

CHAPITRE 2

SPÉCIFICATIONS ET CHOIX DE L'ARCHITECTURE

2.1 Introduction

La compréhension des besoins du client, en termes des spécifications du noyau programmable, est très importante pour un concepteur de circuits logiques. Cette compréhension lui permet de choisir l'architecture la plus apte à répondre aux spécifications désirées. Ainsi, dans ce chapitre, les différentes spécifications auxquelles doit répondre le noyau programmable et une comparaison des architectures matérielles sont présentées.

2.2 Spécifications

Cette section expose les spécifications que le module du noyau programmable doit remplir, notamment au niveau des paramètres programmables et de la reconfigurabilité. Le module conçu doit être configurable c'est-à-dire conçu pour résoudre un problème général, portable c'est-à-dire conçu pour différentes technologies et pour différents outils, déboguable c'est-à-dire vérifié avec un haut niveau de confiance et finalement lisible c'est-à-dire documenté clairement au niveau des applications et restrictions.

2.2.1 Langage et technologie pour l'implémentation

Le noyau doit être programmé en langage de type HDL (Hardware Description Language) pour faciliter la compréhension et permettre l'emploi de variables de configuration. Le langage de programmation choisi pour la réalisation du noyau programmable est le VHDL (Very High Speed Integrated Circuit Hardware Description Language) avec un code entièrement synthétisable. Tous les paramètres de

configurations du noyau sont programmables avant synthèse à l'exception du facteur de décimation qui peut changer de valeur après la synthèse. En ce qui concerne la technologie, le noyau est conçu pour cibler les plates-formes de type FPGA (Field Programmable Gate Array).

2.2.2 Ressources et vitesse

Le noyau doit être conçu efficace avec une complexité acceptable. Pour ce faire, les ressources matérielles limitées telles que la logique de calcul et de contrôle ainsi que la mémoire doivent être minimisées. En ce qui concerne la fréquence d'opération, elle doit être maximisée parce que l'objectif principal est de pouvoir filtrer un signal échantillonné à une fréquence de 1.2 GHz. La latence, délai en cycles d'horloge entre l'entrée du premier échantillon et la première sortie du filtre, est négligée au détriment de la fréquence d'opération. Ceci veut dire que la priorité est accordée à la fréquence d'opération et même si l'augmentation de cette dernière entraîne une augmentation de la latence.

2.2.3 Paramètres fixes

Le module du filtre reçoit ses données d'entrée de façon parallèle. La sortie est produite de manière sérielle. Le module doit posséder un port d'entrée et un port de sortie, chacun ayant une largeur spécifiée par l'utilisateur. Le module a aussi une entrée pour l'horloge *clk* et une pour un signal d'activation de l'horloge *nd*. Il doit aussi posséder un mécanisme pour la synchronisation avec des modules extérieurs, tel qu'un signal pour lui indiquer le début d'une séquence et un signal pour informer que les résultats sont valides.

2.2.4 Paramètres programmables

Comme le but est de réaliser un noyau de filtre à décimation parallélisée qui est configurable, portable, déboguable et lisible, alors tous ses paramètres doivent être programmables. Les différents paramètres sont énumérés et détaillés dans les sous-sections suivantes.

2.2.4.1 Ordre du filtre

L'ordre du filtre dépend entièrement des spécifications requises. Ainsi, il faut établir une plage d'ordre du filtre qui permet de couvrir plusieurs applications. De ce fait, l'ordre minimal du filtre permet de connaître la limite inférieure et l'ordre maximal la limite supérieure. Un bon choix de l'ordre permet de minimiser les ressources logiques ce qui pourrait minimiser la complexité.

2.2.4.2 Largeur des mots binaires

Les ressources logiques utilisées par le noyau ainsi que la précision des calculs dépendent entièrement de la largeur des mots binaires. De ce fait, pour avoir une complexité acceptable et une fréquence d'opération efficace, un bon choix de largeurs des mots binaires est nécessaire.

La largeur des ports en entrée comme en sortie doit pouvoir être choisie par l'utilisateur comme bon lui semble. Cependant, vu que le noyau doit pouvoir filtrer un signal échantillonné à 1.2 GHz, la largeur maximale des mots binaires à l'entrée ne peut pas dépasser 16 bits. Ceci est dû au fait que pour l'instant et pour quelques années encore, aucun convertisseur analogique numérique ne peut fournir en sortie plus que 16 bits avec une telle fréquence d'échantillonnage. Ainsi la largeur des mots binaires à l'entrée est programmable entre 8 et 16 bits.

Dans les filtres à décimation, la largeur des mots binaires à la sortie est souvent plus grande que celle à l'entrée ce qui permet d'éviter les débordements à l'intérieur du filtre. Cependant, la largeur à la sortie est tellement grande qu'il faut la tronquer ou l'arrondir ce qui influence beaucoup la précision des résultats. En regardant les besoins de plusieurs applications et pour pouvoir toutes les couvrir, la largeur des mots binaires à la sortie doit être programmable entre 16 et 24 bits.

2.2.4.3 Largeur des bus de données internes

Les largeurs des différents bus de données internes doivent être définies comme paramètres de manière à conserver ou à améliorer la précision des résultats. Pour ce faire, des mécanismes d'augmentation de la largeur des mots binaires pour empêcher le débordement et/ou pour réduire la largeur, tels que l'arrondi ou la troncation, doivent être employés. Ainsi il serait intéressant de créer des fonctions qui réalisent ces mécanismes.

2.2.4.4 Facteur de décimation

Le facteur de décimation est principalement constitué d'un simple compteur qui permet de réduire la fréquence d'échantillonnage. Ce qui veut dire que la fréquence à la sortie du noyau est donnée par la fréquence d'échantillonnage divisée par le facteur de décimation. Une attention particulière mérite d'être portée au facteur de décimation car ce dernier est strictement lié à la largeur des bus à l'intérieur du filtre. Une augmentation du facteur de décimation entraîne une augmentation de la largeur des bus pour prévenir le débordement ce qui entraîne à son tour une augmentation des ressources logiques et une diminution de la fréquence d'opération. Pour couvrir une bonne partie des applications, le facteur de décimation doit être programmable entre 4 et 1024. Il serait aussi attrayant si le facteur de décimation est reconfigurable ce qui veut dire qu'il peut être programmable après synthèse. Pour ce faire, il faut utiliser une architecture

compacte c'est-à-dire une architecture dont la variation du facteur de décimation n'entraîne aucun changement au niveau de l'architecture elle-même. Ceci est faisable en réservant un port d'entrée pour le chargement du facteur de décimation après synthèse.

2.2.4.5 Type de facteur de décimation

Le noyau doit accepter deux types de facteur de décimation qui sont : fixe ou programmable. Une décimation est dite fixe si le facteur de décimation est choisi seulement avant synthèse. Une décimation est dite programmable si le facteur de décimation est modifiable avant et après synthèse. Donc le noyau doit offrir à l'utilisateur le choix d'un facteur de décimation fixe ou programmable. Si l'utilisateur choisit un facteur de décimation programmable, alors il doit spécifier sa valeur maximale pour pouvoir déterminer la largeur maximale des bus internes qui permet d'éviter le débordement.

2.3 Comparaison des architectures matérielles

Dans le cadre de la réalisation d'un noyau programmable avec une complexité réduite, le choix d'une bonne architecture peut faire une très grande différence au niveau de l'implémentation. Les spécifications requises doivent guider le concepteur à faire un bon choix. Ainsi, trois techniques sont étudiées. La première technique consiste à utiliser des filtres numériques RIF. La deuxième technique utilise des filtres numériques RII. La troisième technique, quand à elle, utilise des filtres Cascaded Integrated and Comb (CIC). Avec ces trois techniques, plusieurs structures sont analysées et évaluées. Cependant, il faut noter que ces différentes structures sont comparées seulement au niveau de la réalisation matérielle. Ainsi, après comparaison, on pourra faire le choix de l'architecture qui répond le mieux aux spécifications.

2.3.1 Filtres numériques à réponse impulsionnelle finie (RIF) à décimation

Cette section présente la théorie des filtres RIF à décimation ainsi que leurs structures de réalisations.

2.3.1.1 Théorie des filtres RIF

Un filtre RIF à coefficients constants est un système linéaire invariant dans le temps. La sortie d'un filtre FIR d'ordre L , quand la série $x[n]$ est présentée à son entrée, est donnée par l'équation (2.1) :

$$y[n] = x[n] \otimes f[n] = \sum_{k=0}^{L-1} f[k]x[n-k] \quad (2.1)$$

avec $f[k]$, k allant de 0 à $L-1$, les L coefficients du filtre. Ces L coefficients correspondent à la réponse impulsionnelle du filtre RIF. La transformée en z de l'équation (2.1) donne :

$$Y(z) = F(z)X(z) \quad (2.2)$$

$F(z)$ représente la fonction de transfert du filtre RIF qui est donnée par :

$$F(z) = \sum_{k=0}^{L-1} f[k]z^{-k} \quad (2.3)$$

La structure de forme directe d'un filtre RIF d'ordre L est présentée à la figure 7. Elle est constituée de délais, d'additionneurs et de multiplicateurs.

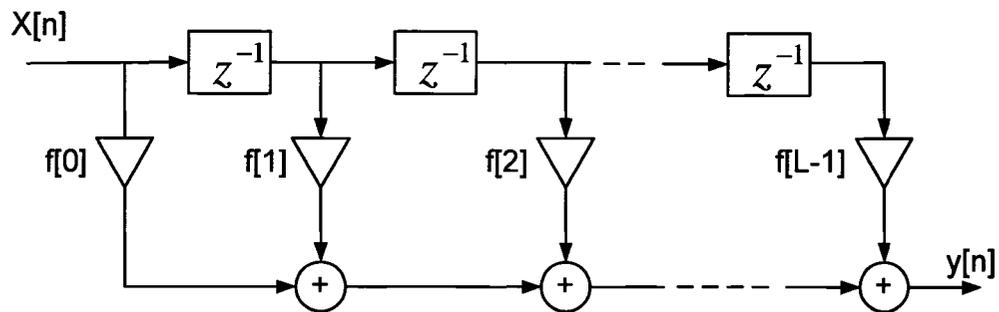


Figure 7 Structure de forme directe d'un filtre RIF d'ordre L

Cette structure de forme directe peut subir une certaine variation pour donner la structure transposée présentée à la figure 8. La structure transposée est, en général, la structure d'implémentation préférée pour les filtres RIF. Les avantages de cette structure résident dans le fait que les entrées n'ont plus besoin de registres supplémentaires pour leur décalage. Cette structure n'a pas besoin non plus d'étages de pipeline supplémentaires pour les additionneurs pour augmenter la fréquence de fonctionnement [16].

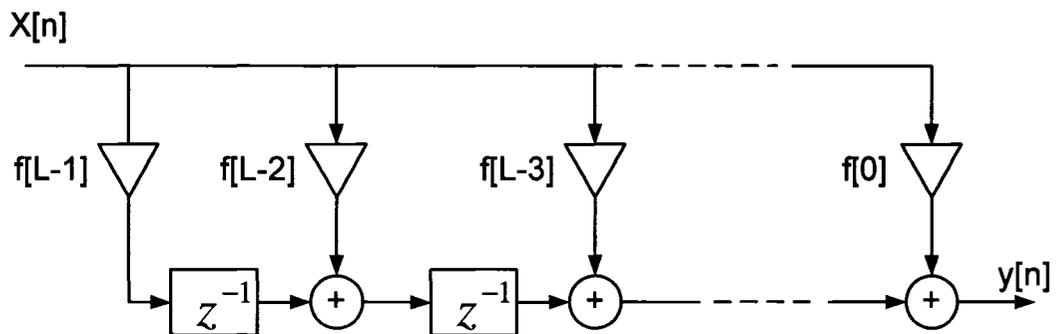


Figure 8 Structure transposée d'un filtre RIF d'ordre L

Le centre de la réponse impulsionnelle d'un filtre RIF est un point important de symétrie. Si ce point est considéré comme le point 0, alors la fonction de transfert s'écrit :

$$F(z) = \sum_{k=\frac{-(L-1)}{2}}^{\frac{(L-1)}{2}} f[k]z^{-k} \quad (2.4)$$

La réponse en fréquence est donnée en posant $z = e^{j\omega T}$ dans l'équation (2.4) :

$$F(\omega) = F(e^{j\omega T}) = \sum_k f[k]e^{-j\omega k T} \quad (2.5)$$

La réponse en phase du filtre est donnée par :

$$\Phi(\omega) = \arctan\left(\frac{\Im(F(\omega))}{\Re(F(\omega))}\right) \quad (2.6)$$

Maintenir l'intégrité de la phase sur une plage de fréquence est très désirée dans beaucoup d'applications comme en communication ou en traitement de l'image. Ceci veut dire que le filtre conçu doit avoir une réponse en phase linéaire. Cette linéarité est mesurée par le délai de groupe défini comme suit :

$$\tau(\omega) = \frac{d\Phi(\omega)}{d\omega} \quad (2.7)$$

Un filtre RIF a une réponse en phase linéaire sur une plage de fréquence si le délai de groupe est constant sur cette plage de fréquence. Le délai de groupe d'un filtre RIF est constant si et seulement si la réponse en fréquence $F(\omega)$ est purement réelle ou purement imaginaire. Ceci veut dire que la réponse impulsionnelle doit être symétrique de façon paire ou impaire, c'est-à-dire :

$$f[n] = f[-n] \quad \text{ou} \quad f[n] = -f[-n] \quad (2.8)$$

Cette propriété de symétrie peut être utilisée pour réduire le nombre de multiplicateurs dans la structure directe de L à $\frac{L}{2}$.

2.3.1.2 Structure de réalisation des filtres RIF à décimation

La décomposition polyphasée du filtre RIF ou RII est très importante dans l'implémentation d'un filtre à décimation ou à interpolation. Prenons, comme exemple, la décomposition polyphasée de la structure du filtre FIR présentée à la figure 1. Le filtre FIR est suivi d'un facteur de décimation R . Ceci veut dire qu'une sortie sur R est choisie à la sortie du filtre. Donc, il n'est pas nécessaire de calculer toutes les sommes de produits $x[n]f[n-k]$ de la convolution. Pour éviter ces calculs inutiles, les identités de Noble, présentées à la figure 9, sont utilisées [16].

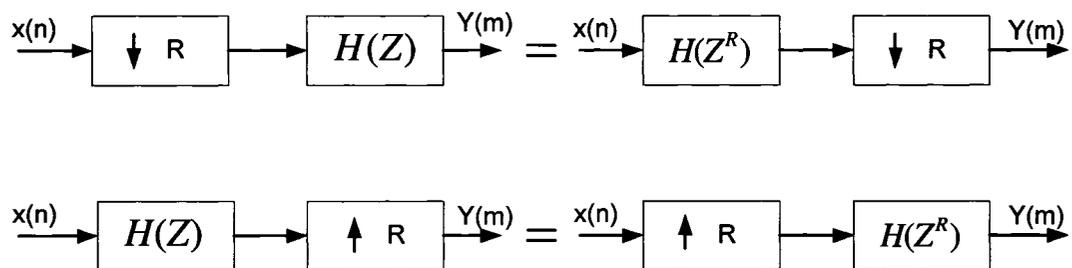


Figure 9 Identities of Noble

Ces identités montrent que $x[0]$ est multiplié par $f[0]$, $f[R]$, $f[2R]$, seulement. Les coefficients du filtre sont aussi multipliés par $x[R]$, $x[2R]$, Seulement, d'où l'intérêt de diviser la séquence d'entrée en R séquences comme suit :

$$x_0[n] = \{x[0], x[R], \dots\}$$

$$x_1[n] = \{x[1], x[R+1], \dots\}$$

$$x_{R-1}[n] = \{x[R-1], x[2R-1], \dots\} \quad (2.9)$$

et le filtre $f[n]$ en R séquences comme suit :

$$f[n] = \sum_{r=0}^{R-1} f_r[n]$$

$$f_0[n] = \{f[0], f[R], \dots\}$$

$$f_1[n] = \{f[1], f[R+1], \dots\}$$

$$f_{R-1}[n] = \{f[R-1], f[2R-1], \dots\} \quad (2.10)$$

La figure 10 montre l'architecture polyphasée d'un filtre FIR décimateur. Cette architecture est R fois plus rapide que celle du FIR suivi du décimateur.

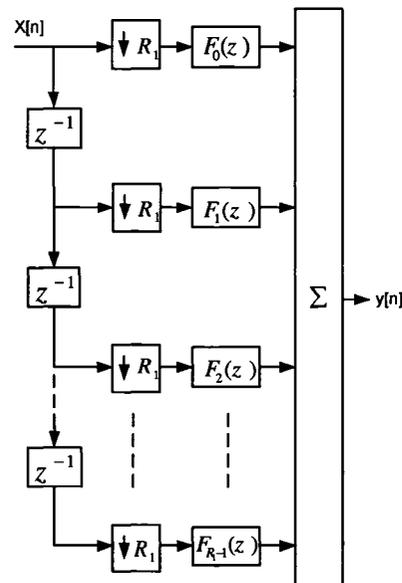


Figure 10 Architecture polyphasée d'un filtre RIF avec un facteur de décimation R_1

Comme indiqué dans le chapitre 1, tout processus de décimation est précédé d'un filtre passe bas avec une fréquence de coupure égale à $\frac{\pi}{R}$ pour limiter la largeur de bande du signal. Ainsi, si le facteur de décimation est grand, alors la largeur de bande du filtre est étroite. Sa bande de transition est préférablement étroite aussi. Ceci nécessite un filtre RIF de grand ordre. L'équation aux différences d'un filtre RIF d'ordre L avec des coefficients symétriques autour de $\frac{L}{2}$, est donnée par [17] :

$$y(n) = f\left(\frac{L}{2}\right)x\left(n - \frac{L}{2}\right) + \sum_{k=0}^{\frac{L}{2}-1} f[k] \{x(n-k) + x(n-L+k)\} \quad (2.11)$$

Cette équation montre que le filtre a besoin de stocker $\frac{L}{2} + 1$ coefficients, d'effectuer $\frac{L}{2} + 1$ multiplications et L additions par sortie. Donc, un filtre RIF de grand ordre entraîne un grand nombre de multiplications. Malgré le gain R en vitesse par rapport à la structure de forme directe, la décomposition polyphasée du filtre RIF est très limitée pour des applications qui utilisent de hautes fréquences d'échantillonnage.

2.3.2 Filtres numériques à réponse impulsionnelle infinie (RII) à décimation

Comparés aux filtres numériques à réponse impulsionnelle finie (RIF), les filtres RII ayant le même ordre offrent des performances beaucoup plus intéressantes. Ceci est dû au fait que les filtres RII ont des boucles de rétroaction de la sortie sur l'entrée. Ces boucles leur permettent de réaliser les zéros et les pôles de la fonction de transfert d'où la réalisation de fonctions de filtrage beaucoup plus sélectives que les RIF. Les avantages des filtres RII sont : leur design standard en utilisant les prototypes de filtres analogiques est bien compris; des fonctions de filtrage très sélectives peuvent être

réalisées avec des filtres de petits ordres ce qui permet de réduire la complexité. Pour les mêmes spécifications, les filtres RII requièrent moins de calculs comparés aux filtres RIF. Cependant, les filtres RII présentent des caractéristiques indésirables qui sont : la non linéarité de la réponse en phase c'est-à-dire qu'il est difficile d'obtenir une réponse en phase linéaire. Un filtre passe tout pourrait être utilisé pour compenser, ce qui contribuerait à l'augmentation de la complexité. Les boucles de rétroaction peuvent non seulement introduire de l'instabilité mais rendent difficiles les techniques pipelines qui permettent d'augmenter la fréquence de fonctionnement. Ainsi, pour mieux comprendre les tenants et les aboutissants de ces filtres, leur théorie, leur implémentation et des architectures destinées à fonctionner en haute vitesse seront présentés dans cette section.

2.3.2.1 Théories des filtres RII

Les filtres RII sont des filtres récursifs c'est-à-dire qu'ils ont des boucles de rétroaction ce qui fait qu'ils ont une réponse impulsionnelle infinie. La figure 11 montre la forme canonique d'un filtre RII.

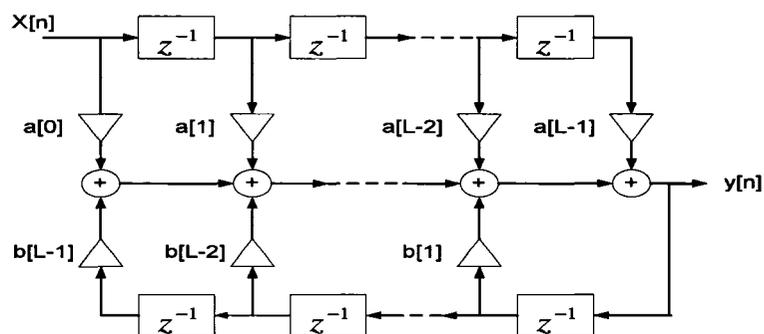


Figure 11 Forme canonique du RII

La fonction de transfert de la forme canonique du RII est donnée :

$$F(z) = \frac{\sum_{l=0}^{L-1} a[l]z^{-l}}{1 - \sum_{l=1}^{L-1} b[l]z^{-l}} \quad (2.12)$$

L'équation aux différences, quand à elle, est donnée par :

$$y(n) = \sum_{l=0}^{L-1} a[l]x[n-l] + \sum_{l=1}^{L-1} b[l]y[n-l] \quad (2.13)$$

L'équation aux différences montre que la sortie d'un filtre RII dépend non seulement des L entrées précédentes mais aussi des $L-1$ sorties précédentes. Pour étudier les propriétés les plus importantes des filtres RII, écrivons la fonction de transfert en fonction des pôles $p_{\infty l}$ et des zéros p_{0l} et en remplaçant z par $e^{j\omega T}$.

$$F(\omega) = |F(\omega)|e^{\theta(\omega)} = \frac{\prod_{l=0}^{L-2} (p_{0l} - e^{j\omega T})}{\prod_{l=0}^{L-2} (p_{\infty l} - e^{j\omega T})} \quad (2.14)$$

En analysant l'équation (2.14), nous pouvons déduire les propriétés suivantes :

Un zéro sur le cercle unité, c'est-à-dire $p_0 = e^{j\omega_0 T}$ produit un zéro dans la fonction de transfert à la fréquence ω_0 .

Un pôle sur le cercle unité, c'est-à-dire $p_{\infty} = e^{j\omega_0 T}$, produit un gain infini dans la fonction de transfert à la fréquence ω_0 .

Pour avoir une phase linéaire, c'est-à-dire un délai de groupe constant, tous les pôles et les zéros doivent être symétriques au cercle unité donc à $z = 0$.

Donc, en résumé, pour qu'un filtre RII soit stable et ait une réponse en phase linéaire, il faut que tous les zéros soient symétriques au cercle unité et tous les pôles à $z = 0$. Cependant, en réalité, un filtre RII a des zéros et des pôles simples sur le cercle unité. Ceci fait que les filtres RII ne peuvent avoir que des réponses en phase approximativement linéaire. Pour ce faire, un filtre passe tout est utilisé. Ce dernier a un gain unité et un gain en phase non nul qui est utilisé pour assurer la linéarisation dans la bande passante du filtre.

2.3.2.2 Méthode de calcul des coefficients

Habituellement, le calcul des coefficients des filtres RII n'est pas effectué d'une manière directe. Il se fait en utilisant une fonction modèle (gabarit). Celle-ci est une fonction réelle définie sur l'axe des fréquences. Les fonctions modèles, telles que les fonctions de Butterworth, Tchebychev et les fonctions elliptiques, sont des fonctions connues pour leurs propriétés de sélectivité.

Ces fonctions modèles peuvent être choisies comme étant le carré du module de la fonction de transfert à obtenir. Cependant, un obstacle apparaît à leur utilisation au niveau du calcul des coefficients des filtres numériques. En fait, ces fonctions modèles ne sont pas périodiques alors que les fonctions à obtenir ont la période f_e . Pour remédier à cela, il faut établir une correspondance entre l'axe réel et $[0, f_e]$. Une telle correspondance est fournie par une transformation conforme dans le plan complexe et qui respecte les contraintes suivantes :

- Transformer une fraction rationnelle de la variable complexe s en une fraction rationnelle de la variable complexe z .
- Conserver la stabilité

Une transformation simple qui utilise la règle trapézoïdale pour faire l'intégration d'une fonction temporelle continue, connue sous le nom de transformation bilinéaire, est obtenue en faisant correspondre un élément du plan s à un élément du plan z en respectant la relation suivante [16] :

$$s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (2.15)$$

Trois types de fonctions modèles, à savoir Butterworth, Chebychev et elliptiques, sont très intéressants à investiguer :

La fonction de Butterworth d'ordre N est donnée par [18] :

$$|F(w)|^2 = \frac{1}{1 + \left(\frac{w}{w_s}\right)^{2N}} \quad (2.16)$$

L'ordre N du filtre est déterminé à partir de son gabarit. Soit à réaliser un filtre dont la réponse en fréquence est supérieure ou égale à $1 - \delta_1$ dans la bande $[0, f_1]$ et inférieure ou égale à δ_2 dans la bande $\left[f_2, \frac{f_e}{2}\right]$. Ainsi N a pour expression [19] :

$$N \geq \frac{\log\left(\frac{1}{\delta_2 \sqrt{2\delta_1}}\right)}{\log(\operatorname{tg}(\pi f_2 T)) - \log(\operatorname{tg}(\pi f_1 T))} \quad (2.17)$$

Cette fonction de transfert, donnée par l'équation 2.16, est N fois dérivable à $w = 0$ ce qui la rend douce autour de 0 Hz.

Les fonctions de Chebychev de type I ou II sont déduites du polynôme de Chebychev $V_N(w) = \cos(N \cos(w))$. Ce polynôme force les pôles du filtre à être sur une ellipse. Ainsi les fonctions de Chebychev de type I et II sont données respectivement par [18] :

$$|F(w)|^2 = \frac{1}{1 + \varepsilon^2 V_N^2\left(\frac{w}{ws}\right)} \quad (2.18)$$

$$|F(w)|^2 = \frac{1}{1 + \left(\varepsilon^2 V_N^2\left(\frac{w}{ws}\right)^{-1}\right)} \quad (2.19)$$

La fonction elliptique, quand à elle, est définie selon les termes de la solution de la fonction elliptique jacobéenne, $U_N(w)$. Elle est donnée par [18] :

$$|F(w)|^2 = \frac{1}{1 + \varepsilon^2 U_N^2\left(\frac{w}{ws}\right)^{-1}} \quad (2.20)$$

L'ordre N du filtre numérique correspondant se calcule aussi à partir de son gabarit et est donné par [19] :

$$N \cong 1.076 \log\left(\frac{2}{\delta_2 \sqrt{\delta_1}}\right) \log\left(\frac{f_e}{\Delta f} \frac{4}{\pi} \sin\left(2\pi \frac{f_1}{f_e}\right)\right) \quad (2.21)$$

où δ_1 représente l'ondulation dans la bande passante, δ_2 l'ondulation dans la bande coupée, f_1 la fréquence de coupure, f_2 la fréquence de début de bande affaiblie, $\Delta f = f_1 - f_2$ la bande de transition et f_e la fréquence d'échantillonnage.

En se basant sur ces différentes fonctions modèles, les conclusions suivantes peuvent être tirées selon les spécifications données :

Tableau I
Comparaison des différentes fonctions modèles

	Ordre du filtre	Caractéristiques De la bande passante	Caractéristiques De la bande coupée	Caractéristiques De la bande de transition
Petit	Elliptique			
Moyen	Chebychev			
Grand	Butterworth			
Équiripple		Elliptique Chebychev I	Elliptique ChebychevII	
Plate		Butterworth ChebychevII	Butterworth ChebychevI	
Étroite				Elliptique
Moyenne				Chebychev I et II
Large				Butterworth

Puisque c'est la réalisation d'un filtre qui fonctionne à une très grande fréquence d'échantillonnage qui nous intéresse, un aperçu sera donné uniquement sur les structures directes car ces dernières utilisent le minimum d'opérations élémentaires. Ceci les rend plus rapides comparées aux autres structures. Cependant, le prix à payer est la grande sensibilité aux erreurs d'arrondi dans le cadre d'un arithmétique à point fixe. Les structures directes de type I et II sont présentées aux figures 12 et 13.

La structure directe de type I calcule d'abord le dénominateur puis le numérateur. Ainsi elle utilise deux blocs de multiplicateur. Un bloc de multiplicateur pour les coefficients du dénominateur et un autre bloc de multiplicateur pour ceux du numérateur.

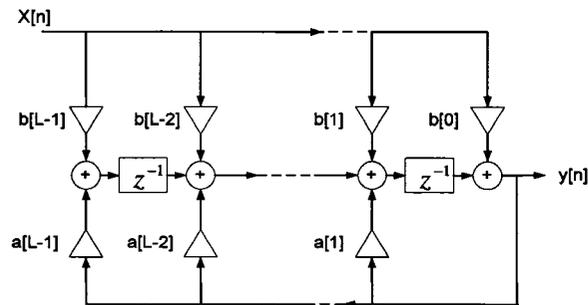


Figure 12 Structure directe de type I

La structure directe de type II n'est autre que la transposée de la structure directe de type I. Cependant, elle présente une particularité intéressante : chaque $y[n]$ ou $x[n]$ est multiplié par tous les coefficients ce qui fait que le numérateur et le dénominateur se calculent simultanément. Ceci simplifie beaucoup la mise en œuvre de la multiplication car au lieu de deux blocs de multiplicateur, cette structure n'en utilise qu'un seul. Cette structure directe de type II est plus rapide que celle de type I mais nécessite beaucoup plus de ressources que cette dernière.

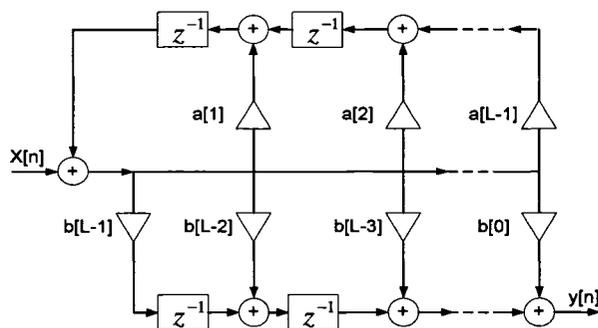


Figure 13 Structure directe de type II

2.3.2.3 Structures de réalisation des filtres RII décimateurs

Dans un processus de décimation, si le facteur de décimation se situe après le filtre passe bas, alors le résultat d'un certain nombre de calculs n'est pas utilisé à la sortie. Pour éviter ces calculs inutiles, on utilise des systèmes qui ne font des calculs que pour les sorties désirées. Les filtres, issus de ces systèmes, ne produisent pas de sortie pour chaque signal d'entrée. Ceci nous incite à croire que la décimation ne peut être effectuée que par des filtres non récursifs. Après tout, un filtre non récursif n'a pas besoin de ses sorties antérieures pour calculer la prochaine sortie, ce qui nous permet d'éliminer certaines sorties. Un filtre récursif, quand à lui, a besoin de ses sorties antérieures pour calculer la prochaine sortie. Donc, un filtre récursif, pour son propre profit, a besoin de toutes ses sorties même si le facteur de décimation ne les utilise pas toutes.

Les filtres récursifs peuvent, en effet, être utilisés comme des filtres décimateurs. La plupart des structures développées pour les filtres non récursifs décimateurs sont applicables aux filtres récursifs décimateurs. Cependant, le fait d'avoir accès directement à la réponse impulsionnelle des filtres non récursifs facilite leur décomposition polyphasée. Pour les filtres récursifs, la décomposition polyphasée est possible en utilisant d'autres techniques car l'accès à la réponse impulsionnelle n'est pas direct.

Martinez et Parks [20] ont proposé une classe de filtres récursifs à décimation qui regroupe les avantages des filtres RIF et ceux des filtres elliptiques. Ces filtres ont leurs zéros sur le cercle unité, des déviations dans la bande passante et dans la bande coupée. L'idée est d'avoir une fonction de transfert avec un dénominateur qui n'a que des puissances de z^R , R étant le facteur de décimation. Cette fonction de transfert est donnée par :

$$F(z) = \frac{\sum_{k=0}^L a[k]z^{-k}}{1 + \sum_{k=1}^K b[k]z^{-R}} = \frac{N(z)}{D(z)} \quad (2.22)$$

Dans la bande passante $|F(z)|$ oscillera autour de 1 au moins $k+1$ fois. Dans la bande coupée, $|F(z)|$ oscillera autour de $\delta_s/2$ au moins $L+1$ fois. Quand la fonction de transfert de l'équation (2.22) est implémentée avec une structure de forme directe, la sortie $y(n)$ peut s'écrire comme suit :

$$y(n) = -\sum_{k=1}^K b(k)y(n-kR) + a(L/2)x(n-L/2) + \sum_{k=0}^{L/2-1} a(k)[x(n-k) + x(n-L+K)] \quad (2.23)$$

L'équation (2.23) montre que cette structure nécessite $(K+1+L/2)$ multiplications par sortie. Mais, comme une sortie sur R est calculée, alors le nombre de multiplications par seconde est de $(K+1+L/2)/R$ mult/s. Le taux d'addition est de $(K+L)/R$ adds/s. Le nombre de coefficients à stocker est de $(K+1+L/2)$.

La question, maintenant, est de savoir comment concevoir un filtre récursif avec un dénominateur ne contenant que des puissances de z^R et comment déterminer leur ordre. Dans [21], un algorithme itératif pour le design de filtres récursifs à rebonds égalisés (*equiripple filter*) avec tous ses zéros sur le cercle unité est présenté. Il fonctionne comme suit : si δ_p (l'ondulation dans la bande passante), f_p (La fréquence de coupure), f_c (début de la bande coupée) et l'ordre du filtre sont donnés, alors la valeur de δ_s (l'ondulation dans la bande coupée) est choisie de telle sorte que le filtre ait le comportement d'un filtre à rebonds égalisés. Un filtre récursif avec un dénominateur ne

contenant que des puissances de z^R peut être conçu en utilisant cet algorithme. La seule différence dans l'algorithme est que le module au carré du dénominateur est de la forme :

$$D(e^{j2\pi f})D(e^{-j2\pi f}) = \sum_{k=0}^K \tilde{b}(k) \cos(2\pi Rkf) \quad (2.24)$$

Si le numérateur $N(e^{j2\pi f})$ contrôle l'ondulation dans la bande coupée et est égale à 1 à $f = 0$, alors une valeur $M_1 \approx L/2$ peut être donnée par :

$$M_1 = \frac{\cosh^{-1}(1/\delta_s)}{\cosh^{-1}(X_0)} \quad (2.25)$$

avec $X_0 = (3 - \cos 2\pi f_c)/(1 + \cos 2\pi f_c)$. Si le numérateur $N(e^{j2\pi f})$ contrôle l'ondulation dans la bande coupée mais a la valeur 1 à $f = f_p$, alors une valeur $M_2 \approx L/2$ peut être donnée par :

$$M_2 = \frac{\cosh^{-1}(1/\delta_s)}{\cosh^{-1}(X_p)} \quad (2.26)$$

avec

$$X_p = \left[\frac{X_0 + 1}{2} \right] \cos(2\pi f_p) + \left[\frac{X_0 - 1}{2} \right]$$

Une approximation de L , l'ordre du numérateur, peut être obtenue comme suit :

$$L_0 = M_1 + M_2 \quad (2.27)$$

Pour ce qui concerne l'ordre K du dénominateur, il doit être choisi de façon à avoir un grand nombre de pôles pour respecter l'ondulation désirée dans la bande passante. Un bon point de départ est de choisir $L = L_0$ et $K = 2$ ou $K = 4$ et de jouer avec les valeurs de L et K pour contrôler respectivement l'ondulation dans la bande arrêtée et celle dans la bande passante. Le tableau II présente, pour les mêmes spécifications, une comparaison entre un filtre FIR décimateur, un filtre elliptique décimateur et le filtre RII avec dénominateur à puissance de z^R .

Tableau II
 Comparaison entre FIR décimateur, elliptique décimateur
 et RII avec dénominateur à puissance de z^R

	Facteur de décimation	Ordre du filtre	Multiplication par seconde	Nombre de coefficients
FIR (un étage de décimation)	$D = 20$	$M = 652$	16.4	327
Elliptique (un étage de décimation)	$D = 20$	$N = 7$	11	11
RII à puissance z^R (un étage de décimation)	$D = 20$	$L = 116$ $K = 4$	3.15	63
FIR (deux étages de décimation)	$D_1 = 10$ $D_2 = 2$	$M_1 = 34$ $M_2 = 72$	3.65	55
Elliptique (deux étages de décimation)	$D_1 = 10$ $D_2 = 2$	$N_1 = 3$ $N_2 = 7$	6.1	16
RII à puissance z^R (deux étages de décimation)	$D_1 = 10$ $D_2 = 2$	$L_1 = 26$ $K_1 = 1$ $L_2 = 10$ $K_2 = 6$	2.1	27

Le tableau II montre le gain du filtre RII avec dénominateur à puissance de z^R par rapport aux autres structures. Donc il est le plus intéressant pour des applications à hautes fréquences d'échantillonnages. Cependant, sa structure telle que définie, ne nous permet pas de filtrer un signal de l'ordre des GHz. La question qui se pose à ce stade est : est-ce possible d'utiliser la technique pipeline dans la structure du RII avec dénominateur à puissance de z^R . En fait, dans n'importe quelle architecture de filtre, la section en amont peut être en pipeline pour augmenter son débit en rajoutant des registres ce qui entraîne une augmentation de la latence. La technique pipeline ne peut pas, en général, être utilisée dans la boucle de rétroaction parcequ'elle change le délai de cette dernière. Ceci, en conséquence, change la fonctionnalité de la boucle de rétroaction. 'Scattered lookahead' est une technique qui transforme un filtre à pôle avec une boucle de rétroaction de délai K en un filtre à pôle et zéro avec une boucle de rétroaction dont le délai est un multiple de K . Ceci permet d'utiliser la technique pipeline, pour les additionneurs et les multiplicateurs dans la boucle de rétroaction, par K . La boucle de rétroaction désirée est générée en prenant chaque pôle du filtre original et générer $K-1$ pôles/zéros qui ont la même module que le pôle original et localisés à $z_n = pe^{jm/K}$. Les $K-1$ pôles ajoutés seront annulés par les zéros de la section en amont du filtre. L'exemple suivant illustre bien cette technique.

Soit un filtre à pôles d'ordre 2 avec $p_1 = 0.5$ et $p_2 = 0.25$. La fonction de transfert est donnée par :

$$F(z) = \frac{1}{1 - 3/4z^{-1} + 1/8z^{-2}}$$

La technique 'Scattered lookahead' introduit deux étages de pipelines additionnels en ajoutant une paire de pôles/zéros à $0.25e^{\pm j2\pi/3}$ et $0.5e^{\pm j2\pi/3}$. Cette addition de pôles/zéros à cette location transforme la fonction de transfert comme suit :

$$\begin{aligned}
F(z) &= \frac{1}{1 + 0.5z^{-1} + 0.25z^{-2}} \times \frac{(1 + 0.5z^{-1} + 0.25z^{-2})(1 + 0.75z^{-1} + 0.5625z^{-2})}{(1 + 0.75z^{-1} + 0.5625z^{-2})(1 - 0.75z^{-1} + 0.125z^{-2})} \\
&= \frac{1 + 1.25z^{-1} + 1.1875z^{-2} + 0.4687z^{-3} + 0.1406z^{-4}}{1 - 0.5469z^{-3} + 0.0527z^{-6}} \\
&= \frac{512 + 640z^{-1} + 608z^{-2} + 240z^{-3}}{512 - 280z^{-3} + 27z^{-6}}
\end{aligned}$$

Cette technique permet d'augmenter le débit du filtre mais son point faible est qu'elle augmente la complexité de la section en amont du filtre.

Les filtres RII avec dénominateur à puissance de z^R sont très intéressants mais le prix de leur pipeline et la grande largeur des coefficients les rendent très limités pour des applications à hautes fréquences d'échantillonnage.

2.3.3 Filtres CIC décimateurs

Généralement, dans un filtre numérique, le nombre de multiplication à faire par seconde est exprimé par $M_r = Kf_e$ où f_e représente la cadence à laquelle se font les calculs donc la fréquence d'échantillonnage. Le paramètre K dépend du type de filtre, de son ordre et de ses performances. Dans ce contexte, Eugène B Hogenauer [7] a présenté en 1981 des filtres particulièrement efficaces. Ce sont les filtres Cascaded Integrator and Comb (CIC). L'absence de multiplicateur, le non stockage des coefficients du filtre et la simplicité d'implémentation permettent aux filtres CIC de jouer un rôle très important dans les applications exigeant de hautes fréquences d'échantillonnage comme les communication sans fil où les signaux sont échantillonnés directement en RF ou IF. Dans les applications à bande étroite comme la radio mobile, des taux de décimation de l'ordre de 1000 sont souvent requis. Une autre application des filtres CIC est dans le domaine des convertisseurs $\Sigma\Delta$ [22].

2.3.3.1 Description générale

Les filtres CIC [7] sont des filtres utilisés dans le domaine de traitement multiscalaire du signal. Ils sont utilisés pour augmenter (interpolation) ou diminuer (décimation) le rapport d'échantillonnage. Ils se basent sur le fait qu'une élimination parfaite de pôle/zéro peut être obtenue. Ceci est possible en utilisant seulement une arithmétique exacte comme le complément à deux. Une attention particulière sera accordée aux filtres CIC décimateurs.

La structure de base des filtres CIC décimateurs (CIC d'ordre 1) [7] indiquée à la figure 14 montre que ces filtres sont constitués essentiellement de deux blocs en cascade. Le premier bloc est composé d'un étage d'intégrateur numérique et le second d'un étage de différentiateur. Ces deux blocs sont séparés par un facteur de décimation R . La figure 15 montre une structure équivalente à la structure de base, obtenue à l'aide des identités de Nobles.

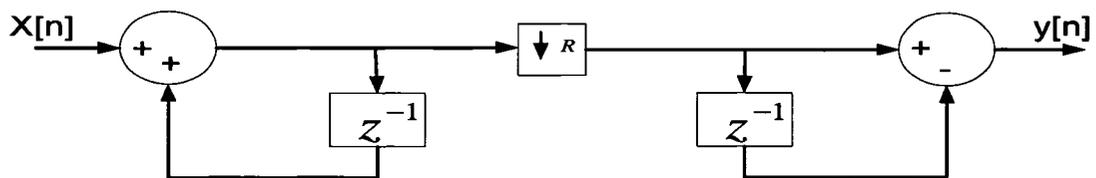


Figure 14 Structure d'un filtre CIC d'ordre 1

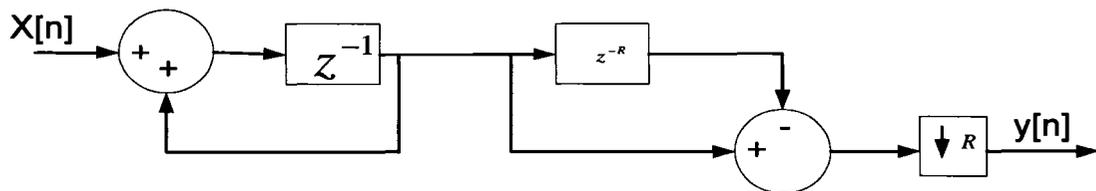


Figure 15 Structure équivalente à la structure d'un filtre CIC d'ordre 1

Un intégrateur est réalisé par un simple filtre à un seul pôle avec une contre réaction unitaire. Les intégrateurs des filtres CIC opèrent à la fréquence d'échantillonnage f_s . Ils peuvent être considérés comme des accumulateurs. La figure 16 représente un simple intégrateur.

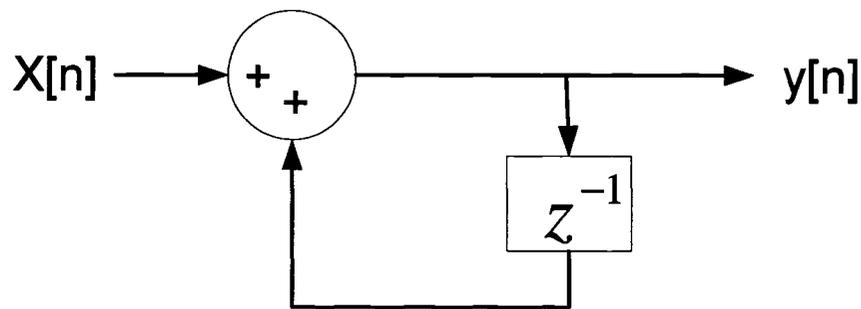


Figure 16 Schéma d'un simple intégrateur

L'équation aux différences de l'intégrateur numérique est donnée par :

$$y(n) = y(n-1) + x(n) \quad (2.28)$$

ce qui donne une fonction de transfert sous la forme suivante :

$$H_I(Z) = \frac{1}{1 - Z^{-1}} \quad (2.29)$$

Les différentiateurs des filtres CIC, quand à eux, fonctionnent à une fréquence beaucoup plus basse f_e et qui est égale à $\frac{f_s}{R}$ avec R le facteur de décimation. Les différentiateurs utilisent un paramètre de délai différentiel noté M et qui sert à contrôler la position des zéros dans la réponse fréquentielle du filtre. La figure 17 représente un différentiateur.

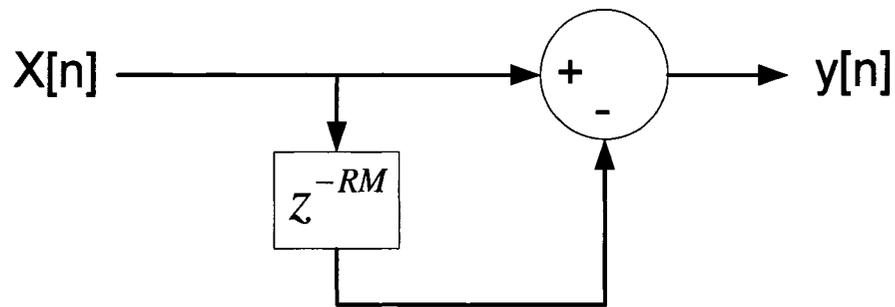


Figure 17 Schéma d'un simple différentiateur

L'équation aux différences du différentiateur est donnée par :

$$y(n) = x(n) - x(n - RM) \quad (2.30)$$

ce qui donne une fonction de transfert sous la forme suivante :

$$H_c(Z) = 1 - Z^{-RM} \quad (2.31)$$

avec R le facteur de décimation et M le délai différentiel.

Le filtre CIC d'ordre N est constitué alors de N intégrateurs en cascade qui fonctionnent à la fréquence d'échantillonnage f_s , suivi d'un facteur de décimation R qui réduit la fréquence de f_s à $\frac{f_s}{R}$ puis de N différentiateurs en cascade qui fonctionnent à la fréquence $\frac{f_s}{R}$. La figure 18 présente la structure d'un filtre CIC d'ordre N .

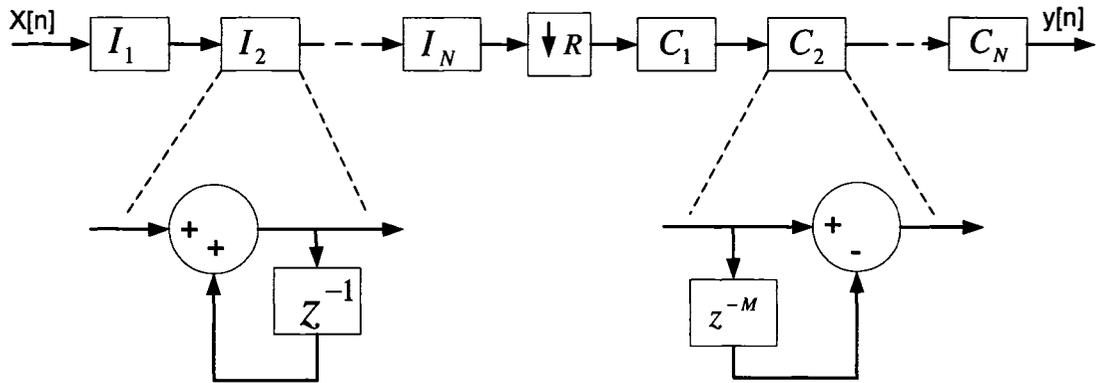


Figure 18 Structure d'un filtre CIC décimateur d'ordre N

La fonction de transfert du CIC décimateur d'ordre N est donnée par le produit de la fonction de transfert de l'intégrateur (2.29) par la fonction de transfert du différentiateur (2.31) :

$$H(Z) = H_I(Z) \times H_C(Z) = \left(\frac{1}{1-Z^{-1}} \right)^N \times (1-Z^{-RM})^N \quad (2.32)$$

$$H(z) = \left(\frac{1-Z^{-RM}}{1-Z^{-1}} \right)^N$$

La division euclidienne du numérateur par le dénominateur de l'équation (2.32) donne

$$\frac{1-Z^{-RM}}{1-Z^{-1}} = 1 + Z^{-1} + Z^{-2} + Z^{-3} + \dots + Z^{-RM+1} \quad (2.33)$$

Donc la fonction de transfert de l'équation (2.32) peut s'écrire sous la forme suivante :

$$H(z) = \left[\sum_{k=0}^{RM-1} Z^{-k} \right]^N \quad (2.34)$$

L'équation (2.34) montre qu'un filtre CIC d'ordre N n'est rien d'autre que N filtres FIR en cascade. L'intérêt de cette équivalence réside dans le fait que le filtre est stable et la phase est linéaire ce qui entraîne un délai de groupe constant. Ces propriétés sont primordiales pour le filtrage.

2.3.3.2 Réalisation des filtres CIC décimateurs

Le développement spectaculaire de la microélectronique a rendu les FPGA ``Field Programmable Gate Array`` très populaire dans le domaine de traitement du signal. Cependant, la vitesse des opérations élémentaires de ces derniers est encore limitée à environ 500 MHz ce qui rend le filtrage à hautes fréquences d'échantillonnage difficile. La structure régulière du filtre CIC décimateur ne permet pas de filtrer un signal échantillonné avec une fréquence de l'ordre des GHz. Il faut donc trouver une architecture de réalisation appropriée qui utilise les filtres CIC décimateurs et qui permettra aux FPGA d'implémenter un filtre à haute vitesse. Différentes architectures sont présentées dans les sections suivantes.

➤ Structure pipeline

La structure pipeline est l'une des premières structures utilisée pour augmenter la vitesse des filtres CIC [7]. Cette technique consiste à couper les longues chaînes d'additionneurs en insérant des registres (P0, P1, P2, ...) entre les additionneurs. Ceci donne un gain en vitesse mais augmente le temps de chargement de toutes les bascules du circuit (la latence). La figure 19 montre la structure pipeline d'un filtre CIC décimateur d'ordre 2.

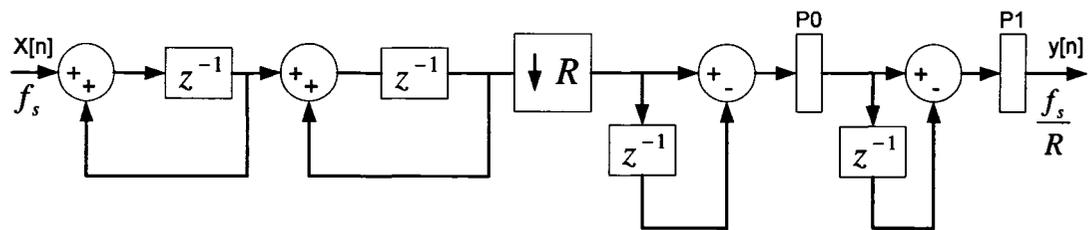


Figure 19 Structure pipeline d'un CIC décimateur d'ordre 2

Cette structure est très facile à implémenter mais reste limitée en terme de rapidité. Ceci fait que son utilisation pour des applications à haute vitesse devient impossible.

➤ Structure parallèle

Le traitement parallèle a pour but d'augmenter la rapidité d'exécution d'un processus. En traitement numérique du signal, le traitement parallèle est la technique qui permet d'atteindre la plus grande vitesse d'exécution pour les circuits arithmétiques tels que les multiplieurs et les additionneurs [23].

Le temps nécessaire à l'exécution d'une addition parallèle dépend du temps de propagation de la retenue. Tous les bits des opérations sont transmis simultanément aux opérateurs parallèles ce qui réduit le temps de propagation de la retenue d'où une augmentation de la vitesse de calcul. Le résultat reste disponible après la détection et la correction du dépassement.

La première approche, pour paralléliser la structure régulière du filtre CIC décimateur, est de considérer des entrées parallèles de l'intégrateur [19]. La figure 20 montre le concept de parallélisme.

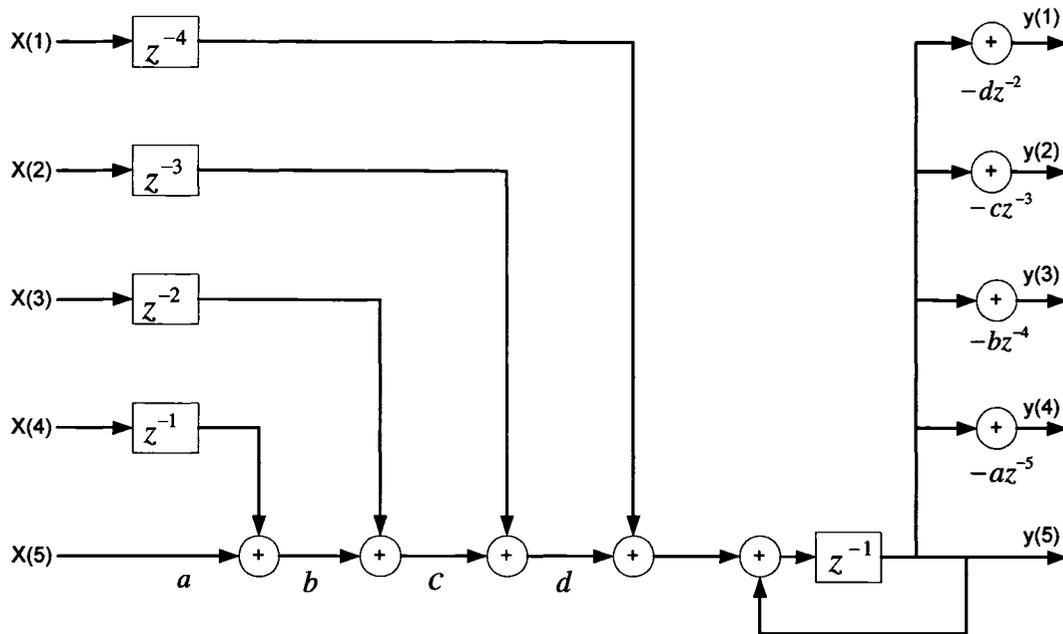


Figure 20 Structure parallèle de l'intégrateur du CIC décimateur

En fait, les 5 entrées $X(1)$, $X(2)$, $X(5)$ sont générés simultanément et additionnées une à une de façon à obtenir une somme totale qui sera ensuite intégrée par l'intégrateur parallèle. Afin de retrouver les images des entrées, il faut soustraire de la sortie de l'intégrateur parallèle, les valeurs $X(1)$, $X(2)$, $X(5)$. Des accumulateurs sont utilisés pour le stockage des résultats intermédiaires. Ensuite vient le facteur de décimation et le différentiateur.

En regardant la structure parallèle de la figure 20, la première chose qui apparaît est la longue chaîne d'additionneurs qui est étroitement lié aux nombres d'entrées parallèles. Ceci limite la vitesse d'exécution.

➤ Structure parallèle améliorée

Cette structure permet de couper la longue chaîne d'additionneurs en utilisant des additionneurs en arbre au lieu qu'ils soient en série comme le montre la figure 21 [19]. Cette structure réduit nettement la latence. Elle passe de $P+2$ à $\log_2(P)$, P étant le nombre d'entrées parallèles.

La complexité d'une architecture de traitement numérique de signal est définie comme étant la quantité de matériel nécessaire pour réaliser cette architecture sous un ensemble de contraintes données. Pour la structure parallèle et la structure parallèle améliorée, les paramètres principaux de complexité sont le nombre d'additionneurs et la taille des mémoires.

Dans le cadre de l'architecture parallèle, l'addition se fait à un résultat précédent disponible dans un registre. La mise en mémoire de ce résultat est ensuite effectuée. Ce qui fait que, si le nombre d'entrées parallèles est P alors la complexité est donnée par :

$$C = P(P + 2) \times b \quad (2.35)$$

b représente la largeur binaire des entrées.

Quand à la structure améliorée présentée à la figure 15, la complexité est donnée par :

$$C = P \log_2(P + 2) \times b \quad (2.36)$$

Les équations (2.35) et (2.36) montrent bien que la complexité de la structure parallèle améliorée est nettement réduite ce qui engendre automatiquement une amélioration de la latence.

Cependant, cette structure montre que son implémentation occuperait une surface CMOS assez grande. Cette surface augmenterait en fonction du nombre de branches

parallèles. L'occupation d'une grande surface CMOS entraîne une consommation de puissance importante. Cette structure réduit la complexité mais n'est pas une solution économique. Donc, il faut se retourner vers d'autres techniques beaucoup plus intéressantes et économiques.

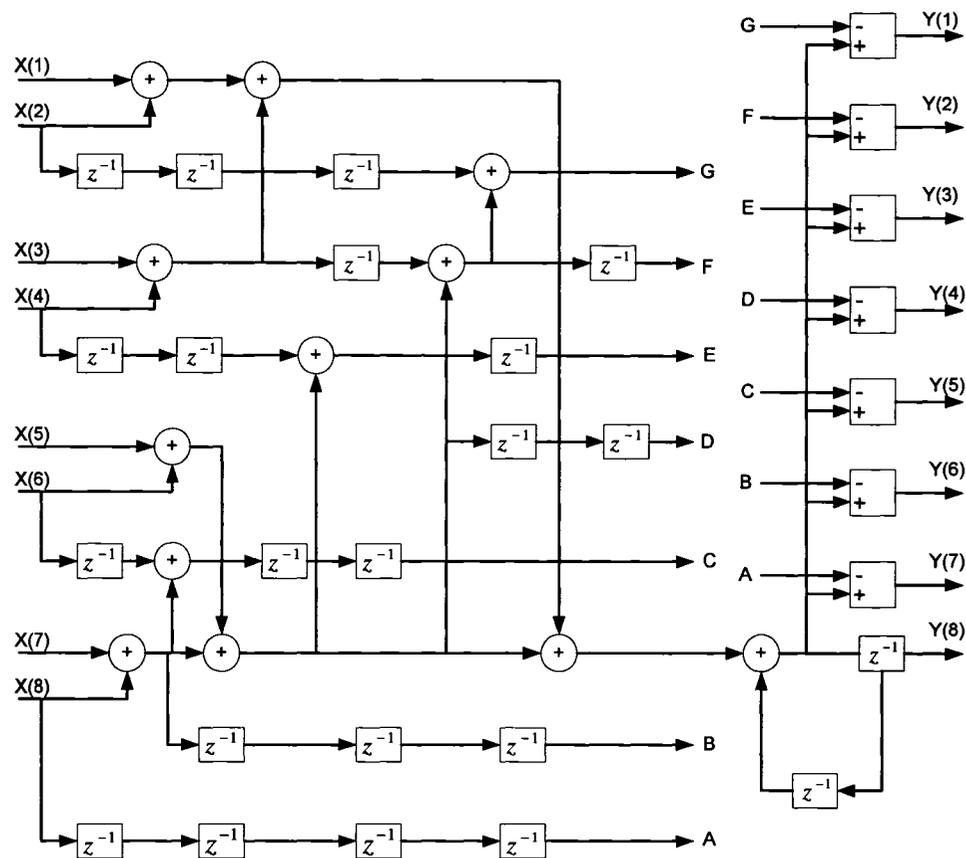


Figure 21 Structure parallélisée améliorée avec 8 entrées

➤ Structure parallèle utilisant la décomposition polyphasée

La fonction de transfert d'un filtre CIC d'ordre N avec un facteur de décimation R est donnée par l'équation (2.32) et s'écrit comme suit :

$$H(z) = \left(\frac{1 - Z^{-RM}}{1 - Z^{-1}} \right)^N$$

Cette fonction de transfert peut être décomposée en deux fonctions de transfert [19]. La première fonction de transfert est d'ordre N_1 avec un facteur de décimation R_1 et la deuxième est d'ordre N_2 avec un facteur de décimation R_2 . Cependant, pour avoir le même taux de décimation, R doit absolument être le produit de R_1 par R_2 . Ainsi, la fonction de transfert peut s'écrire comme suit :

$$H_P(z) = \left(\frac{1 - z^{-R_1}}{1 - z^{-1}} \right)^{N_1} \left(\frac{1 - z^{-R_2 M}}{1 - z^{-R_1}} \right)^{N_2} = H_1(z) \times H_2(z^{R_1}) \quad (2.37)$$

avec

$$H_1(z) = \left(\frac{1 - z^{-R_1}}{1 - z^{-1}} \right)^{N_1} \quad (2.38)$$

$$H_2(z) = \left(\frac{1 - z^{-R_2 M}}{1 - z^{-1}} \right)^{N_2} \quad (2.39)$$

Étant donné qu'un filtre CIC d'ordre N n'est rien d'autre que N filtres FIR en cascade, alors en utilisant l'équation (2.34), l'équation (2.38) peut s'écrire comme suit :

$$H_1(z) = \left(1 + z^{-1} + z^{-2} + \dots + z^{-(R_1-1)} \right)^{N_1} \quad (2.40)$$

L'équation (2.40) montre que $H_1(z)$ équivaut à N_1 filtres FIR en cascade. Ainsi, une partie de la fonction de transfert du filtre CIC est transformée en un filtre FIR. La fonction de transfert $H_1(z)$, qui est un filtre FIR, peut être décomposée en structure polyphasée et donne :

$$H_1(z) = \sum_{i=0}^{R_1-1} z^{-i} E_i(z^{R_1}) \quad (2.41)$$

$E_i(z)$ représente la composante polyphasée avec i le nombre de phases ou de branches parallèles. En d'autres termes, i est le nombre d'entrées parallèles. Le choix de N_1 et de R_1 doit être judicieux. Le tableau III présente les valeurs des coefficients des composantes polyphasées pour différents valeurs de N_1 et de R_1 .

Tableau III
Coefficients des composantes polyphasées

N_1	2			3		
R_1	2	4	8	2	4	8
$E_0(z)$	(1, 1)	(1, 3)	(1, 7)	(1, 3)	(1, 12, 3)	(1, 42, 21)
$E_1(z)$	(2, 0)	(2, 2)	(2, 6)	(3, 1)	(3, 12, 1)	(3, 46, 15)
$E_2(z)$		(3, 1)	(3, 5)		(6, 10, 0)	(6, 48, 10)
$E_3(z)$		(4, 0)	(4, 4)		(10, 6, 0)	(10, 48, 6)
$E_4(z)$			(5, 3)			(15, 46, 3)
$E_5(z)$			(6, 2)			(21, 42, 1)
$E_6(z)$			(7, 1)			(28, 36, 0)
$E_7(z)$			(8, 0)			(36, 28, 0)

Le tableau III montre que tous les coefficients sont des entiers. Ceci est très important du point de vu réalisation car les composantes polyphasées ne nécessitent pas de multiplications. La figure 22 montre un exemple de réalisation pour $E_0(z) = 1 + 3z^{-1}$.

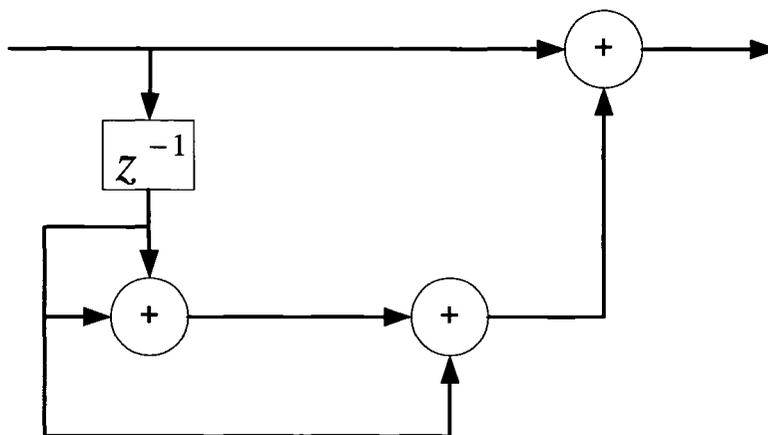


Figure 22 Exemple de réalisation pour $E_0(z) = 1 + 3z^{-1}$

La structure parallèle utilisant la décomposition polyphasée présente beaucoup d'avantages par rapport aux autres structures. Elle ne nécessite pas de multiplications ni de stockage des coefficients. Sa structure est compacte ce qui permet de varier le facteur de décimation sans changer la structure globale. Elle effectue une décimation multiétage qui permet de réduire considérablement la complexité par rapport à une décimation à un seul étage. Elle est aussi facile à implémenter. Ainsi, la structure parallèle du CIC utilisant la décomposition polyphasée est l'architecture retenue pour l'implémentation du filtre décimateur et est présentée à la figure 23.

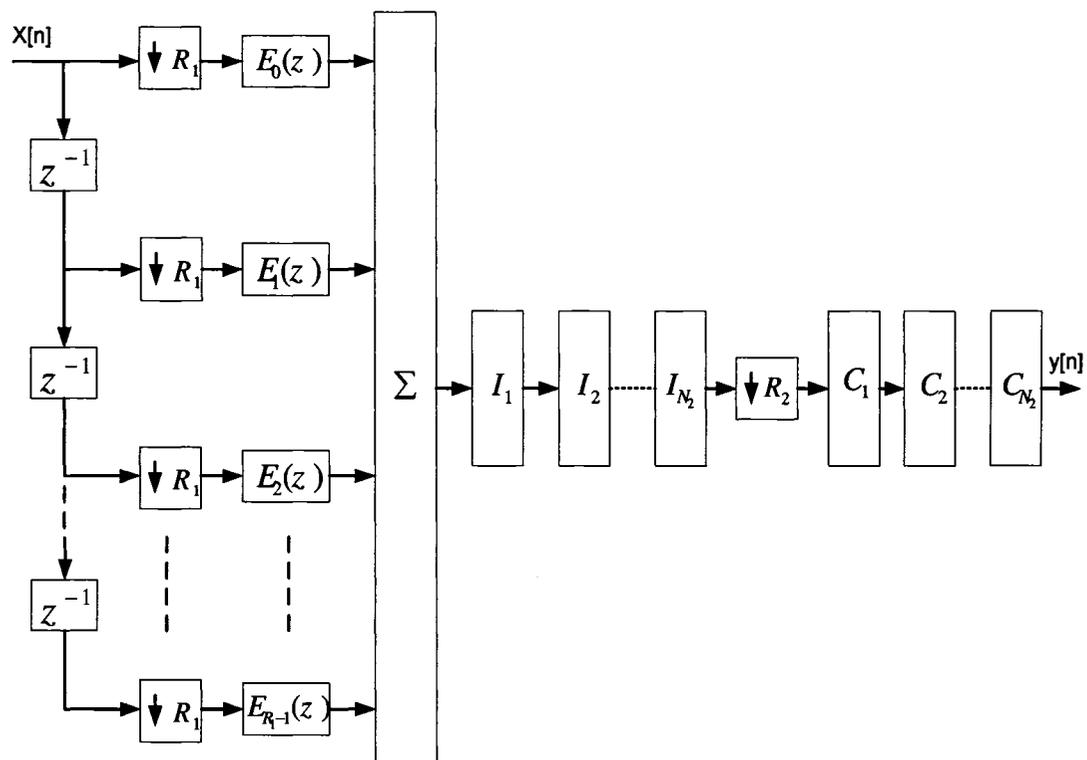


Figure 23 Structure parallèle du CIC utilisant la décomposition polyphasée

La structure du CIC polyphasé est composée de deux étages de décimation. Le premier étage de décimation est un filtre FIR qui fonctionne à la fréquence d'échantillonnage divisée par le nombre de branches parallèles. Le deuxième étage de décimation est un filtre CIC régulier dont les intégrateurs fonctionnent à la même fréquence que le filtre FIR polyphasé et les décimateurs, quand à eux, fonctionnent à la fréquence des intégrateurs divisée par le deuxième facteur de décimation R_2 . Cette structure est très intéressante pour les applications qui utilisent de hautes fréquences d'échantillonnage car elle est facile à implémenter et est capable de filtrer un signal échantillonné à la fréquence $f_s = f_{\text{filtre}} \times R_1$. Ceci veut dire que si la fréquence du filtre est $f_{\text{filtre}} = 150$ MHz et le nombre de branches parallèles est $R_1 = 8$, alors le filtre peut filtrer un signal échantillonné à la fréquence $f_s = 150 \text{ MHz} * 8 = 1.2 \text{ GHz}$.

2.4 Conclusion

Ce chapitre a été consacré aux spécifications du noyau et à la recherche d'une architecture qui répond le mieux aux spécifications désirées.

Dans un premier temps, une étude a été accomplie sur les filtres numériques à réponse impulsionnelle finie (RIF). La théorie de base de ces filtres et différentes structures de réalisation ont été présentées. Ces dernières sont assez intéressantes mais sont très limitées en vitesse. De ce fait, une autre approche a été présentée. Elle consiste à utiliser une décomposition polyphasée qui permet de paralléliser l'entrée. Cette technique augmente considérablement le débit du filtre. Malgré cette augmentation de débit, le filtre reste toujours limité pour des applications en hautes fréquences.

Dans un deuxième temps, les filtres numériques à réponse impulsionnelle infinie (RII) sont présentés. La théorie de base de ces filtres et différentes structures de réalisation ont été présentées. Ces dernières sont plus rapides que celles des filtres RIF mais sont aussi limitées pour des applications à hautes fréquences.. De ce fait, une autre approche a été présentée. Elle consiste à utiliser des filtres numériques à réponse impulsionnelle infinie avec un dénominateur ne contenant que des puissances de z^R . Les inconvénients de cette approche sont la grande sensibilité des coefficients et l'augmentation de la complexité de la section en amont du filtre. Ainsi, l'ensemble des techniques décrites utilisant les filtres RII ne permet pas de répondre adéquatement aux spécifications très exigeantes requises pour le filtrage à haute fréquence.

Les filtres CIC ont été présentés en dernier. La structure de base des filtres CIC est en pipeline. L'utilisation de cette structure en haute fréquence est impossible. Par conséquent, comme le parallélisme est le meilleur moyen d'augmenter la vitesse de traitement, il fallait trouver une architecture parallèle. Ainsi, trois architectures parallèles ont été présentées. La première architecture est une structure parallèle qui présente une

longue chaîne d'additionneur qui augmente au fur et à mesure que le nombre d'entrées parallèles augmente. L'amélioration de cette structure parallèle constitue la deuxième architecture. Cette dernière occupe une très grande surface CMOS. L'occupation d'une grande surface CMOS entraîne une consommation de puissance importante. En fin, une structure parallèle utilisant la décomposition polyphasée a été présentée. Elle est avantageuse car en plus de la simplicité d'implémentation, elle permet d'atteindre des performances que les architectures classiques n'offrent pas. De ce fait, c'est la structure retenue pour la réalisation du noyau programmable.