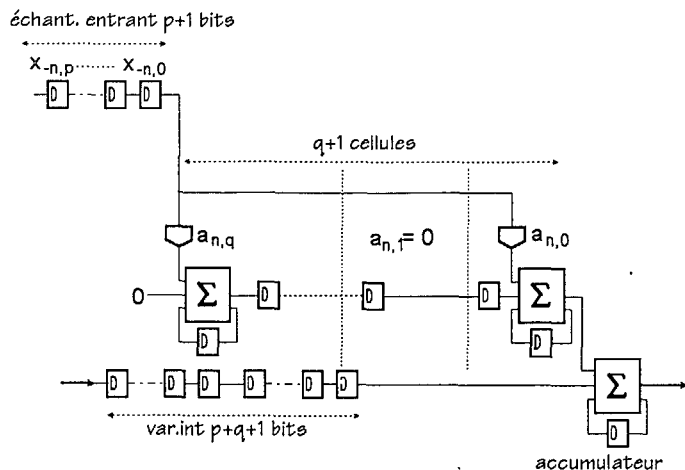


Figure 8 - Structure simplifiée pour les bits de coefficients valant 0

5. BILAN D'IMPLANTATION

La figure 9 représente la structure de la cellule de base pour les structures envisagées. On y note qu'il faut considérer des fonctions dépendantes de 4 entrées au maximum ce qui correspond bien aux cellules des composants les plus simples.

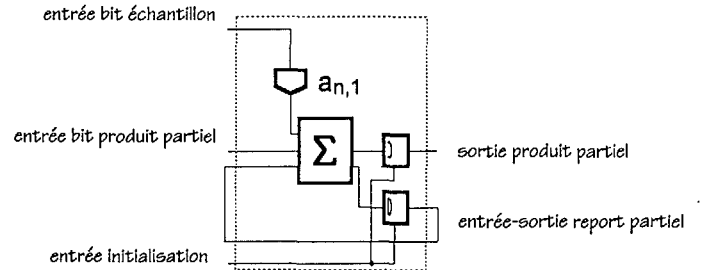


Figure 9 - Fonctionnalités implantées dans les cellules de base

Désignant respectivement par:

- N , l'ordre du filtre
- k , le pourcentage de bits des coefficients différents de 0 (égaux à 1 ou -1)
- τ , le délai moyen par cellule de base

on peut établir le tableau général des coûts d'implantation de la figure 10.

On notera qu'une modification mineure de la structure de ces cellules de base permet d'implanter une arithmétique "à troncation du module" évitant l'existence de cycles limites.

Implantation	Coût en cellules	Période T_e
transposée FIR	$3N(q+1)+N(p+2)+(p+1)$	$(p+q+2)\tau$
transposée IIR	$4N(q+1)+N(p+5)+2(p+1)$	$2(p+q+2)\tau$
transposée optimisée FIR	$(2+k/100)N(q+1)+N(p+2)+(p+1)$	$(p+q+2)\tau$
transposée optimisée IIR	$(3+k/100)N(q+1)+N(p+5)+2(p+1)$	$2(p+q+2)\tau$

Figure 10

Un gain additionnel peut être envisagé en considérant une étude de bruit de calcul limitant les représentations des variables internes ce qui compliquera le séquencement de l'ensemble.

6. OUTILS DE DEVELOPPEMENT ASSOCIES

Dans le cadre des réalisations considérées ci-dessus, il est clair que pour optimiser le coût des implantations, il faut minimiser le nombre de bits non nuls dans la représentation de chaque coefficient des filtres FIR ou IIR.

La solution la plus simple mais non optimale consiste à calculer dans chaque cas le nombre de bits nécessaires pour représenter la partie fractionnaire des coefficients (grâce à une étude de



sensibilité [1]) et ensuite à les coder individuellement avec des bits à 0, 1 ou -1.

A l'opposé, la solution minimale sera obtenue par une minimisation du nombre de bits non nuls dans un espace discret des paramètres. Cette solution est trop complexe pour pouvoir être envisagée et une solution intermédiaire va être développée pour chaque type de filtre.

Dans le cas des filtres FIR, comme chaque coefficient y présente une sensibilité comparable [1], une procédure itérative consiste à quantifier les coefficients un à la fois et à recalculer les coefficients non quantifiés par un algorithme de type programmation linéaire qui, comme le problème est dans ce cas linéaire, converge très rapidement [2].

Dans le cas des filtres IIR, la sensibilité des coefficients d'une structure cascade est bien connue [1] et on commence donc par quantifier les uns après les autres les coefficients du numérateur qui fixent la position des pôles de l'affaiblissement en sachant que l'influence de ces coefficients sur la bande passante du filtre est faible.

On quantifie ensuite les pôles de la fonction de transfert en commençant par les plus sensibles (ceux situés le plus près du cercle unité); chaque fois que l'on quantifie un coefficient de la structure cascade, on recalcule les autres pour obtenir l'approximation optimale des spécifications imposées grâce à un algorithme décrit en [3] et légèrement modifié [2].

Un processus de "routage" automatique basé sur le langage VHDL et prenant en compte une description "langage à transfert de registres" adaptée aux cellules rencontrées termine la phase d'implantation, la base en est une bibliothèque reprenant l'implantation d'une cellule de base sur chaque type de composant.

On dispose ainsi d'un outil de C.A.O complet adapté à ces structures de composants.

7. EXEMPLE

Considérons le cas d'un filtre passe-bas dont la fréquence d'échantillonnage est de 10 kHz, la bande passante s'étend de 0 à 2500 Hz avec un affaiblissement maximum de 0.2 dB et qui présente une bande atténuée à partir de 3000 Hz avec un affaiblissement minimum de 60 dB.

Dans le cas des filtres IIR, on trouve un filtre de degré 7 qui, après une quantification avec 7 bits de partie fractionnaire, vérifie les spécifications avec les coefficients suivants:

A0	A1	A2	B1	B2
1	0.648438	1	-0.085938	0.625
1	0.921875	1	0.085938	0.890625
1	1.53125	1	-0.390625	0.28125
1	1	0	-0.289063	0

• $A_k = 0.0353095$

qui représente 37 bits non nuls (constante multiplicative exclue). Après utilisation de la procédure itérative de

minimisation du nombre de bits non nuls, on obtient les coefficients suivants:

A0	A1	A2	B1	B2
1	0.625	1	-0.140625	0.625
1	0.9375	1	0.0585938	0.8945313
1	1.5	1	-0.40625	0.25
1	1	0	-0.25	0

• $A_k = 0.03668667$

qui représente 29 bits non nuls et respecte les spécifications.

Les mêmes spécifications ont été utilisées pour un filtre FIR qui est de degré 54, demande 97 bits non nuls après quantification utilisant 13 bits pour les parties fractionnaires et 77 bits non nuls après optimisation.

8. CONCLUSIONS

Les décompositions structurelles envisagées s'adaptent parfaitement aux cellules de base rencontrées sur les composants considérés.

Les structures transposées conduisent à un accroissement de coût peu sensible vis-à-vis des structures directes tout en permettant des performances en fréquence d'échantillonnage des plus intéressantes.

L'optimisation des coefficients sur base des facteurs [-1,0,1] pour les puissances binaires permet de réduire sensiblement le coût en cellules de base.

Les algorithmes d'optimisation des coefficients des filtres IIR et FIR permettent de réduire le nombre de bits non nuls de 25 à 30 % en moyenne

9. BIBLIOGRAPHIE

- [1] R. BOITE, H. LEICH, "Les filtres numériques : Analyse et synthèse des filtres unidimensionnels", 3e édition MASSON, 1990,
- [2] H. LEICH, "Optimisation des filtres FIR et IIR dans un espace discret des coefficients", Rapport interne F.P.Ms, 1994
- [3] H. LEICH, "Approximation simultanée de l'amplitude et du temps de propagation de groupe des filtres numériques récursifs", TRAITEMENT DU SIGNAL, Vol 7, n° 16, 1990, pp. 491-498
- [4] S.D. BROWN, R.J. FRANCIS, J. ROSE, Z. G. VRANESIC, "Field-Programmable Gate Arrays", KLUWER ACADEMIC PUBLISERS, 1992,
- [5] H. LEICH, J. HANCQ, "Implantation de filtres numériques sur composants logiques programmables et reconfigurables", Actes du Workshop "Adéquation Algorithmes Architectures" Grenoble, janvier 1994, pp.127-134
- [6] Documents constructeurs XiLinX (XC2000, XC3000, XC4000), Altera (FLEX8000), Plessey, Concurrent Logic, Algotronix